ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL BALDIATI PARIZI

SUPPORTING THE DECISION-MAKING OF THE DESIGN THINKING TECHNIQUES SELECTION TO USE IN SOFTWARE DEVELOPMENT THROUGH A RECOMMENDATION SYSTEM

Porto Alegre
2023

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

SUPPORTING THE DECISION-MAKING OF THE DESIGN THINKING TECHNIQUES SELECTION TO USE IN SOFTWARE DEVELOPMENT THROUGH A RECOMMENDATION SYSTEM

RAFAEL BALDIATI PARIZI

Doctoral Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Profa. Sabrina dos Santos Marczak, Ph.D.
Co-Advisor: Profa. Dra. Tayana Uchôa Conte

Porto Alegre
2023

Ficha Catalográfica

P234s Parizi, Rafael Baldiati

Supporting the Decision-making of the Design Thinking Techniques Selection to Use in Software Development through a Recommendation System / Rafael Baldiati Parizi. – 2023.
244 p.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina dos Santos Marczak.
Coorientadora: Profa. Dra. Tayana Uchôa Conte.

1. Software Engineering. 2. Design Thinking. 3. Techniques. 4. Recommendation Strategies. 5. Decision-making. I. Marczak, Sabrina dos Santos. II. Conte, Tayana Uchôa. III. , . IV. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

RAFAEL BALDIATI PARIZI

SUPPORTING THE DECISION-MAKING OF THE DESIGN THINKING TECHNIQUES SELECTION TO USE IN SOFTWARE DEVELOPMENT THROUGH A RECOMMENDATION SYSTEM

This Doctoral Thesis has been submitted in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science, of the Computer Science Graduate Program, School of Technology of the Pontifical Catholic University of Rio Grande do Sul.

Sanctioned on March 22, 2023.

COMMITTEE MEMBERS:

Prof. Dr. Afonso Sales (PPGCC/PUCRS)

Profa. Dra. Edna Dias Canedo (PPGI/UNB)

Prof. Dr. José Carlos Maldonado (PPGCC/IME-USP)

Profa. Dra. Monalessa Barcellos (PPGI/PUCRS)

Profa. Dra. Sheila Reinehr (PPGI/PUCPR)

Profa. Dra. Tayana Uchôa Conte (PPGI/UFAM - Co-Advisor)

Profa. Sabrina dos Santos Marczak, Ph.D. (PPGCC/PUCRS - Advisor)

“Design is the process of changing existing situations into preferred ones”
(Herbert A. Simon)

ACKNOWLEDGMENTS

I want to start by expressing my gratitude to my supervisors, Professors Sabrina Marczak and Tayana Conte, who guided me through my graduate road, supporting me and fostering me to research in-depth Design Thinking in Software Development. All the support they gave me was fundamental to obtaining the outcomes of our study. I learned countless other things during all the meetings we had in the last 4 years working together.

I would also like to thank all the researchers who I could collaborate with during my graduation period. It also includes the reviewers of my thesis. In particular, I would like to thank Professor Falk Uebernickel and Danielly the Paula, who welcomed me as a guest researcher at the Hasso-Plattner Institute at the Potsdam University and who extremely collaborated with my work, giving me new perspectives on the use of Design Thinking not only in software development but also in innovation and research.

In special, I would like to share my gratitude to all the undergraduate students who were in touch with our research (Igor, Marina, Lucas, Wagner), making lots of efforts to support not only the development of our research outcomes but also being part of the research process.

It is also important to thank the financial support from different financial agencies. We thank PUCRS BPA 2019, 2020 and 2021 projects (Programa de Bolsas Pesquisa Alunos da PUCRS/Chamada Geral 1/2019, 1/2020 and 1/2021) and CNPq PIBIC 2019/2020 project (Programa Institucional de Bolsas de Iniciação Científica do CNPq) for the undergraduate research assistant scholarships. We thank CNPq (Grant 314174/2020-6, Grant 307177/2018-1, Grant 200372/2022-0). We also thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil) for the financial support (Code 001) and IFFar (Farroupilha Federal Institute) for granted work license (grant 158/2021).

Finally, and most importantly, I would also like to thank my family and friends, who supported me during my graduation. A special thanks go to Luciana and Cleci for supporting me whenever I needed them. You made it possible. Furthermore, I would like to express my gratitude to all those who, directly or indirectly, contributed to the completion of this thesis.

APOIO À TOMADA DE DECISÃO DA SELEÇÃO DE TÉCNICAS DE DESIGN THINKING PARA USO EM DESENVOLVIMENTO DE SOFTWARE ATRAVÉS DE UM SISTEMA DE RECOMENDAÇÃO

RESUMO

Design Thinking (DT) é uma abordagem de solução de problemas utilizada por empresas de software que posiciona o usuário como centro do processo de desenvolvimento para entender suas necessidades e desenvolver soluções que atendam as reais necessidades do usuário. DT fomenta a empatia, a colaboração entre os membros da equipe e o usuário, a formação de equipes multidisciplinares, a ideação de múltiplas soluções e a rápida avaliação das soluções propostas. DT é percebido pelas equipes como um conjunto de técnicas para de forma prática engajar os participantes, gerar ideias inovadoras e ter contato frequente entre time e usuário. Portanto, a seleção de quais técnicas usar se mostra como uma atividade importante, porém desafiante. No entanto, há uma falta de estudos que investiguem como profissionais que usam DT tomam decisões para a seleção de quais técnicas de DT utilizar e de recursos que possam dar suporte à tal decisão. Desta forma, inspirada na metodologia de pesquisa *Design Science Research*, esta Tese de Doutorado traz como contribuição prática o desenvolvimento de um sistema de recomendação colaborativo para prover suporte a profissionais de TI na seleção de técnicas de DT e como contribuições teóricas a modelagem da tomada de decisão da seleção de técnicas de DT e a caracterização de DT no desenvolvimento de software. Estudos empíricos de avaliação mostraram que o sistema de recomendação de técnicas de DT auxilia a decisão de quais técnicas os profissionais podem selecionar. Assim, esta tese defende que a seleção de técnicas de DT é baseada em elementos de contexto e que recursos computacionais possibilitam contribuir para a seleção de técnicas de DT.

Palavras-Chave: Engenharia de Software, Design Thinking, Técnicas, Estratégias de Recomendação, Tomada de decisão.

SUPPORTING THE DECISION-MAKING OF THE DESIGN THINKING TECHNIQUES SELECTION TO USE IN SOFTWARE DEVELOPMENT THROUGH A RECOMMENDATION SYSTEM

ABSTRACT

Software companies have been using Design Thinking (DT) as a problem-solving approach to put the user at the center of the development process, to understand and to develop solutions that meet the real users' needs. DT encourages empathy, team members and users collaboration, the composition of multidisciplinary teams, ideation of multiple solutions and rapid evaluation of the proposed solutions. Literature on the Software Engineering field points out that in addition to a mindset or a process, teams have perceived DT as a set of techniques to engage participants, generate innovative ideas and have frequent contact between team and user. Therefore, selecting which techniques to use is an essential but challenging activity. However, there is a lack of studies investigating how DT practitioners make decisions for selecting which DT techniques to use as well as a lack of resources providing support to decisions in software development. Thus, inspired by the Design Science Research methodology, this Doctoral thesis brings a practical contribution by developing a collaborative recommendation system to support IT professionals in the selection of DT techniques and as theoretical contributions by modeling the decision-making of DT practitioners behind the selection of DT techniques as well as the characterization of the use of DT in software development. Finally, this thesis argues that the selection of DT techniques is context-based and that computational resources contribute to the selection of DT techniques.

Keywords: Software Engineering, Design Thinking, Techniques, Recommendation Strategies, Decision-making.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API – Application Programming Interface

CF – Collaborative Filtering

CBF – Content-based Filtering

CEO – Chief Executive Officer

CODICE – Codisigning DIgital Cultural Encounters

DCIDT – Divergent-Convergent Inquiry-based Design Thinking

DM – Decision-making

DSR – Design Science Research

DSS – Decision-making Support Systems

DT – Design Thinking

DTA4RE – Design Thinking Assistant for Requirement Engineering

EC – Exclusion Criteria

FSSO – Feature Set Score Obtained

GT – Grounded Theory

HCAW – Human-centered Agile Workflow

IC – Inclusion Criteria

IT – Information Technology

JI – Judgement Scale Interpretation

KNN – K-Nearest Neighborhood

ML – Machine Learning

MFSS – Max Feature Set Score

MVP – Minimum Viable Product

OQ – Open Questions

OS – Overall Score

PEU – Perceived Ease of Use

PU – Perceived Usefulness

RE – Requirements Engineering

RQ – Research Question

RS – Recommendation System

RSSE – Recommendation System for Software Engineering

SB – Service Blueprint

SF – Sub-feature

SFIL – Sub-feature Level of Importance

SLM – Sytematic Literature Mapping

SLR – Sytematic Literature Review

TAM – Technology Acceptance Model

TR – Technological Rule

TS – Theoretical Sampling

UCD – User-centered Design

UI – User Interface

UX – User Experience

WSS – Number of Working Spaces

CONTENTS

1. INTRODUCTION

1.1 Context and Research Problem

Design Thinking is a problem-solving approach that brings human-centered design principles at its core [73]. Driven by the search for innovation, it fits the development of new products, services, or processes, from startups to large and complex environments [35]. Defined as a User-centered Design (UCD) approach [154], DT has been integrated with software development aiming to boost the creation of user-centered solutions, transforming users into active participants in the development process [2]. When used by software development teams, DT fosters problem exploration iteratively and emphatically, contributing to user engagement and team collaboration [68, 107].

Design Thinking can be used in the early phases of the software development process to identify what the customer needs are, providing better support for downstream development activities, mainly those related to identifying a proper solution for the problem at hand. The multidisciplinary view offers software teams a collaborative problem-solving environment, bringing all involved parties closer together [235, 237].

Integrated with Agile, DT fosters the search for a solution that meets the user's needs, while agile methods are intensely collaborative, focused on speed, simplicity, quick and continuous deliveries, feedback collection and reaction to changes [103, 243].

Literature reporting on the use of DT techniques in software development has been growing. Souza *et al.* (2017) [65] show that IT professionals have used more than 50 techniques associated with DT. Dobrigkeit and De Paula (2019) [68] studied the perceptions of different functions (developers, managers) and concluded that the knowledge and the application of DT might be different within the same team, arguing that implementing the appropriate techniques is a key success factor.

Kahan *et al.* (2019) [124] argue that DT supports requirements elicitation and using the appropriate set of techniques might be a key element. Hehn *et al.* (2020) [107] formulated a framework integrating DT and Requirements Engineering (RE), proposing 40 artifacts where DT can be used as the first step of RE activities (upfront), integrated into the RE activities (infused) or along the software development process (continuous). However, the authors indicate that selecting what techniques to use and when to integrate each technique into RE activities are open issues when integrating DT into RE.

However, little is discussed about the selection of DT techniques for software development, turning the selection of which techniques to use in a complex decision-making endeavor. Pessoto (2017) [203] states that "*during the Design Thinking process, designers find themselves continuously in a situation where making a decision is fundamental to* *proceed and develop a solution. Conditions of certainty develop better-taken decisions; however, there are numerous cases where uncertainty dominates the choice"*. Thus, a computational mechanism that supports IT professionals in the decision-making of which DT techniques to use can be of help.

In this scenario, selecting what DT techniques to use in software development opens the doors to Recommendation Systems (RSs). RSs perform information filtering, providing recommendations of relevant items based on the user's previous interests [101]. Recommendation systems allow users to obtain information to aid decision-making in complex (large amounts of data) scenarios [226] and filter the most valuable information from that available [209, 250].

1.2 Goals and Research Questions

This thesis seeks to support the decision-making of the Design Thinking techniques selection by proposing, developing and evaluating a DT techniques recommendation system for software development. For us, supporting means giving access to qualified and updated information about DT techniques, taking into account a collaborative environment of experiences of DT practitioners.

In addition, we defined the following specific goals:

1. Characterizing the state-of-the-art of DT in software development;
2. Exploring the state-of-the-practice on the use of DT in software development;
3. Modeling the decision-making of DT practitioners for selecting DT techniques to use in software development;
4. Developing and evaluating a DT techniques recommendation system to support the selection of DT techniques in software development.

To achieve the research goals, we posed the following main research question (RQ): "***How can we support software development professionals to select Design Thinking (set of) techniques?***"

Inspired by the RQ and basing our research agenda on Design Science Research (DSR), we also posed the following complementary research questions:

RQ1) How has DT been integrated into software development, and what are DT models and techniques being used?

- Conducted studies: a Systematic Literature Mapping and an Exploratory Survey with IT professionals;

RQ2) What challenges do IT professionals face when using DT techniques in software development and how difficult do they consider the selection of DT techniques?

- Conducted studies: a Systematic Literature Mapping, an Exploratory Survey with IT professionals and an Interview-based study with DT practitioners;

RQ3) What is the decision-making of IT professionals behind selecting DT techniques to use in software development?

- Conducted studies: an interview-based study with DT facilitators;

RQ4) How might a DT techniques recommendation system support the decision-making of selecting DT techniques in software development?

- Conducted studies/activities: a Meta-Design Thinking Session, a requirements elicitation activity, an early tool evaluation, a DESMET comparison method evaluation, a requirements refining activity, a Technology Acceptance Model (TAM) tool evaluation, an interview-based study for modeling the decision-making of the selection of DT techniques, and an empirical study for validating the DT techniques recommendation system with DT practitioners.

1.3 Publications

We list below the papers that we published in Conferences (national and international) and the articles published in Journals. We sorted the publications considering the timeline of the studies that we conducted. Table 1.1 summarizes all publications and the metadata related to them, sorted by data of publication.

- **Publication 1:** (ERES 2020): Kryvoruchca, G.; Pereira, L.; Parizi, R.; dos Santos Marczak, S. The Use of Design Thinking in a Global Information Technology Company. Proceedings of the Regional Software Engineering School, 2020.
  - This study aimed to describe how does the adoption of Design Thinking with software development take place in a global information technology company. We conducted an interview-based case study with 16 professionals.
- **Publication 2:** (JSS 2022): Parizi, R.; Plautz, M.; Marczak, S.; Conte, T. How has design thinking being used and integrated into software development activities? A systematic mapping. Journal of System and Software, 2022.
  - In this article, we report a Systematic Mapping Study to investigate the use of DT in software development. We evaluated 127 papers from 2010 to 2021. We

analyzed how DT is integrated in software development, what are the models and techniques, what are criteria used for selecting DT techniques, and what are the key points that DT practitioners should be aware of when using DT.

- **Publication 3:** (XP 2020): Prestes, M. P.; Parizi, R.; Marczak, S.; Conte, T. On the Use of Design Thinking: A Survey of the Brazilian Agile Software Development Community. In: Agile Processes in Software Engineering and Extreme Programming, Springer, Copenhagen, Denmark. 2020.
  - In this paper we aimed to characterize how the software companies have been implemented DT. It presents the results of a survey answered by 127 professionals from the Brazilian software industry.
- **Publication 4:** (CIBSE 2020) – nominated as best paper - Parizi, R.; Silva, M.; Couto, I.; Trindade, K.; Prestes, M.; Marczak, S.; Conte, T.; Candello, H. Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool. In: Ibero-American Conference on Software Engineering, Springer.
  - In this paper, we presented the the meta-DT session in order of developing a collaborative tool to provide recommendations about potential DT techniques to be used in support of requirements engineering activities. We described our DT session that identified the collaborative tool as a proposed solution, a requirements elicitation activity to define the tool scope, and an interview-based early experimental study with professionals that use DT in industry.
- **Publication 5:** (SBQS 2020): Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. A Design Thinking Techniques Recommendation Tool: An Initial and On-Going Proposal. Brazilian Symposium on Software Quality, 2020.
  - In this paper, we focused on comparing Helius features with other related tools using the DESMET method.
- **Publication 6:** (CSBC-WTD 2020): Parizi, R.; Marczak, S. A Context-based Recommendation Model for Design Thinking Techniques Selection in Software Development. In: Proceedings of the Brazilian Conference on Software: Practice and Theory (WTD). Online, 2020.
  - In this paper, we presented our research agenda that involved the characterization of the state-of-the-practice and the development of a DT techniques recommendation tool. Our goal was to collect feedback from different researchers and validate our research proposal.

- **Publication 7:** (SAC 2021): Pereira, L.; Parizi, R.; Prestes, M.; Marczak, S.; Conte, T. Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering In: Symposium on Applied Computing. 2021.
  - In this paper, we combined two qualitative methods, a focus group to collect and understand the professionals' DT usage opinions and a survey to confirm the professionals' challenges and benefits of using DT. This paper was produced in collaboration with another Ph.D. student.
- **Publication 8:** (ICEIS 2021): Pereira, L.; Parizi, R.; Marczak, S.; Conte, T. Design Thinking Techniques Selection in Software Development: On the Understanding of Designers and Software Engineers Choices. In International Conference on Enterprise Information Systems, Prague, Czech Republic. 2021. p. 353-360.
  - In this paper, we conducted an empirical investigation to identify how Designers and Engineers are adopting DT, and understanding their decision-making. We had 39 practitioners who used DT for software development in seven focus groups among 35 distinct organizations. This paper was produced in collaboration with another Ph.D. student.
- **Publication 9:** (WCO 2021 - Poster): Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. Proposta de um Sistema Colaborativo de Recomendação de Técnicas de Design Thinking em Projetos de Software. In: Workshop de Colaboração Online, Online. 2021.
  - In this poster, we introduced the idea of implementing a community of practices in Helius, to support the decision-making and the selection of DT techniques to use in software development.
- **Publication 10:** (WER 2021): Parizi, R.; Couto, I.; Hanauer, L.; Conte, T.; Marczak, S. Helius: On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals' Collaboration. In: Requirements Engineering Workshop, PUC-Rio, Brasília, DF. 2021.
  - In this paper, we further the presentation of our Design Thinking Techniques recommendation system, comparing it to other decision-support systems.
- **Publication 11:** (WER 2021): Filho, J. C. d. S. D.; Damian, A. L.; Parizi, R.; Marczak, S.; Conte, T. Aplicando Técnicas de Design Thinking para a Especificação de Cenários na Elicitação de Requisitos. Workshop de Engenharia de Requisitos, 2021.

  - In this paper, we worked to understand the role of DT techniques in constructing Scenarios in Requirements Engineering and learn more about DT techniques present in Helius. This paper was produced in collaboration with another Master's student from UFAM.

- **Publication 12:** (REIC 2021): Kryvoruchca, G.; Parizi, R.; Correa, L.; Marczak, S. On the Understanding of the Benefits and Challenges of DT Adoption in Software Development: A Cross-data Analysis. Rev. Eletrônica de Iniciação Científica em Computação, 2021.

  - This paper was an extended article (Publication 1), where er further investigate the perceived benefits and challenges of adopting DT in software development. We performed a cross-analysis on data we collected in 2 additional studies: a Survey with 158 IT professionals and a focus-group-based study with 39 IT professionals. Our analysis compares the benefits and challenges of adopting DT by ORG with those we collected from other professionals, serving as a guide for practitioners on the use of DT in software development.

- **Publication 13:** (SBSC 2022): Parizi, R.; Leal, L.; Marczak, S.; Conte, T.; Uma Proposta de Comunidade de Prática a partir da Experiência de Uso de Técnicas de Design Thinking no Desenvolvimento de Software. Simpósio Brasileiro de Sistemas Colaborativos, 2022.

  - This paper presents a detailed proposal of a community of practice of DT practitioners implemented in Helius. Our goal was to collect feedback from researchers from the Collaborative Systems community.

- **Publication 14:** (JSERD 2022): Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. A Tool Proposal for Recommending Design Thinking Techniques in Software Development. Journal of Software Engineering Research and Development, 2022.

  - This publication presents in detail the process from the ideation to the requirements refining and validation of our Design Thinking Techniques recommendation system.

1.4 Thesis Outline

The remainder of the thesis is outlined as follows: Chapter 2 introduces Design Thinking in software development, decision-making and recommendation systems. Chapter 3 details our research design based on the Design Science Research methodology, showing the activities we performed in each DSR iteration. Chapter 4 presents the problem understanding space, where we collected data through exploratory studies to define a research problem, while Chapters 5, 6 and 7 describe and detail the 3 iterations in the DSR for proposing and evaluating a solution to solve the identified problem. Finally, Chapter 8 concludes this thesis, starting with an overall overview, followed by the contributions that this doctorate brings to the field, a summary of the opportunities for future research, and the publications reached as an outcome of our research efforts.

Table 1.1: Publications of this doctoral research

ID	Qualis	Venue	Publication
			2022
15	–	PROFES (Journal First)	Parizi, R.; Plautz, M.; Marczak, S.; Conte, T. How has design thinking being used and integrated into software development activities? A systematic mapping. *International Conference on Product-Focused Software Process Improvement, Jyväskylä, Finland*, 2022.
14	A1	Journal of System and Software (JSS)	Parizi, R.; Plautz, M.; Marczak, S.; Conte, T. How has design thinking being used and integrated into software development activities? A systematic mapping. *Journal of System and Software*, 2022.
13	B1	Journal of Software Engineering Research and Development (JSERD)	Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. A Tool Proposal for Recommending Design Thinking Techniques in Software Development. *Journal of Software Engineering Research and Development*, 2022.
12	B1	Brazilian Symposium on Collaborative Systems (SBSC)	Parizi, R.; Leal, L.; Marczak, S.; Conte, T.; Uma Proposta de Comunidade de Prática a partir da Experiência de Uso de Técnicas de Design Thinking no Desenvolvimento de Software. *Simp. Bras. de Sistemas Colaborativos*, 2022.
			2021
11	C	Revista Eletrônica de Iniciação Científica em Computação (REIC)	Kryvoruchca, G.; Parizi, R.; Correa, L.; Marczak, S. On the Understanding of the Benefits and Challenges of DT Adoption in Software Development: A Cross-data Analysis. *Revista Eletrônica de Iniciação Científica em Computação*, 2021.
10*	A4	Workshop on Requirements Engineering (WER)	Filho, J. C. d. S. D.; Damian, A. L.; Parizi, R.; Marczak, S.; Conte, T. Aplicando Técnicas de Design Thinking para a Especificação de Cenários na Elicitação de Requisitos. *Workshop de Engenharia de Requisitos*, 2021.
9	A4	Workshop on Requirements Engineering (WER)	Parizi, R.; Couto, I.; Hanauer, L.; Conte, T.; Marczak, S. Helius: On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals' Collaboration. In: Requirements Engineering Workshop, PUC-Rio, Brasília, DF. 2021.
8	–	Workshop on Online Collaboration (WCO)	Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. Proposta de um Sistema Colaborativo de Recomendação de Técnicas de Design Thinking em Projetos de Software. In: Workshop de Colaboração Online, Online. 2021.
7*	A3	International Conference on Enterprise Information Systems (ICEIS)	Pereira, L.; Parizi, R.; Marczak, S.; Conte, T. Design Thinking Techniques Selection in Software Development: On the Understanding of Designers and Software Engineers Choices. In International Conference on Enterprise Information Systems, Prague, Czech Republic. 2021. p. 353-360
6*	A2	Symposium on Applied Computing (SAC)	Pereira, L.; Parizi, R.; Prestes, M.; Marczak, S.; Conte, T. Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering In:*Symposium on Applied Computing*. 2021.
			2020
5	A3	International Conference on Agile Software Development (XP)	Prestes, M. P.; Parizi, R.; Marczak, S.; Conte, T. On the Use of Design Thinking: A Survey of the Brazilian Agile Software Development Community. In: *Agile Processes in Software Engineering and Extreme Programming*, Springer, Copenhagen, Denmark. 2020. p. 73-86
4	B2	Ibero-American Conference on Software Engineering (CIBSE)	Parizi, R.; Silva, M.; Couto, I.; Trindade, K.; Prestes, M.; Marczak, S.; Conte, T.; Candello, H. Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool. In: *Ibero-American Conference on Software Engineering*, Springer, Curitiba, Brazil. 2020.
3	–	Regional Software Engineering School (ERES)	Kryvoruchca, G.; Pereira, L.; Parizi, R.; dos Santos Marczak, S. The Use of Design Thinking in a Global Information Technology Company. *Proceedings of the Regional Software Engineering School*, 2020.
2	B1	Brazilian Symposium on Software Quality (SBQS)	Parizi, R.; Moreira, M.; Couto, I.; Marczak, S.; Conte, T. A Design Thinking Techniques Recommendation Tool: An Initial and On-Going Proposal. *Brazilian Symposium on Software Quality*, 2020
1	–	Brazilian Conference on Software: Practice and Theory (CBSOFT - WTD)	Parizi, R.; Marczak, S. A Context-based Recommendation Model for Design Thinking Techniques Selection in Software Development. In: *Proceedings of the Brazilian Conference on Software: Practice and Theory (WTD)*. Online, 2020.

2. BACKGROUND

This chapter introduces Design Thinking in the context of software development, highlighting the perspective of DT as a set of techniques. The chapter also introduces decision-making, its strategies and criteria for problem-solving and the role of recommendation systems in that context.

2.1 Design Thinking in Software Development

Design Thinking is a human-centered problem-solving approach that explores the users' and businesses' needs, transforming ideas into acceptable and validated solutions [35]. DT can be understood as "*a way of describing a designer's methods that is integrated into an academic or practical management discourse*" [120]. Hiremath and Sathiyam (2013) [111] argued that DT is increasingly used in software development companies as a tool for innovation. It offers iterative learning from the beginning of the development cycle, including continuous improvement [108].

In software development, DT supports the understanding of the problem to be solved and the proposal and validation of solutions that meet the users' needs [8, 167, 140]. DT also collaborates from the early stages of software activities–from the elicitation of requirements [107] to the creation of an innovative mindset in developers, engineers, and managers [68].

By bringing the user needs to the center of the discussion, DT also improves team communication and facilitates knowledge domain acquisition, which is a well-known issue in software development [154]. DT is also considered an easy-in integration and a way to boost agile development [213].

Brenner, Uebernickel and Abrell (2016) [33] present Design Thinking from 3 perspectives - DT as a mindset, process, or as toolbox. This characterization has been widely accepted in the literature [140, 148, 107, 162]. DT as a mindset considers that humans make innovation for humans, combines divergent and convergent thinking, promotes the philosophy of fail often and early, fosters the creation of prototypes that can be experienced, and tests early with customers [68]. Dobrigkeit and De Paula (2019) [68] also argue that DT as a mindset facilitates the professional to work on teams composed by diverse professionals. Hehn *et al.* (2020) [107] mention that the company's success includes changes in the development team's mindset. The authors argue that team members must be empathic to participate in co-creation activities since the value can be obtained only by understanding the customers' needs.

Design Thinking as a process is structured as a set of iterative working spaces, exploring both divergent and convergent thinking [33]. Literature reports a span of DT processes, also known as DT models. Each model defines DT as a set of working spaces to understand the problem and produce innovative solutions [35, 13]. A set of techniques can be applied in each working space, configuring the third perspective of DT: as a toolbox, [107, 53, 51]. DT as a toolbox refers to the use of design methods and techniques from engineering, computer, and psychology to solve a problem [140]. DT as a toolbox provides practitioners with multiple mechanisms to aid the creation of a solution in the design process. The use of appropriate methods is a core factor of success [68]. In this thesis, we consider the perspective of DT as a toolbox.

Literature discussing DT as a set of techniques has been growing. Liedtka (2015) [151] suggests a list of techniques associated with DT working spaces. The author summarizes a wide span of DT techniques, grouping them into visualization, ethnography, collaborative sense-making, assumption surfacing, prototyping, co-creation, and field experiment techniques. Meireles *et al.* (2022) [170] propose a categorization of DT techniques including the input, control (how to use), mechanism, and output terms, indicating that it is important to take into account what techniques to use. Rozante, Amancio and Flores (2020) [64] argue that the key to conducting DT sessions with no quality loss is by choosing the proper techniques. They also point out that the DT techniques allow the stakeholders to gather different points of view, boosting creativity and innovation.

Therefore, considering the DT leans and how it can support the discovering of user needs and scoping of a solution [107], well-known activities of the Elicitation phase of the Requirements Engineering discipline, our long-term research goal is to support software development professionals' decision in the selection of which DT techniques to use in a certain development scenario.

2.2 Decision-making

Decision-making (DM) is the *"process of choosing a preferred option or a course of actions from among a set of alternatives on the basis of given criteria or strategies"* [267]. As a process, a decision involves activities that lead the decision-maker to identify the best alternative among the available ones to solve an identified problem [254]. Bock (2015) [27] argues that "*an individual who must make a decision is thought to have to choose between a set of alternative courses of action. A good or "optimal" choice [...] is seen as one that maximizes the presumed attainment of pursued goals.*"

Decision-making is composed of intentional elements, combining technologies, evaluation of alternatives and the decision itself. Tello *et al.* (2019) [254] abstractly draw the decision-making process comprising the following steps (Figure 2.1): a) identify a prob-

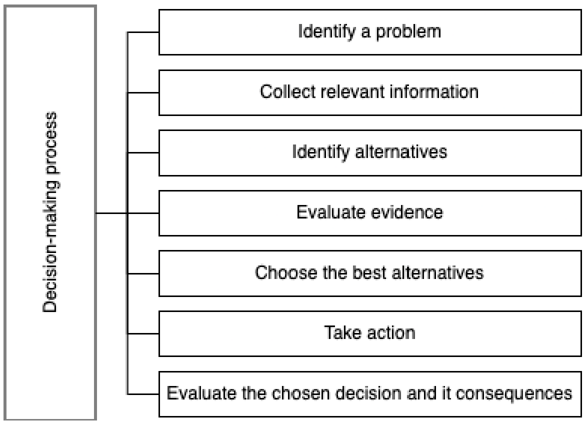


Figure 2.1: Decision-making process drawn by Tello *et al.* (2019) [254]

lem, b) collect relevant information, c) identify alternatives, d) evaluate the evidence, e) choose the best alternative, f) put it into practice, and g) evaluate the chosen decision and its consequences. The authors also suggest that these steps are stimulated by a starting point: understanding that a decision needs to be made.

Figure 2.2 shows a step-wise DM process proposed in the Guide to the Software Engineering Body of Knowledge (SWEBOK) [30], containing the following steps: understanding the real problem, defining the selection criteria or identifying all reasonable technically feasible solutions, evaluating each proposal against selection criteria, selecting the preferred proposal and monitoring the performance of the selected proposal.

Lehto *et al.* (2012) [144] argue that DM can not be unplugged from the context in which it is required. For instance, decisions made for daily activities may be simpler than others for specific moments, or those required under a scenario of severe pressure may be more complex.

Tonetto *et al.* (2006) [256] associate decision-making with human thinking, arguing that making a decision involves a set of heuristics, i.e., general rules of influence that the decision-maker uses to simplify their judgments in decision-making tasks [256]. Rangel (2008) [222] points out that decision-making is a complex process in which the decision-maker (a human being) considers his emotions, motivations, perceptions and

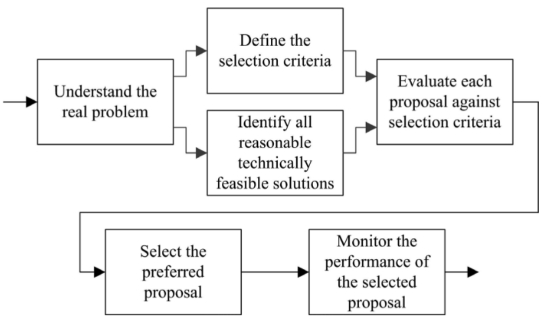


Figure 2.2: Decision-making process proposed in the SWEBOK Guide [30]

previous experiences as key elements to achieve a decision. Zachary *et al.* (1982) [273] argue that each decision comprehends 3 elements (also called constituents): (1) decision-situation (where, when and for what the decision is made), (2) decision-maker (who are making a decision), and (3) decision process (what are the steps of the decision-making).

Hastie (2001) [106] advocates that "*Good decisions are those that effectively choose means that are available in the given circumstances to achieve the decision-maker's goals.*" The author also mentions that a decision problem might be graphically represented as a decision tree (Figure 2.3), highlighting three major components: alternative courses of action, consequences, and uncertain conditioning events. The author presents definitions for each concept of decision-making (Table 2.1).

Wang and Ruhe (2007) [267] define DM as a set of 3 essences: (1) a decision is made when a decision goal is defined, (2) a set of alternatives or choices are known, and (3) a set of strategies and decision-criteria are established by the decision-maker to select

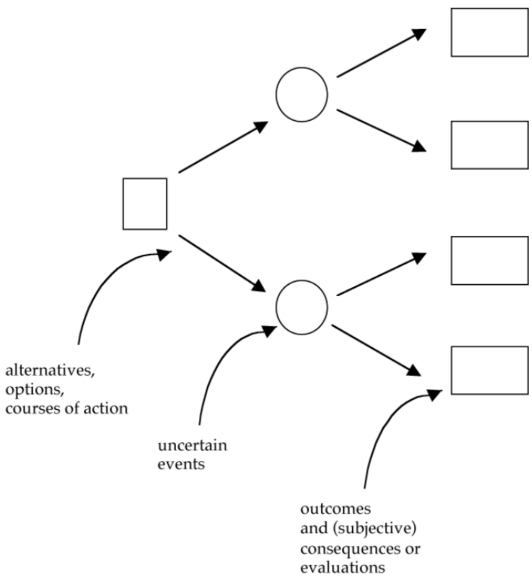


Figure 2.3: Template for a decision [106]

Table 2.1: Decision-making concepts presented by Hastie (2001) [106]

Type	**Description**
Decision-making	the entire process of choosing a course of action
Judgment	the components of the larger decision-making process that are concerned with assessing, estimating, and inferring what events will occur and what the decision-maker's evaluative reactions to those outcomes will be
Uncertainty	the decision-maker's judgments of the propensity for each of the conditioning events to occur
Preferences	behavioral expressions of choosing (or intentions to choose) one course of action over others
Outcomes	the publicly describable situations that occur at the end of each path in the decision tree (of course, outcomes may become mere events if the horizon of the tree is extended further into the future)
Consequences	the subjective evaluative reactions (measurable on a good-bad, gain-loss scale) associated with each outcome

one or more alternatives. The authors also categorize the decision-making processes into 2 categories, depending on the decision strategy used:

- Descriptive DM: illustrates empirical investigations that allow observing the behavior behind a choice;
- Normative DM: indicates that the decision-maker uses his rationale and follows his own preferences. It means that the decision-maker may consider elements like minimum cost, maximum benefit, and others for making a decision.

Wang and Ruhe (2007) [267] also proposed a taxonomy of decision-making. The authors classify a decision into 4 categories: intuitive, empirical, based on heuristics, and rational. Table 2.2 summarizes the decision-making taxonomy. For each decision category, one or more decision strategies might be used, as well as the criteria used by the decision-maker to select the alternative (or set of) that solves the problem.

Lehto *et al.* (2012) [144] argue that decision-making is connected to problem-solving. Therefore, decision-making, problem-solving, and Design Thinking are related concepts since DT is a problem-solving approach that considers the human aspect involved in deciding the most suitable solutions.

In this scenario, computational resources might be used as mechanisms to support decision-making, the use of DT, and problem-solving as well. In Section 2.3 we introduce the role of recommendation systems in the support of decision-making.

Table 2.2: Decision-making taxonomy proposed by Wang and Ruhe (2007) [267]

Category	**Strategy**	**Criteria**
Intuitive	Arbitrary	Based on the most easy or familiar choice
	Preference	Based on propensity, hobby, tendency, expectation
	Common sense	Based on axioms and judgment
Empirical	Trial and error	Based on exhaustive trial
	Experiment	Based on experiment results
	Experience	Based on existing knowledge
	Consultant	Based on professional consultation
	Estimation	Based on rough evaluation
Heuristic	Principles	Based on scientific theories
	Ethics	Based on philosophical judgment and belief
	Representative	Based on common rules of thumb
	Availability	Based on limited information or local maximum
	Anchoring	Based on presumption or bias and their justification
Rational	Minimum Cost	Based on minimizing energy, time, money
	Maximum cost	Based on maximing gain usability, functionality, reliability, quality, dependability
	Maximum utility	Based on cost-benefit ratio
	Interactive events	Based on automata
	Games	Based on conflict
	Decision-grids	Based on a series of choices in a decision grid

2.3 Recommendation Systems

Recommendation systems (RSs) support overcoming information overload by filtering out relevant information to the user [209, 3, 181]. In addition, recommendation systems have collaborated effectively in decision-making processes since they enable users to get personalized information based on their interests [117, 25, 137]. Sharma *et al.* (2019) [239] mention that RSs improve the decision-making process by presenting the users with the most suitable suggestions.

RSs integrate computing and engineering methods to provide suggestions that meet users' needs and preferences [226]. Taghavi *et al.* (2018) [250] discuss that RSs have been used in domains such as entertainment, education, and service, by recommending music, TV shows, friends on social networks, books, articles, and so on, including companies such as Netflix, Facebook, and Amazon [172, 264].

Developing an RS involves defining a recommendation approach, i.e., whether the RSs will provide identical recommendations for all its users (non-personalized) or whether it will provide specialized recommendations for a particular user (personalized) [250, 209, 3, 137]. Non-personalized recommendation systems do not consider user information for recommending items [25]. They use statistical analysis, such as the top-rated items and the most-used items, such as the top-N new products [101, 209, 226]. Non-personalized RSs provide identical recommendations for all users [209, 250].

On the other hand, Personalized RSs provide recommendations considering the characteristics of the items (content-based recommendations) and/or characteristics of the users (collaborative filtering-based recommendations). Personalized RSs analyze user ratings for items to determine a user profile, which is composed of the user's desires and preferences [209]. For instance, personalized RSs establish the user's preferences and uses previous evaluations for items or identify users similar to the active user (the user for whom the recommendation is intended) to generate the recommendation. Machine Learning (ML) algorithms such as *K-Nearest Neighbors* (KNN) and *k-means* are used to search for similar users, supporting personalized recommendations[188].

RSs request data from users or collect the data implicitly. For instance, implicit data are collected from users' clicks on items, while explicit data are asked of the users as ratings to items [209].

The most well-known approaches of RSs are classified as content-based, collaborative filtering, and hybrid [117, 250]. Each approach is based on different mechanisms for handling information and assessments that users provide to items as follows.

- Content-based Filtering (CBF) - consider the characteristics of the items for the recommendations. The recommendation of items to the active user (the user who requests or must receive the recommendation) is based on the similarity of items with others previously evaluated positively by the user [255].

- Collaborative Filtering (CF) - explore information about past actions or opinions of users of a specific community. CFs use a user-items rating matrix as input and produce as output values indicating how much the active user may like or dislike a specific item or even a list of N recommended items, characterized as top N-list [171].

- Hybrid Recommendations - use the combination of two or more recommendation techniques to increase the accuracy and performance of the recommendations. The combination of the approaches is used to enhance the advantages and mitigate the weaknesses of each approach [117].

Robillard *et al.* (2010) [226] introduce recommendation systems for Software Engineering (RSSE). The authors argue that RSSEs can provide relevant information for performing tasks foreseen in SE activities. RSs are decision-support systems. Liang (2008) [150] argues that "*as the concept of decision support systems (DSS) has evolved from aiding decision makers to perform analysis to providing automated intelligent support, the recent proliferation of recommendation systems has shown the power of DSS in enhancing performance*". Zeebaree and Aqel (2019) [274] point out that DSS systems reduce uncertainty and improve decision-making.

Pérez *et al.* (2010) [217] define the goal of DSS as follows: "*The central goal of decision support systems is to process and provide suitable information to support individuals or organizations in their decision-making tasks.*" The authors refer to decision-making as a selection process, which "*consists of how we can obtain the solution set of alternatives from the opinions on the alternatives given by the experts.*"

In this context, this thesis proposes Helius, a collaborative recommendation system to support IT professionals in selecting DT techniques for software development. Helius provides both personalized and non-personalized recommendations of DT techniques. In addition, this thesis also proposes a DT techniques decision-making model, taking into account the decision elements of DT facilitators when selecting DT techniques. The decision-making model serves as a source of information for proposing recommendation mechanisms of DT techniques.

Next, Chapter chapter 3 shows the research methodology of this thesis.

3. DESIGN SCIENCE RESEARCH AS METHODOLOGY

This chapter presents the research methodology that we designed for this thesis. Our goal is to answer the following research question: *How can we support software development professionals to select DT (set of) techniques during requirement engineering?*

Given its nature, our research can be characterized as qualitative. Qualitative research aims to understand and find explanations for a phenomenon, while quantitative research focuses on collecting data from an observed phenomenon, characterizing its analysis based on mathematical and statistical models [80].

Our research agenda consists of empirical methods considering that we aim to understand a phenomenon in the software development context. Wohlin *et al.* (2003) [270] argue that empirical methods provide a relevant scientific basis for the area by considering software development as a human activity.

The research in this thesis follows the Design Science Research methodology [228, 110, 268]. We inspired our research agenda in the DSR framework proposed by Runeson *et al.* (2020) [228]. By using DSR, we started by identifying and defining a problem instance to next propose and validate artifacts to solve the problem at hand. As a result, our long-term research goal is to contribute to both state-of-the-art and state-of-the-practice on the use of DT techniques in software development[1].

3.1 Design Science Research

DSR promotes problem-solving by exploring instances of the problem in practice, creating artifacts that result in better constructs (human-made designs) in specific contexts. Wieringa (2014) [268] introduced Design Science in SE, arguing that "*Design Science is the design and investigation of artifacts in context. Artifacts are designed to interact with a problem context to improve something in that context*".

Runeson *et al.* (2020) [228] presented an iterative framework for DSR in SE research. The framework has 3 components (or activities): *Problem conceptualization activity* (problem understanding), which expresses the understanding of a general problem as instances (concrete problems); *Solution Design activity* (solution design approach): which represents a creative activity of problem-solving where solutions for the problem-at-hand are ideated, and; *Empirical validation activity* (validation approach): which allows assessing whether the solution proposal is feasible for the problem, providing a room to extend the acquired knowledge.

[1] Our research was submitted and approved by the Ethics Committee Board.

The DSR framework includes the concept of Technological Rule (TR). TR describes the desired effect of a proposed solution in a particular context. It expresses the scope of validity of the solution and it helps researchers to determine practical (problem-solving) and theoretical (generalization) contributions of the research. Novelty is another element of the DSR framework [228], which represents a refinement of the technological rule to summarize the contributions achieved with the research.

The framework is grounded in the 3 cycles for Design Science proposed by Hevner (2007) [110], namely: Relevance cycle, Design cycle, and Rigor cycle. The Relevance cycle seeks to identify and understand the application context, the research problem, and the acceptance criteria that will evaluate the research results. The Design cycle contemplates the research activities that iteratively enable the construction and validation of artifacts. The Design Cycle also allows feedback collection to refine the design in the next iteration. The Rigor cycle aims to guarantee that the artifacts being produced are valid and contribute both to practice and theory. The Rigor cycle suggests the use of empirical methods for validating the research constructs and their application in a particular context.

3.2 Research Design

Figure 3.1 illustrates our DSR-based research methodology. We conducted the Design Science Research framework iteratively, starting with the Problem Understanding activity followed by 3 iterations of the Solution Design and Validation activities.

Figure 3.2 shows a timeline of our research design[2].

3.2.1 Problem Understanding Approach

Problem Understanding aims to explore, identify and conceptualize a relevant and novel research problem. Runeson *et al.* (2020) [228] argue that problem conceptualization is a typical activity when employing DSR since the method aims to address real problems as problem instances in a specific context. The authors assume that (Runeson *et al.*, 2020, p. 130):

> Understanding a general problem in terms of one or more concrete problem instances is a basis for understanding how this general problem may be solved. While exploring a specific problem instance, it may become clearer what the core of the problem is, thus focusing the potential solution design on these areas.

For the Problem Understanding, we conducted 2 exploratory studies: a Systematic Literature Mapping and a Survey with professionals from the agile software develop-

[2] The complementary material and artifacts generated for this research can be accessed in https://github.com/rafaelparizi/PHD_repository/blob/main/README.md

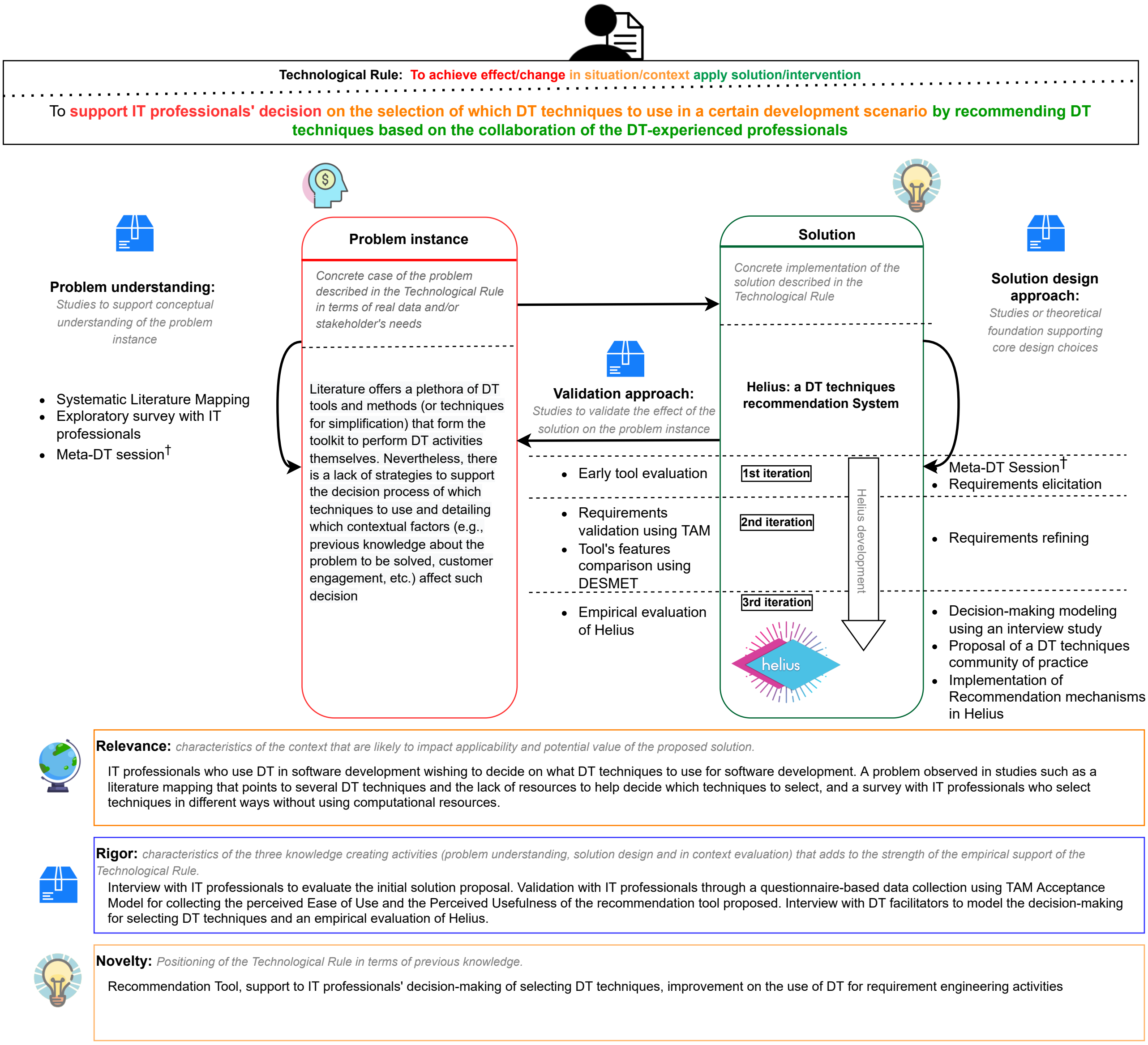


Figure 3.1: DSR method based on Runeson *et al.* (2020) [228]

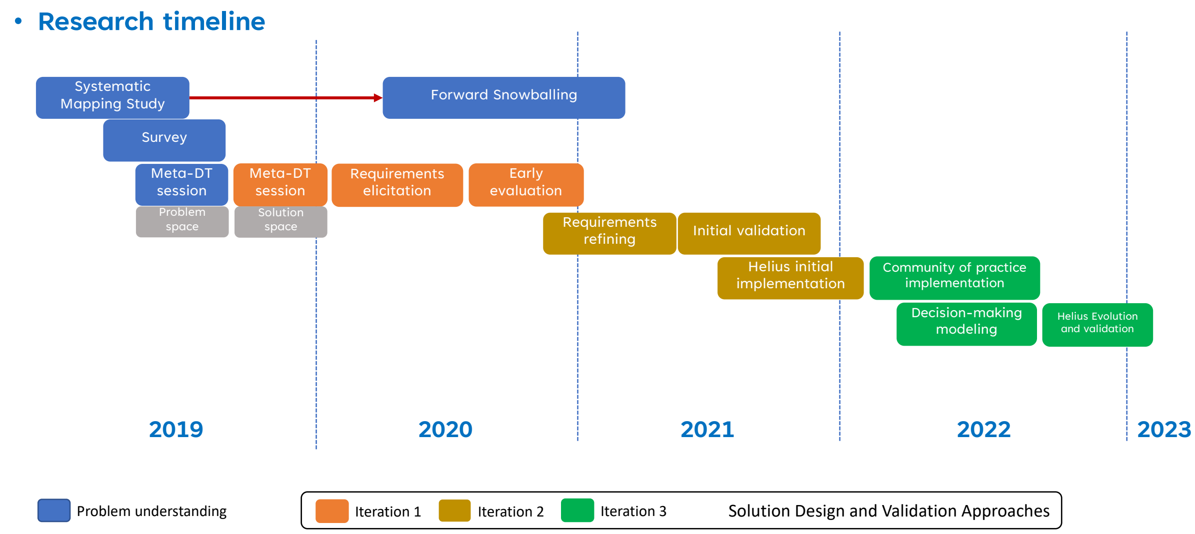


Figure 3.2: Research design - a timeline perspective

ment industry. Exploratory studies seek to find out what is happening, seek new insights and generate ideas and hypotheses for further research [229]. We also conducted a DT session to consolidate the results of our exploratory studies to establish a problem.

Systematic Literature Mapping

Systematic Literature Mapping is a procedure that follows a strict process for searching, selecting, and extracting data from publications, mapping in a broad view the existing evidence on a research question [204]. We performed our Systematic Literature Mapping following the guidelines proposed by Petersen *et al.* (2008) [204], aiming to identify publications that report the use of DT in software development. In addition, we performed a forward snowballing review to supplement our literature mapping [81].

Our literature mapping aimed to answer the following research question: "*How has DT been integrated into software development, what models and techniques are used, how are DT techniques selected, and what are the key points that DT practitioners should be aware of when using DT for software development?*". In Section 4.1, we detail our Systematic Literature Mapping study.

Survey

Looking for answers to know how industry professionals use Design Thinking in software development, we conducted a survey to seek a more in-depth understanding of the Brazilian software development community. Our survey is characterized as exploratory [206, 125], seeking an understanding of the phenomenon through the information collected. In Section 4.2, we detail our exploratory Survey with IT professionals.

Meta Design Thinking Session

Motivated by the findings that we obtained with the exploratory studies, we worked on the definition of a technological rule for our research problem: to support IT professionals' decision to select which DT techniques to use in a specific development scenario. Then, aiming to further the problem understanding, we performed a DT session. We named the session like "meta-DT" since we conducted a DT session to support the process of selecting DT techniques.

After the problem understanding activity was completed, we could determine the relevance of the research [228, 110]. We identified that our research is relevant to IT professionals who use DT in software development wishing to decide on what DT techniques to use for software development. The problem observed in our SLM points out that exist several DT techniques and there is a lack of resources to help decide which techniques to select. At the same time, our survey showed that there is no consensus on how IT professionals select techniques, lacking in considering the context information and the experience of other professionals. Once we defined the relevance of our research, we moved on to Solution Design and Solution validation activities. We performed these activities in 3 iterations, as it is described next.

3.2.2 Solution Design and Validation Approaches

Iteration 1: Solution Proposal and Early Evaluation

The first iteration of the Solution Design was embedded within the meta-DT session. It took place in the second half of the session, which referred to the solution space. Our goal was to propose an initial solution to the problem at hand. As a result, we proposed Helius, a DT techniques recommendation system. Next, still in the Solution Design activity, we focused on eliciting requirements for the recommendation tool solution.

Moving towards the Validation activity in the DSR framework, in Iteration 1 we performed an early evaluation of the initial artifact proposed. This evaluation aimed to evaluate our tool specification with industry practitioners to identify whether we were missing any relevant feature in the artifact.

Iteration 2: Solution Refining, Initial Validation and Initial Features Implementation

Motivated by the results of Iteration 1 and aiming to improve the artifact, we performed a second iteration in the DSR framework. We refined the requirements of the proposed solution and then we validated the solution with industry professionals.

We used the results of the early evaluation with DT practitioners (Iteration 1 - Validation Approach) and our defined artifacts, such as user journeys, service blueprints, and low-level prototypes (Iteration 1 - Solution Design Approach) as input to the Refining requirements activity. As a result, we obtained the macro-features and features that represent the requirements of our recommendation system.

Next, we moved forward to evaluate our DT techniques recommendation tool requirements. We conducted the validation activity by collecting data via a questionnaire based on the Technology Acceptance Model (TAM) [59]. TAM uses a Likert scale [7] and it allows evaluate users' perceptions of the acceptance of technology. The model considers the Perceived Ease of Use (PEU) and the Perceived Usefulness (PU) factors in assessing this acceptance. We present the results of the TAM-based validation in Section 6.2

We also compared the features that we proposed for Helius in Iteration 2 to features of two other related tools by using the DESMET method [131]. Using DESMET, we aimed to demonstrate that Helius innovates in the field. As a result, we figured out that

Helius advances in the recommendation of DT techniques for considering the combined use of techniques, for allowing the evaluation of the techniques used and for considering the user experience in the recommendations. Section 6.2 shows the results of the evaluation of the features using DESMET.

Still in Iteration 2 and inspired by the results we obtained through the validation activities (TAM and DESMET), we started to implement Helius using Flutter as the programming Language. Section 6.3 presents the first implementation of Helius.

Iteration 3: Solution Evolution and Validation

Next, we run the third iteration in our DSR-based methodology to improve Helius and empirically evaluate it. In Iteration 3, we conducted 3 activities in the solution design approach: i. proposal of a community of practice of DT techniques experiences, ii. modeling the decision-making of the DT techniques selection and, iii. implementation of the DT techniques recommendation mechanisms in Helius. In addition, moving to the validation approach, we conducted an empirical study with IT professionals to evaluate Helius, our DT techniques recommendation system.

We started Iteration 3 by proposing a community of practice of DT techniques as a resource for supporting DT practitioners in selecting DT techniques. This collaborative feature of Helius aims to help Helius' users to know about other professionals' experiences in DT techniques in software development (see Section 7.1).

Next, we conducted an interview-based study to model the DT facilitators' decision-making behind selecting DT techniques. We used Grounded Theory as the research method. Our goal was to collect insights to enable the recommendations provided in Helius. Therefore, once we proposed a descriptive decision-making model of the DT techniques selection and still in the solution design approach, we converted the decision-making strategies into recommendation mechanisms. We implemented such recommendation mechanisms into Helius. The mechanisms include non-personalized and personalized recommendations of DT techniques (see Section 7.2).

Finally, after implementing the recommendation mechanisms in Helius, we conducted an empirical study with DT practitioners to evaluate Helius. Section 7.3 details how we conducted the study and the results of its evaluation.

Next, Chapter 4 starts presenting the problem-understanding approach.

4. PROBLEM UNDERSTANDING APPROACH

This chapter presents the exploratory studies that we conducted aiming to identify a research problem related to the use of Design Thinking in software development. We conducted a Literature Mapping Study to characterize the state-of-the-art and also conducted a Survey with professionals in the software development industry to collect the state-of-the-practice of the use of DT for software development. In addition to these studies, we conducted a Design Thinking session to frame and define the problem constructs to be solved through this doctoral research.

4.1 Design Thinking in Software Development: A Literature Mapping

Given its iterative approach to problem-solving, DT has been integrated with agile methods for boosting software development [200, 160]. While DT fosters the understanding of the problem and the search for a solution that meets the user's needs, Agile methods focus on speed, simplicity, continuous and fast deliveries, frequent feedback, and quick reaction to changes [103].

Nevertheless, the nature of software development teams and structure associated with the lack of training on the design subject, and the number of DT models and techniques available, using DT becomes challenging. Therefore, it is important to investigate how DT has been used to support software development and what resources are available to meet user needs by delivering solutions that address the problem at hand.

Literature review studies such as the one by Souza *et al.* (2017) [65] represent research efforts for reporting the use of DT in software development aiming to help practitioners how to use DT. The authors evaluated 22 papers and mapped 11 models and 55 techniques of DT. Results also show that DT is a dynamic approach that does not define an order to its working spaces, allowing adaptation according to the context of the problem.

Waidelich *et al.* (2018) [265] performed a literature review and analyzed 35 documents, including journal papers, textbooks, and web documents in German and in English, to provide an overview of DT models used in practice, but not limited to software development. The authors conclude that there is no standardized model for use. They also pointed out that there is flexibility in the steps to be followed, indicating that studying which techniques can be used in each working space is a research possibility.

Pereira (2018) [200] presents a literature review that evaluated 29 papers between 2008 and 2017 to identify how the DT approach and Agile methods are integrated into the development process, which strategies are used, and which models exist to carry out this integration. The authors figured out that the DT integrated into Agile seeks further to capture users' needs in the early stages and ensure the usability of the software.

On the other hand, literature also presents that DT can not be considered a silver bullet. For instance, Pereira *et al.* (2021) [201] investigated not only the benefits of using DT in software development, but also the challenges that IT professionals might face with the use of DT. The authors conducted a focus-group study with 39 professionals from distinct companies' roles and pointed out that the use of DT requires attention to points such as the time pressure, the lack of participants' engagement, the resistance to applying DT, and the lack of empathy. De Paula, Amancio and Flores (2020) [64] discuss points and counterpoints of applying DT in the software industry. They figured out that in addition to knowing the right problem and identifying the appropriate solutions, the use of DT might include some risks such as a lack of participants' commitment, or a high effort to conduct DT activities that can be considered a waste of time. Therefore, IT professionals should be aware of some attention points when using DT in software development in order to explore it effectively.

Thus, this section aims to further the understanding of the use of DT in software development. Our goal is to identify not only the DT models and techniques, but also what are the DT integration strategies in systems development, what criteria are considered by professionals in making decisions about which DT techniques to use, and what are the points that the DT practitioners have to pay attention when deciding to use DT.

Our Systematic Literature Mapping aimed to answer the following Research Question (RQ): *How has DT been integrated into software development, what models and techniques are used, how are DT techniques selected, and what are the key points that DT practitioners should be aware of when using DT?*"[1].

This procedure includes the following steps that support the identification, evaluation, and interpretation of available papers for an established research topic:

- Research protocol definition:
  - Definition of research questions;
  - Definition of the search strings;
  - Definition of the inclusion and exclusion criteria for selecting publications;
- Search execution:
  - Execution of automatic search in digital libraries;
- Studies selection, data extraction and results analysis:
  - Selection of the studies using the inclusion and exclusion criteria;

[1]This study resulted in a publication in the Journal of Systems and Software. Ref.: [195]

- Data extraction from the selected studies;
- Data analysis and results presentation.

4.1.1 Study Design

We followed the guidelines for Literature mappings proposed by Petersen *et al.* (2008) [204], aiming to identify publications that report the use of DT in software development (Item A). In addition, we performed a forward snowballing review as a way to supplement our literature mapping as reported by Felizardo *et al.* (2016) [81] (Item B).

A - Systematic Literature Mapping

Figure 4.1 illustrates the process that we followed in our SLM, which was composed of 7 activities (activities i to vii). We started by defining a research protocol for our mapping study. The protocol is composed of the research questions, the search approach, and the criteria to include or exclude publications. Then, starting from the main RQ, we derived the research questions (activity i):

RQ1. What strategies for integrating DT in software development have been adopted?

RQ2. What DT models are used in software development?

RQ3. What DT techniques are used in software development?

RQ4. What is reported about the selection of DT techniques in software development?

RQ5. What are the key points to be aware of when using DT in software development?

We formulated the search strings (activity ii) using the keywords strategy, as defined by Petersen *et al.* (2015) [205]. We used Software Development and Design Thinking as 2 keyword categories (see Table 4.1). The keywords in each category were combined

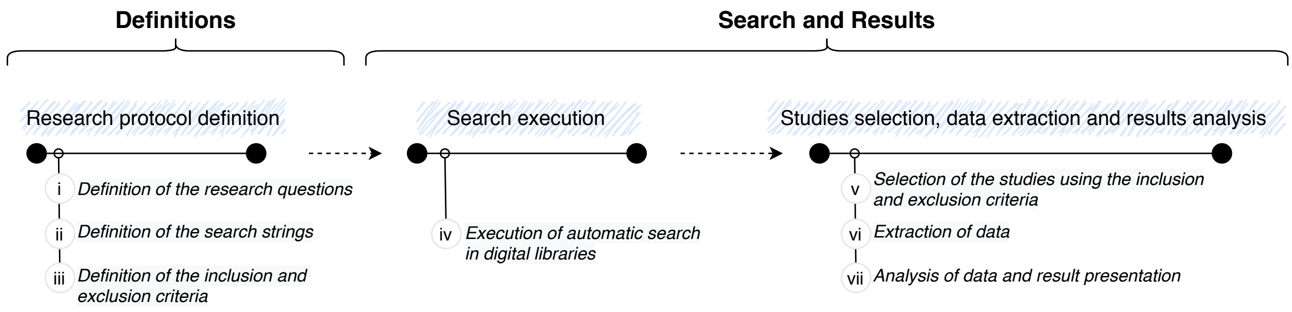


Figure 4.1: Systematic Literature Mapping activities

with a Boolean operator “OR”, and the categories were combined using a Boolean operator “AND”. The search string defined was:

(("software engineering") OR ("software development") OR ("software industry") OR ("software construction") OR ("software project") OR ("software process") OR ("Software project management")) AND (("design think*") OR ("design session")).

Table 4.2 shows the criteria that we used for selecting relevant publications. This study focuses on publications reporting DT usage in software development (Inclusion Criteria 1 – IC1). The exclusion criteria allow us to not include publications not available or duplicated, publications not written in English (EC2, EC3, and EC4), or publications not peer-reviewed (EC5).

Next, we performed automatic searches for publications in August 2020[2] on the following digital libraries (activity iv): ACM Digital Library[3], IEEE Xplore[4], Science Direct[5], Scopus[6], Springer Database[7], and Wiley Online Library[8]. The total of retrieved publications was 3,386 (see Table 4.3).

Table 4.1: Keywords used in the SLM

Category	**Keywords**
Software Development	software engineering
	software development
	software industry
	software construction
	software project
	software process
	software project management
Design Thinking	design thinking
	design session

Table 4.2: Inclusion and Exclusion Criteria for the SLM

Type		**Description**
Inclusion	IC1	publications related to DT in software development
Exclusion	EC1	Publications that do not attend the IC1, i.e., that do not discuss DT in software development
	EC2	Duplicated publications
	EC3	Publications not available for download
	EC4	Publications not written in English
	EC5	Publications that were not peer reviewed

[2]We run the automatic searches on the digital libraries in August 2020. Then, aiming to supplement the set of publications about the use of DT in software development, we performed a forward snowballing as suggested by [81].

[3]https://dl.acm.org/

[4]https://ieeexplore.ieee.org

[5]https://www.sciencedirect.com/

[6]https://www.scopus.com

[7]https://link.springer.com/

[8]https://onlinelibrary.wiley.com/

Table 4.3: Publications retrieved from the digital libraries

Search Engine	Result (# of Publications)
ACM	159
IEEE Xplore	592
Science Direct	295
Scopus	1592
Springer	551
Wiley	197
Total	**3,386 publications**

Figure 4.2 shows the publications' selection process (activity v). We performed the steps: 1) exclusion of duplicated, not available for download, not written in English, or not peer-reviewed publications, 2) evaluation of agreement using Cohen's Kappa coefficient [141], 3) reading the title, the abstract, and the keywords, and 4) reading the publications' full-text.

1st step: Exclusion of duplicated, not available, not written in English or not peer-reviewed publications

In the first step of the publications' selection process, we worked on removing duplicated publications by using the StArt tool[9], which provides technical support for the systematic research process. We also excluded publications not peer-reviewed, not available for downloading, and publications not written in English, according to criteria EC2 to EC5, respectively. In this first step, we removed 299 publications remaining 3,087 publications. The exclusion rate of publications was 8.83%.

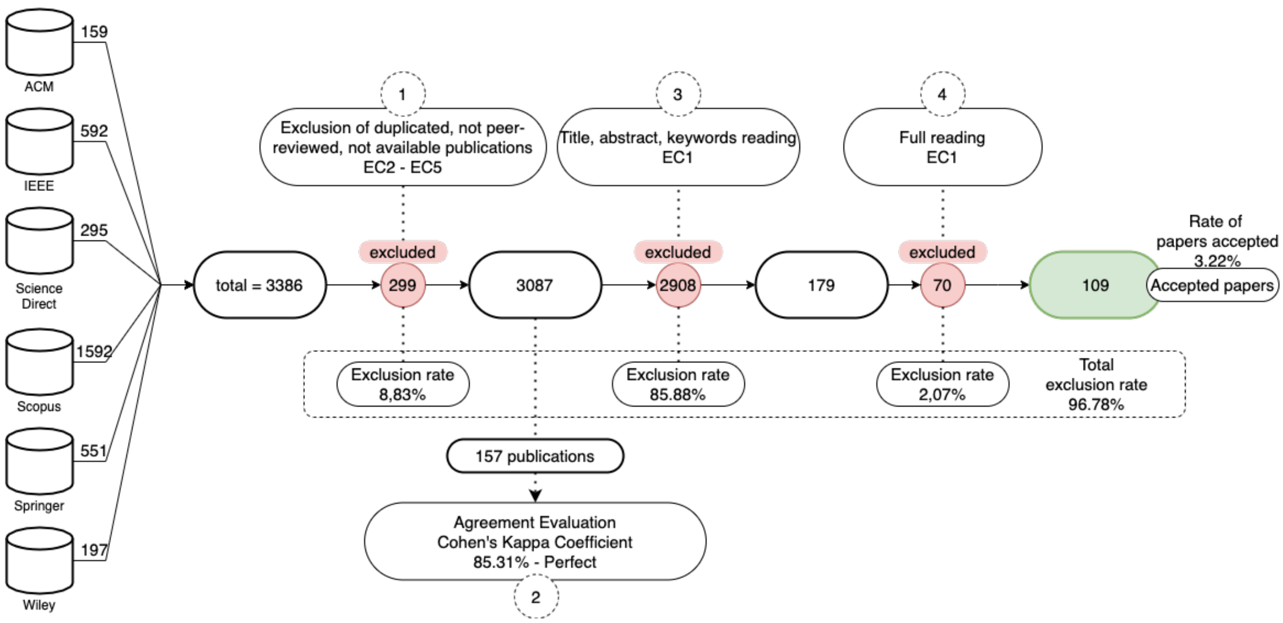


Figure 4.2: Systematic mapping study publications' selection process

[9]http://lapes.dc.ufscar.br/tools/start_tool

2nd step: Agreement evaluation using Kappa coefficient

In the second step, we performed an agreement evaluation using Cohen's Kappa coefficient [141]. Kappa provides a coefficient for estimating the degree of agreement between two reviewers [259]. Table 4.4 presents Kappa's agreement values.

In this step, we selected 5% of the publications (157 publications) to compute Cohen's Kappa agreement coefficient. The publications were selected randomly and 2 researchers examined them. Through this step, we obtained an agreement coefficient of 0.8531, considered perfect according to Kappa's agreement indexes. Table 4.5 illustrates the Kappa coefficient result. Researcher 1 accepted 33 publications and rejected 124 publications, while researcher 2 accepted 37 and rejected 120 publications, resulting in a perfect degree of agreement. Thus, we proceeded with publication selection considering the inclusion and exclusion criteria, based on Cohen's Kappa coefficient result.

3rd step: Reading the title, abstract and keywords

In this step, we performed the publications' selection by reading the title, keywords, and abstract. We read the abstract of the 3,087 publications and applied the IC1 or EC1 criteria. As a result, we excluded 2,908 publications that did not satisfy the inclusion and exclusion criteria, remaining 179 publications of the total retrieved publications from the automatic searches. The exclusion rate was 85.88%.

4th step: Reading the full text

In this step, we read the full text of the 179 remaining publications. We excluded 70 publications that did not correspond to DT in software development, resulting in an ex-

Table 4.4: KAPPA Coefficient Indexes [263]

Kappa	**Agreement description**
< 0	Without agreement
0.01 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 0.99	Perfect agreement

Table 4.5: KAPPA Coefficient result for the SLM

	Publications Accepted	**Publications Rejected**	**KAPPA Coefficient**
Author 1	33	124	0.8531
Author 2	37	120	Perfect agreement

clusion rate of 2.07%[10]. Thus, the publications' selection process resulted in 109 selected publications. The total exclusion rate was 96.78%, and the acceptance rate was 3.22%.

B - Forward Snowballing

Figure 4.3 shows the forward snowballing procedure that we followed. Felizardo *et al.* (2016) [81] argue that forward snowballing is an alternative procedure for updating literature review studies. We performed the forward snowballing procedure in 3 iterations. We used the set of 109 publications selected in the SLM as the start seed. So, we looked for citations from each of the 109 publications in the start seed. We used Google Scholar[11] to find citations for each paper.

We downloaded the citations and analyzed each one according to the inclusion and exclusion criteria (see Table 4.2). We started the selection process by excluding duplicated publications, not peer-reviewed, not written in English or not available for download. The remaining publications were analyzed by title, abstract, and keywords.

Iteration 1

In iteration 1, we found a total of 1,387 citations for the 109 publications of the starter seed. Initially, we excluded 967 citations using the EC2-EC5 criteria, remaining 420 citations. Then, we read the title, abstract and keywords of the 420 citations. We excluded more 404 publications using EC1. Next, we did the full reading and accepted 16 publications in iteration 1.

Iteration 2

In iteration 2, considering as the start seed the 16 publications that we selected in iteration 1, we found 82 citations. Initially, we excluded 58 citations using the EC2–EC5 exclusion criteria, remaining 26 citations. Then, we read the title, abstract and keywords of the 26 citations. We excluded more 24 publications using EC1. Next, we did the full reading and accepted 2 new publications in iteration 2.

Iteration 3

In iteration 3, considering as the start seed the 2 publications that we selected in iteration 2, we did not find citations. So, we ended the forward snowballing iterations. Therefore, we have selected 127 publications in our mapping study. Section 4.1.2 presents the results we obtained performing the data extraction of the selected publications.

[10] The values of percentage take into account the total of retrieved publications.
[11] https://scholar.google.com

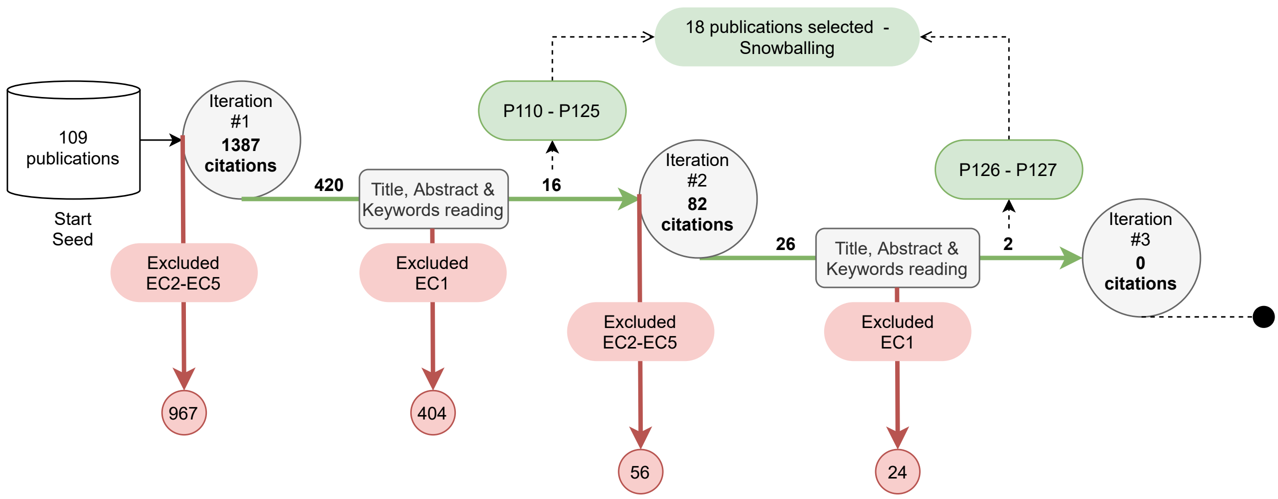


Figure 4.3: Forward snowballing iterations

4.1.2 Literature Mapping Results and Discussions

This section presents the results of this Systematic Literature Mapping. Initially, we show the results' metadata that we extracted from the selected publications. Next, we discuss each research question that we posed in Section 4.1.1.

Tables 4.6 and 4.7 list the 109 accepted publications through the SLM, and Table 4.8 shows the 18 publications that we selected through the forward snowballing procedure. Both tables include the publication's ID, classification (Book chapter, Conference paper, or Journal article), title, and authors. We use the publication ID to identify the publication in the analysis of the results. We also classified the selected publications in empirical and non-empirical research [55], as follows:

- Empirical research: publications with methodological procedures applied in a real context, such as:
  - Academic context: students participated in the study in an academic context;
  - Industrial context: the paper was developed in an industrial environment;
  - Innovation context: the paper was developed for social innovation;
- Non-empirical research: publications discussing theoretical and philosophical aspects, not applied in a practical context.
  - Theoretical context: the publication was not applied to any context.

Results Metadata

Figure 4.4 shows the publications' frequency by year related to DT in software development, categorized as book chapters, conference publications, and journal articles.

Table 4.6: Selected publications in the SLM

ID	C	Title	Authors
001	○	Integrating the Design Thinking into the UCD's methodology	[97]
002	◍	A heuristic approach for supporting innovation in requirements engineering	[78]
003	○	Design thinking for search user interface design	[20]
004	○	Increasing Kenyan Open Data Consumption: A Design Thinking Approach	[180]
005	●	From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking	[261]
006	○	Design thinking methodology for the design of interactive real-time applications	[230]
007	○	Fast train to DT: A practical guide to coach design thinking in software industry	[111]
008	○	Reframed contexts: Design thinking for agile user experience design	[2]
009	○	CoDICE: Balancing software engineering and creativity in the co-design of digital encounters with cultural heritage	[76]
010	◍	Design thinking: Expectations from a management perspective	[224]
011	●	From product development to innovation	[169]
012	○	Guiding novice database developers in database schema creation	[4]
013	○	Lean UX - The next generation of user-centered Agile development	[152]
014	○	Design thinking for usability evaluation of cloud platform service	[123]
015	○	The Role of Design Thinking and Physical Prototyping in Social Software Engineering	[185]
016	○	Trends in the Use of Design Thinking for Embedded Systems	[13]
017	○	Eliciting Requirements Using Personas and Empathy Map to Enhance the User Experience	[82]
018	○	Design thinking methods and tools for innovation	[48]
019	◍	A Brief Introduction to Design Thinking	[157]
020	○	Design Thinking Framework to Enhance Object Oriented Design and Problem Analysis Skill in Java Programming Laboratory: An Experience	[221]
021	○	Can Metamodels Link Development to Design Intent?	[89]
022	○	Aligning healthcare innovation and software requirements through design thinking	[44]
023	○	IBM design thinking software development framework	[156]
024	●	LODPRO: learning objects development process	[219]
025	●	Models as bridges from design thinking to engineering	[253]
026	○	OnTimeCargo: A smart transportation system development in logistics management by a design thinking approach	[17]
027	○	Pet empires: Combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment	[63]
028	○	The origins of design thinking and the relevance in software innovations	[119]
029	○	An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation	[103]
030	◍	Are We Ready for Disruptive Improvement?	[227]
031	◍	Communication Breakdowns in the Integration of User-Centred Design and Agile Development	[28]
032	○	Increasing the Quality of Use Case Definition Through a Design Thinking Collaborative Method and an Alternative Hybrid Documentation Style	[168]
033	○	Developing High-Performing Teams: A Design Thinking Led Approach	[127]
034	○	Embedded Design Thinking in Co-Design for Rapid Innovation of Design Solutions	[1]
035	◍	From the Real to the Virtual: Developing Improved Software Using Design Thinking	[164]
036	○	Design Thinking Framework for Project Portfolio Management	[232]
037	○	The Use of Design Thinking in Agile Software Requirements Survey: A Case Study	[40]
038	●	Framing Design Thinking: The Concept in Idea and Enactment	[43]
039	●	Applying design thinking methods to ecosystem management tools: Creating the Great Lakes Aquatic Habitat Explorer	[98]
040	○	Promoting the Elicitation of Usability and Accessibility Requirements in Design Thinking: Using a Designed Object as a Boundary Object	[145]
041	○	Question-answer analysis in design thinking at the conceptual stage of developing a system with a software	[244]
042	○	The Agile Manifesto, design thinking and systems engineering	[57]
043	○	Hackathons, semesterathons, and summerathons as vehicles to develop smart city local talent that via their innovations promote synergy between industry, academia, government and citizens	[16]
044	○	An entrepreneurial narrative media-model framework to knowledge building and open co-design for smart cities	[143]
045	○	The Students' Perspectives on Applying Design Thinking for the Design of Mobile Applications	[260]
046	○	Infusing Design Thinking into a Software Engineering Capstone Course	[191]
047	○	Identifying Design Features Using Combination of Requirements Elicitation Techniques	[178]
048	○	FATHOM: TEL Environment to Develop Divergent and Convergent Thinking Skills in Software Design	[223]
049	○	A2BP: A method for ambidextrous analysis of business process	[231]
050	○	Coupling design thinking, user experience design and agile: Towards cooperation framework	[184]
051	○	Design thinking methods and techniques in design education	[134]
052	○	Introducing 'Human-Centered Agile Workflow' (HCAW) – An Agile Conception and Development Process Model	[94]
053	○	The best of three worlds -The creation of innodev a software development approach that integrates design thinking, scrum and lean startup	[67]
054	○	Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation	[51]
055	○	The Importance of Empathy for Analyzing Privacy Requirements	[147]
056	○	Policy Recommendations to Induce Behavioural Changes through Interactive Energy Visualization	[178]
057	○	A first implementation of a design thinking workshop during a mobile app development course project	[207]

*Continue on the next page

Paper Category(C): ○ Conference proceedings | ◍ Book chapter | ● Journal article
Research Type (T): Empirical -{000 Academic | 000 Industry | 000 Innovation} | Non-empirical -000 Theoretical

Table 4.7: Selected Papers (continued from the previous page)

ID	C	Title	Authors
058	○	Adopting design thinking practices to satisfy customer expectations in agile practices: A case from Sri Lankan software development industry	[210]
059	○	CPM / PDD in the context of Design Thinking and Agile Development of Cyber-Physical Systems: Use cases and methodology	[158]
060	○	Designing human-centric information systems: Towards an understanding of challenges in specifying requirements within design thinking projects	[109]
061	○	Educating for empathy in software engineering course	[146]
062	●	Juicing the game design process: towards a content centric framework for understanding and teaching game design in higher education	[142]
063	○	Adding Scrum-style project management to an advanced Design Thinking class	[72]
064	○	Software 4.0: 'How' of building 'Next-Gen' systems	[198]
065	○	The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems	[108]
066	○	Visual analytics for cyber-physical systems development: Blending design thinking and systems thinking	[102]
067	◍	Using Design Thinking for Requirements Engineering in the Context of Digitalization and Digital Transformation: A Motivation and an Experience Report	[42]
068	●	Effective Design Methodologies	[15]
069	●	Integrating cell and molecular biology concepts: Comparing learning gains and self-efficacy in corresponding live and virtual undergraduate laboratory experiences	[99]
070	●	Advanced agile approaches to improve engineering activities	[39]
071	○	The Product Backlog	[234]
072	○	Dual-track agile in software engineering education	[216]
073	○	Design thinking in practice: Understanding manifestations of design thinking in software engineering	[68]
074	○	Design Thinking and Acceptance Requirements for Designing Gamified Software	[208]
075	○	Design Thinking in a Nutshell for Eliciting Requirements of a Business Process: A Case Study of a Design Thinking Workshop	[148]
076	○	A Step by Step Methodology for Software Design of a Learning Analytics Tool in Latin America: A Case Study in Ecuador	[189]
077	○	Definition of Indicators in the Execution of Educational Projects with Design Thinking Using the Systematic Literature Review	[9]
078	●	How digital transformation can influence business model, Case study for transport industry	[91]
079	●	Creating an innovative digital project team: Levers to enable digital transformation	[100]
080	●	Implementing Experience Sampling Technology for Functional Analysis in Family Medicine – A Design Thinking Approach	[56]
081	●	Design thinking: Challenges for software requirements elicitation	[167]
082	○	Towards applying design-thinking for designing privacy-protecting information systems	[18]
083	○	Design Thinking's Resources for in-situ Co-Design of Mobile Games	[46]
084	○	A Lean Design Thinking Methodology (LDTM) for Machine Learning and Modern Data Projects	[5]
085	○	Combining challenge-based learning and design thinking to teach mobile app development	[88]
086	○	A Hackathon Methodology for Undergraduate Course Projects	[87]
087	○	SMARTD Web-Based Monitoring and Evaluation System	[38]
088	○	Design Thinking and Scrum in Software Requirements Elicitation: A Case Study	[32]
089	○	Mobile application based on design thinking for teaching kinematics	[14]
090	○	CALDET: A TRIZ-Driven Integrated Software Development Methodology	[31]
091	●	Challenges in Requirement Engineering: Could Design Thinking Help?	[124]
092	○	Design Thinking Versus Design Sprint: A Comparative Study	[12]
093	●	A Value-Centered Approach for Unique and Novel Software Applications	[237]
094	○	Innodeck: Card based innovation support - A modular human-centered approach to facilitate innovation workshops	[104]
095	●	Using Agile Approaches to Drive Software Process Improvement Initiatives	[182]
096	●	Integrating design thinking into extreme programming	[243]
097	○	Integrating Design Thinking into Scrum Framework in the Context of Requirements Engineering Management	[8]
098	○	A LX (learner experience)-based evaluation method of the education and training programs for professional software engineers	[126]
099	○	Design Thinking Approach for Mobile Application Design of Disaster Mitigation Management	[249]
100	●	Design Thinking in Industry	[64]
101	●	Three Phases of Transforming a Project-Based IT Company Into a Lean and Design-Led Digital Service Provider	[140]
102	●	On Integrating Design Thinking for Human-Centered Requirements Engineering	[107]
103	●	Migrating a Software Factory to Design Thinking: Paying Attention to People and Mind-Sets	[162]
104	●	When Does Design Help Thinking, and When Does Design Thinking Help?	[199]
105	●	Designing mangrove ecology self-learning application based on a micro-learning approach	[45]
106	●	Embracing Quality with Design Thinking	[251]
107	●	Operationalizing Design Thinking in Business Intelligence and Analytics Projects	[50]
108	●	"StoryWeb": A storytelling-based knowledge-sharing application among multiple stakeholders	[197]
109	◍	Inherent Mapping Analysis of Agile Development Methodology Through Design Thinking	[161]

Paper Category(C): ○ Conference proceedings | ◍ Book chapter | ● Journal article
Research Type (T): Empirical -{000 Academic | 000 Industry | 000 Innovation} | Non-empirical - 000 Theoretical

Table 4.8: Selected publications through a forward snowballing

ID	C	Title	Authors
110	○	Experimenting with design thinking in requirements refinement for a learning management system	[85]
111	○	Promoting creativity and innovative thinking in software engineering teaching: a case study	[96]
112	●	An experimental study of the use of design thinking as a requirements elicitation approach for mobile learning environments	[62]
113	●	Systems thinking approach to implementing kanban: A case study	[236]
114	○	Empowering Project Managers in Enterprises-A Design Thinking Approach to Manage Commercial Projects	[136]
115	●	Supporting the teaching of design thinking techniques for requirements elicitation through a recommendation tool	[245]
116	●	Technique for representing requirements using personas: a controlled experiment	[83]
117	●	A Brief Study on Enhancing Quality of Enterprise Applications using Design Thinking	[61]
118	●	Innovation through Design Thinking, User Experience and Agile: Towards Cooperation Framework	[183]
119	●	DT@IT Toolbox: Design Thinking Tools to Support Everyday Software Development	[71]
120	○	Design Thinking Use in Agile Software Projects: Software Developers' Perception	[41]
121	○	User Experience Design for Disaster Management Mobile Application using Design Thinking Approach	[66]
122	●	InnoDev: a software development methodology integrating design thinking, scrum and lean startup	[70]
123	○	InnoDev Workshop: A One Day Introduction to Combining Design Thinking, Lean Startup and Agile Software Development	[69]
124	○	Requirement Engineering and the Role of Design Thinking	[114]
125	○	The design thinking of co-located vs. distributed software developers: distance strikes again!	[121]
126	●	Designing a Persuasive Application for Behaviour Change with Children	[214]
127	○	A Novel Perspective to Threat Modeling using Design Thinking and Agile Principles	[60]

Publication Category(C): ○ Conference proceedings | ● Book chapter | ● Journal article
Research Type (T): Empirical -{000 Academic | 000 Industry | 000 Innovation} | Non-empirical -000 Theoretical

Most of them were published in 2019 (29 publications), followed by 2016 and 2020 (20 publications), and 2018 (19 publications). We also classified the publications following the classification proposed by Wieringa *et al.* (2006) [269], such as:

- Evaluation research: publications that describe results from the investigation of a problem in practice, showing the use of a method in practice.
- Proposal of a solution: publications that propose a solution technique and argue about its relevance without complete validation. The technique must be new or at least an improvement on an existing technique.
- Validation research: publications that investigate the properties of a proposed solution that has not yet been implemented in practice. It also includes publications that involve experiments, simulation, prototypes or mathematical analysis.
- Philosophical publications: publications that present discussions about ways to understand the phenomenon conceptually.
- Opinion publications: publications that have the author's opinion about what is wrong or right about a given topic.
- Personal experience publications: publications that report experiences from applying some method or technique, including lessons learned.

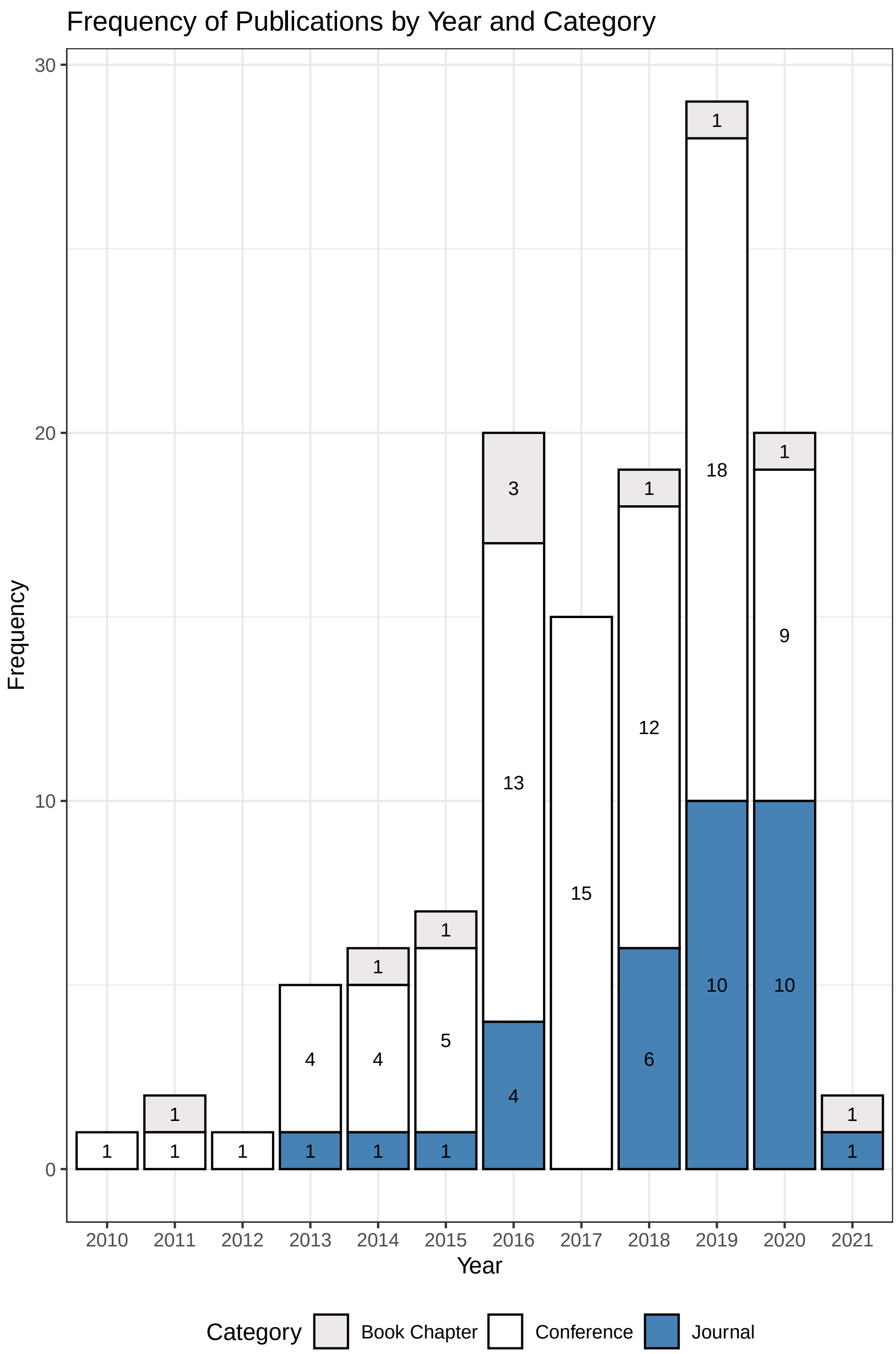


Figure 4.4: Frequency of publications x Year of each category of publications

Wieringa *et al.* (2006) [269] mention that the same paper can be classified in more than one class. Following this classification, we correlated each paper according to the 2 research types, including empirical research in the academic, industry, or innovation contexts, and non-empirical research in a theoretical context. Figure 4.5 shows that most of the studies were performed as empirical research in the industry context for evaluation purposes (33 publications), followed by experience reports in an academic context (21 publications) and proposals of a solution in the industry (20 publications).

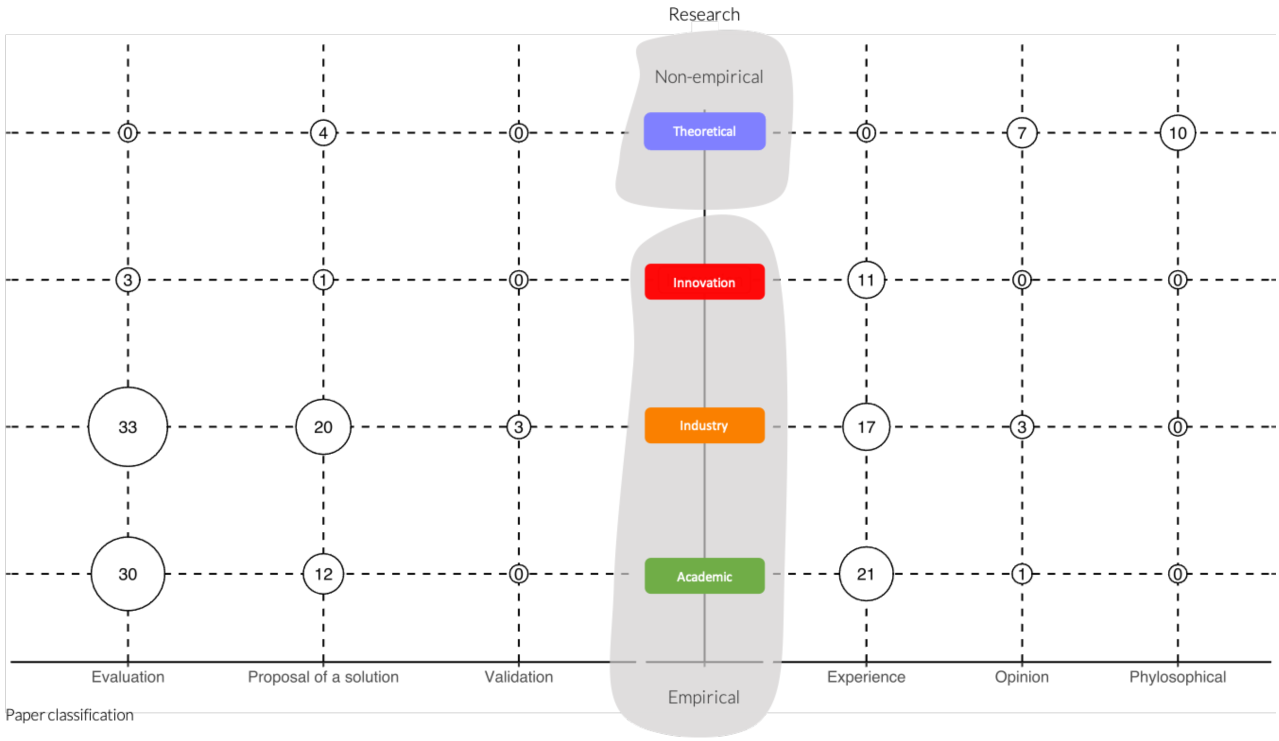


Figure 4.5: Publications' classification x research type

Results

RQ1: What strategies for integrating DT in software development have been adopted?

This research question classifies the use of DT integrated into software development following the integration strategies defined by Hehn *et al.* (2020) [107]:

1. Upfront DT: DT is considered a starter activity of a software project. It serves to understand the customer and to identify features to be implemented in the software;

2. Infused DT: DT is considered a toolbox for supporting existing RE activities;

3. Continuous DT: DT is seen as a whole integrated approach in software development, involving the customers along the full development cycle.

Figure 4.6 illustrates the 3 integration strategies of DT in software development. We mapped the integration strategies for each research context: empirical research in the Academic, Industry and Innovation context and non-empirical research in Theoretical publications. It is worth mentioning that we do not include in the plot the paper which introduced the strategies (e.g, Hehn *et al.* (2020) [107]).

The results show Upfront DT is the most cited strategy for integrating DT into software development. It means that DT supports the teams in discovering the users' needs and validating the candidate solution proposals at the beginning of the Requirement

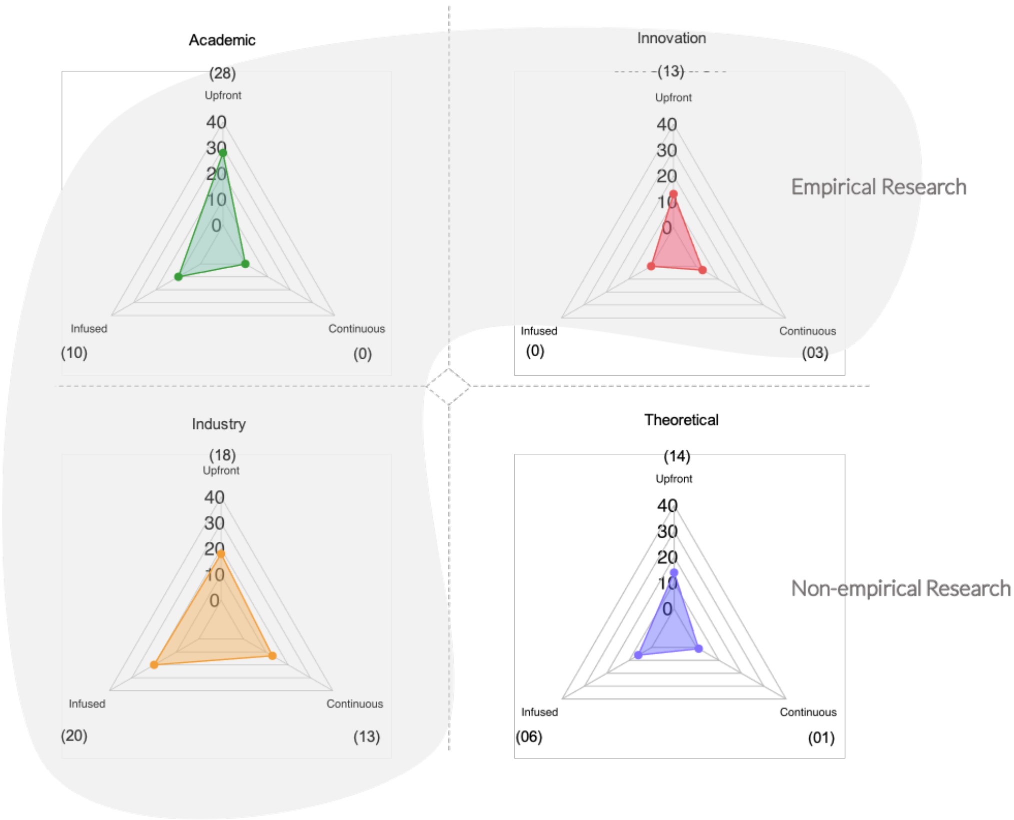


Figure 4.6: Strategies for integrating DT in software development

Engineering activities in a software development process. The studies using DT in an Upfront strategy consider it to understand and define the problem, ideate it into several candidate solution ideas, prototype the candidate solutions, and get feedback from the stakeholders. For instance, Dobrigkeit and De Paula (2019) [68] performed an interview-based study and collected the team's perceptions about DT in software development. The interviewees reported that they use DT as a pre-development phase for understanding the problem, gathering users' needs proposing solutions and validating prototypes before starting the development process itself.

Peráire (2019) [216] used the Infused DT strategy for integrating DT in a Dual-track Agile approach, combining Human-Computer Interaction and Requirements Engineering in a course with students for software development. The authors used practices such as ideation and experimentation to generate alternatives of viable interaction design concepts that can satisfy stakeholders' needs.

Mahe *et al.* (2019) [162] used the Continuous DT integration strategy. The authors applied DT to change a company's mindset. They applied DT to a company aiming to modify the traditional software development methodology that had been used and that had been resulting in difficulties such as the reduction of employees' motivation, lack of communication between consultants and developers, and poor requirements elicitation. Thus, taking DT as a continuous strategy, the company improved the requirements elicitation processes. It helped consultants and analysts to identify the most relevant features, improve the relationship between customers and managers and company consultants, foster team engagement, promote employee creativity, encourage a strong collaboration, and lead to a “fail fast, learn fast” philosophy.

RQ2: What DT models are used in software development?

In this question, we show the DT models that we found in the literature. A DT model represents a set of DT working spaces, encompassing activities that provide a way to start from the problem space and to evolve to the solution space [68]. Brown and Wyatt (2010) [37] advocate that the DT process is best thought of as a system of overlapping spaces rather than a sequence of orderly steps. Table 4.9 shows the DT models that we identified in our Systematic Mapping Study, considering the number of working spaces themselves, the set of working spaces, and the studies that mentioned them.

Table 4.9: Design Thinking Models, Working Spaces and Publications

WSs	**Model**	**Working Spaces**	**Publications**
2	DCIDT	Divergent - Convergent	008
3	Brown	Inspiration - Ideation - Implementation	045 046 085 086 089 051 002 078 073 103 115 008
	Souza e Silva	Immersion - Ideation - Prototyping	017 112 077 088
	Daniëls et. al	Understand - Explore - Materialize	080 087
	Codice	Resources - Ideas - Design Products	009
4	Dunne and Martin	Generate Ideas (abduction) - Predict Consequences (deduction) - Test - Generalize(Induction)	008
	IBM DT	Understand - Explore - Prototype - Evaluate	043 023 050 118
	HCAW	Research - Ideation - Prototyping - Evaluation	052
	Double Diamond	Discover - Define - Develop - Deliver	061 083 113 102 106 055
	Luchs	Discover - Define - Create - Evaluate	019
5	d.school	Empathy - Define - Ideate - Prototype - Test	011 014 020 054 057 076 104 099 107 116 004 028 036 038 049 050 058 065 066 067 073 075 091 096 098 101 109 124 018 022 039 105 120 126 016 035 041 042 068 082 097 106 117 127
	Meinel and Leifer	- (Re) define - Needfinding and Synthesis - Ideate - Prototype - Test	060 033 034
6	Driving board	Approach - Develop - Present and Provoke - Explore - Reflect - Escape	015
	Hasso Plattner Institute (HPI)	Understand - Observe Point of view - Ideate - Prototype - Test	063 052 059 070 073 114 123 030 122 003 029 092 053
	Nordstrom (de Paula & Araújo)	Define - Observe - Form Insights - Frame Opportunities - Brainstorm - Experiment	027
7	Hiremath & Sathyiam	Scoping - Research - Synthesis - Ideation - Prototyping - Validation - Implementation	007
	Sandino	Define - Explore - Ideate - Prototype - Select - Implement - Review	006 037

Research type: Empirical{000 Academic 000 Industry 000 Innovation} | Non-empirical - 000 Theoretical
* Wss = Number of Working Spaces

 # DT model with 2 working spaces

Divergent-Convergent Inquiry-based Design Thinking model (DCIDT) is a DT model that organizes DT exploring divergent and convergent spaces associated with 2 activities: inquiry and questioning. Adikari, McDonald and Campbell (2013) [2] proposed DCIDT integrated with a method called A2BP to systematize the analysis phase of the business process modeling. The method allows the exploration of internal and external business process opportunities.

 # DT models with 3 working spaces

Brown's DT model [35] was the second most cited DT model in literature. Brown's DT model has 3 working spaces: Inspiration, Ideation, and Implementation. Inspiration is a working space that helps to understand the problem, the needs, and the challenges of end-users. Ideation is a working space that allows analyzing previously collected data to turn it into ideas to prototype the solution. Implementation is a working space performed to test the generated prototypes and collect user feedback.

Souza and Silva DT model [246] contains 3 working spaces: Immersion, for capturing and processing users' information; Ideation, working on the ideas most referenced in the information provided by users, enabling the identification of requirements and characteristics of the proposed solution, and Prototyping, focused on transforming the requirements and characteristics into a real solution, and on verifying with the user whether requirements match the users' needs. Ferreira, Conte and Barbosa (2015) [82] used this set of working spaces to present Pathy, a technique for exploring empathy, and Braz *et al.* (2019) [32] applies DT with Immersion, Ideation, and Prototyping working spaces integrated to Scrum, focusing on requirements elicitation activities.

Daniëls DT model [56] represents DT into Understand, performed to empathize with users and define the problem, Explore, ideate solutions and prototype them, and Materialize, to test and evaluate the solution created. The model was used to redesign an e-health application. Budiarto *et al.* (2018) [38] applied a similar model to develop an application for the Indonesian government's agriculture sector.

CoDICE (Codesigning DIgital Cultural Encounters) DT model [76] organizes Design Thinking into 3 working spaces to co-design smart objects for enhanced encounters with cultural heritage to contribute to a joint European project. CoDICE's DT model working spaces are: Resources, or Situated Resources Gathering, a working space where co-designers can collect useful or inspirational material while visiting the physical environment where the digital artifact is going to be deployed; Ideas, or Divergent Inquiry and Ideation, aimed at generating ideas; and Design Products, or Convergent Design, focused on producing solutions for selected ideas.

 # DT models with 4 working spaces

Dune and Marting (2016) [75] suggest a DT model with 4 working spaces: Generate Ideas, Predict Consequences, Test and Generalize. It explores abductive, deductive, and inductive thinking in addition to testing. The authors integrated DT into a framework that enhances the user experience's current design by integrating three design approaches - DT, design for the user experience, and agile software development.

IBM DT model is a 4-working spaces DT model that provides a framework to write requirements, organize teams, and track project progress, including constant end-user feedback [156]. This model comprises the following working spaces: Understand, which establishes empathy with users to help to understand them and their problems' context; Explore, which focuses on the generation of new innovative ideas; Prototype, which generates artifacts intended to answer questions to solve the problem; and Evaluate, which solicits users for feedback about the prototype created.

IBM DT introduces the hill concept, creating a new way to express user needs into project requirements. Each hill articulates a clear objective and contains a defined scope to be achieved in a release and must be written to solve a specific and clearly defined user problem. The hills also describe intersections between user expectations and business requirements and consist of: "Who": Describes a specific user; "What": describes a problem that needs to be resolved; and "How": a measurable target resulting from the completion of the hill [156].

HCAW (Human-centered Agile Workflow) model [94] organizes the working spaces in cycles, where an initial Cycle Conception is proposed and organized into a Research phase that starts by receiving from business personnel a problem scenario used by the team to research to understand the problem better. This understanding is recorded in an Insight Report, the foundation for the Ideation phase, in which a Co-Creation Workshop is hosted to generate ideas that are later prototyped in the Prototyping phase and evaluated in the Evaluation phase.

Double Diamond DT model organizes DT in 4 working spaces: Discover, Define, Develop and Deliver [52]. Challiol *et al.* (2019) [46] explain that Discover and Develop are the 2 working spaces exploring divergent activities since they allow open possibilities (problem thinking and solutions development). The other 2 working spaces, Define and Deliver, explore convergent thinking, defining the problem, and selecting the suitable solution. For instance, Levy (2018) [146] used the Double Diamond in the academic context to teach and practice empathy activities with students from 2 different courses. The authors aimed to assess how empathy is explored in projects linked to wicked problems. Challiol *et al.* (2019) [46] used this model with students as an approach aiming to teach mobile application development for the domain of co-location games. Hehn *et al.*

(2020) [107] discuss the Double Diamond in the industry domain proposing a framework for integrating DT and Requirements Engineering.

Luchs (2015) [157] proposes a DT model with 4 working spaces: Discover, for capturing customer's insights, followed by Define working space when the team creates a definition of the problem based on the insights gathered previously. In the Create working space, the team develops a solution to next evaluate them with the stakeholders getting feedback in the Evaluate working space.

#DT models with 5 working spaces

D.school DT model from Stanford University [74] was the most cited DT model in the literature. *D.school DT* is composed of 5 working spaces: Empathize, Define, Ideate, Prototype and Test. Empathize aims to observe and view users and their behavior in the context of their lives, interacting with them and living their experiences; Define is used to synthesize the empathy findings into insights and scope a specific and meaningful challenge; Ideate relates to the generation of design and solution alternatives, while Prototype is the phase in which ideas are transformed into physical artifacts for getting feedback from users and refining solutions in the Test phase.

D.School DT model was used by Mutuku and Colaco (2012) [180]. The authors applied D.school DT model to create and design solutions based on insights related to government data, focusing on Open Data solutions for the government of Kenya. Pham *et al.* (2018) [207] present another example of *D.school DT* model, using it in the format of workshops with students to learn how to develop solutions and generate ideas for mobile application development.

Meinel and Leifer DT model organizes DT with the working spaces [127]: (re) Define, to establish the problem to be solved; Needfinding and Benchmarking, to understand the users and explore the design space; Brainstorm, to generate ideas; Prototype, to build prototypes of the ideas, and; Test to learn with the generated solution. This model shapes DT in a cyclic mode, e.g., the first activity (re)Define helps to restart the process seeking improvement in the solution. For instance, Keighran and Adikari (2016) [127] introduced this model in a study involving DT to develop strategies that satisfy a high-performance team structure. Adikari *et al.* (2016) [1] used these DT working spaces to explore how the inspired co-design approach on DT could be used to create ideas and prototyping to design solutions agreed by the user for an intended information system.

 # DT models with 6 working spaces

Driving Board [185] represents DT as an iterative cycle organized into 6 working spaces: Develop, Present and Provoke, Explore, and Reflect. This cycle is preceded by an Approach phase, in which stakeholders and team share information about the problem context and user experiences, and concludes with Escape. This working space allows the participants to conclude that the problem has been sufficiently explored and that the list of elicited requirements reflects the solution for the identified problem. The working spaces aim to define low-fidelity prototypes to give a taste of what could be used to explore the discussed ideas (Develop); present the latest versions of prototypes to encourage further ideation through the participants' feedback (Present and Provoke); use the prototypes as boundary objects to help focus in the problem space and provide a technological scope to facilitate the further exploration of the problem (Explore), and promote the reflection upon provided feedback aiming to improve or propose new prototypes until the solution is reached (Reflect).

Hasso Plattner Institute (HPI) model organizes DT into 6 working spaces [105]: Understand, when the team sets the problem space; Observe, when the team gains a consistent view and empathizes with the users and stakeholders; Define the Point of View, which serves to define the point of view and in which the knowledge gained will be collated and summarized; Ideate, when the team subsequently generates a variety of solution possibilities, then selects a focus; Prototype, which serves in the development of concrete solutions that can then be tested on the appropriate target group (Test). For instance, Junior and Do Nascimento (2016) [122] applied HPI model to create e-learning objects. The authors argue that the integration of this model with Scrum allowed the discovery of the users' needs and the production of a solution that was tested by the users. Luedeke (2018) [158] used the HPI model to develop a tool for the automotive industry.

Nordstrom DT model proposes an integration between DT, Lean Startup, and Agile software development exploring the following 6 working spaces: Define the challenge, when a multidisciplinary team of developers and designers gathers information about the challenge to be solved and defines the sequence of activities to seek the solution; Observe People, when the team records what stakeholders do and think; Form Insights working space is used to generate insights to drive the discussion of potential solutions; Frame opportunities is a working space used to create a visual representation to understand the collected data by highlighting key relationships and developing the solution strategy. Next, the Brainstorming Ideas working space allows brainstorming solution ideas and testing the chosen solution under real conditions to learn how it works in practice (Try Experiments). De Paula and Araújo (2016) [63] followed that model for developing a mobile game.

 # DT models with 7 working spaces

Hiremath and Sathiyam (2013) [111] introduced a DT model for the creation of innovative solutions, setting DT in 7 working spaces: Scope, which is concerned with planning customer and stakeholder interactions, project member's availability, and planning the DT activities efficiently; Research, working space for the team learning about the problem space by observing and interviewing end-users and finding their needs and motivations; Synthesis, working space used by the teams to identify connections on the data gathered from research and making statements about their understanding of the problem; Ideation, working space for ideating, sharing, voting, selecting and generating feasible ideas; Prototyping, working space used by the team to create physical ideas to represent the solution; Validation, working space used to meet the users and validate the prototyped ideas, and Presentation, the last working space used to show the produced design blueprint for the stakeholders and establishing the solution ready for production.

Sandino *et al.* (2013) [230] propose a DT model for the development of interactive real-time applications. It organizes DT in 7 working spaces: Define is a working space used to the definition of a series of constraints (e.g., how faithfully a simulator must reflect reality) that will guide the subsequent work; Explore is used for the team to gather information about potential users, their needs, and previous solutions to the same problem; Ideate is a working space when the team collaboratively identifies relevant issues and generates as many ideas as possible to get answers to the identified needs. Next, Prototype working space is when the ideas are prototyped and refined through iterative discussions; Select is the working space used to choose the candidate solution that might solve the problem; Implement is the working space used by the team to implement the candidate solution, and in Review working space the team keeps track of whether it fits the purposes that it was conceived for, once the product is introduced to the users, and identifies possible areas of improvement.

RQ3: What DT techniques are used in software development?

This Research Question maps what DT techniques are used in software development. DT as a set of techniques typifies the perspective of DT as a toolbox as proposed by Brenner *et al.* (2016) [33].

Pusca and Northwood (2018) [215] discuss that DT can be understood by two main and core spaces: problem and solution spaces. The former focuses on problem identification and formulation, while the latter, the solution moment, seeks to propose and validate solutions. The authors also argue that the problem space “is considered an analytic sequence in which the designer determines all of the elements of the problem and specifies all of the requirements and the constraints”, while the solution space “consists of synthesis and analysis sequences in which several possible concepts are evaluated to find the best solution for the problem”. Therefore, following the statements presented by Pusca and Northwood (2018) [215], we organized the DT techniques according to the problem space and solution space. Also, we point out a column “Not mentioned”, which lists the publications that mention DT techniques but without indicating if the technique serves the problem or the solution space.

Table 4.10 also lists 85 DT techniques that we mapped in literature for both problem and solution spaces. It means that teams using DT in software development are concerned with understanding the problem and proposing the right solution [234].

Table 4.10: DT techniques used in software development

Id	Technique	Problem Space	Solution Space	Not mentioned
1	AEIOU	109		
2	A Beginner's mind		119	
3	Acceptance Test		109 037 092	
4	Affinity Diagram	115	099	
5	As-is scenario map	126	109	100 111
6	Behaviour Map	006 115		
7	Behavioural Archeology	006 115		
8	Benchmarking			111
9	Bodystorming	118	115 003 122	
10	Brainstorming	112 115	020 043 045 051 054 057 076 088 089 099 107 116 095 002 004 007 023 075 081 115 123 015 022 039 087 105 126 016 068 092 122	024 073 079 120 026 121 055
11	Brainwriting		085 086 083 101 091	
12	Blueprint	081 087 101 018 087 115		120
13	Business Model Canvas	115	101	
14	Card sorting	007		111
15	Cost-benefit matrix		083	
16	Conceptual Map (cognitive)	115	088	024
17	Crazy eights		079 101	
18	Customer Journey Map	046 054 099 116 049 065 075 101 102 091 115 119 018 035 092	081	098 120 055
19	Day in the life	085		
20	Desk Research	088 081 092		

21	Dot Voting		086 107 091	
22	Eliminate-reduce-raise-create grid		085 002	
23	Empathy Map	017 045 051 083 098 050 075 081 091 109 115 118 003 080 035		111 120 055
24	Epic	097		
25	Error Analysis	006		
26	Ethnography	006 115 116		031 084
27	Expectation Test		109	
28	Exploratory Research	054 088 112 081 102 115 092		
29	Feedback matrix		054 023 050 065 091	
30	Field Studies	009 102		
31	Fishbone			100
32	Five Fingers		119	
33	Five Human Factors	060		
34	Five Whys	037 119		016
35	Fly on the wall	006 115		
36	Focus Group	076 114	080	031
37	Generative Sessions	081		
38	How can (might) We?	054 107 099 081 053	124	121
39	Ideas Menu	081		
40	Interview	054 076 088 089 107 023 037 065 067 099 110 116 074 075 081 091 115 124 015 003 022 039 105 126 025 097 122 123	076 107 112 123 080	014 071 073 120 121 025 117
41	Insight Cards	014 112 115 092		024 120
42	I Wish/I like feedback		057	120
43	Letter to grandMa	119		
44	Matriz CSD			120
45	Mind Mapping	107 112 081 109 022	115 123	024 111 120

46	Motivation Matrix	115		
47	Now, How, Wow		091	
48	Observation	085 002 023 067 035 097 122	076 123 105	073 079 025
49	Personal Inventory	119		
50	Personas	017 045 046 054 083 085 086 112 116 007 009 049 059 065 074 075 091 098 101 102 115 118 123 003 018 078 126 053 122		014 031 010 071 079 113 120 026 108 121 055 117
51	PEST	060		
52	(Pitch) Presentation		085 091	120
53	Positioning Matrix		081	
54	Priorization Grid		050 118	100
55	Proof-of-concept		051 058 053	111
56	Prototyping (paper or low fi)		046 051 054 057 083 085 086 107 116 002 007 023 034 049 067 075 081 102 119 123 003 015 018 126 005 035 053 068 092	062 073 079 113 120 026 005 084
57	Protoyping (medium-fi)		076 102 039 105 097	025 026
58	Prototyping (physical/hi-fi)		054 076 085 099 110 112 090 102 114 115 123 015 039 126	062 113 026 005 025 111
59	Power of ten		119	
60	Questionnaire	088 006 023 081 115 105	076 099 116 006	
61	Role playing		107 091 092	073
62	Sailboat			100
63	Service Walkthrough		101	
64	Shadow	124 025 122		
65	SIPOC Review			100
66	Social network Mapping	006		

67	Stakeholder Map	050 102 115 118 018 122	081	111 071
68	StoryBoard	046 023 075 073	054 060 049 067 034 091 115 118	024 031 072 120 055
69	Storytelling	066 115 022 035 122	003	120
70	Survey	089 076 081 006 123	089 006	
71	(Systematic) Literature Review	085 080 025		
72	Touchpoint matrix	115		
73	Trends Matrix	060		
74	Try it yourslef	006	115	
75	Usability (user) test		046 076 116 023 065 102 114 126	
76	Use case diagram		087	
77	User Stories		097	010
78	Venn Diagrams			122
79	Workshop	088 089 004 081 015 092	089 004 052 100 105 097	031 025
80	World cafe	057		
81	Yes, but/ Yes and then.. game	057		
82	5w2h	049		
83	30 Second Sketch		119	
84	2x2 Matriz			122
85	6-3-5 method		123	

Divergent Space: 000 Academic 000 Industry 000 Innovation | 000 Theoretical

Convergent Space: 000 Academic 000 Industry 000 Innovation | 000 Theoretical

Space Not Mentioned: 000 Academic 000 Industry 000 Innovation | 000 Theoretical

Brainstorming, Empathy Maps, Interviews, and Personas are the most cited techniques in the search for a better problem understanding (or problem space). For proposing solutions and validating them (or solution space), the most cited techniques were Prototyping (on different fidelity levels), Usability testing, and Workshops. Prototyping is the DT technique that assumes the role of the protagonist. Although the prototypes' level of fidelity varies, they offer support to DT practitioners to visualize the ideas and learn with them to improve a solution [68]. Hehn *et al.* (2020) [107] claim that Prototyping helps to promote user interaction with the proposed solution. Araújo *et al.* (2019) [12] advocates that prototypes are a well-known technique that helps validating the ideas created in DT activities. Alhazmi and Huang (2020) [8] mention that prototypes allow the stakeholders to interact with the solution.

Still looking at the DT techniques, we found 19 techniques used in both problem and solution spaces. Such techniques are: Affinity Diagram, As-is Scenario Map, Bodystorming, Brainstorming, Business Model Canvas, Conceptual Map, Customer Journey Map, Focus Group, How Can We?, Interview, Mind Mapping, Observation, Questionnaire, Stakeholder Map, Storyboard, Storytelling, Survey, Try It Yourself, Workshop. It means that DT techniques are flexible, helping the practitioners to solve the problem using the techniques according to their needs.

Also, well-known techniques are used in software development activities, such as Epics, Use case diagrams and User stories. Although these are not DT techniques, they were reported as adopted techniques in the use of DT in software development. This finding "blend in" techniques from DT itself and software development, like Requirement Engineering, for example, [67].

Given the DT techniques that we listed in Table 4.10, choosing among them may be challenging. Thus, in RQ4, we present our findings of how the literature reports the decision criteria practitioners use for choosing DT techniques.

RQ4: What is reported about the selection of DT techniques in software development?

This research question aimed to explore how literature describes the factors that contribute to the decision-making of Design Thinking techniques selection to use in the context of software development.

We did not find any paper explicitly discussing how professionals decide which techniques to use. On the other hand, we have figured out that the DT techniques are often chosen according to the goal that the DT practitioner wants to achieve (or DT activity). Goals in this context are encapsulated by the DT working space being conducted by the teams. Table 4.11 lists the publications which mention that DT techniques were selected considering the DT working space being conducted. This means that the authors directly associated the DT techniques used with the working space.

Table 4.11: Selection of DT techniques

Technique selection	**Publications**
According to the goal to be achieve (encapsulated as DT working spaces)	001 014 017 020 024 043 045 046 051 054 057 076 077 083 085 086 088 089 099 107 112 002 004 006 007 009 023 034 037 049 050 052 058 059 065 066 067 074 075 081 090 091 093 095 098 101 102 109 115 003 015 018 022 039 078 080 087 105 005 016 019 025 030 035 053 068 092 094 097

Research type: Empirical research - 000 Academic 000 Industry 000 Innovation context| Non-empirical research - 000 Theoretical

We have also investigated what possible criteria practitioners use in selecting DT techniques. The selection criteria for the techniques we have identified are:

- Application goal: what the DT practitioner aims to achieve using a DT technique;
- Application time: the time required to apply a technique;
- Familiarity: level of knowledge about a technique, but without using it;
- Comfortable to use: how comfortable the practitioner feels with using a technique;
- Stakeholder's information: information available about the stakeholder and how to explore her participation in the DT activities;
- Problem information: information available about the problem and how to explore it;
- Previous experience: level of experience in using a technique.

Table 4.12 shows the criteria for selecting DT techniques that we identified in the literature and the respective papers that mention them.

Souza *et al.* (2020) [245] argue that even DT specialists recommend the use of DT techniques toolkits to guide DT practitioners on the selection of DT techniques based on DT models' working spaces. The authors also mention that the selection of techniques might be made by exploring questions such as "What tools can I use to understand people?". It indicates that the Design Thinking practitioners' goals determine what techniques to use.

Dobrigkeit *et al.* (2020) [71] point out in a study with practitioners that the selection of DT techniques uses information that allows identifying which techniques require a short application time, are easy to put into practice and understand how to use, and developers feel comfortable using.

Chasanidou, Gasparini and Lee (2015) [48] discuss that the DT techniques help to facilitate the creation of innovative software solutions. The authors consider selecting the methods and tools that are most fit, and knowing them may improve the results generated for software engineering activities through the possibility of choice between the alternatives that involve the user more effectively in the software development process. Brenner,

Table 4.12: Criteria used by Practitioners for Selecting DT Techniques

Criteria	Publications
Application goal	115
Application time	100 119
Familiarity	119 018 094
Comfort of use	119 094
Stakeholders' information	100 094
Problem information	094
Previous experience	073

Research type: Empirical 000 Academic 000 Industry 000 Innovation Non-empirical 000 Theoretical

Uebernickel and Abrell (2016) [33] claim that the deployment of appropriate methods is a success factor of DT in projects, while Carlgren, Rauth and Elmquist (2016) [43] argue that the use of the proper techniques allows collecting the “right” solution’s requirements.

De Paula, Amancio and Flores (2020) [64] consider that it is important to do an efficient pre-work on the selection of DT techniques, investigating the stakeholders in detail, taking the time to know their needs, and choosing the appropriate innovation techniques for ensuring the production of solutions meeting the user’s expectations. The authors accepted IBM’s challenge to apply a DT light-version, where time constraints were decisive for selecting the techniques.

Harriet, Monika and Verena (2019) [104] proposed a card-based set of DT techniques called Innodeck. Innodeck provides information about the techniques supporting the use of DT for innovation. The authors mention that DT is composed of various techniques and selecting which technique to use in the working spaces depends on the problem and the people involved in the project. For instance, the authors mention it is worth the professionals selecting those methods that they are familiar, comfortable with, and consider suitable for their challenges. Hehn *et al.* (2020) [107] proposed a framework comprising 40 artifacts to be generated using DT integrated into RE that indicates a set of techniques that can be used and when to select each one.

Dobrigkeit and De Paula (2019) [68] investigated a global software company and concluded that the professional’s experience plays an essential role in understanding DT, in how it is used or how the techniques are selected. The authors mention that the developers who participated in the study described DT as a process or toolbox because they had, in general, less DT experience than the managers or designers. On the other hand, the designers consider DT a mindset that fosters the professionals’ problem-solving.

Also, the DT techniques usage in software development is discussed by Souza *et al.* (2019) [245]. The study presents an experience report addressing the challenges faced by software engineering students in selecting and using DT techniques. Initially, the authors introduced 15 DT techniques to a group of students. After the execution of a mobile application development challenge, the students used only 6 techniques. The authors reported that the students did not select more techniques due to the lack of clarity in understanding how a certain technique works and how to use the techniques in the established scenario. Then, to handle these challenges, in a second round, the authors proposed DTA4RE (Design Thinking Assistant for Requirement Engineering)[12], a tool to aid the selection of DT techniques. DTA4RE suggests DT techniques according to user needs based on a question-answering form. After the second round, the students mentioned that the tool helped them to choose techniques to solve the problem they had at hand. The students selected 12 techniques of the 15 presented.

[12] https://sites.google.com/site/dta4re/pagina-inicial

RQ5: What are the key points to be aware of when using DT in software development?

DT fosters an open-minded environment for problem-solving, explores collaboration, and promotes creativity. It also aims to boost the user's participation in the software development process. However, DT is not a silver bullet and its use might establish a complex environment. For instance, De Paula, Amancio and Flores (2020) [64] discuss points and counterpoints based on their experiences of using DT in the software industry. The authors indicate not only the benefits but also the risks they figured out in projects with DT. Pereira *et al.* (2021) [201] also investigated the perceived benefits and the challenges faced by IT professionals when using DT to understand the user's needs and propose innovative software solutions. However, both studies do not indicate what the key points DT practitioners have to pay attention to when using DT in software development. Therefore, this question aims to bring some light through the map of the points to be aware of when using DT in software development.

Table 4.13 presents a set of points that DT practitioners should consider when using DT in software development. We categorized the key points into 4 categories, as fol-

Table 4.13: Attention points for using DT in software development

Attention points	Publications
Problem and solution preconceiving	
Preconceiving problems (lack of problem understanding)	071 106
Preconceiving solutions (outdated ideas)	071 100
Organization and stakeholders participation	
Reach end users	086 007 081 114 065 071 073 097
Time pressure	086 071 100 102 114 119 029 097 106
Unavailability of resources (dedicated space for creativity, materials)	102 055 106
Lack of higher management engagement	071 073 029 106
Lack of collaboration	058 114 125
Lack of employee commitment	057 007 102 114 119 097 106
Mindset changing	007 102
Lack of knowledge in DT	051 058 125 106
DT techniques selection and results' sharing	
Select a correct combination of artifacts and their proper use	081 100 114
Share results from DT activities and ensure an effect on the final product	073
Requirement Engineering integration	
Lack of requirements traceability	065 081
Neglect of non-functional requirements	065 081
Imprecise effort estimates	065 081
Prioritization of requirements	065 081
Changing of requirements	081
Lack of documentation	054 081

Research type: Empirical 000 Academic 000 Industry 000 Innovation 000 Theoretical Non-empirical lows: Problem and solution preconceiving, organizational and stakeholders' participation, DT techniques selection and results sharing, and Requirements Engineering integration.

The first category of attention points shows that DT practitioners have to be aware that 2 types of participants' preconceiving might difficult the adoption of DT in software development. Problem preconceiving is the first type of preconceiving [234, 251]. It means that the customers or even team members may have a pre-formed understanding of the problem to be solved, which is known as Entrained Thinking. Snowden and Boone (2007) [242] define Entrained Thinking as a pre-established mindset that drives the team to an already conceptualized solution (lacking innovation) or even to a solution that does not solve the right problem. The second preconceiving that the DT practitioners have to pay attention to is the solution preconceiving [234, 64]. It happens when the customer presents the problem to be solved and also presents the solution she wants in advance.

The second category of attention points refers to the organizations' and stakeholders' participation in DT activities. It means that when using DT in software development, the DT practitioners have to pay attention to how the organizations commit to DT and also to the user participation in DT activities. Literature reports that it might be difficult to reach the "proper" end-users for the problem that the team wants to solve due to several constraints such as people agenda, ethical issues, organization setup on business environment [111, 234, 68].

The time required for using DT techniques or conducting DT workshops is also a point to be considered in using DT in software development. Kongot and Pattanaik (2017) [136] argue that "DT techniques need a certain amount of time to be carried out. Customer's time is precious, and often there is a need to improvise on the techniques and develop a proposal that works given the time and audience". Alhazmi and Huang (2020) [8] advocate that DT requires a lot of effort to address the users' needs. In addition, time pressure is provoked by customers that do not understand the design process and the value it has on the construction of an innovative and desired solution [234].

The unavailability of resources might become a challenge that compromises the use of DT in software development. Therefore, it is an important point to be aware of before starting the use of DT [107, 102, 146]. DT fosters a creative environment that looks for innovative solutions, breaking paradigms and proposing solutions that might have never been thought of before. However, a proper exploration of DT requires investment in dedicated spaces or materials for research, user analysis, brainstorming, ideas generation, prototyping, and feedback collection.

Literature also shows that for using DT in software development, a DT practitioner should be prepared to deal with a lack of higher management engagement [234, 68, 103, 251], pressure to the teams to converge [234], lack of collaboration, when the participants are not opened to give their ideas, or lack of experience in working on a multidisciplinary environment proposed by DT [210, 136, 121]. In addition, another key point to consider when using DT are the lack of employee commitment that represents the team members may think that discussing the problem and proposing solutions is a waste of time for those working on the technical aspects of the solution [8]; lack of knowledge in DT, indicating that the use of DT in software development requires practice to be used properly. For example, Prasad *et al.* (2018) [210] points out that organizations fail in using DT because they do not have enough theoretical knowledge in DT or the integration of agile and DT principles. In addition, DT may be challenging to be performed because it requires a mindset change, which is a key element in the proper exploration of DT, but that might be achieved when the professionals have years of experience in DT [68].

The third category of attention points to consider when using DT refers to the selection of DT techniques and how to share the results obtained in DT activities. Using DT requires a decision-making process for selecting what artifacts the team should elaborate on a project [167, 64, 251]. For Tannian (2020) [251], DT is not a cooking recipe since it evolves several dimensions, such as problem domain, team composition, level of DT expertise, cooperating users, available materials, and market timing, among others. These dimensions promote a complex scenario of problem-solving. On the other hand, Dobrigkeit and De Paula (2019) [68] argue that once a result is achieved with DT, it needs to be shared properly with the stakeholders. The authors argue that "results need to be shared with the whole team in a time-effective manner and are only interesting to most of the team once a certain feature is under development". Therefore, there is still room for research about how to explore the decision-making of DT techniques in software development effectively.

The last but not less important category of key attention points that DT practitioners consider when using DT is the integration of DT into RE activities. Researchers and DT practitioners have considered DT an easy-in integration approach to perform RE better. However, although there are different integration strategies between DT and RE as proposed by Hehn *et al.* (2020) [107] such as Upfront DT, Infused DT or Continuous DT, when using DT, the practitioners have to pay attention to the requirements traceability, to the requirements prioritization, to the requirements changing, and to the effort estimation [167, 108]. Minimal or no documentation are reasons that collaborate to fail since DT does not foster the creation of extensive documentation of requirements [51, 167, 108]. Furthermore, non-functional requirements are usually not in the spotlight when using DT, even though usability feedback is collected with end-users.

In this context, the main findings of our study are:

Finding 1: Design Thinking is integrated into software development through 3 integration strategies (Upfront, Infused and Continuous). Upfront is the most cited strategy.

Taking into account the integration strategies of Design Thinking in software development proposed by Hehn *et al.* (2020) [107], we identified that the DT Upfront integration strategy is the most cited strategy if compared to the Infused and Continuous strategies (73/127 publications).

This result shows that DT has been integrating into software development as a starter activity to the Requirements Engineering, highlighting the search for the problem and the proposition of solutions. Therefore, development teams have been perceiving DT as a way to compose multidisciplinary teams, search for a deep understanding of the problem, ideate and prototype various possible solutions, and collect feedback from users before thinking about the technical requirements of the solution to be built. This perception indicates that software development teams have moved forward to understand what is needed to develop before starting the development process.

The use of the Upfront integration strategy has also shown that development teams recognize that even with the benefits resulting from the use of agile methods and the well-known Requirements Engineering techniques, there is still room for what the solution is expected. Thus, teams can put into practice the philosophy “fail fast and fail often” to develop the proper solution.

DT can also be integrated into software development through the Infused DT strategy, which boosts Requirements Engineering activities. In this strategy mentioned in 36 of the 127 publications selected, DT assists in activities already carried out by the development team to extract, analyze, and evaluate requirements. The infused strategy sets DT as a toolbox that supports the practices adopted by teams without requiring new steps in the software development process or even without forcing teams to modify the activities already performed.

Using the Infused strategy also indicates that teams have realized the need to improve their Requirements Engineering activities. Thus, DT has been included as a mechanism that encourages exploration of the mindset of designers, promoting a deep understanding of the problem and prototyping the solution that suits the user’s needs.

DT was integrated into software development by the Continuous DT strategy in 17 of the 127 publications selected. In this strategy, DT integration exceeds the process of developing a solution and achieves levels of change in the organization’s mindset. DT is seen not only by development teams as a set of techniques or as a process that foresees a set of working spaces, but as a practice to think in disruptive and innovative solutions that meet the user’s needs. Continuous DT provokes the development team, managers, and professionals who perform decision-making functions to realize that innovating is not an isolated activity but a way of acting and thinking.

Continuous DT has been discussed as a DT integration strategy in software development obtained due to DT practitioners’ experience. Dobrigkeit and De Paula (2019) [68] show that the perception of DT as a mindset beyond a process or a toolbox is a perception of professionals with more experience using DT or who perform management functions. On the other hand, the authors point out that for developers or professionals with less experience in DT, it is seen as a specific activity to solve a particular problem. Thus, this result indicates that companies will no longer use DT only as an upfront activity or as a means of improving what is already performed by Requirements Engineering in an infused strategy. Companies will explore DT to adapt thinking around how they work to produce innovative solutions.

Finding 2: Design Thinking models organize DT in distinct working spaces and make it a flexible and dynamic approach for problem-solving.

Brown and Wyatt (2010) [37] define the Design Thinking process as “a system of overlapping spaces rather than a sequence of orderly steps”. Based on this statement, we identified a spam of DT models in this systematic mapping study. We found DT models ranging from 2 to 7 working spaces. However, despite the difference in the number of working spaces, the models use convergent and divergent thinking to understand the problem and evaluate the proposed solution.

D.school and Brown were the most DT models reported in the literature, respectively. The former organizes DT in 5 working spaces, starting from empathy, which encourages the interaction and observation of users and their behavior in the context of the problem, going to the synthesis of the problem representing the foundation of the understanding of the previous working space. Also, the d.school DT model suggests ideation as a space for proposing several ideas to find proposals for innovative solutions, prototyped and evaluated by the user. These 5 working spaces lead teams to move from the problem to the solution space.

The DT model proposed by Brown organizes the transition from the problem space to the solution space, starting with the Inspiration working space, which is similar to the first two working spaces of the D.school model. The goal is to understand the problem according to the needs and challenges of the end users. In Brown’s model, the ideation working space acts similarly to the ideate and prototyping working spaces in d.school, where ideas and prototypes for these ideas are generated. Finally, the Implementation working space foreseen in the Brown model corresponds to the Test working space of the d.school model, in which end-users evaluate ideas.

Thus, despite having differences between DT models, DT presents flexibility regarding working spaces, allowing the teams to generate innovative solutions. The amount of working spaces of a DT model does not limit the use of DT for understanding the user’s needs, creating team engagement, exploring creativity and proposing appropriate solutions to the problem presented. Therefore, there is no single silver bullet DT model that must be followed, as also reported by Waidelich *et al.* (2018) [265], where the authors performed a literature review on DT models applied for general purposes. It is worth mentioning that no studies are comparing different DT models regarding the working spaces used to generate innovative software solutions.

*Finding 3:*A large number of DT techniques can be applied to software development in the problem space or the solution space. Techniques can be selected independently of the integration approach used, and a large number of techniques creates a decision-making problem.

DT practitioners also explore DT as a toolbox [33]. In our mapping study, we identified 85 techniques used in DT working spaces when integrated into software development for both problem exploration (problem space) and solution proposal (solution space). It reinforces the dynamic and flexible nature of DT that explores the designers' mindset for problem-solving.

The most cited DT techniques in software development are Brainstorming, Empathy Map, Interviews, Personas, and Prototyping. These techniques allow the collaborative proposition of various ideas to solve the problem, map the user's needs, obtain data about the problem and the solution, represent the user through a fictional character, and sketch the solution, respectively. However, the wide variety of DT techniques gives the practitioner the option to explore them as needed and test them in conjunction with other software development practices already performed by RE or by the development method used, such as the Agile method.

Our results also show that the DT techniques are not linked to DT integration approaches in software development. If the team wishes to integrate DT using the Upfront, the Infused, or the Continuous strategy, it can explore the full set of techniques and select those it considers suitable for the problem at hand. In addition, the flexibility provided by DT has also allowed the integration of techniques already known in Requirements Engineering, such as User Stories, Use Cases diagrams and Epics. Thus, we can highlight that DT and ER can be complementary activities so that their techniques can be used together and thus enable the software requirements to be appropriately established.

Therefore, DT as a toolbox helps DT practitioners to understand what the user's needs are and also develop a solution that is viable and feasible. However, although many DT techniques collaborate with teams in software development activities, it creates an endeavor in regard to the decision-making process for selecting the techniques. Thus, understanding how the selection of techniques is performed by professionals and researchers is important for using DT in software development.

Finding 4: DT practitioners usually consider the goals they want to achieve for selecting DT techniques. Goals in this context are encapsulated as DT working spaces. Application time, familiarity, comfortable to use, stakeholder's information, problem information and previous experience are also criteria used to guide the selection of DT techniques.

Despite the importance that professionals have given to DT in software development, we did not identify any work in the literature that investigates in depth what is the decision-making process of DT techniques. Our study identified that some publications indicate that the practitioners have selected the techniques based on the working space of a selected DT model. This finding confirms that the DT models help the choice of techniques and that the perspective of DT as a process is taken into account by teams [33].

However, it is necessary to know which criteria contribute to the decision-making of which DT techniques the practitioner will put into practice in order to help the software development. We identified that practitioners use the information they have about the techniques as criteria. This information can include the application time required by the technique, which indicates how the team should prepare itself and how much time, for example, the stakeholders should plan to participate in the activities of DT, the familiarity that the practitioner has with the technique, that indicates how much the practitioner knows the technique even without having applied it before, and comfortable to use, that indicates how apt the professional feels to apply the technique in practice with the team multidisciplinary available.

Other criteria that support practitioners with the decision of which DT techniques to select involve the information that the practitioner has about the stakeholders, allowing one to explore the most of each participant, or about the problem, which enables the practitioner to choose techniques that better explore the problem and propose effective solutions to it. Previous experience is a criterion cited for selecting DT techniques, which considers the result of earlier uses of certain techniques helping to use a technique again.

The technique selection criteria we have identified in this systematic mapping study is an initial set of indications of how practitioners decide which techniques they intend to use. However, we emphasize that there is a lack of studies investigating what are the criteria used by professionals to select the techniques that could encourage the development of solutions for helping novice professionals to establish techniques that foster the exploration of DT in software development in the most effective way.

Finding 5: DT boosts software development, but it is not a silver bullet. Although DT has been easy-in integrated into software development, there are key attention points that DT practitioners have to be aware of when using DT.

Although the literature has been showing that DT brings benefits to software development, it also shows that IT professionals have to be aware of key points to extract such benefits of the DT's usage (see Table 4.13).

Preconceiving problems or preconceiving solutions are 2 points that need certain attention in the use of DT in software development. DT practitioners have to handle problems already defined or solutions pre-proposed by customers, even before starting the use of DT. A mindset change could be a way to walk around these preconceiving issues, but it may be just achieved after a long time of experiencing DT in software development. Pressure provoked by the market time or higher management also represents endeavors to be solved when using DT in software development.

Design Thinking has collaborated to improve Requirement Engineering activities by providing a space to do a deeper understanding of the users' needs [107, 64]. However, reaching the "right" end-users could require a big effort from the organization due to time or domain constraints. In addition, the integration between RE and DT is not all a bed of roses. Literature has pointed out that DT for RE requires attention to the lack of requirements traceability, the neglecting of non-functional requirements, the non-ability to do a proper effort estimation, and the difficulty of doing a requirements prioritization by do not consider detailed documentation of requirements.

Literature also points out that it is important to develop a creative mindset and establish empathy with the user, as well as to move from a specific problem solution to a problem-solving mindset [154]. Creating a multidisciplinary team composed of professionals with different backgrounds might be a way to achieve the benefits of DT. However, some companies are still resistant to bringing their employees to work full-time in DT activities. Bringing the user together is yet another point to consider when using DT. In addition, is also necessary to find proper tools to support the use of DT, that will provide the proper setup to understand the real problem and propose innovative solutions.

Therefore, it is essential to identify which models and techniques are the best fit for a certain scenario, customer profile, and application areas (e.g., the problem is blurry, the problem is well defined but no indication of the solution is previously known, a solution already in place is no longer attending the user needs and requires reconsideration).

Our mapping study contains limitations and threats to validity inherent to a literature review [129, 271]. As a construct validity, our research questions may not provide complete coverage of all the papers that present DT and software development. As internal validity, related to studies selection and data extraction, two authors performed the process of paper selection [129]. We searched 6 digital libraries containing the majority of the high-quality papers in SE. Two senior researchers reviewed this research strategy. As conclusion validity, we extracted data and had discussions following the protocol which we defined before kicking off the extraction process.

Next, we present a Survey with IT professionals aiming to investigate the research gap in the selection criteria of DT techniques that explore the decision-making scenario of DT practitioners. Our goal is to further the understanding of how DT can be exploited to collaborate with software development and produce solutions that meet the user's needs and are feasible, viable, and desirable.

4.2 Design Thinking in Software Development: A Survey in Industry

Our Systematic Mapping Study showed that DT has been chosen as an approach to support problem-solving by many software development companies. However, little is mentioned about how these professionals use DT concerned with which techniques are performed, which steps are followed, and the way to implement this approach, as it proposes itself, to be divergent to generate numerous alternatives and, also, convergent, to find a solution. Thus, we executed a survey to further our understanding of how the Brazilian community of IT professionals uses DT in software development.

Our survey is characterized as exploratory [206, 125], seeking an understanding of the phenomenon through the information collected. We posed the following research question: "*How do professionals in the Brazilian software development industry use Design Thinking?*". Our goal was to collect data for characterizing the DT models and techniques used in software development, how professionals select the techniques and their level of difficulty in selecting DT techniques[13].

This section is structured as follows: we start by presenting how we carried out the planning and design of the survey, proceeding with its prior validation. Next, it describes the details of the survey's execution. Next, we show the outcomes gathered with our survey, discussing the findings and limitations.

4.2.1 Survey design

We started to build this study by designing the Survey according to guidelines proposed by Kitchenham (2001) [206]. We performed the following activities: planning and design, prior validation; survey execution, data analysis and results presentation.

Next, we show details of each survey's activities.

Planning, Design and Prior Validation

We started building a questionnaire as a data collection instrument, using the Qualtrics tool[14] (see Appendix A)[15]. Table 4.14 shows the questions that we posed in the questionnaire. We invited IT professionals to answer questions related to DT, such as DT methods, techniques, decisions and difficulties of using DT. We also questioned the professionals about their jobs and DT experiences, aiming to draw a profile of the respondents (demographic questions). We created the questions of the survey based on data gathered previously through our systematic literature mapping.

Before starting to distribute the questionnaire, we performed a prior validation process. Following the recommendations given by Kitchenham *et al.* (2002) [133], a pilot test was performed to evaluate the consistency and correctness of our instrument. We

[13]This study was published in the International Agile Conference (XP). Ref.: [211]
[14]Available in https://www.qualtrics.com
[15]This study was conducted in partnership with a Master's student at PUCRS.

Table 4.14: Questionnaire structure

#	Question	Type
	Demographic Questions	
D1.	What is your experience, in years, using Design Thinking?	closed
D2.	What is your current organizational role or function?	open
	Questions about Design Thinking in Software Development	
Q1	There are several process models, that abstract working spaces when using Design Thinking. Do you use any of these models as a reference in your activities?	closed
Q2	Several techniques can be used to support the use of Design Thinking. What techniques do you usually use?	closed
Q3	How do you usually decide which techniques to use?	closed
Q4	On a scale of 0 (No difficulty) to 10 (Extreme difficulty), how difficult do you feel in deciding which techniques to use in a given situation?	closed

sent the instrument to a researcher that has experience in the Software Engineering field and the use of DT in the software development industry. She participates in the activities of our research and also knows the results of the literature mapping that supported the planning and design of the survey.

Execution

Following the survey's planning and design process, we defined the target audience as those professionals working on DT in software development. We used LinkedIn[16] to reach out to our target audience and to distribute the questionnaire electronically. LinkedIn is a professional-oriented social network where IT professionals share their experiences and curricula. We applied the following strings as filters to reach our target audience: "*design thinking*" *and* "*software*" and "*design thinking*".

We surveyed the participants between December 2019 to January 2020. In the meantime, we invited 466 professionals to participate in our survey. As a result, 158 participants accepted to participate in our study, representing a response rate of 33.90%. Figure 4.7 shows the number of answers per period.

4.2.2 Survey Results and Discussions

D1. What is your experience in years using Design Thinking?

Figure 4.8 shows the experience of the professionals who answered the questionnaire. Regarding software development, 78 professionals mentioned having more than 8 years of experience, followed by 39 professionals who reported having between 4 and 7 years of experience. Thus, 74% of the survey's participants have more than 4 years of experience in software development (Figure 4.9).

[16] Available in https://www.linkedin.com

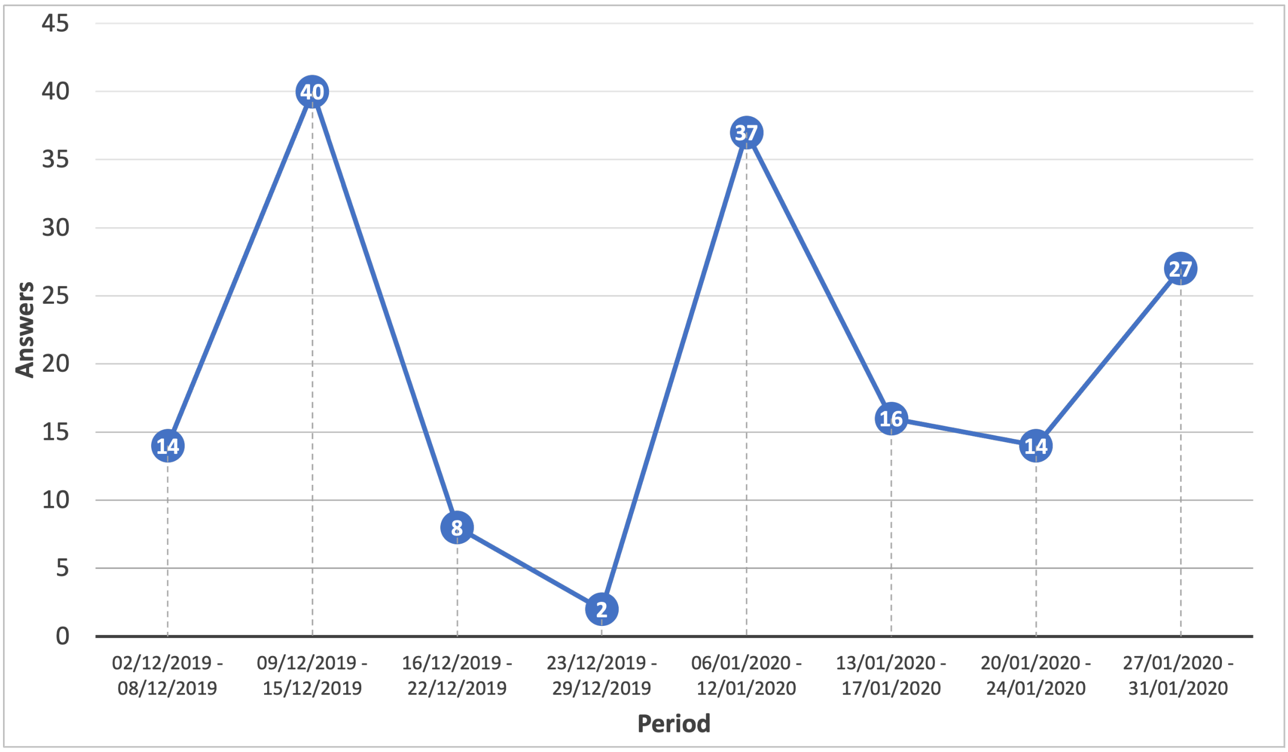


Figure 4.7: Survey participants' answers x period

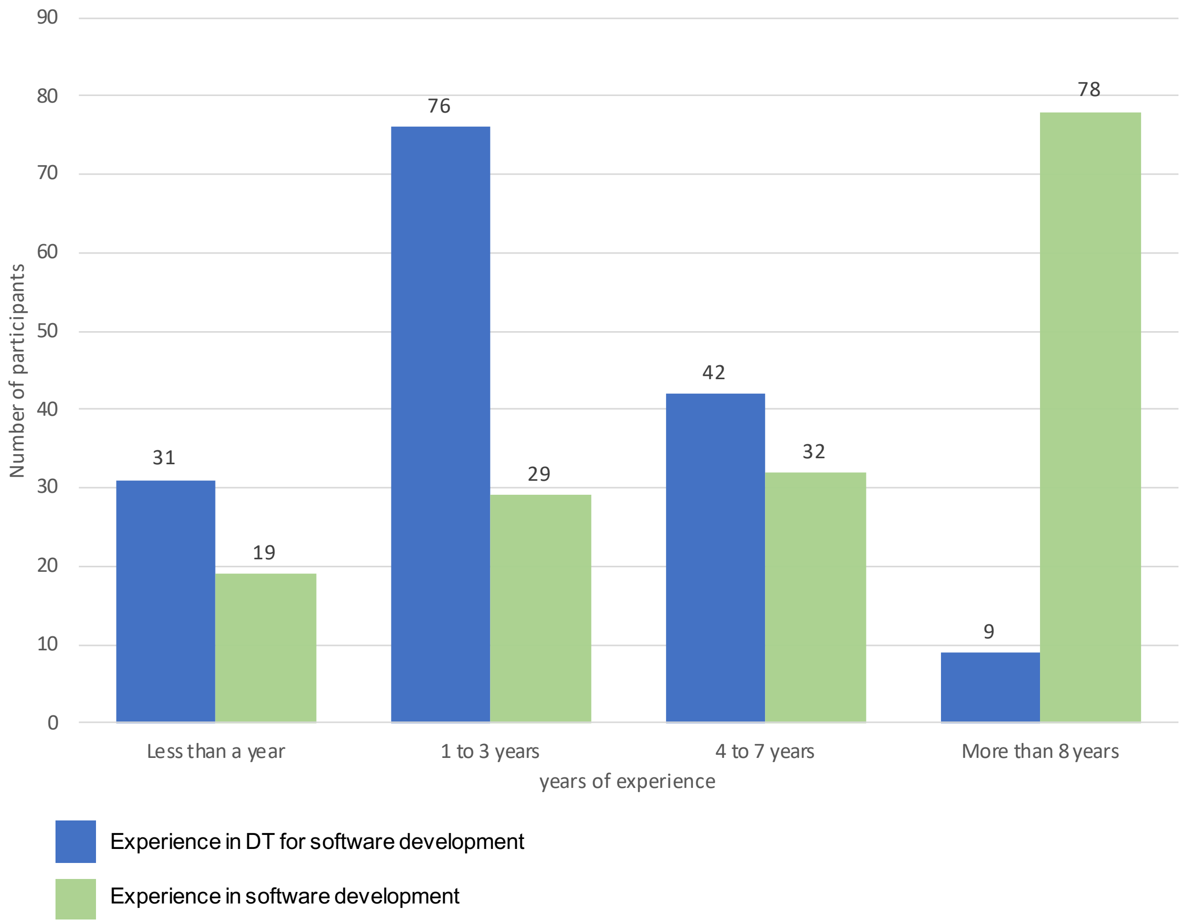


Figure 4.8: Survey participants' experience in software development x DT

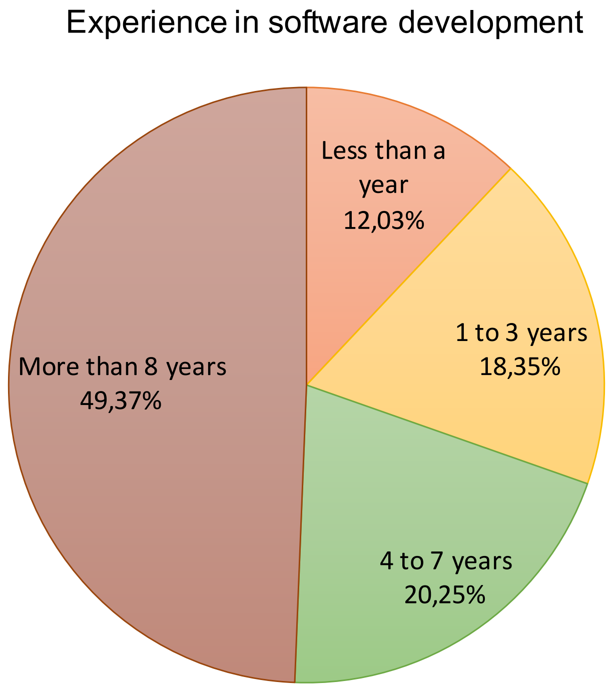


Figure 4.9: Survey participants' experience in software development

Regarding the experience in the use of DT in software development, the largest number of participants (76) responded having between 1 and 3 years of experience, followed by professionals (42) with 4 to 7 years in the use of DT for software development (Figure 4.8). This shows that 80.1% of the participants have at least 1 year of experience with DT (Figure 4.10).

QD2. What is your role or function?

In question QD2, we asked the participants about their jobs. Figure 4.11 summarizes the most cited roles played by the professionals who participated in our study. The participants most cited the following roles: Facilitator (16 citations), Researcher, UX/UI Designer, Developer, Specialist, Analyst, Agile Coach, Consultant, Product Owner, and Engineer. In addition, some participants reported playing the role of Head of Digital Transformation, Design and Innovation Manager, Service Designer, Product manager, Chief Executive Officer (CEO), Scrum Master, and Design Leader, among others.

Agile Coach, UX/UI Designer and Facilitator were the most cited roles, with 27, 21 and 16 professionals, respectively. This result indicates that DT is used by both Design professionals (UX/UI designers and DT Facilitators) and development professionals, such as Agile Coaches. Furthermore, based on the results for the 2 demographic questions, we can understand that the study reached diverse professionals with different experiences in software development and in using DT for such purposes.

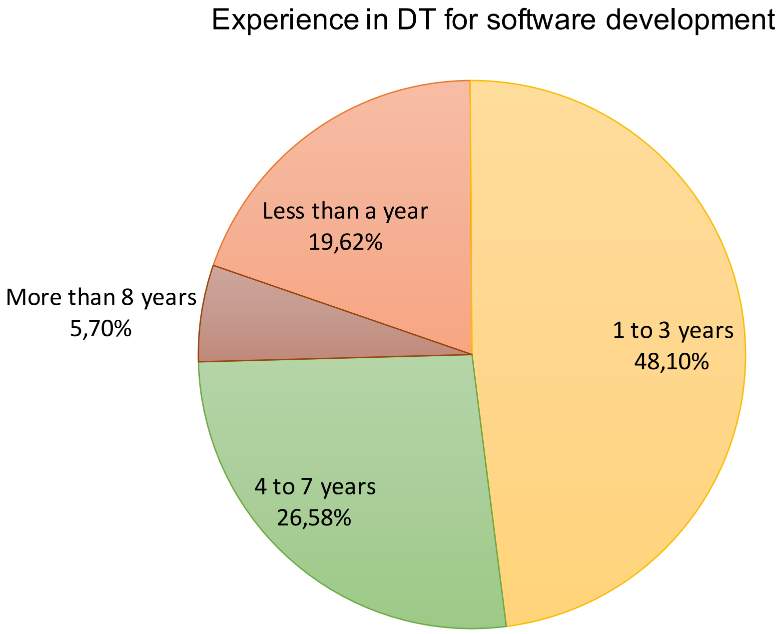


Figure 4.10: Survey participants' experience in DT for software development

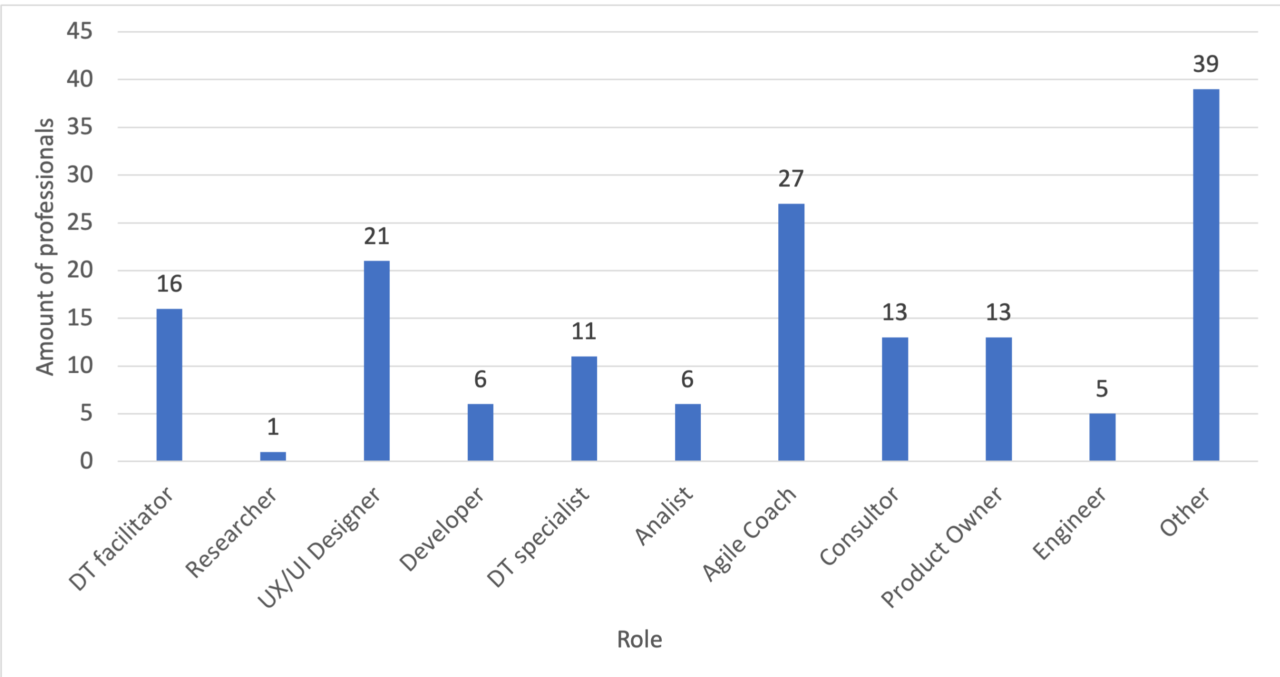


Figure 4.11: Survey participants' role in software development industry

Q1. There are several process models that abstract working spaces in the use of DT. Do you use any of the DT models as a reference in your activities?

In question Q1, we asked the participants if they use DT models as a resource for helping them to apply DT in software development. We listed in the questionnaire the DT models we found through our Systematic Literature Mapping. Figure 4.12 exemplifies a subset of DT models and the respective set of working spaces of each model presented to the survey's participants.

Figure 4.13 shows the result obtained in the Survey for Q1. The results indicate that the DT practitioners use different DT models, meaning that there is no consensus on the set of working spaces when applying DT in software development. This result is aligned with that obtained through the literature mapping (Section 4.1), which summarizes different DT models that have been used as a guide for integrating DT in software development and for conducting DT workshops or projects.

The answers the participants provided to the Q1 also show that DT models such as Divergent and Convergent, the Stanford d.school Model and the Hasso Plattner Institute Model (HPI) were the most cited ones. Therefore, this result confirms that professionals understand DT as a set of working spaces, as mentioned by Brenner *et al.* (2016) [33]. In addition, the result clarifies that the view of DT as a model is dynamic and reinforces the flexible nature of DT.

Q2. Several techniques can support the use of DT. Which techniques do you use?

In Q2, we aimed to identify the DT techniques DT practitioners use when integrating DT into software development. Figure 4.14 shows the DT techniques the participants mentioned they use and the respective amount of responses for each technique.

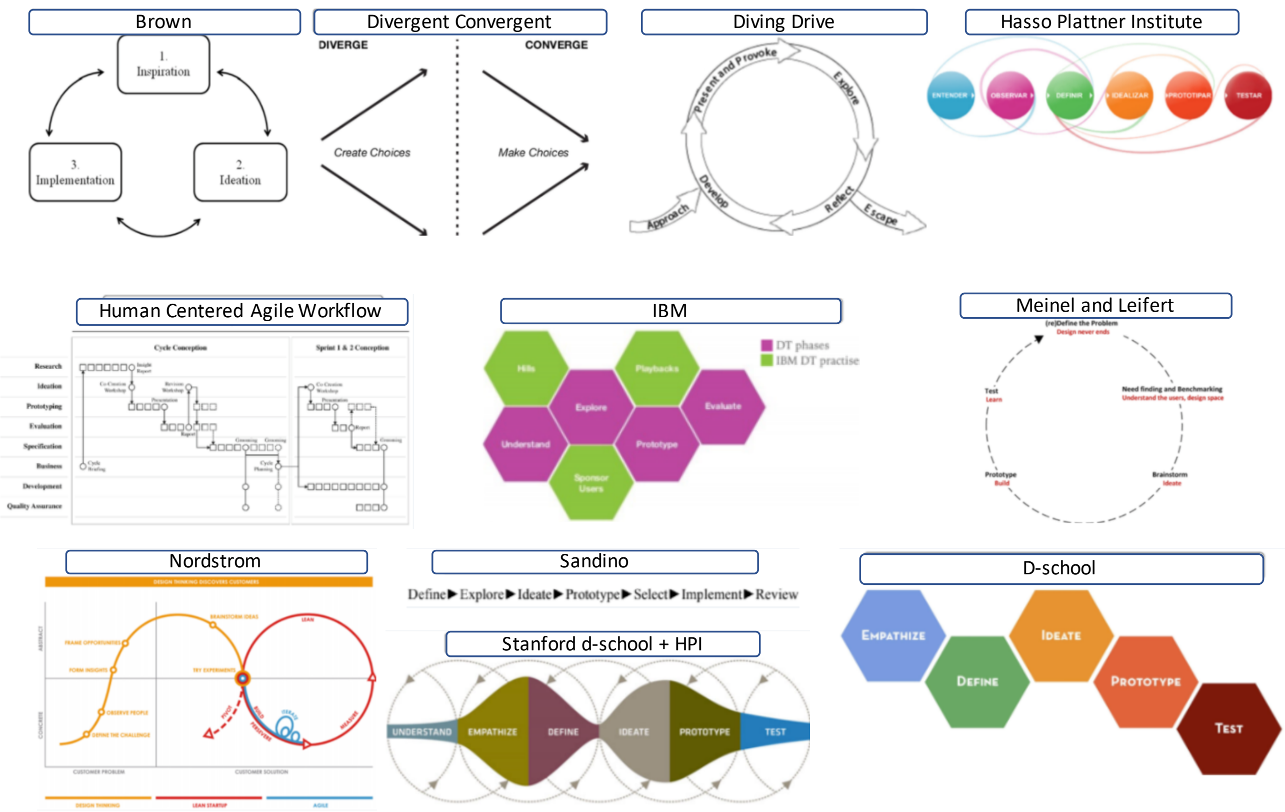


Figure 4.12: DT models presented to the survey's participants

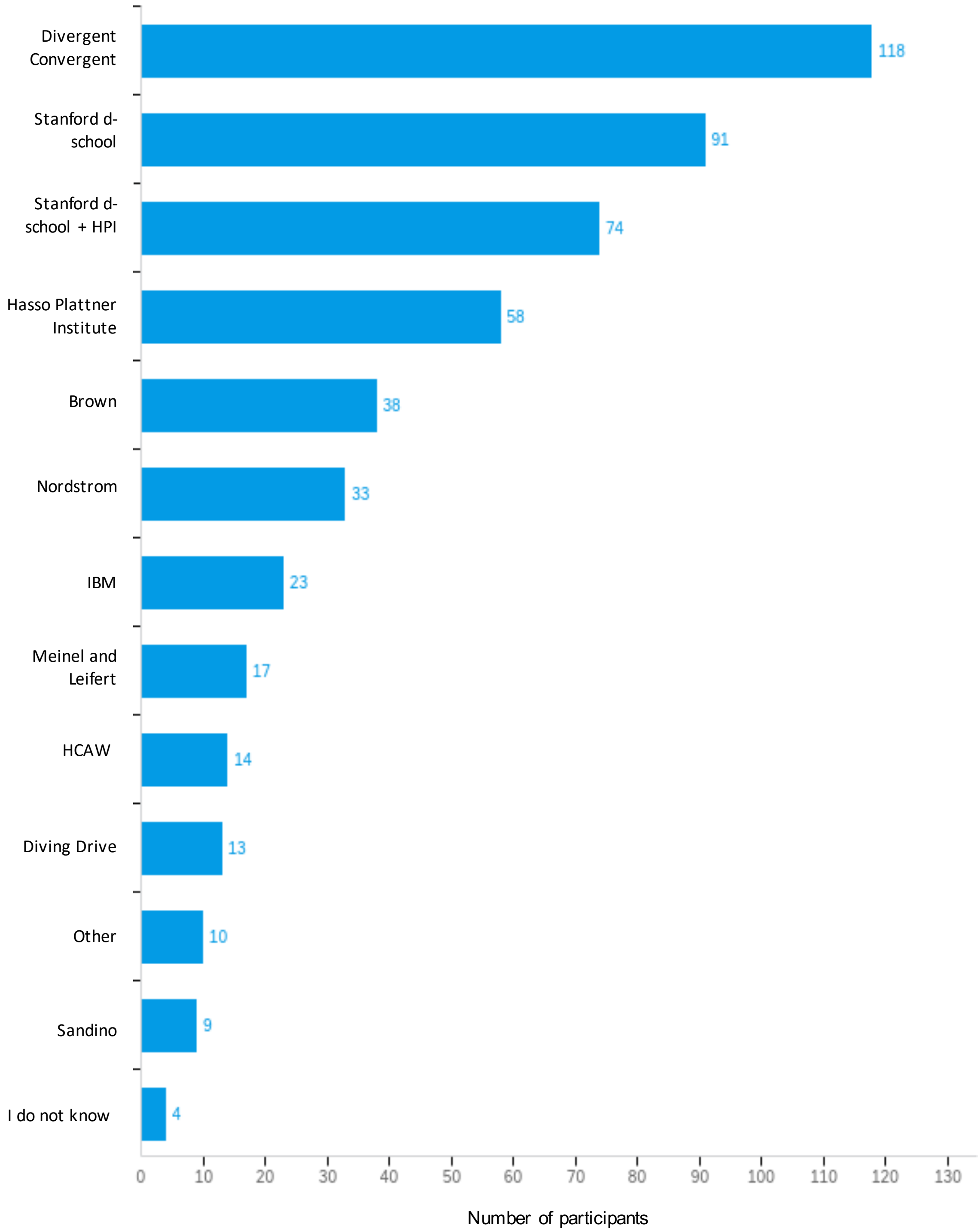


Figure 4.13: DT models indicated by the survey's participants

The professionals who participated in our survey indicated 49 different associated with DT as techniques they use in software development activities. The participants mentioned Brainstorming (151 participants), Personas (149 participants), and Empathy Map (123 participants) as the 3 most cited DT techniques that they use.

The results allow us to understand that IT professionals use diverse DT techniques, indicating that the selection of which techniques to use in software development must configure a challenging endeavor.

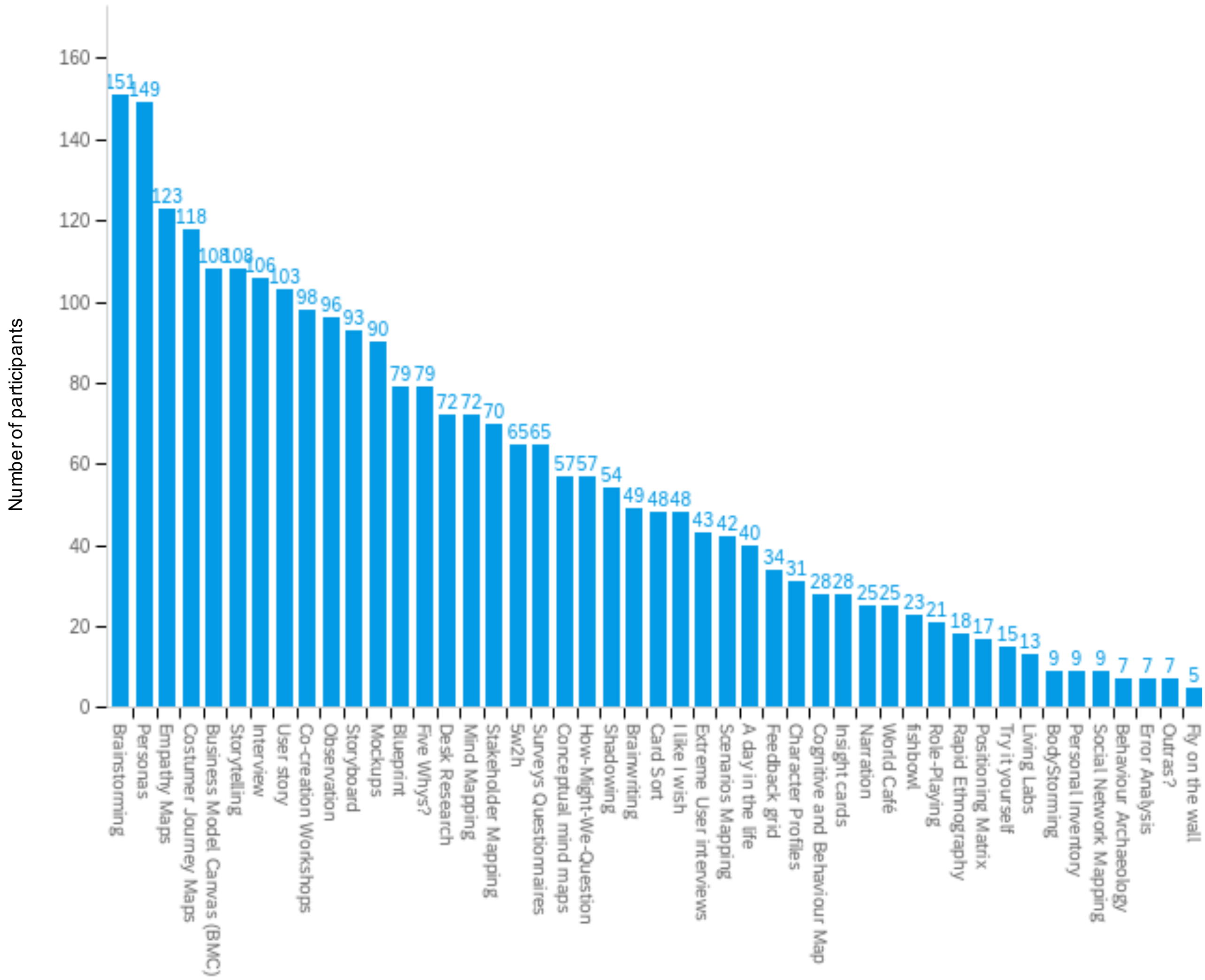


Figure 4.14: DT techniques the survey's participants use in software development

Q3. How do you (IT professional) usually decide which techniques to use?

In Q3, we asked how the participants decided which DT techniques to use in software development. Our goal was to identify the elements considered by IT professionals as determinants for selecting DT techniques. The questionnaire included some answers' alternatives as we found in literature, such as: based on the previous experience of the professional; it depends a lot on the context that the professional will use; the professional chooses the techniques according to the space/stage of DT, where each space/stage has its own techniques; the professional usually needs to study the techniques to know which one is the best for the moment; the professional already has his/her catalog of techniques that he/she uses; the professional selects the techniques by indication of a colleague; the professional selects the technique when the technique fits his/her need, and; the professional selects the technique by recommendation given by his/her company.

Figure 4.15 shows the answers the professionals indicated as decision strategies for selecting DT techniques. The results point out that the selection of techniques is mainly

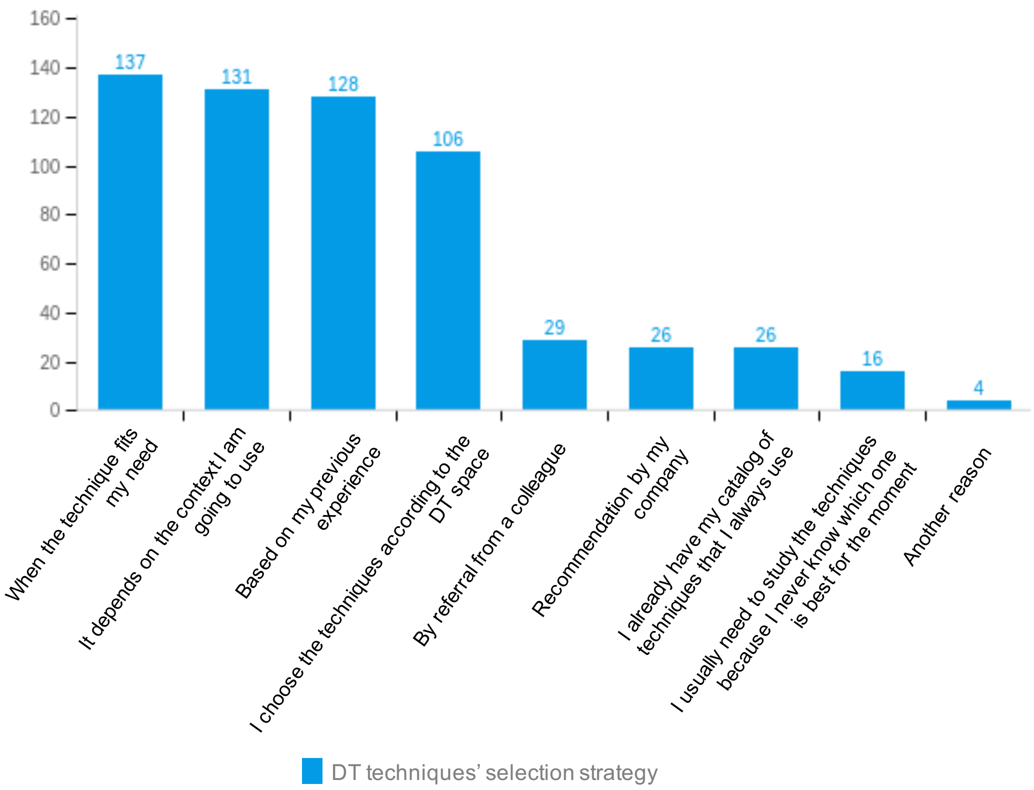


Figure 4.15: Survey participants' decision strategies for selecting DT techniques

linked to the following elements: the need of the professional who will use DT (137 participants), the context of the project in which DT is or will be used (131 participants), based on previous experience in the use of DT techniques by the professional (128 participants), and based on the working space according to the adopted DT model (106 participants)[17].

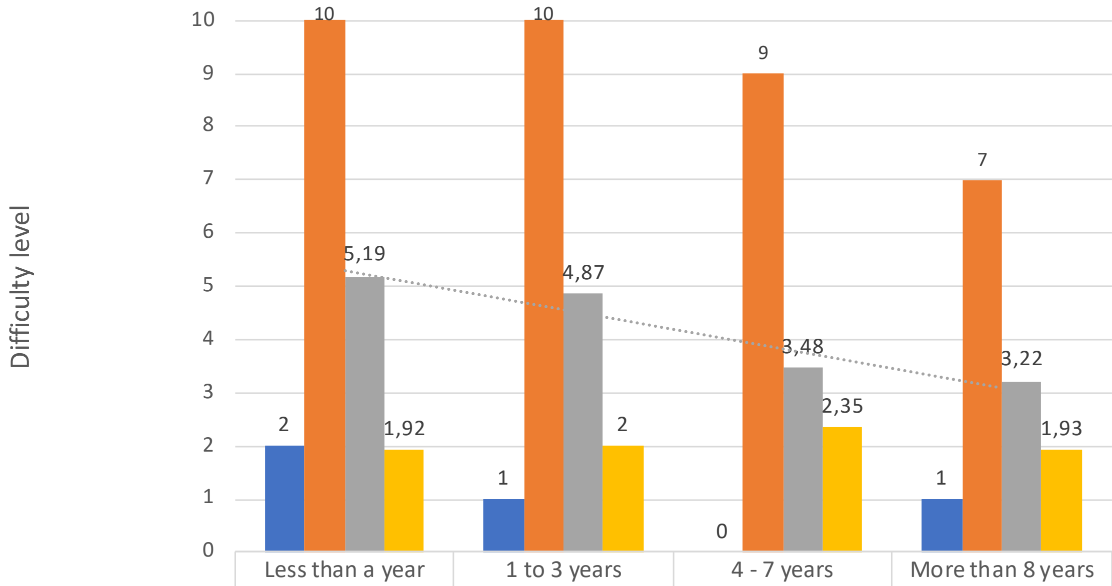
However, we can not assume what decision-making strategies the IT professionals use when selecting DT techniques. Therefore, more in-depth studies are required to model the decision-making of selecting DT techniques, aiming to assist software development professionals interested in using DT. This is the overall goal of this research.

Q4. On a scale of 0 (No difficulty) to 10 (Extreme difficulty), how difficult do you consider to decide which techniques to use in a given situation?

In Q4, we aimed to identify what is the difficulty level of software development professionals when they need to select techniques to be used in DT activities. The results pointed out that, on average, the difficulty of professionals is 4.5 on a scale of 0 to 10, with a standard deviation of 2.18.

Through the perspective of years of experience in DT, the results show that those professionals less experienced in using DT considered it more challenging to select DT techniques (Figure 4.16). The results also show that years of experience in DT are a key

[17] In Q3 the participants could select more that one single option as their answers.



	Less than a year	1 to 3 years	4 - 7 years	More than 8 years
Min	2	1	0	1
Max	10	10	9	7
AVG	5,19	4,87	3,48	3,22
Standard deviation	1,92	2	2,35	1,93

Figure 4.16: Survey participants' difficulty in selecting DT techniques

factor in terms of difficulty in selecting DT techniques. For instance, professionals with less than 1 year of experience in DT have 5.19 (average) on a scale from 0 to 10 as their difficulty in selecting the set of techniques. On the other hand, the difficulty level for selecting DT techniques decreases to 3.22 (average) for professionals who have been using DT for more than 8 years. The mean value trend line in Figure 4.16 shows this result.

Figure 4.16 also shows that even with the increase in experience in years of professionals in using Design Thinking, there are still professionals who present high difficulties (maximum values of each experience range). For example, 9 was indicated as being the level of difficulty for selecting a technique by professionals with 4 to 7 years of experience (max value row), and 7 was indicated as being the level of difficulty for selecting a technique by professionals for professionals with more than 8 years using DT for software development (max value row).

Q4 reinforces that selecting DT techniques in software development might be a challenge. Although accumulating experience in DT, there are IT professionals who have elevated levels of difficulty when selecting DT techniques. Thus, it is important to investigate this challenge in detail and that computational resources that support IT professionals in choosing DT techniques in software development may be of help.

Therefore, the results of the exploratory studies show that there is a large set of DT models, including many techniques, which professionals choose for different reasons and goals. In addition, the exploratory studies' results show that the professionals who want to use DT have a certain level of difficulty in selecting techniques to use in the DT activities. Therefore, there is evidence that selecting DT techniques is a complex decision-making problem.

Our exploratory survey presents limitations in that we cannot generalize to the entire universe of software development since we conducted the survey only in the Brazilian scenario. Therefore, the answers might only represent the participants' point-of-view and not the whole organization of which they are part. To mitigate them, we invited participants who work for global software development companies that are multicultural by nature. Nevertheless, these limitations represent opportunities to replicate this survey in different countries. These replications would allow the community to build a broader view of DT usage and its integration with agile methods.

Thus, to better understand the problem and to seek solutions for it, we performed a meta DT session, as we detail in Section 4.3.

4.3 Meta Design Thinking Session: Problem Space

By bringing the user needs to the center of the discussion, DT improves team communication and facilitates knowledge domain acquisition, a well-known issue in software development [154]. Given its interactive and dynamic nature, DT has also considered an easy-in integration with and a way to boost agile development [213]. Despite the use, navigating in this new world might be challenging.

Literature discussing DT as a set of techniques has grown. It offers many techniques that form the toolkit to perform DT activities themselves. However, despite the large number of studies reporting on the use of DT in software engineering [65], we still have no consolidated knowledge of how developers choose DT techniques and what criteria and sources they consider for supporting such selection.

We identified in our previous literature mapping (see Section 4.1) initiatives such as the Luma Institute's Innovating for People Toolkit [159], which explains the meaning of each technique and suggests related techniques for a certain purpose, or the IDEO Toolkit [115], which recommends techniques based on a predefined set of questions that a professional might ask about on the use of DT. These examples demonstrate that choosing a technique is not trivial.

Although highly used by practitioners, both initiatives fail to consider the context in which DT is being used. For instance, if the team has no previous information about the stakeholders, selecting interviews might be of more interest to learn about their profile rather than a focus group session in which people might not get along and waste effort. DTA4RE - Design Thinking Assistant for Requirements Elicitation [245] is a first step towards aiming to tailor the recommendation process. However, DTA4RE is still limited in the number of contextual factors it considers to recommend a technique and on not take into account any feedback from those who use the recommendations. Nevertheless, there is a lack of studies investigating the decision process of which techniques to use and detailing which contextual factors (e.g., previous knowledge about the problem to be solved, customer engagement, etc.) affect such decisions.

In this section, we further investigate the use of Design Thinking techniques in software development, aiming to support IT professionals in selecting what techniques to use. To achieve our goal, we conducted a Meta Design Thinking session. We named it a meta-DT session since it was a Design Thinking workshop focused on deepening the understanding of the use of DT by itself in software development.

The meta-DT session worked as a bridge between the understanding and the solution proposal approach in our DSR-based research methodology (see Figure 3.1). We split our meta-DT session into 2 spaces: a problem space and a solution space. This section focuses on the problem space. It comprises the framing of a research problem that takes into account the previous results that we obtained through our exploratory studies. The solution space is explored in Section 5.1[18].

4.3.1 Design of the Meta-DT Session

We started the Meta-DT session by posing the following research question: *How can we support software development professionals to select DT (set of) techniques during software engineering activities such as requirement engineering?*

The session was moderated by a Requirements Analyst with 2.5 years of experience in DT. She designed the session into the following activities: i) problem understanding based on the presentation of the working question; ii) problem definition, which aimed to discuss the need for a tool further; iii) ideation using the Brainwriting DT technique to generate ideas; iv) convergence activity using the Affinity Diagram DT technique to find similar features; iv) prototyping the solutions (paper-based prototypes defined by two mixed groups of 5 members); and v) choice of solutions and presentation by each one of the groups through a voting activity.

Considering the moderator's previous experience in conducting DT sessions, she organized the problem understanding and problem definition activities with the following steps: i) presentation of a working question to explore in-depth the problem at hand, and ii) problem definition to further discuss the need for a tool. This section focuses on the steps of problem understanding based on the presentation of the working question and ii) problem definition, which aims to discuss the need for a tool further. Activities iii) to v) are presented in Section 5.1 (Solution Space).

[18]This study was published in the Ibero-American Conference on Software Engineering (CIBSE). Ref.: [194]

4.3.2 Meta-DT session: problem space results

A total of 10 people participated in the activity – 5 of them from industry, with an average of 3 years of experience working with DT, and 5 graduate students with DT as a research topic.

For the Problem Understanding, we used as input the information that we have collected through the conduction of the previous exploratory studies: a Systematic Literature Mapping and an Exploratory Survey with professionals from the agile software development industry.

During 1 hour-long, the participants discussed the following results:

- Systematic Literature Mapping [195]: We learned that there is a large set of DT models, DT techniques, and different DT techniques' selection strategies reported in the literature. We also identified tools (e.g., DTA4RE, IDEO DT) that support the identification and selection of DT techniques. Still, none of them takes the context in which the selection of the techniques takes place.
- Survey with IT professionals from the Agile community [211]: We learned that 1/3 of them had a difficulty of 6 or up points (scale 1 to 10) in selecting a technique. We also learned that 83% of them chose techniques based on the product context, 81% on their previous experience (which includes learning from others), and 67% on the fitness of a certain technique to a certain DT working space phase (e.g., interviews to support discovering). Results showed that selecting a set of techniques might be challenging and shed some light on selection criteria.

At the end of the first half of our meta-DT session (problem space), the participants converged on a problem definition. Section 4.4 presents the problem framed as a result of the meta-DT session, expressed as a Technological Rule of the DSR methodology.

4.4 Research Problem: Framing the Problem's Construct

The problem that we observed in our SLM points to several DT techniques and a lack of resources to help deciding which techniques to select. The Survey showed that IT professionals select techniques in different ways without considering the context information and the experience of other professionals.

Next, following the DSR-based research framework and motivated by the findings we obtained with the exploratory studies, we worked on the definition of a technological rule. The technological rule represents what problem the thesis proposes to solve. We defined the first element of our technological rule following the structure defined by Runeson *et al.* (2020): To achieve effect/change in a situation/context apply intervention.

> To support IT professionals' decision-making to select which DT techniques to use in a specific development scenario.

In this activity, we also defined the relevance cycle in our DSR-based research method. As we mentioned in Section 3, the relevance cycle aims to identify and understand the application context and the research problem. We identified that our research is relevant to IT professionals who use DT in software development wishing to decide on what DT techniques to use for software development.

Since that we defined the relevance of our research, we moved on to Solution Design and to Solution validation activities in the DSR, as we show in Chapter 5.

5. ITERATION 1: SOLUTION PROPOSAL AND EARLY EVALUATION

This chapter describes the first iteration of Solution Design and Validation Approaches in the DSR framework that we followed in this thesis. In total, we run 3 iterations in the DSR framework. It includes the activities that we conducted to propose and validate Helius[1], a DT techniques recommendation system to aid IT professionals in selecting DT techniques for software development.

We started Iteration 1 of the Solution Design activity in the second half of the meta-DT session, referring to the solution space of a DT session that uses the double diamond as the DT model.

5.1 Meta Design Thinking Session: Solution Space

As mentioned in Section 4.3.1, we conducted a Meta-DT session to identify, define, and propose a solution for the research problem. The following Technological Rule guided our meta-DT session: *To support IT professionals' decision to select which DT techniques to use in a specific development scenario.*[2]

In a two-minute time slot and using post-its, the same professionals who participated in the Problem Space activities (see Section 4.3.1) described what she thought was essential to provide relevant information to interested parties about the established problem. This ideation activity resulted in 24 proposed insight cards for the established problem. Then, to gain a deeper understanding of this scenario, an Affinity Diagram was organized (see Figure 5.1). This diagram allows the organization and grouping of results (insight cards) according to their similarities, dependencies, or affinities, clustering macro areas or features that delimit the addressed theme [262].

The 4 resulting identified macro features (in Portuguese, from left to right) are: i) qualified decision-making; ii) prediction-based recommendation; iii) techniques presentation; and iv) dynamic visualization of recommendations.

Next, the participants were separated into two mixed groups of practitioners and graduate students to carry out a prototyping activity. We named these groups as Group 1 and Group 2. The moderator introduced two personas, Joano and Sindi, presented in Figure 5.2 and previously created for the session to present the characteristics of potential tool's users, fostering the proposal of solutions that take into account the users' needs. Each

[1] Helius means the search for clarity, originating from the Greek that represents the personification of the sun. Clarity means supporting decision-making.

[2] This activity was published as an article in the Journal of Software Engineering Research and Development (JSERD).

persona was assigned to each of the groups to work on. Group 1 was assigned to the Joano persona and Group 2 to Sindi.

Through prototyping, for about 30 minutes, each participant individually designed a first round of paper-based low-fidelity prototypes to address the 4 identified macro features. Figure 5.3 shows the prototypes related to the macro feature iii) presents the techniques. Subsequently, each group was instructed to review the designed prototypes, group them by features and vote by posting a yellow post-it on the grouped set of prototypes by the macro feature that most represented what the tool should offer.

Table 5.1 summarizes the identified features per persona grouped by macro feature and the respective amount of votes attributed to them (columns (V_1) and (V_2)). The voting aimed to highlight whether certain macro features stood out and deserved priority attention or a deeper discussion. Each participant could choose between one to three features. Features with zero voting indicate that none of the participants considered it a priority. The groups differ on what they consider to be most relevant. While Group 1 considered that the visualization of techniques was the most relevant macro feature (6 out of 10 votes), Group 2 was torn among 3 of the macro features, with no predominant decision. Given the distinct persona behaviors and needs, this divergent scenario was not considered an issue.

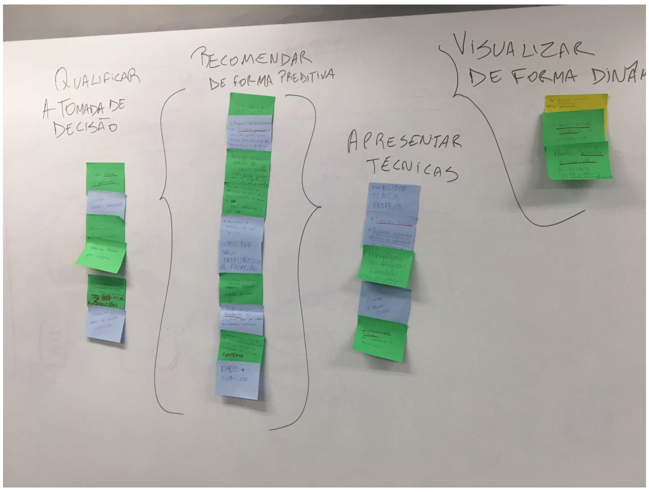


Figure 5.1: Affinity Diagram identifying the macro features for a tool

Name **Joano** 	**Profile** 38 years old Single Father of a 6 years old Designer Taking a fellowship in Design Owns a car
Behaviour Hurried Studies a lot Shy Likes TV series Tries to jog Social media blogger	**Needs** Stay connected 24x7 Optimize time to complete his activities

Name **Sindi** 	**Profile** 24 years old Married Facilitator Uses transportation apps BSc in Software Engineering Travels a lot for work
Behaviour Enjoys travelling Communicative Engaged with the community Friends with lots of people Pays attention to details	**Needs** Trust people Be successful in her area of work

Figure 5.2: Personas Sindi and Joano used in the DT session

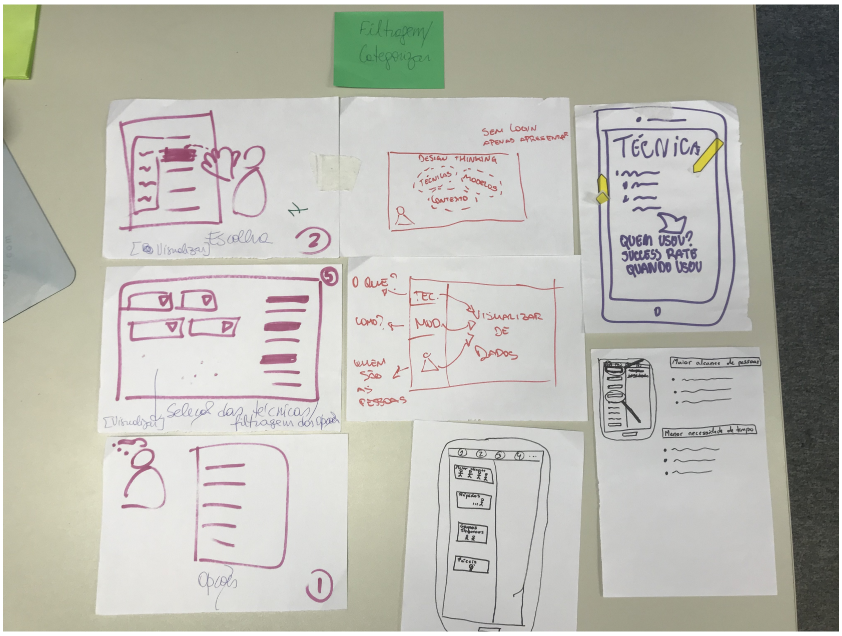


Figure 5.3: First round: low-fidelity prototypes for the macro feature iii)

Table 5.1: Identified features per persona grouped by macro feature

Macro Feature	Joano Persona (Group 1)	V_1	Sindi Persona (Group 2)	V_2
i) Qualified decision making	- Organize Techniques by Category (7a) - Qualify Decision Making (7a, 7b, 7d)	0	- Suggest Complementary Technique (7a) - Organize Techniques by Category (7a) - Qualify Decision Making (7a, 7b, 7d)	4
ii) Prediction-based recommendation	- Get to know similar cases (7b, 7c, 7d) - Consider the context of use of DT (7a, 7c) - Provide feedback (7d)	1	- Suggest tools predictively (7a) - Receive examples of the application of the technique (7b, 7c, 7d) - Consider the context of using DT (7a, 7c) - Provide feedback (7d)	2
iii) Techniques presentation	-Visualize techniques and models (7a, 7b, 7c) - Search for technique recommendations in a systematic way (7a, 7c)	3	- Visualize the techniques and models (7a, 7b, 7c)	0
iv) Dynamic visualization	- Simulate the use of combined techniques for a particular purpose (7c)	6	- Dynamic and interactive visualization (7a, 7b)	2

To conclude, each group presented their results to one another and explained their voting and motivations behind it. This discussion promoted common ground among the meta-DT session participants and made it possible for the group to conclude that they had reached a tool vision.

5.2 Requirements Elicitation Activity

Upon a 1.5-hour-long session, we conducted a requirements specification activity. This activity was conducted by the 5 graduate students who refined the tool's requirements. Based on our previous experience in elicitating requirements, we chose the following techniques: i) User Journeys [149], to represent the needed steps for a user to achieve their (business) goals, ii) Service Blueprints [149] to describe how the 'service' (tool features) may be offered to the user (touch-points), covering the entire journey, identifying points for improvement and business opportunities, and iii) Low-Fidelity Prototypes [149], to define how users will interact with the tool[3]. Specifications were defined by 3 of the students and reviewed by the other 2 until a consensus was reached.

5.2.1 User Journeys

Proceeding with the requirements-gathering exercise for the tool proposed, we started with the definition of 5 User Journeys. Figure 5.4 shows a User Journey developed for Persona Sindi, highlighting her main actions when looking for a technique and aiming to use it in her product development. The identified functional requirements derived from this user journey are highlighted in Figure 5.4, described next:

- Access the system: user access the tool;
- Recommend techniques based on a self-updated visualization graph: user can receive technique recommendations and self-update the tool, collaborating with other system's users;
- View techniques details: user can access detailed information about a certain technique, like when, how, and why to use it;
- Attach/Add techniques from the visualization graph to a new project: user has the opportunity to choose a technique and associate it with a new project;
- Create a new project: user can create a project and manage the used techniques and previous experiences, updating the visualization graph.

5.2.2 Service Blueprints

As mentioned, Service Blueprints (SB's) are used to visually present the detailed specification of aspects of a service (business feature), from the user's perspective, and other relevant parts that may be involved.

Figure 5.5 presents an SB for the persona Sindi related to the User Journey shown in Figure 5.4, aiming to get new insights and a deeper understanding of the problem. In this service blueprint, the user initially accesses system (A) and selects from the suggestion graph some techniques for his DT project (B). The user can also see information

[3]The full report can be found at http://bit.do/CIBSE2020DesignThinking

about the selected technique (C) as well as linking to the running project (D). Figure 5.5 also presents other expected functionalities through the relationships shown in each of the horizontal lanes' physical evidence (i), customer actions (ii), contact with backstage actions (iii), and support processes (iv).

5.2.3 Low-fidelity prototypes

Considering the insights gathered from the Users' Journeys and the Service Blueprints, we further detailed the features through their prototypes. We designed a second round of paper-based prototypes mapping the results from the two referred elicitation techniques to consolidate our understanding as presented here.

Techniques Recommendation Graph Screen (Figure 5.6-(a)) shows the techniques within a graph according to a specific filter. By default, the graph is set to generate the vertices and edges according to the most used technique. Therefore, the starting and central node from this graph present the most recommended technique, followed by the

User Journey – Sindi

UJ001 – Create a new project = ______

1. Sindi wakes-up cheerful to finish this week with this beautiful friday morning;
2. Then, she gets out of the bed and takes a shower;
3. After the shower, Sindi joins her husband in the breakfast table;
4. In the midst of conversations with her husband, Sindi remembers her trip to São Paulo, which involves the projected she was assigned for by her boss;
5. Therefore, she finishes her breakfast, prepare her material and heads to the airport;
6. At the airport, after finishing the check-in, Sindi finds out her flight was delayed due to aircraft maintenance;
7. Hence, Sindi enters in the DT app looking forward to anticipate the project's preparation - **ACCESS THE SYSTEM;**
8. In the Home Screen, she access the feature "recommendation graph" and starts to see the most recommended techniques based on users choice and displayed as a graph - **DISPLAY AS A GRAPH THE RECOMMENDED TECHNIQUES BASED ON A RETRO FEED ALGORITHM;**
9. Curious about some techniques she never heard of, Sindi holds her finger on the screen for a few seconds and a pop up appears with details about the technique in question - **VISUALIZE DETAILS AND INFORMATIONS OF TECHNIQUES;**
10. After navigating through the recommendation graph and visualizing some technique's details, Sindi gather the ones she needs to use for her project and proceeds with its creation - **ATTACH SELECTED TECHNIQUES FROM THE GRAPH TO A NEW PROJECT;**
11. After loading the project creation screen, Sindi fills the name, participants and remaining details about the project;
12. Finally, Sindi reviews her inputs and details about the project and press the create project button - **CREATE A NEW PROJECT**

Figure 5.4: User Journey View techniques in detail for Persona Sindi

next most recommended nodes that relate to the first (e.g., Brainstorming followed by Journey maps and Personas). Thus, our tool presents a network of DT techniques.

In addition, if a user is not familiarized with any of the techniques, each of them contains descriptive information, e.g., definition, case scenarios to use, and others (see Figure 5.6-(b)). When the selected technique is selected, its respective data have to pop up inside a little rounded square, delivering the user's continuous use of the feature, despite having to go back and search manually about the given technique.

Figure 5.6 illustrates two other features. Figure 5.6-(c) provides the concept of a project to store related information held together, whereas having to seek them separately all the time. Therefore, the screen exhibits blank fields to be filled about the project's crucial details (e.g., name, description and participants), a set of available DT models to fit in their project and also a list of the previously selected techniques with the graph.

Finally, Figure 5.6-(d) illustrates a screen for capturing user feedback from the use of DT techniques in their software development projects, chosen through the recommendations made by the tool we are proposing in our work. In this screen, the user can inform the result of the application of a specific technique (e.g., Persona), filling in fields like what was the experience of use, if it suggested any alteration or if it was done alternatively, how many times already used, would be used again, and how to rate it (on a level of stars ranging from 1 (not suitable) to 5 (very appropriate)).

This last screen represents a core feature of our tool: the exchange of information and experience among DT professionals, creating an effective and consolidated communication channel and establishing a community environment, especially for those who apply DT in software engineering. Thus, our proposal is not limited to an information guide but defines it as a collaborative environment that enables the exchange of experiences be-

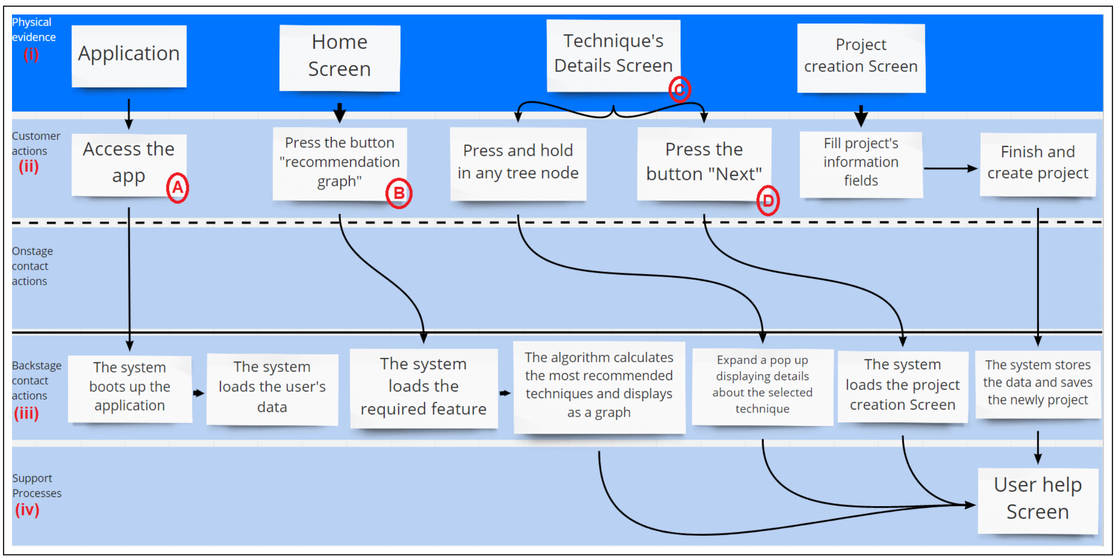
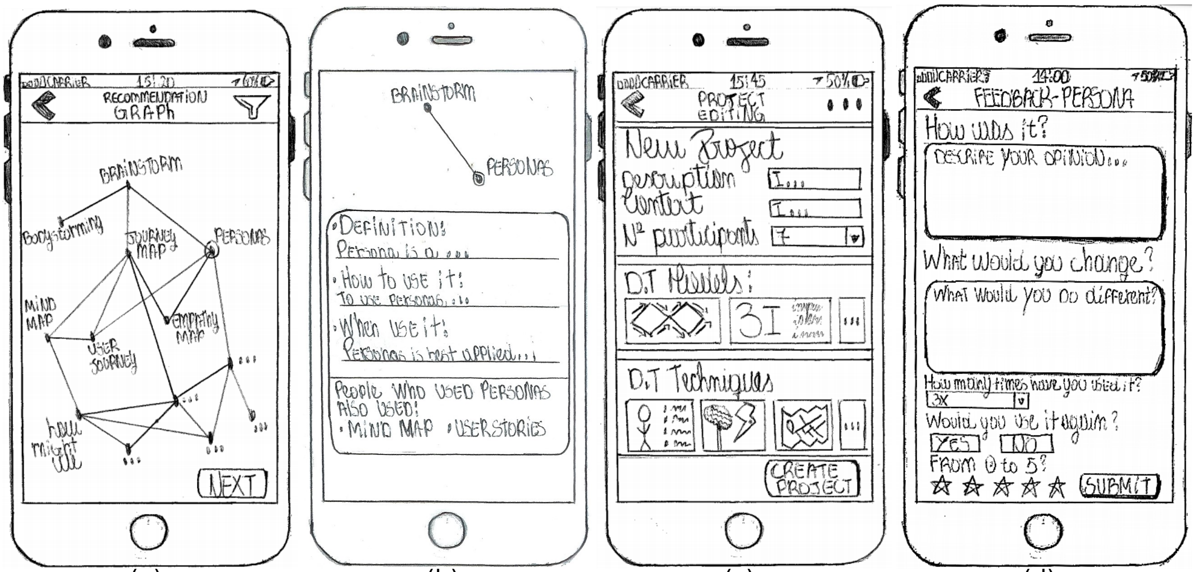


Figure 5.5: Service Blueprint for Persona Sindi



(a) (b) (c) (d)

Figure 5.6: (a) Graph; (b) Technique detail; (c) Project creation; (d) Feedback

tween DT users. This feature should also be further modeled ahead as per the use of the first tool draft to be made available.

5.3 Early Solution Proposal Evaluation

Once we concluded the creation of low-fidelity prototypes, we moved to the validation approach in the DSR framework. In Iteration 1, we conducted an early interview evaluation study with DT practitioners in software development.

5.3.1 Interview-based Evaluation Study

We interviewed 5 DT practitioners (a product designer, a service designer, a product owner, a business analyst, and an IT manager) of 2 multinational IT companies: a large Brazilian TV broadcasting company, a cooperative bank and an IT provider with an average of 3.5 years of experience with DT in software development.

We first openly asked how they select the techniques they use to later present them with our recommendation tool idea and a sample of printouts of low-fidelity prototypes to discuss in detail. Four interviews took place during coffee breaks of the DT track of an industry-based developers' conference. Each interview lasted an average of 20 min. We sent the prototypes to the interviewee by e-mail and then we discussed our proposal.

Here, to maintain the confidentiality of both practitioners, we identify them as P1, P2, P3, P4, and P5. We asked: i) how do you select DT techniques? ii) how do you deal with changes in the technique selection during a DT session, if any? and iii) how would you welcome a tool that recommends your techniques considering your product context and feedback from others?

Regarding question (i) choice of DT techniques, the participants reported that they select considering the customer's knowledge and feeling, according to P2; that they learn to choose from the experience gained from years of application, and that ends up creating a particular set of techniques by the results that have already been obtained in previous applications, as mentioned by P3. In addition, the techniques can be chosen and determined according to the DT working space, either for understanding the problem, for ideation, or for developing the solution. P4 unveiled it:

- –"*We learn to choose techniques over time because each technique has a result. If one wants to understand the problem, one uses certain techniques. One defines one's own toolkit over time. Also, if one has to think about solutions, one has these (certain techniques) here that help one get there*" - P4;

On the need to look for other techniques while conducting a DT session (question ii), interviewees responded that situations arise when users/participants are not engaged or do not understand how the technique works and need to change or adapt the technique for more meaningful results. P3 and P4 stated that:

- –"*There are situations in DT Workshops where the use of some techniques does not work, so the moderator/designer must choose another technique from their experience. There are also instances when participants find it challenging to use a particular technique, so one needs to use others*" - P3;
- –"*It happens that one has to change in the middle of a workshop because the customer does not respond well to a certain activity. In this situation, empathy must be used to understand what is happening and get around the situation. Changing the technique is often helpful in those situations.*" - P4;

P5 and P2 suggest making combinations between different techniques, observing that with the applied variations, going beyond the pre-established models and keeping the freedom for exploring the Designer's mindset, one gets a more efficient result.

- –"*Experiment variations between techniques (a mix of techniques), because this ensures the bias of the Designer and not just replicating methods [...]*" - P5;
- –"*You test if this technique combines with this one.*" - P2;

Regarding the need for a recommendation tool (question iii), the interviews showed that it would be helpful and of great help to professionals who use DT. They suggested that the DT user might input some data such as *"Do you already have the problem* *defined?*" to know if the person (client) already has the scope of the problem defined; *“Do you know users?*" – then suggest a particular set of tools.

For P1, the tool should be geared to the DT community by fostering information exchange (feedback, more effective techniques); should have explanations of concepts, such as whether the solution to the problem is developing a mobile app, providing useful Minimum Viable Product (MVP) tips; and should clearly present the techniques to understanding the problem, for ideation, and for creating the solution. Also, it should establish a sense of community, with a forum for users to interact among themselves, going beyond a simple guide, and provide metrics for evaluation (number of users who accessed in a period, user satisfaction when using).

Therefore, we realized through the early evaluation task with 5 professionals that the selection of DT techniques is made considering the customer’s knowledge and feeling, previous experience, or according to the working space of DT. We have also learned that professionals might change a DT session’s selected techniques due to a lack of participant engagement. In addition, the professionals mentioned that a tool that considers the previous experience of the professionals to recommend techniques seems to help with the decision-making of selecting DT techniques for software development. However, there is still room for the solution’s refinement. In Section 6, we show Iteration 2 in the DSR-based method, where we performed a tool’s requirements refining activity and conducted an initial tool validation.

6. ITERATION 2: SOLUTION REFINING AND INITIAL VALIDATION

Motivated by the results of Iteration 1 (Chapter 5) and seeking improvement to the solution, we performed a second iteration in the DSR framework. We refined the requirements of the proposed solution and then we validated the solution with industry professionals. We also used the DESMET method for features analysis as described next.

6.1 Requirements Refining Activity

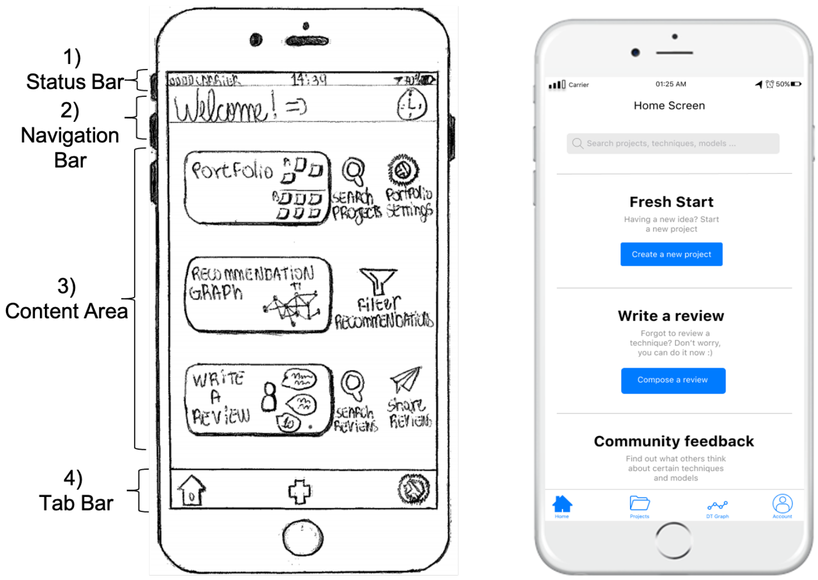
After the early evaluation activity, inspired by the feedback gathered with the DT professionals and based on our defined artifacts (user journeys, blueprints, and low-level prototypes), we performed a requirements refining activity. We transformed the low-level fidelity prototypes into high-level prototypes to further detail each of the previously identified features. For this transformation, we used Figma as a prototyping tool.

Supported by 2 graduate students and reviewed by 2 other researchers (an assistant researcher and a senior researcher), the requirements refining activity was based on the use of the Figma prototyping tool [84]. Figma is a web-based tool that allows the definition of transitions between prototype screens and allows the creation of content into multiple layers' structure. Using Figma, we created high-level prototypes to refine our understanding of the requirements. We transformed the low-level prototypes generated in the Solution Design cycle into high-fidelity prototypes.

6.1.1 High-level prototyping

Figure 6.1 presents an example of our low-level (a) to high-level (b) fidelity prototype transformation. We structured all the high-prototypes into 4 basic areas: (1) a status bar, a thin line containing system information; (2) a navigation bar, to show information about the user's current location within the App; (3) a content area, to represent the content, where the general buttons, inputs and other components are positioned in, and finally; (4) a tab bar, located at the bottom of the canvas–used area to draw–, used to show all tool sections, which in our example are: the home screen itself, the projects, the recommendation graph, and the user profile.

As a result, we transformed the 19 low-level into 62 high-level prototypes, as illustrated in Figure 6.2. We started the high-fidelity prototyping process with the creation of an initial screen, drawing the 4 main areas of the screen structure as previously mentioned–the status bar, the navigation bar, the content area, and the tab bar. We used



(a) Home Screen Low-fidelity prototype

(b) Home Screen High-fidelity prototype

Figure 6.1: Low fidelity redesigned to high-level fidelity prototypes: Home screen

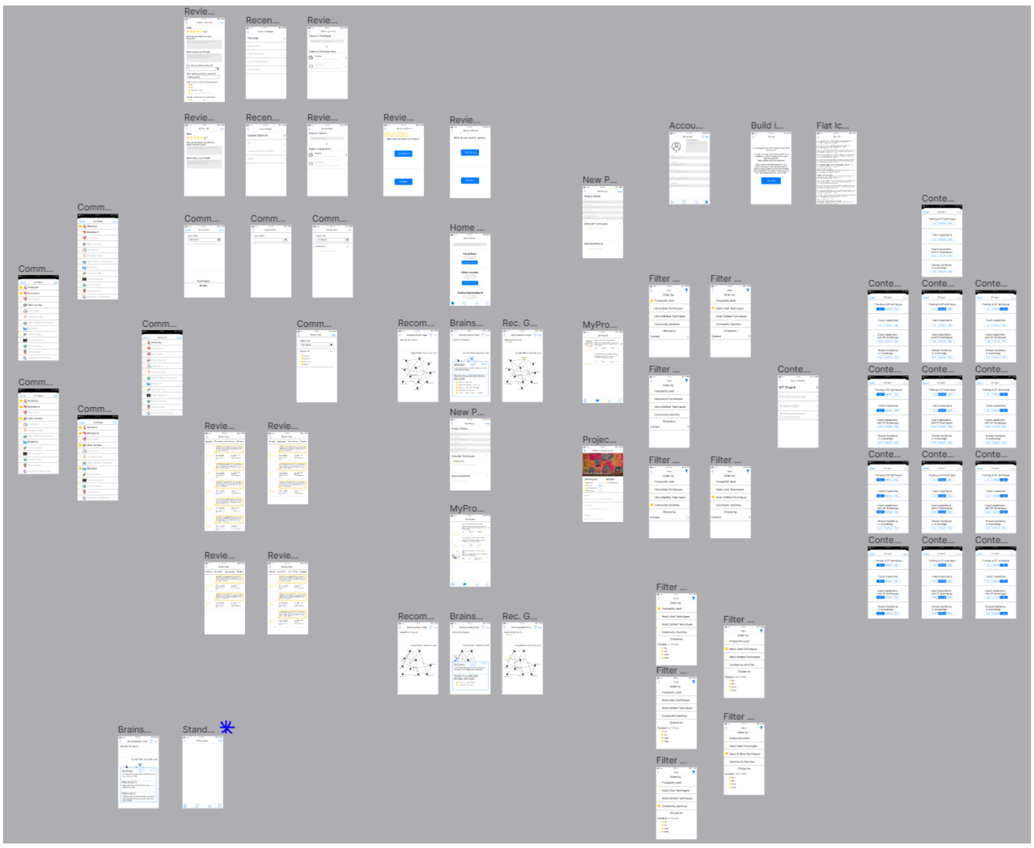


Figure 6.2: Overview of the screens created using Figma

the initial screen as a standard prototype to facilitate the creation of the other screens of the solution. After that, using the initial prototype, we built all the other high-level prototypes in the solution. For each defined feature, we created one or more high-level prototypes (see Figure 6.3). During this process, we peer-reviewed the drafted high pro-

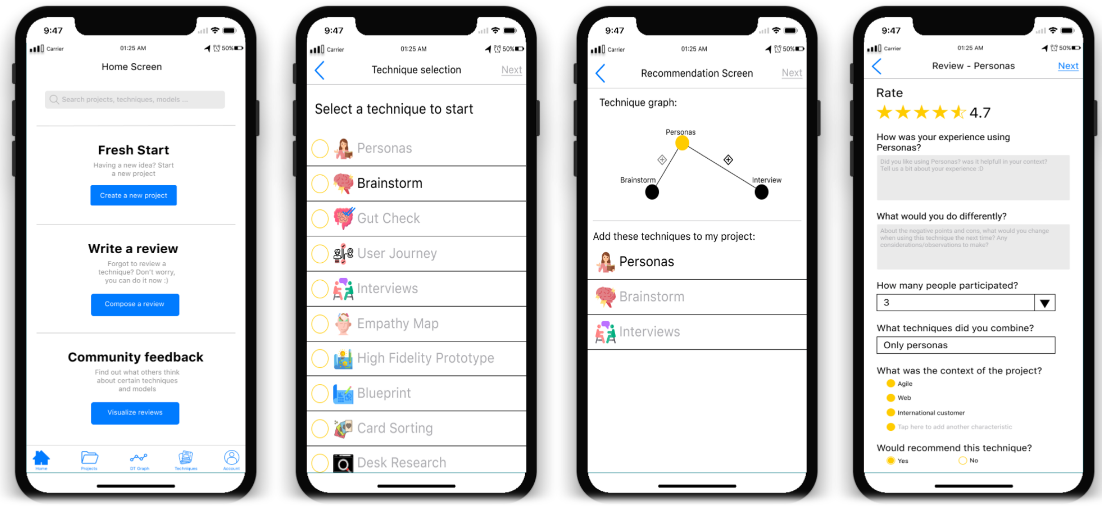


Figure 6.3: High-level fidelity prototypes generated using Figma

totypes among the authors, resulting in improved versions. This process allowed us to obtain a better and more refined understanding of the specified user requirements.

Next, based on the high-fidelity prototypes we generated for our solution, we extracted the features for Helius. As it is a recommendation and a collaborative system, we designed a set of features aiming to assist software development professionals in the selection of DT techniques according to their needs. Thus, Helius is a collaborative recommendation tool that takes project context and previous experiences to recommend DT techniques in software development.

Therefore, we defined the target audience, the features, and a subset of the screen overview to show how Helius works. The tool's target audience is professionals who work with software development (e.g., developers, requirements analysts, UX designers), especially those who are beginners in the use of Design Thinking in the context of software development but not limited to this profile.

Table 6.1 presents a set of features and respective sub-features for our tool proposed based on the DT session and on the high-level fidelity prototyping activities. We organized Helius into 6 main features: (F1) DT Techniques Recommendations, (F2) DT Techniques Filtering, (F3) DT Techniques Evaluation, (F4) DT Techniques Community Feedback, (F5) DT Techniques Information, and (F6) DT Techniques and Project Management.

The DT Techniques Recommendation feature (F1) is Helius' core feature. This will help users to receive recommendations of the techniques most suited to their needs according to the project characteristics and contextual information (e.g. project domain, stakeholders' commitment level, team expertise in Design Thinking). The recommendations are performed by Helius using the DT techniques information and using information from the user, making the tool a collaborative system too. Therefore, once the user evaluates the techniques selected to be used in a project, Helius uses this data for recalculating the recommendation graph.

Table 6.1: Helius Features

ID	Feature set	Sub-feature ID	Sub-feature description
F1	DT Techniques Recommendations	F1-SF01	To consider the combined use of techniques in projects to recommend DT techniques
		F1-SF02	To consider contextual information from the user's project to recommend DT techniques
		F1-SF03	To consider feedback from other users to recommend DT techniques
F2	DT Techniques Filtering	F2-SF01	To consider techniques characteristics to filtering similar DT techniques
		F2-SF02	To consider project context information to filtering similar DT techniques
F3	DT Techniques Evaluation	F3-SF01	To allow evaluation of a single technique by the user
		F3-SF02	To allow evaluation of the techniques used in conjunction in a DT project by the user
F4	DT Techniques Community Feedback	F4-SF01	To allow filtering feedback by technique
		F4-SF02	To show related feedback for projects and techniques used in conjunction
F5	DT Techniques Information	F5-SF01	To show detailed information about each DT technique
		F5-SF02	To show the related uses of a technique with other techniques
F6	DT Techniques and Project Management	F6-SF01	To manage a project which uses DT techniques
		F6-SF02	To share project data with other team members

The DT Techniques Filtering feature (F2) allows the users to filter techniques based on the characteristics of the techniques. This feature works as a catalog of techniques in which the user can filter them using a set of criteria, including the working space, the goal, among others. In this way, Helius, in addition to recommending the techniques (Feature F1), allows filtering DT techniques, helping the user to select the most appropriate techniques for her project. This feature supports professionals who already have experience in DT, instead of requesting a technical recommendation, prefer to filter to learn about other techniques.

The DT Techniques Evaluation feature (F3) allows the users to make evaluations considering the techniques they have used in their projects. Therefore, once the user selects a technique for evaluating, Helius shows a form composed of a field for evaluation rate, and 6 questions: *"How was your experience using the technique"*, *"What would you do differently"*, *"How many people participated in the use of this technique?"*, *''What techniques did you combine together?"* and *"What was the cost of applying a technique?"*.

The DT Techniques Community Feedback feature (F4) combines the evaluations of the users about their experiences using a technique. This feedback allows the users to know each technique based on previous experiences[1].

[1] In this version of Helius, there is no evaluation moderation.

Helius also provides the DT Techniques Information feature (F5). This feature organizes a collection of data about each technique, such as the technique definition, how to use it, and suggestions about when to use it. In addition, Helius includes community feedback on the technique information. Feature F5 is further explained in Section 7.1.

The DT Techniques and Project Management feature (F6) allows the users to create a project, including data such as project name, project description, the participants' profiles, project due date, project tasks, and team members. No information that could compromise the project's confidentiality is requested or exposed. Therefore, once the users have created a project, they can request a recommendation. Also, it is by using this feature that Helius provides the feature for technique evaluations (ii). Users can evaluate the techniques they have used in their own projects.

Figure 6.4 illustrates a subset of screen prototypes of Helius. Screen (a) shows the start user interface, which provides functionalities such as "create a project" (F6-SF01), "write a review" (F3-SF01, F3-SF02), and view the "community feedback" about the techniques and projects (F4-SF01, F4-SF02). Screen (b) illustrates the creation of a project to insert DT techniques. It is also possible to insert DT models. Additionally, on Screen (b), there is the functionality: "Recommend me a technique" used to request DT techniques recommendations (F1-SF01, F1-SF02, F1-SF03). Once the user has selected the recommendation feature, Helius shows Screen (c).

On Screen (c), the user is able to select one technique. After that, Screen (d) is opened, showing a graph of DT techniques and their relationship, i.e., techniques recommended according to their usage in previous projects (F1-SF01, F1-SF02, F1-SF03). For example, on Screen (d), Personas was the selected technique, and the other two techniques, Brainstorm and Interviews, were present in the graph. It means that Personas x Brainstorm and Personas x Interview were already used in conjunction before within a project (F5-SF01, F5-SF02). In addition, on Screen (d), the users may select which techniques want to insert into their project (we show the selection of Personas to be added to

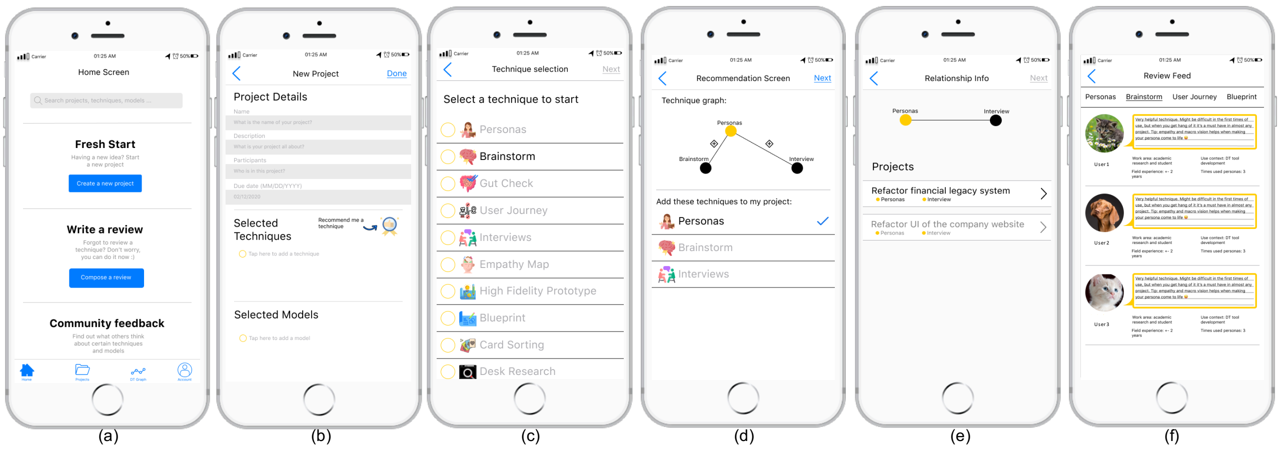


Figure 6.4: Recommendation tool's screens overview

a project, marked with a blue 'v'). Once the user has clicked in the square over a graph edge, which creates a relation with two techniques, Helius presents Screen (e). On Screen (e), there is information related to the relationship between the two techniques, such as the projects that they already were used in conjunction with (F5-SF01, F5-SF02). For example, on Screen (e), we show the relationship between Personas and Interviews and the project where these two techniques were applied in conjunction. Helius also builds a graph from filters applied by users on the characteristics of the techniques and of the project (F2-SF01, F2-SF02).

Finally, on Screen (f), we show the feedback registered by the Helius tool users about the DT techniques they had used (F4-SF01, F4-SF02). This feedback helps other users describe previous experiences with the use of DT techniques.

6.2 Requirements Validation Activity

After the requirements refining activity, we moved to the validation activity in Iteration 2 of the DSR-based approach. We used the high-level fidelity prototypes generated with Figma as input for designing and running a requirements validation activity. Thus, we organized the validation activity into 2 major tasks: (Section 6.2.1): validation environment setup using Quant-UX, aiming to design and prepare the environment for validating the specified requirements through prototyping, and (Section 6.2.2): feedback collection with DT practitioners, through a questionnaire-based data collection, aiming to validate our DT techniques recommendation tool for future implementation. As mentioned, the questionnaire was based on TAM considering the Perceived Ease of Use and Perceived Usefulness factors.

6.2.1 Validation Using Quant-UX and TAM

We used Quant-UX to simulate the interaction of the high-fidelity prototypes as a means to validate the requirements with potential users, i.e, Design Thinking practitioners. Quant-UX is a free prototyping and validation tool that allows to design and measure the users' interactions with a solution. Thus, we created the screens interactions aiming to collect experiences from the Design Thinking practitioners by using our DT techniques recommendation tool prototypes [218].

For validation purposes, Quant-UX includes features such as screen recordings, A/B testing, and QR Codes to share the solution with users and test the solution. Quant-UX also allows the creation of user interface flows, transforming the prototypes into interactive ones, and providing a functional perspective of the solution. It also includes features like (a) heatmaps, highlighting the points of user interaction on the screen; (b) user journey, showing which prototypes are executed by the user and in what flow, helping to discover usability problems, and (c) test evaluation, a statistical report describing useful data about the user interactions, like scroll visibility, number of views, and dwell time.

Using Quant-UX, we first imported the 62 high-level fidelity prototypes from Figma as images. Next, we created the interactions between the prototypes, defining the logical entries for generating user flows. Figure 6.5 illustrates an overview of the logic flow. The arrows portray the paths between an element and the screen that is summoned when that element is pressed.

Feedback collection with DT practitioners

Once the validation environment was set up, we collected the feedback with DT practitioners[2]. We invited 80 out of 158 respondents who have authorized us to contact them for follow-up studies. Seven of them (8.75%) accepted our invitation and participated in this validation study. Table 6.2 shows the participants (named validators here), identified as V1 to V7, their roles and their DT's years of experiences.

We sent a 4 minutes video to the practitioners explaining our research goals and introducing in general words how the tool will work. The video did not contain explanations about how the practitioners should use the features of the proposed tool, aiming to avoid

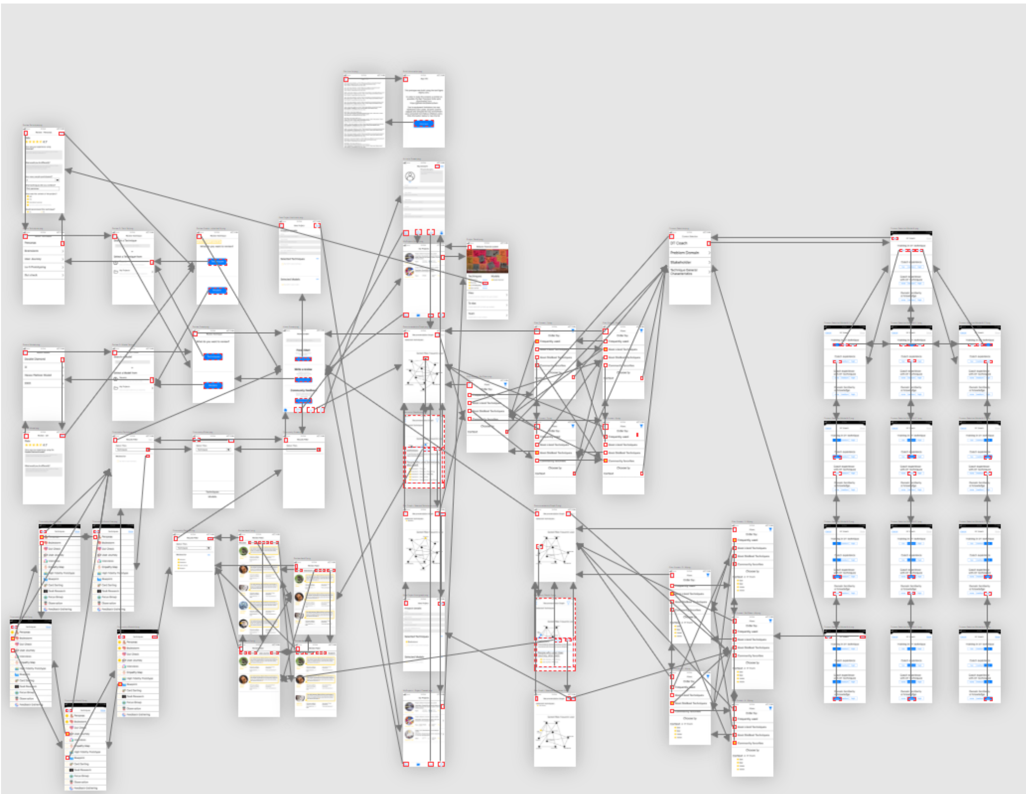


Figure 6.5: Overview of the screens flow built using Quant-UX

[2]These are distinct individuals from those who participated in the early evaluation study.

feedback bias. We also sent the link for the prototypes created using Quant-UX, and the TAM-based questionnaire link.

Table 6.3 shows the questionnaire that we created for collecting validation feedback based on the TAM model and on the Likert scale (Totally agree, Strongly agree, Partially agree, Partially disagree, Strongly disagree, Totally disagree). We posed statements considering Perceived Ease of Use (PEU) and Perceived Usefulness (PU) factors. Also, we posed 3 open questions (OQ) for collecting if the practitioners would use our recommendation tool, which are the positive features they perceived, and which are the features we should improve in our tool proposal. The questionnaire was peer-reviewed by a senior researcher who has conducted several studies using TAM over the years and piloted with a graduate student who also works in the industry.

Figure 6.6 shows the TAM's results of the Perceived Ease of Use, while Figure 6.7 shows the Perceived Usefulness of the recommendation tool, respectively.

- *Perceived Ease of Use:* The DT practitioners who participated in our validation activity indicated that the DT techniques recommendation tool can be considered an easy-to-use tool. With different agreement levels, the participants considered it easy to learn how to use the recommendation tool (PEU #1), acquire the tool's ability

Table 6.2: Participants of the tool's validation step

Participant ID	Role	DT Years*
V1	Head of Innovation	10
V2	Innovation Manager	3
V3	Engineer Support	5
V4	UX Designer	5
V5	Project Manager	5
V6	Software Engineer	3
V7	Designer	4

*Years of Experience using DT in software development

Table 6.3: Validation Questionnaire based on TAM model

Perceived Ease of Use statements for the recommendation tool:	
PEU #1	It was easy to learn how to use the DT techniques recommendation tool
PEU #2	I was able to use the recommendation tool to request recommendations for DT techniques
PEU #3	It was easy to acquire the ability to use the recommendation tool
PEU #4	It was easy to remember how to request DT techniques recommendation
PEU #5	I think the recommendation tool is an easy to use tool
Perceived Usefulness statements for the recommendation tool:	
PU #1	The tool allows to create projects and include DT techniques
PU #2	The tool allows to require recommendations of DT techniques
PU #3	The tool allows to evaluate the DT techniques used
PU #4	The tool allows to view information about the related techniques in the graph
PU #5	The tool allows to view other users' comments on DT techniques
PU #6	I consider the recommendation tool useful for selecting DT techniques
Open questions about the use of the recommendation tool prototypes:	
OQ1	Would you use this tool for selecting DT techniques in software development? Why?
OQ2	Which features of the tool do you consider positive for the selection of DT techniques?
OQ3	Which features of the tool do you think need to be improved for the selection of DT techniques?

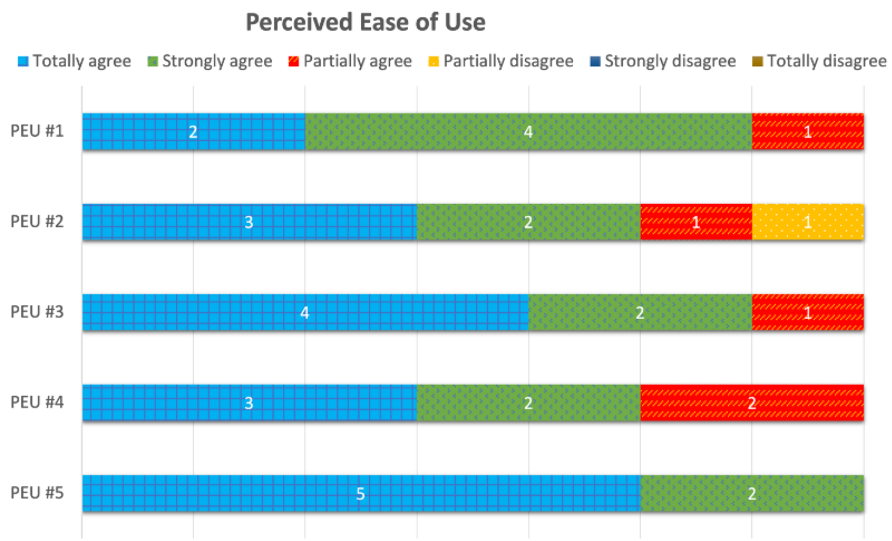


Figure 6.6: TAM results for Perceived Ease of Use

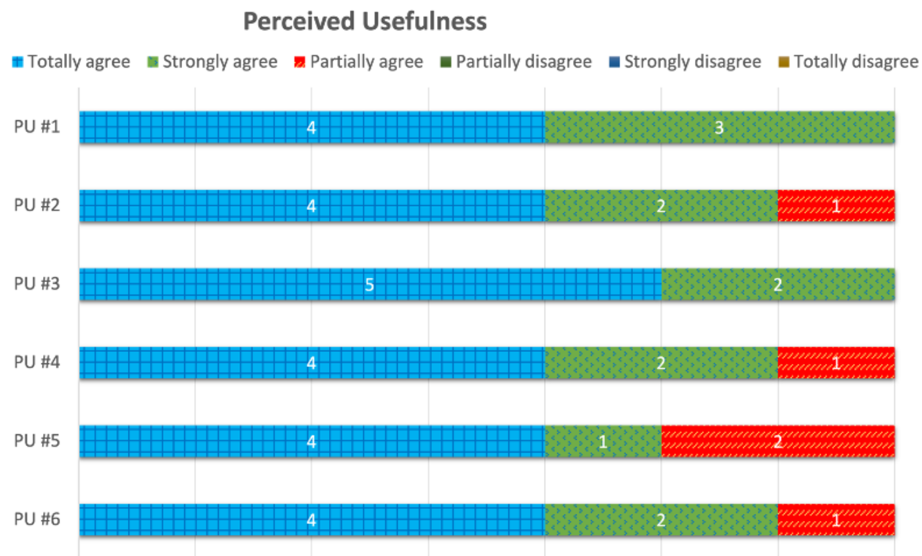


Figure 6.7: TAM results for Perceived Usefulness

(PEU #3), remember how to request DT techniques recommendation (PEU #4), and in general, the tool was easy to use (PEU #5). Only for the statement related to how easy it was to request recommendations for DT techniques (PEU #2), 1 DT practitioner partially disagrees with the ease of using the tool.

- *Perceived Usefulness:* Regarding the usefulness of the recommendation tool, most of the DT practitioners totally agreed that the tool allows: creating projects and including DT techniques (PU #1), to require recommendations of techniques (PU #2), to evaluate the DT techniques used (PU #3), to view information about the related techniques in the graph (PU #4), and to view users' comments on DT techniques (PU #5). The DT practitioners also agree that the recommendation tool is useful for selecting DT techniques (PU #6).

Regarding the questions about the features provided by the tool, the DT practitioners answered the following:

OQ1: Would you use this tool for selecting DT techniques for software development? Why?

The DT practitioners pointed out they would use the recommendation tool because it provides data about the experiences of professionals who used the tool, allowing them to better know the practices around DT techniques:

–"*The tool will become a data pool with several experiments and combined models of tools. This is cool to bring inspiration and best practices*" - V1;

–"*We can create shortcuts through the experience of other users*" - V2;

–"*This tool will help my choice of techniques to the point that I can learn about other users' experience with certain techniques*" - V7;

The participants considered the tool appropriate for selecting DT techniques and for engaging DT practitioners during DT sessions:

–"*The tool would be very useful to assist when I need to think about DT techniques to engage participants in the DT session*" - V5;

The participants also considered the tool useful since it provides information to select a DT technique according to how much time is required to use it, and also it was considered useful to professionals who do not have any DT experience:

–"*(The tool) allows me to see how much time is required to apply a DT technique because in order to select the techniques I always consider the time available I have to apply DT.*" - V4;

–"*I found it very useful for those who have no previous experience with DT*" - V6;

OQ2: Which features do you consider positive for the selection of DT techniques?

Two participants considered the recommendation graph as a positive feature:

- –"*The Recommendation graph [...] for a quick consultation (of the techniques) is amazing, especially for more experienced professionals*" - V1;
- –"*The suggested graph is an interesting feature. By clicking on it, it is possible to know what other techniques are related. The technique recommendation is great, it's like we have someone helping us to choose techniques*" - V7;

Other participants (V2, V3, V4, and V5) mentioned the reviews of the techniques as a positive feature since it provides information to the professional for choosing the appropriate DT technique. For instance, one participant mentioned:

- –“*(The tool) provides a view of the comments of other participants, which allows me to see what the experience with the techniques was like*” - V4;

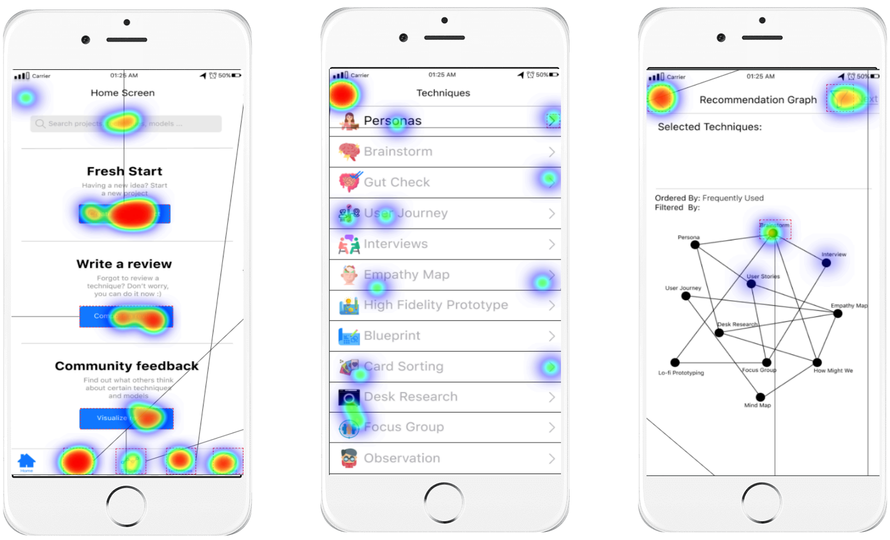
In question *OQ3: Which features of the tool do you think need to be improved for the selection of DT techniques?*, the DT practitioners pointed out improvements for some features of the proposed tool, such as:

i) techniques visualization: –“*To improve visualization of techniques and details of techniques.*” - V5;

ii) reviews presentation: –“*The organization by tabs could be changed*” - V1;

iii) project creation including DT techniques: –“*I think that when we describe a project, we have to start with the selected techniques. I thought the prototype didn’t give me this option to see if it worked*” - V3;

Still in the validation activity, using Quant-UX we captured heatmaps of the prototypes’ areas that the users clicked. These heatmaps helped us to analyze the user experience and to identify improvements to the proposed tool’s screens. Figure 6.8 shows examples of the heatmaps for the Home screen (a), for the technique selection screen (b), and for the graph recommendation screen (c). In the Home screen (a), the participants clicked on the main areas, such as buttons and the bottom tab bar, which allowed them to access the features of the tool. In the Graph recommendation screen (c), the



(a) Home screen (b) Techniques selection (c) Graph recommendation

Figure 6.8: Heatmaps of the use of the tool’s prototypes

participants were able to access the information of the technique shown in the graph, and also they were able to access the filtering features. On the other hand, the heatmap of the Techniques selection screen (b) shows that some users had some difficulty selecting a technique to see its information since there are clicks over the technique's name, but the area on the screen for this purpose is the icon placed on the right side of the techniques' name. Thus, this result indicates that its necessary to highlight the area to be clicked.

Thus, based on the results that we obtained in this study, we list some insights as a set of initial takeaways about our recommendation tool proposal:

- The tool seeks to provide information on DT techniques, going beyond the existing user's toolkit. The tool must allow the selection of other techniques through its recommendation system that should take into account a set of items for the recommendation such as previous use of a certain technique, user feedback, product context, and project characteristics;
- The tool should be valuable and able to assist both on-boarding novice users as well as expert ones during their DT sessions in software development. Both profiles found the tool idea useful;
- The tool should represent an innovative solution presenting a recommendation tool associated with a community-building environment through feedback. Thus, industry practitioners might collaborate with their community to foster the improvement of DT in software development.

In this study, we collected with the validation study the need of exploring the sense of community as a future direction of our tool, allowing the users to retro-feed the recommendation system with their experiences on the use of DT techniques, fostering an opportunity to learn with other professionals and to establish some shortcuts when selecting DT techniques. The results of the early tool evaluation and the requirements validation activities also indicated that our DSR-based research presents a theoretical contribution (Relevance element of the DSR).

We figured out that IT professionals select DT techniques based on the investigation of the customer's knowledge and feeling, on the learning with previous experience on the application of DT tools, on the creation of their own set of DT techniques, or based on the analysis of the DT working space and techniques related to it. We also discovered that the lack of participant engagement, the lack of participant's understanding of how the technique works, or the professionals might change the selected techniques during a DT session because of low meaningful results.

Next, in order to reinforce the contributions of our proposal when compared to other tools to recommend DT techniques in software development, we conducted a comparison between Helius' features and the similar tools that we found in our SLM.

6.2.2 Feature Analysis using DESMET method

Before starting to implement Helius, we conducted a validation of the tool scope. Our first step towards this validation–the focus of this section, aims to demonstrate that the tool feature set indeed brings new contributions with regards to those tools already available, i.e., it innovates[3].

The related tools which we compared to our project were identified through a Systematic Literature Mapping (Section 4.1) and through a Survey performed with IT professionals (Section 4.2). Thus, aiming to compare Helius with other tools, we performed feature analysis using DESMET [131]. In this study, we compared the other tools with a prototyped version of Helius, previously validated by industry professionals (see Section 5). The Helius' features were intentionally defined as a baseline for the comparison, following the procedures proposed in Kitchenham *et al.* (1996) [131].

DESMET is a method that provides an algorithm to compare tools when we intend to identify the best alternative for a specific domain [131]. For instance, Marshall [165] performed a study comparing tools for Systematic Literature Reviews (SLRs). Starting with the definition of the features, sub-features, and their respective degrees of importance, the authors conducted comparisons feature by feature, obtaining the most suitable tool for conducting an SLR.

Using the qualitative analysis provided by DESMET, we compare Helius features (Table 6.4) with two related tools: DTA4RE [245] and IDEO DT [115]. Next, we start presenting an overview of similar tools, and we show the DESMET evaluation, extracting the percentage of agreement for the defined features expected in the recommendation system of DT Techniques.

DTA4RE is a tool that provides recommendations for DT techniques for software development. Through a form-based algorithm, the tool presents the users with a set of questions and, once they have answered, the tool exposes the applicable DT techniques. DTA4RE focuses on DT techniques for requirements engineering, recommending techniques according to three working spaces of DT: inspiration, ideation, and prototyping.

Although providing information about the techniques, such as how to use a technique, and when to use it, DTA4RE does not offer features related to DT techniques evaluations and also does not recommends techniques based on community feedback.

IDEO DT is a web-based tool that provides DT techniques suggestions according to filters that the user has applied. Once in the IDEO DT system, it is possible to select some needed characteristics, such as what working space the user is in, and the scenario of use, among others. For each technique filtered, IDEO DT gives an overview, the steps to be done, and the difficulty level for using it.

[3]This study was published as a paper in the Workshop on Requirements Engineering (WER 2021) [193]

Table 6.4: Helius Features used in the DESMET evaluation

ID	Feature set	Sub-feature ID	Sub-feature description	Subfeature: level of importance	Feature set Importance Weighting
F1	DT Techniques Recommendations	F1-SF01	To consider the combined use of techniques in projects to recommend DT techniques	Mandatory	0.3
		F1-SF02	To consider contextual information from the user's project to recommend DT techniques	Mandatory	
		F1-SF03	To consider feedback from other users to recommend DT techniques	Mandatory	
F2	DT Techniques Filtering	F2-SF01	To consider techniques characteristics to filtering similar DT techniques	Highly Desirable	0.2
		F2-SF02	To consider project context information to filtering similar DT techniques	Highly Desirable	
F3	DT Techniques Evaluation	F3-SF01	To allow evaluation of a single technique by the user	Highly Desirable	0.2
		F3-SF02	To allow evaluation of the techniques used in conjunction in a DT project by the user	Highly Desirable	
F4	DT Techniques Community Feedback	F4-SF01	To allow filtering feedback by technique	Desirable	0.1
		F4-SF02	To show related feedback for projects and techniques used in conjunction	Desirable	
F5	DT Techniques Information	F5-SF01	To show detailed information about each DT technique	Nice to have	0.1
		F5-SF02	To show the related uses of a technique with other techniques	Desirable	
F6	DT Techniques and Project Management	F6-SF01	To manage a project which uses DT techniques	Nice to have	0.1
		F6-SF02	To share project data with other team members	Nice to have	

Similarly to DTA4RE, IDEO DT does not provide runtime evaluations of the recommended techniques nor consider the users' feedback to calculate the recommendations. Helius is a system that proposes to innovate using the users' collaborations to recommend suitable DT Techniques, including the previous experiences from the users (community feedback-based) to support DT in software development projects.

DESMET Evaluation process

DESMET is an evaluation method that requires the designation of scores for the features and sub-features, such as sub-features importance levels, features weights, and judgment scales. Thus, following Marshall [165], each score was initially designated by the first author, and after they were analyzed by the fourth and fifth authors.

Using the DESMET method initially is needed to define the importance level for each sub-feature (Table 6.5) (SFIL). It designates a multiplier associated with each sub-feature representing its importance in the solution.

Table 6.5: Sub-features Importance levels (SFIL)

Importance level	Multiplier
Mandatory (M)	*4
Highly Desirable (HD)	*3
Desirable (D)	*2
Nice to have (N)	*1

Next, we evaluated the set of sub-features in our proposal and attributed the corresponding importance level (Table 6.4 – column Sub-feature level of importance). For instance, we consider sub-feature F1-SF01 *“To consider the combined use of techniques in projects to recommend DT techniques”* as one of the most important sub-feature of Helius. Therefore we defined it as a Mandatory (M) sub-feature. We marked it as M.

In the sequence, in DESMET are defined importance weights for each feature set (FIW). It means the importance of each set of features for a tool. For our study, we defined the importance weight, as shown in Table 6.6. For example, we defined F1 as the most important feature set, with an importance weight of 0.3, which represents 30% of importance in this context. All importance weights can be seen in Table 6.4 – column Feature set importance weighting).

Next, we have to calculate the Max Feature Set Score (MFSS) for each macro-feature. Equation 6.1 shows that MFSS is calculated by adding up the Sub-feature Importance Levels (SFIL) that belong to a macro-feature. Thus, $SFIL_i$ represents the importance level of the subfeature i (see Table 6.5), and n represents the number of sub-features that belong to a macro-feature. For instance, in Table 6.7 - column D), 12 is the MFSS of the macro-feature F1, 6 is the MFSS of the macro-feature F2, and so on.

Table 6.6: Feature Set Importance Weight (FIW)

ID	Feature Set	FIW
F1	DT Techniques Recommendations	0.3
F2	DT Techniques Filtering	0.2
F3	DT Techniques Evaluation	0.2
F4	DT Techniques Community Feedback	0.1
F5	DT Techniques Information	0.1
F6	DT Techniques and Project Management	0.1

Table 6.7: Feature analysis results for three tools (Helius, DTA4RE, and IDEO DT)

				Helius				DTA4RE				IDEO DT			
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**
Feature Set	**Subfeatur**	**Subfeatur importance level (M,HD,D,N**	**Max Feature set Score**	**JI score Helius (JIS)**	**Weigh Score**	**Featur set Score obtained**	**% Feature Set Score**	**JI score DTA4RI (JIs)**	**Weigh Score**	**Featur set Score obtained**	**% Feature Set Score**	**JI score IDEO (JIS)**	**Weigh Score**	**Featur set Score obtained**	**% Feature Set Score**
F1	F1 - SF01 F1 - SF02 F1 - SF03	4 4 4	12	1 1 1	4 4 4	12	100%	1 1 0	4 4 0	8	66.67%	0 0.5 0	0 2 0	2	16.67%
F2	F2 - SF01 F2 - SF02	3 3	6	1 1	3 3	6	100%	1 1	3 3	6	100%	1 0	3 0	3	50.0%
F3	F3 - SF01 F3 - SF02	3 3	6	1 1	3 3	6	100%	0 0	0 0	0	0.0%	0 0	0 0	0	0.0%
F4	F4 - SF01 F4 - SF02	2 2	4	1 1	2 2	4	100%	0 0	0 0	0	0.0%	0 0	0 0	0	0.0%
F5	F5 - SF01 F5 - SF02	1 2	3	1 1	1 2	3	100%	1 0	1 0	1	33.3%	1 0	1 0	1	33.3%
F6	F6 - SF01 F6 - SF02	1 1	2	1 1	1 1	2	100%	0 0	0 0	0	0.0%	0 0	0 0	0	0.0%
				Overall score (OS)			100%	Overall score (OS)			43.33%	Overall score (OS)			20.83%

$$MFSS = \sum_{i=1}^{n} SFIL_i \tag{6.1}$$

Another definition that needs to be done using DESMET is the Judgement scale and its Interpretation (JI). JI means to consider if each sub-feature is in the tool evaluated. We defined JI following Marshall [165]: *Yes* – score 1, if a sub-feature was fully identified in the tool; *Partly* – score 0.5, if a sub-feature is not fully present or was implemented differently, and; *No* – score 0, when the sub-feature can not be found in a tool.

Table 6.7 – columns E, I, and M show the JIs we attributed for each sub-feature of Helius, DTA4RE, and IDEO DT tools, respectively. For example, we fully identify the sub-feature F1-SF01 in Helius (score 1 in column E); F1-SF01 was also fully identified in DTA4RE (score 1 in column I); but F1-SF01 was not identified in IDEO DT (score 0 on column M). It is important to mention 2 points: (i) as described by Kitchenham [131] this JIs evaluation is according to the evaluator's perception, and; (ii) all sub-features have JI with a score 1 for Helius because we used the prototype-based version of the tool, considering all features we intend to implement on it.

Consequently, based on the previous definitions, we performed the DESMET evaluation. Table 6.7 shows the complete data of this evaluation. Table 6.7-(D) shows the maximum possible feature score for a feature set. It is the sum of the importance levels of the sub-features from each feature set. For example, 12 represents the max feature set score reachable in the feature set F1, 6 for F2, and so on.

Next, based on the Judgment scores and their Interpretations (JIs), shown in Table 6.7 – columns E (Helius), I (DTA4RE), and M (IDEO DT), a weighted score for each feature of each tool is calculated (Table 6.7 – F (Helius), J (DTA4RE), N (IDEO DT)). The weight score is the multiplication of the JI with the respective sub-feature Importance level (Table 6.7 – (C)) (Equation 6.2). For example, for F1-SF01, the weight score for Helius is 4 (Table 6.7 – (F)), obtained with Table 6.7 – (E)(JIS) * (C)(sub-feature importance level).

$$WS_i = JIS_i * SFIL_i \tag{6.2}$$

Next, is calculated a Feature Set Score Obtained (FSSO) (see Table 6.7 – (G (Helius), K (DTA4RE), O (IDEO DT))). This feature set score obtained is the sum of the weight scores for a feature set, as shown in Equation 6.3. For example, in Table 6.7 – (K) we show the feature score (FSSO) obtained for the F1 by DTA4RE, resulting in 8 (4+4+0).

$$FSSO = \sum_{i=1}^{n} WS_i \tag{6.3}$$

In the sequence, it is calculated the percentage of the feature set score reached (FSS) (Table 6.7 – (H (Helius), L (DTA4RE), P (IDEO DT))). The values are calculated by dividing the feature score obtained (FSSO) (Table 6.7 – (G (Helius), K (DTA4RE), O (IDEO DT))) by the max possible feature score for each Feature set (MFSS) (Table 6.7 – (D)). For instance, for F1 DTA4RE reached to 66.67% of the max possible feature set score (Table 6.7 – (L)); DTA4RE reached, and IDEO DT reached to 16.67% (Table 6.7 – (P)).

Equation 6.4 shows how FSS is calculated.

$$FSS = FSSO/MFSS * 100 \tag{6.4}$$

Finally, the overall score (OS) for each tool is calculated (Equation 6.5). OS is obtained with the sum of each % feature set score, considering the feature set importance weight (Table 6.6). For example, Equation 6.6 shows how the overall score for DTA4RE was obtained. Considering the importance weight of F1 as 0.3 (Table 6.7), the % feature set score for F1 for DTA4RE (Table 6.7 – (L)) is 66,67% * 0,3 = 20%. For F2, the importance weight is 0,2, then it is calculated using 100% * 0,2 = 20%, and so on. In the end, all these values are summed and the overall score is obtained.

$$overall\ score\ (OS) = \sum_{i=1}^{6}(FSS_i * FIW_i) \tag{6.5}$$

$$DTA4RE\ OS = (66.67*0.3) + (100*0.2) + (0*0.2) + (0*0.2) + (33.3*0.1)+(0*0.1) \tag{6.6}$$

The overall score for DTA4RE was 43.33% and for IDEO DT was 20.83%. These results show Helius as an appropriate recommender system for the features we defined (Table 6.1), aiming to support the use of DT techniques in software development.

Therefore, motivated by the results of our initial tool validation and of the feature analysis activity using DESMET, we were able to complete the novelty element of the DSR. Our research contributes to the knowledge base brought by this study are:

> The recommendation tool itself, the support to IT professionals on the decision-making for selecting DT techniques, and an improvement on the use of DT for RE activities.

6.3 Helius' Initial Features Implementation

Since the results showed that Helius represent an innovative solution, we started to implement it. We used Flutter[4] as the programming language, which provides support to develop multi-platform solutions, including mobile applications for IOS and Android devices, as well as for use directly on the web browser. Figure 6.9 shows Helius running on 3

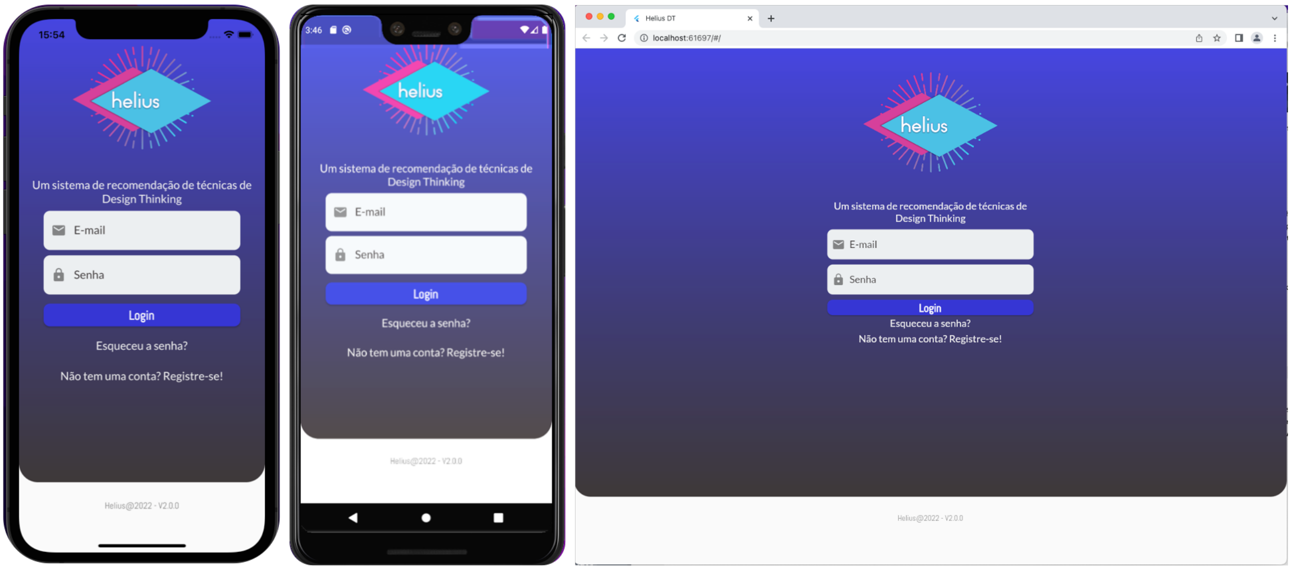
[4] Available in: http://flutter.dev

different platforms: (a) iPhone simulator – IOS[5]; (b) Android Simulator – Google Android[6]; (c) Chrome Web Browser – Internet Browser[7]. These platforms represent the front-end layer in Helius' architecture (see Figure 6.10).

We developed Helius following the mobile-first approach [177] since we considered that DT practitioners could easily access DT techniques recommendations and DT techniques information, as well as other professionals' experience in using DT techniques in software development. Therefore, we also developed a back-end layer composed of an Application Programming Interface (API) in order to connect the front-end layer to the persistence layer. We used NestJS[8], a NodeJs-based API, aided for the Firebase platform[9] to support features like authentication, portfolio management, and so on. We used MongoDB as the database in Helius for storing the data. Figure 6.10 illustrates Helius' architecture.

To recommend DT techniques, Helius provides a set of features, as we showed in Table 6.1. We implemented the DT techniques recommendation module using Python language. We used FastApi framework[10] to support the development of the recommendation module. However, before implementing the recommendation mechanisms in Helius, we started by implementing the features F2 to F6[11], as shown in Table 6.8.

Figure 6.9 shows the opening screen in Helius. The user might log in or create a new account. Once the user is logged in, Helius shows the main screen (Figure 6.11–(a)).



(a) (b) (c)

Figure 6.9: Helius' multi-platform version

[5]https://apps.apple.com/br/app/helius-design-thinking/id1596960889
[6]https://play.google.com/store/apps/details?id=com.helius.helius_v21
[7]https://heliustool.github.io/helius-web/
[8]Available in: http://nestjs.com
[9]Available in: http://firebase.google.com
[10]Available in: https://fastapi.tiangolo.com/
[11]*Sub-feature F6-SF02 was not implemented in this version of Helius. We decide to include it as future work in the Helius' implementation.

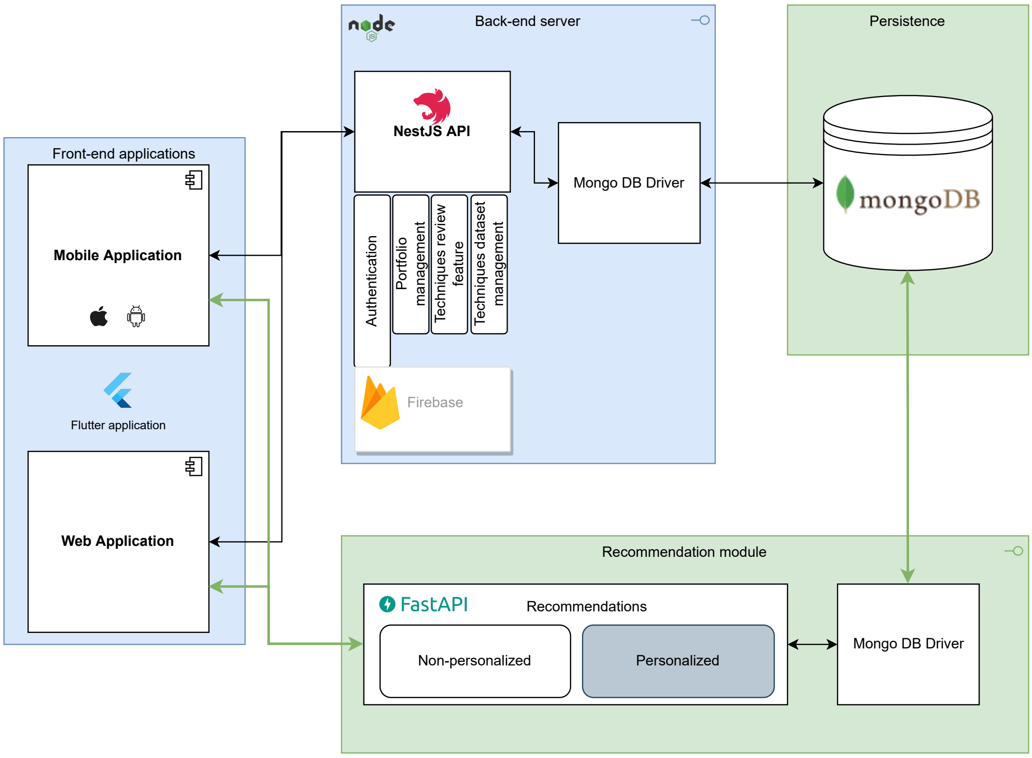


Figure 6.10: Helius' architecture

Table 6.8: Helius' initial features implemented in Iteration 2

ID	Feature set	Sub-feature ID	Sub-feature description
F2	DT Techniques Filtering	F2-SF01	To consider techniques characteristics to filtering similar DT techniques
		F2-SF02	To consider project context information to filtering similar DT techniques
F3	DT Techniques Evaluation	F3-SF01	To allow evaluation of a single technique by the user
		F3-SF02	To allow evaluation of the techniques used in conjunction in a DT project by the user
F4	DT Techniques Community Feedback	F4-SF01	To allow filtering feedback by technique
		F4-SF02	To show related feedback for projects and techniques used in conjunction
F5	DT Techniques Information	F5-SF01	To show detailed information about each DT technique
		F5-SF02	To show the related uses of a technique with other techniques
F6	DT Techniques and Project Management	F6-SF01	To manage a project which uses DT techniques
		F6-SF02	To share project data with other team members*

The main screen in Helius shows 3 categories of features: 1) Start features, 2) Recommendations, and 3) Favorite Users.

Category 1) "Start Features" contains the following features:

(i) *NEW DT techniques project (ask for recommendations) (feature F6):* to create projects (portfolios) of DT techniques to conduct DT workshops/projects. It allows one to ask for DT techniques recommendations and add the techniques recommended to a project;

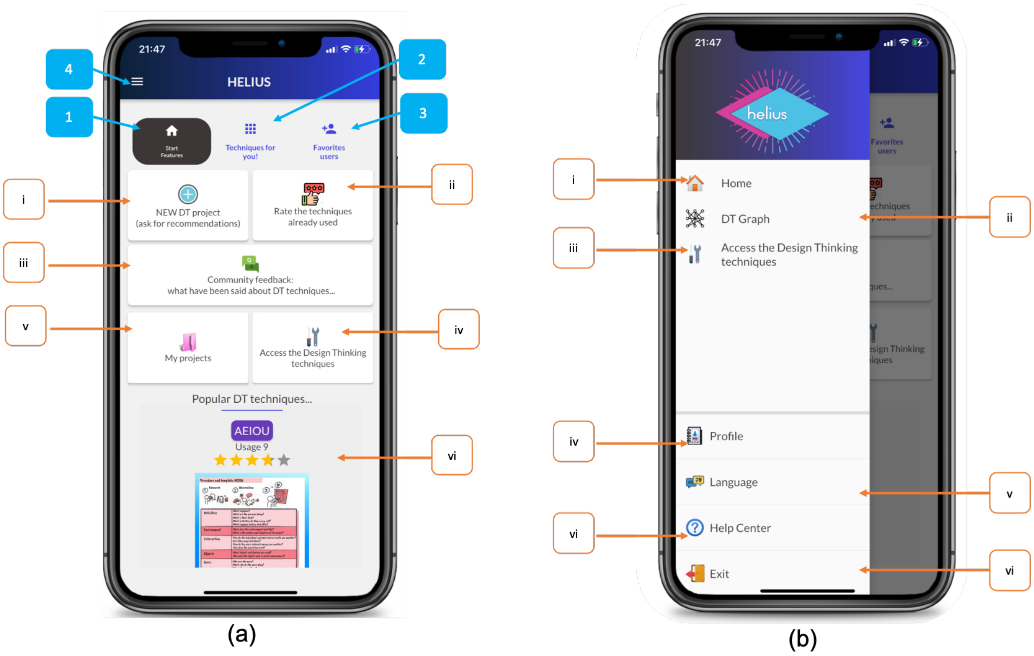


Figure 6.11: Helius' home screen (a); Drawer menu (b)

(ii) *Rate the techniques used (feature F3):* to collaborate and retro-feed the recommendation mechanisms. Helius is a collaborative recommendation system that uses DT techniques reviews to calculate recommendations to the users;

(iii) *Community feedback: (feature F4):* to access experiences of other DT practitioners on the use of DT techniques in software development;

(iv) *Access the DT Techniques: (features F2 and F5):* to gather information about DT techniques such as name, when to use and how to use;

(v) *My Projects: (feature F6):* to quickly access the repository of DT techniques portfolios (projects) the user has created in Helius;

(vi) *Popular DT techniques area (feature F5):* to quickly know what are the most used DT techniques in Helius.

Category 2) "Recommendations", includes a summary of the recommendations to be provided in Helius and category 3) "Favorite Users' includes a feature that allows to access information about favorite users in Helius. These categories are further explored in Section 7.2.2 and Section 7.1, respectively.

On the Main screen, the user might access a drawer menu (item 4 in Figure 6.11–(a)). The drawer menu provides access to the following features (Figure 6.11–(b)):

i. *Home:* To access the Main screen;

ii. *DT graph:* To access a DT techniques' graph. The graph represents the use of techniques in combination (techniques used together in a project). It allows the user to identify what techniques are being used in software development;

iii. *Access to DT techniques:* To access the DT techniques information;

iv. *Profile:* To access the user's profile information (name, e-mail, etc.);

v. *Language:* To set up the Helius' language. Helius provides support to English and Portuguese (default language);

vi. *Help Center:* To access a set of screens that provide support on the use of Helius.

vii. *Exit:* To logging out of Helius.

(i) NEW DT project (ask for recommendations)

In Helius, a DT practitioner can create a Portfolio of DT techniques (project for simplification). This feature represents feature F6 (DT Techniques and Project Management). Figure 6.12 shows the user journey in Helius for creating a DT project. It can be started through the Main Screen (i), by clicking on the "New DT project" button.

By creating a project, the DT practitioner might keep the historical data about her projects using Helius. A DT project in Helius includes a name, a description, the project domain (e.g., Embedded Systems, E-healthy systems, Software for startups, etc), the project context (e.g., contact with the customer/user: [No, Yes], the place to perform the technique: [organization, client], number of participants: [individual (1), small group (2 to

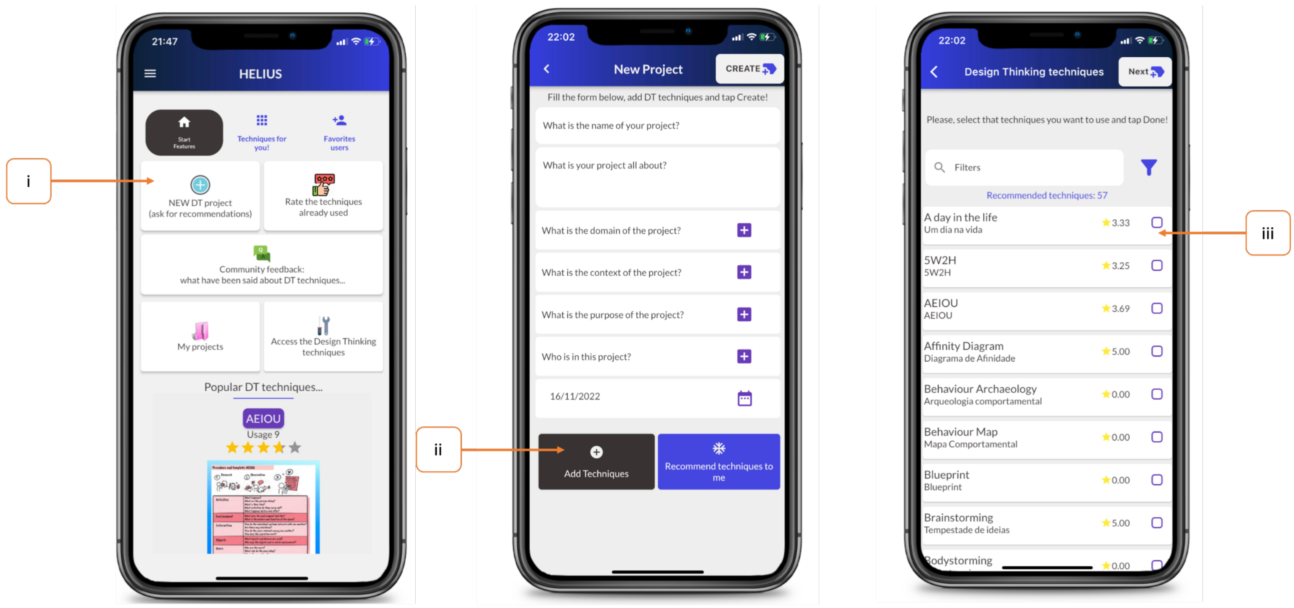


Figure 6.12: User journey for DT techniques portfolio creation in Helius

5), large group (+ 5)], among others), the project purpose (e.g., discussing and simulating various ideas based on the information collected, making associations between the ideas generated, experimenting other solutions to gather new ideas, among others), the participants and the creation date.

The DT techniques portfolio feature allows the Helius user to ask for DT techniques recommendations or add the DT techniques on her own (Figure 6.12-(ii)). Once the user has decided to add techniques that she already knows or has already used before by clicking on the “Add Techniques” button, Helius shows a list of DT techniques, including the technique’s name (in English and in Portuguese) and the average rating attributed by other DT practitioners to the techniques. For instance, Figure 6.12–(iii) shows that the technique “A day in the life” was rated by Helius’ users as 3.33 on average. On the other hand, if the user wants to ask for recommendation (Figure 6.12-(ii)), Helius opens a screen containing different mechanisms of DT techniques recommendations (see Section 7.3).

(ii) Review of DT techniques

Helius uses the reviews for techniques made by its users as a feed to recommend DT techniques to DT practitioners. The reviews that DT practitioners provide to techniques are also used as experiences of using DT explored in the Community feature (see Section 7.1). Therefore, once the user has created a DT techniques portfolio (project) containing DT techniques (added on her own or by asking for recommendations), she is able to evaluate the techniques that she has used, retro-feeding the recommendation mechanisms. This feature represents feature F3 (DT techniques evaluation).

Figure 6.13 shows the user journey to review DT techniques in Helius. To do a review for a DT technique, the user needs to click on the “Rate the techniques used” button on the Main Screen (Figure 6.13–(i)). Then, Helius shows the user’s projects containing pending reviews (e.g., projects containing techniques that she has not evaluated yet). For instance, Figure 6.13–(ii) shows the project “Projeto Web” containing 4 techniques to be reviewed by the user. Figure 6.13–(iii) presents the Helius’ screen where the user can see the techniques to be reviewed. The screen also shows that the technique “Try it yourself” has already been evaluated.

If the user decides to review a technique that she has used by clicking on the “Review” button, Helius presents the technique Review page (Figure 6.13–(iv)). Once on the Review page, the user might rate the technique (a quantitative evaluation from 0 to 5 in a star-like range), inserting a description about the experience she has using such technique, the participants present when the technique was conducted, if other and which techniques were used in combination and an evaluation in terms of costs (also presented as a star-like range). In terms of costs, Helius presents to the user a list of suggestions about how to evaluate the cost of a technique, including workshop length (hours), the number of employee participants per workshop, hours of scoping, scheduling, and logistics to plan workshops, internal costs for design thinking workshops [116].

(iii) Community Feedback

Community feedback represents the feature F4 (DT Techniques Community Feedback) in Helius. It aims to provide access to the professionals' experiences in using DT techniques in software development. The Community feedback feature uses the reviews for techniques to support the decision-making of DT techniques. The feature is explored in detail in Section 7.1.

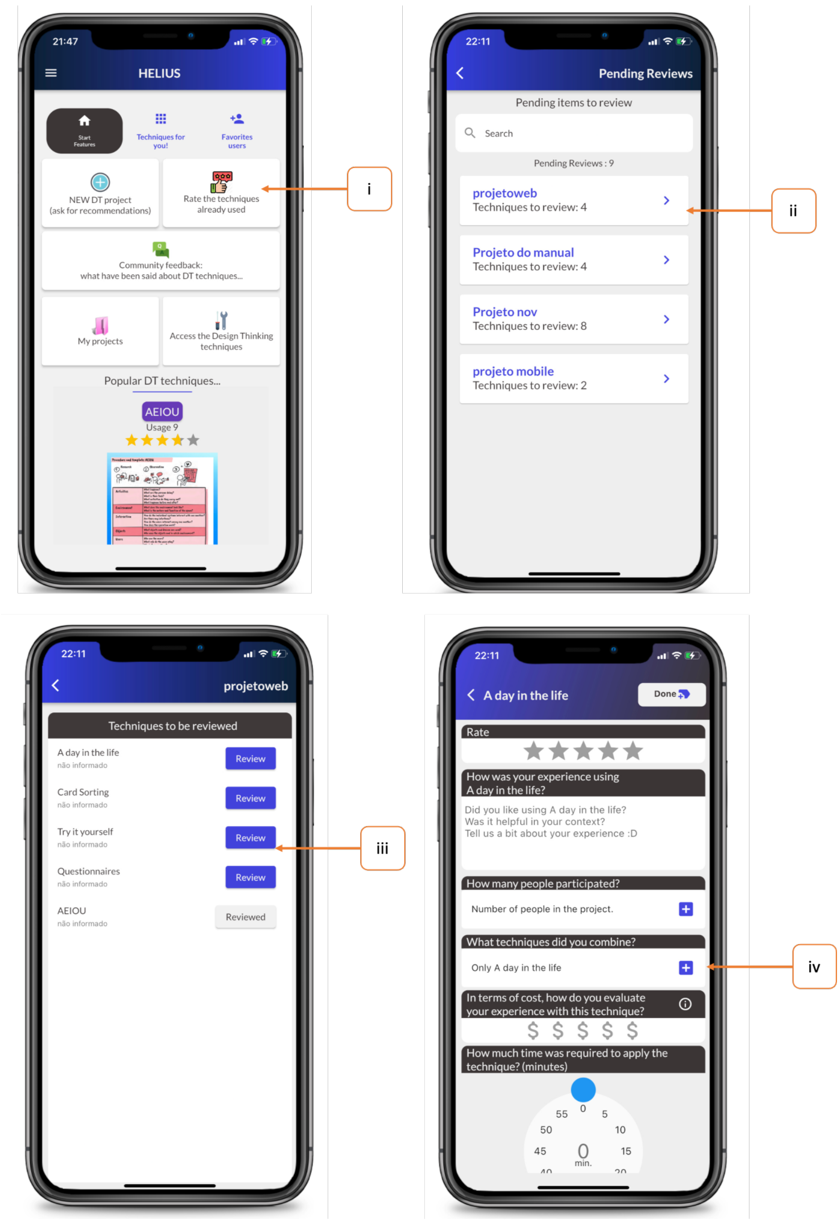


Figure 6.13: User journey for DT techniques review in Helius

(iv) Techniques information

Helius also provides a feature that presents DT techniques information (feature F5 in Table 6.8). Helius provides information about 57 DT techniques that we identified through our literature mapping study presented in Section 4.1[12].

The user can access the DT techniques information screen by clicking on the "Access the Design Thinking Techniques" button on the Main screen (Figure 6.14–(i)). Helius shows the list of DT techniques (Figure 6.14–(ii)), allowing the user to filter the DT techniques by clicking on the filter button (Figure 6.14–(iii)). The filters of DT techniques in Helius include "General" working spaces (problem space and solution space)[13], "Brown"

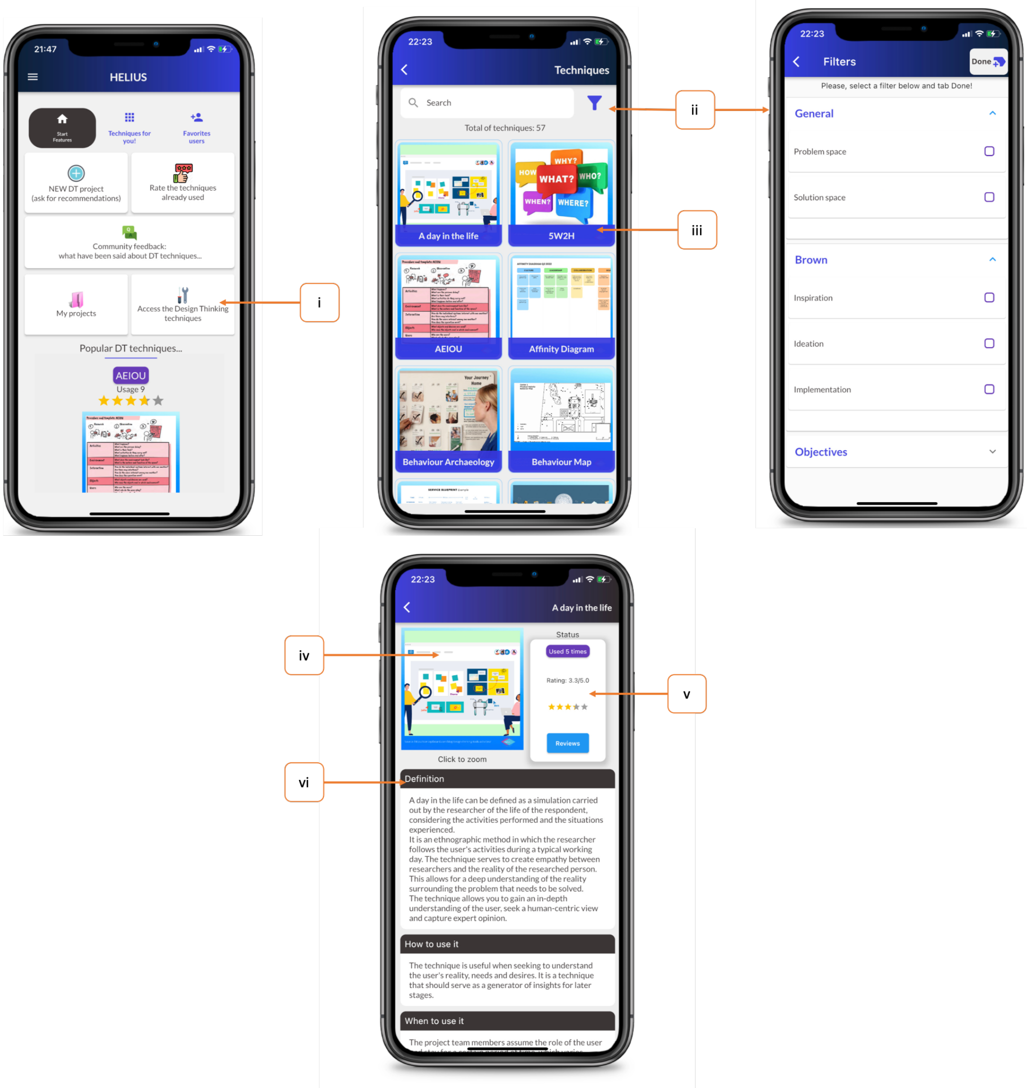


Figure 6.14: User journey for DT techniques info visualization in Helius

[12]We randomly selected 57 techniques from the SLM to start the implementation of Helius.
[13]We have classified all techniques we found in literature through the literature mapping study in solution space or problem space. To see details about this classification, see Section 4.1 – Table 4.10.

DT model, which includes 3 working spaces (Inspiration, Ideation and Implementation)[14], and Objectives, which includes a set of techniques' goals (e.g., knowing more about users (with interaction, user's workplace), knowing more about users (with interaction, everywhere), discovering the users, discovering problem details or information, understating the users-organization relationship, among others)[15].

(v) My projects

In Helius, the user can also view information about her projects. This feature represents feature F6 (DT Techniques and Project Management). Figure 6.15 shows the user journey to see the projects and their details.

Once the user clicks on the "My projects" button in the Main screen (Figure 6.15–(i)), Helius opens a screen showing the user's project list (Figure 6.15–(ii)). By clicking on a project, the user can see the project's details, including the DT techniques used and its reviews (reviews made by the own user), the description, domain, purpose, context, participants and creation date. If the user clicks on the "See Review" button, Helius shows the review the user made to the technique on that project (Figure 6.15–(iii)).

(vi) DT graph

Helius also provides a graph of DT techniques. This feature refers to Feature F5 – DT techniques information (see Table 6.1). We designed and implemented a Graph of DT techniques that represents how the DT techniques have been used in combination. Vertices represent the techniques and the edges represent the techniques' combined usage. The larger the size of a technique, the more it has been used.

Figure 6.16 shows the user journey for accessing the DT techniques graph in Helius. The user might access the graph through the Drawer menu on the Home screen (Figure 6.16-(a)). Once she has clicked on the Graph Button, Helius opens the screen given by Figure 6.16-(b), building a graph that contains the five most used techniques and the combinations between them. The combination indicates the projects that the techniques were used together. To see those projects, the user can click on item (ii)–(Figure 6.16-(b)). If the user clicks on the relationship between two techniques (over the edge), Helius shows the projects that use such techniques, as shown in screen given by Figure 6.16-

[14] We classified the DT techniques into Inspiration, Ideation or Implementation following the work of Souza *et al.* (2020) [245] and considering the Brown's model was the most cited DT model in literature.

[15] We extracted the DT techniques goals by consulting the work of Souza *et al.* (2020) [245], Ignácio and Benitti (2020)[116], and Lewrick, Link and Leifer (2020) [149].

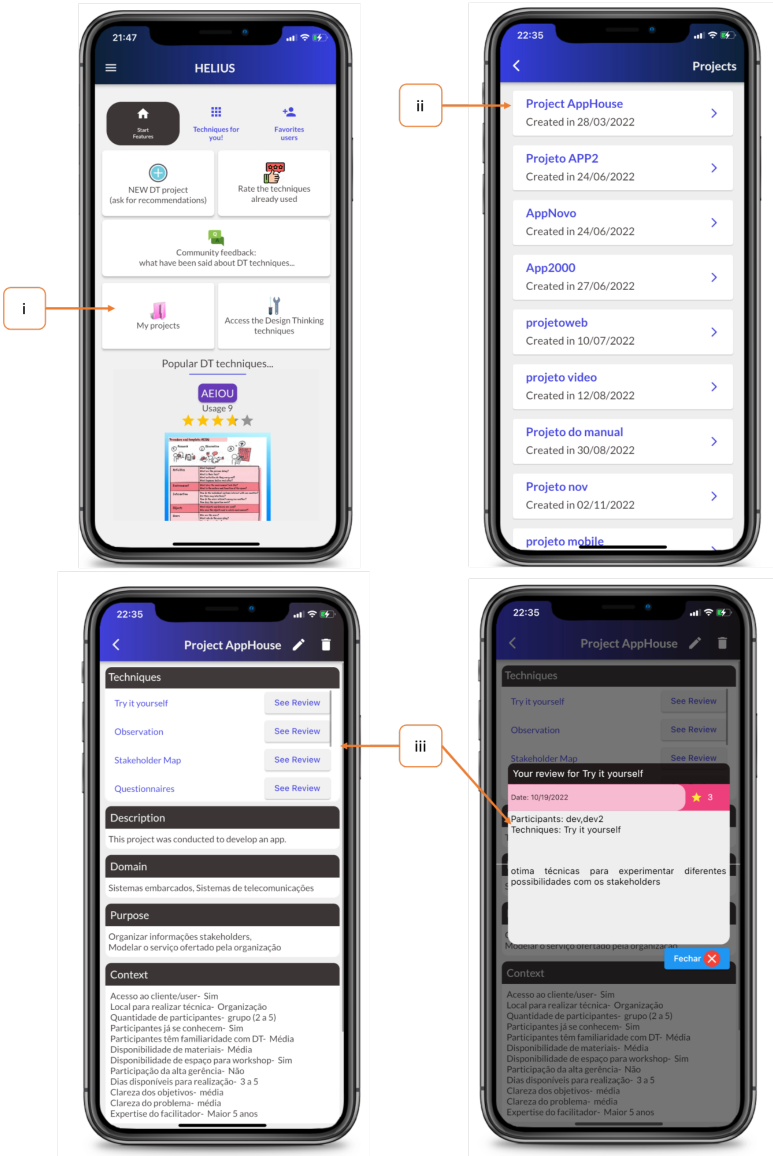


Figure 6.15: User journey for DT techniques portfolio visualization in Helius

(c)). For instance, screen (c) shows the projects using the techniques “Feedback grid” and “Brainstorming”.

Back in Figure 6.16-(b), once the user clicks on a technique name (item (iv)), Helius shows the technique’s details. The user also might see the list of DT techniques in the graph by clicking on item (iii)–screen (b). Then, Helius shows screen given by Figure 6.16-(d)), listing the DT techniques in the graph. The user might see the technique’s details by clicking over its name (item vi).

However, since our long-term research is to support the decision-making of the Design Thinking techniques selection through the development and evaluation of a DT techniques recommendation system for software development, Helius needs to include features capable of supporting such decision-making in the selection of DT techniques. Therefore, the next section shows Iteration 3 that we performed on the DSR-based methodology. Our goal is to explore in depth the features of Community feedback (F4) and

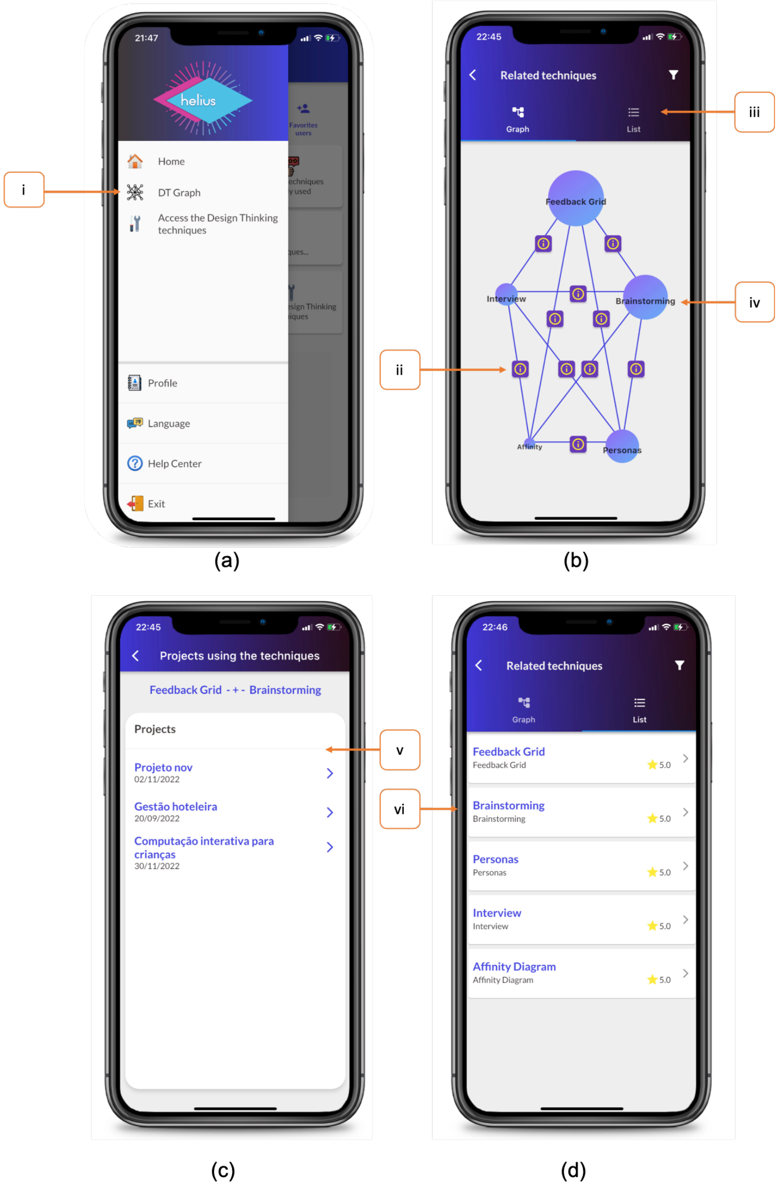


Figure 6.16: User journey visualizing the Graph of DT techniques

DT techniques recommendations (F1), which turn Helius into a recommendation system (Table 6.1). In addition, we conducted an empirical study in the validation approach activity on the DSR-based method to validate Helius with industry professionals.

7. ITERATION 3: SOLUTION EVOLUTION AND VALIDATION

This chapter presents Iteration 3 in our DSR-based research method. Here our goal is to explore in depth how Helius works to support IT professionals in the decision-making of the DT techniques selection in software development. Therefore, we structured this chapter as follows:

- Section 7.1 presents a community of DT techniques practices as a way to implement feature F4 – DT techniques community feedback (See Table 6.1), highlighting the collaborative aspect of Helius;
- Section 7.2 shows a decision-making model that we grounded by interviewing DT facilitators. The goal is to identify and implement the recommendation mechanisms capable of recommending DT techniques (feature F1 in Table 6.1);
- Section 7.3 shows the implementation of the recommendation mechanisms and the empirical study that we conducted to validate Helius with DT practitioners.

7.1 Community of DT techniques experiences

Considering the initial features that we implemented in Helius and the results that we obtained during Iterations 1 and 2 in the DSR method, we realized that Helius lacked a collaborative feature where professionals could share their experiences by using DT techniques, aiding others on the decision of which techniques to use (feature suggested by the participants in the Early Tool Evaluation activity - Iteration 1, Section 5.3).

Therefore, to propose a new feature in Helius, we posed the following research question: *"How to promote collaboration among IT professionals who use DT techniques in software development, to foster the creation of a community of practice and assist in deciding which DT techniques to use?"*. Here we refine the collaboration features proposed for Helius by creating a community of practices of DT techniques[1].

Communities represent groups of individuals who share similar interests in real or virtual environments [175]. Cho and Wash (2021) [49] define virtual communities as "groups of people who interact with each other, mediated by technology such as computers and the Internet". The authors point out that a virtual community contains 3 elements: 1) the technology which provides a communication channel between members, 2) a group of individuals that participates in the community (members), and 3) the content shared in the community and is available to members.

[1] This study was published in the Brazilian Symposium on Collaborative Systems (SBSC 2022).

Prilla, Blunk and Chounta (2020) [212] argue that community-based systems provide mechanisms for professionals to share experiences, learn from them and make decisions. Fuks *et al.* (2008) [86] introduced the 3C model for instantiating collaboration in the software development context. The model abstracts links between 3 elements: Communication, Cooperation and Coordination. The authors assume that "computational support for collaboration may be realized through the interplay between communication, coordination, and cooperation tools".

Kniberg and Ivarsson (2012) [135] proposed the Spotify model for a community of practice (Figure 7.1), aiming to increase productivity and innovation in Agile teams. The model gives attention to the team's autonomy and communication, fostering the development of products with high quality [135]. The Spotify model defines development teams in *Squads*. Squads are cross-functional and self-organized teams consisting of 4 to 12 professionals [241]. *Squads* contain *Tribes*, which can be formed by up to 200 people. Each Tribe has a clear set of goals and the responsibility of developing a specific *feature*. Each Tribe contains *Chapters*, which are groups of experts who focus on the social and professional development of the team's members. Lastly, the model introduces *Guilds*, which represent groups of team members with similar interests and who share their experiences to foster collective knowledge.

Therefore, we inspired the community feature on the concept of *Guilds*. We intended to include in Helius a resource capable of allowing DT practitioners to share their experiences of using DT techniques, fostering the creation of a community of DT practitioners and collaborating with the decision-making of the selection of DT techniques for software development. We argue that experiences of other practitioners' use of DT techniques can aid the decision-making of which DT techniques to use.

In this context, we consider communities as groups formed by individuals who collaborate from similar interests [175], and that they are composed of 3 components [49]: 1) technology, 2) participating members and interactions, and 3) shared content (Figure 7.2). Technology is the recommendation system for DT techniques. Members are the IT professionals that use DT techniques for software development and interactions

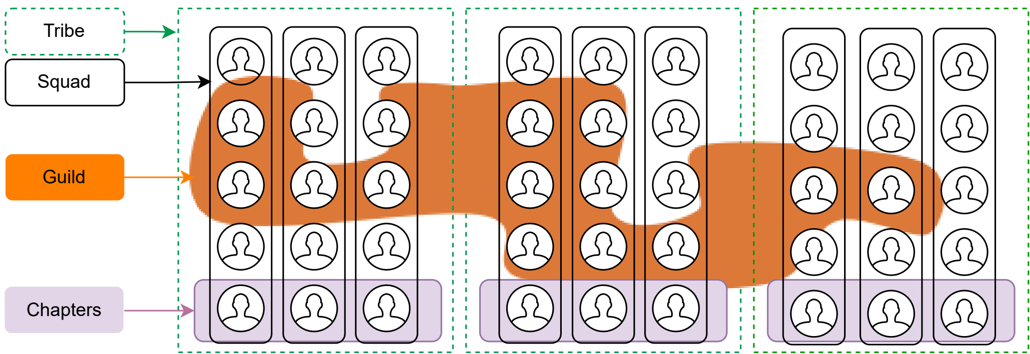


Figure 7.1: The Spotify model for Scaling Agile – Kniberg and Ivarsson (2012) [135]

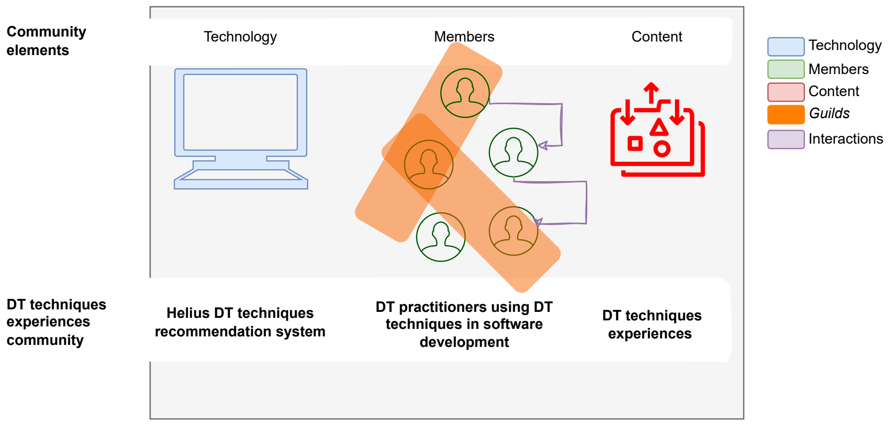


Figure 7.2: Components of a DT techniques community of practice

are *Guilds*, a set of professionals with similar interests, or even that can interact with each other (interaction arrows). Content is given by the experiences of using techniques and by the recommendations provided by the system to the professionals, allowing the collection of information for decision-making about which techniques to use. Thus, we aim to support professionals from different organizations, even if not structured by the Spotify model, to access other professionals' experiences of using DT techniques.

Figure 7.3 details the DT techniques community of practice. In Helius, after using DT techniques that have been recommended to her (by the other *features* of the recommendation system), the user will be able to evaluate her experience of using each technique. By rating a technique (1 to 5 stars), including a description of experience, the purpose of use and the context of the application of the technique, the user collaborates with the community of practice. In addition to the evaluation data, Helius collects information about the user's profile. The evaluation of a technique plus the user's profile information composes a deposit of experience in the recommendation system's database.

Once the DT practitioners have deposited their experiences using DT techniques, Helius stores a database of DT techniques experiences. Thus, other Helius users might collect the DT practitioners' experiences. By collecting experiences, the user completes a cycle in the Helius community of practice. Therefore, she can request other experiences, make decisions based on such experiences, and then use the techniques and evaluate them to retro-feed the experiences' database in Helius.

In addition, the community of practice in Helius allows the users to follow other users. Thus, a user can create a subset of professionals that she wants to follow and see the experiences of using DT techniques, composing an environment similar to *Guilds*, used as a reference in this proposal. Thus, when following other professionals, the user might collect the experiences of using DT techniques from other professionals. Helius also presents the experiences of users considered similar to the active user (the user logged in). Helius uses the KNN algorithm to calculate similar users (see Section 7.3.1).

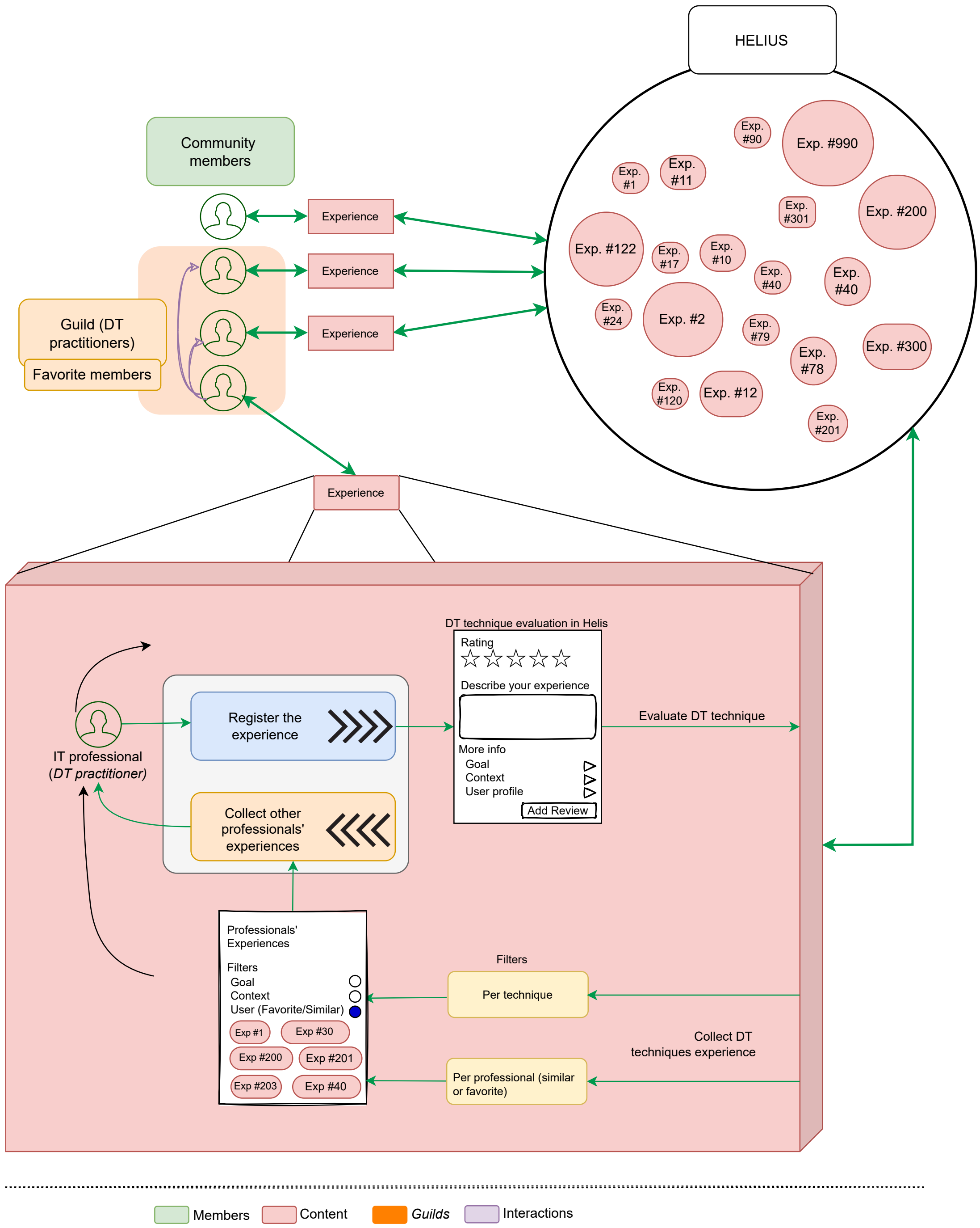


Figure 7.3: DT techniques experience community of practice in Helius

Table 7.1 shows the features of the DT techniques community of practice. These features were implemented into Helius. We classified the features taking into account the community elements: members, interactions and content. The component technology is the recommendation system itself.

Therefore, Helius offers 2 alternatives to the user filtering DT techniques experiences: (1) by a user (similar or favorite) and (2) by technique. Figure 7.4 shows the user journey in Helius to access the community of practices of DT techniques experiences by choosing filter (1). In the Main Screen, once the user has clicked on the Community Feed-

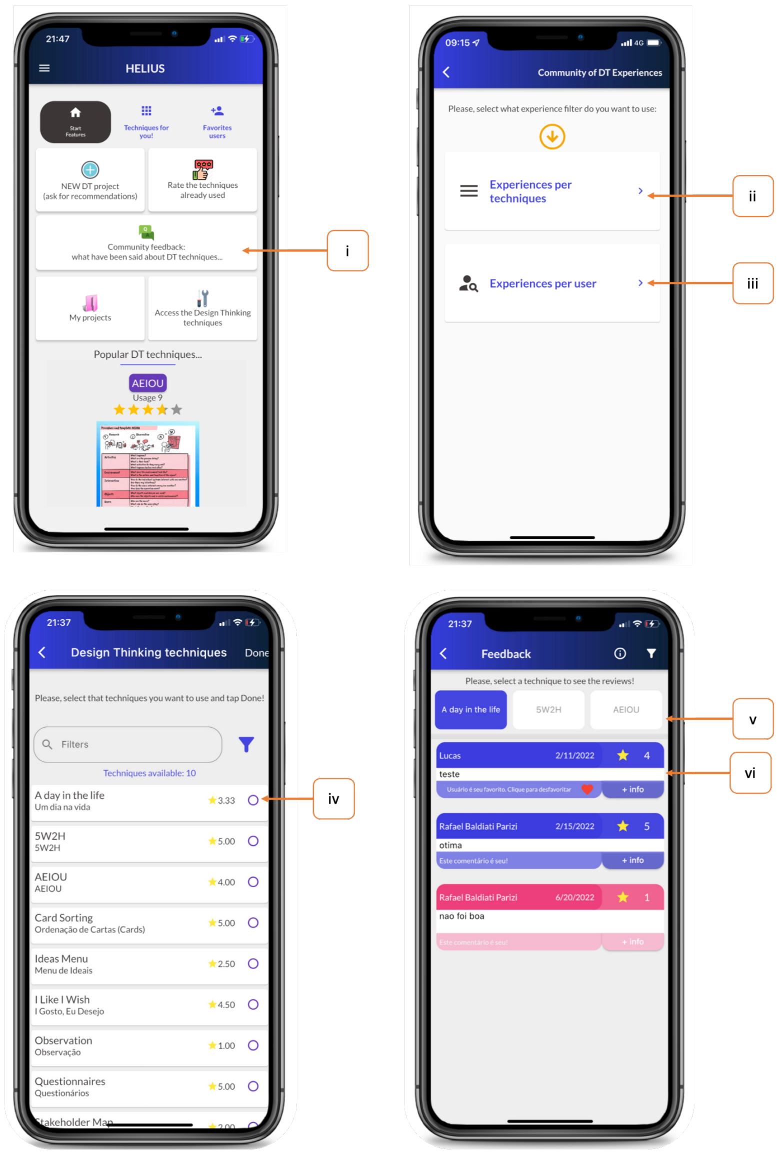


Figure 7.4: User journey for DT techniques experiences collection in Helius

Table 7.1: Community elements in Helius DT techniques recommendation system

Community element	level	Feature	Description
Members and interactions	DT practitioners	To search for similar professionals and their evaluations for DT techniques	To search for professionals who have similar interests in terms of DT techniques (similar contexts, similar evaluations for DT techniques)
		To allow the creation of a connection with other DT practitioners	To allow the user to follow other DT practitioners for knowing their experiences on using DT techniques
		To register DT techniques experiences as evaluation of techniques	To allow the DT practitioner to evaluate DT techniques and depositing her experience, creating a community of DT techniques experiences
Content	DT Techniques	To allow the evaluation of DT techniques	To allow users to evaluate DT techniques, registering the experiences of using DT
		To show DT techniques evaluated by similar professionals	To visualize the experience in using DT techniques by similar professionals
		To show DT techniques evaluated by favorite professionals	To visualize the experience in using DT techniques by favorite professionals

back button (i), Helius opens a screen with the filtering options (ii) and (iii). By clicking on the “Experiences per user” button (ii), Helius shows a new screen presenting 2 lists of users (iv): Favorite users and Similar Users. Favorite users include the DT practitioners that the current user follows. Similar users include the users that have similar evaluations to DT techniques. Helius computes similar users using the KNN algorithm.

For instance, if the user selects the “Favorite Users” button, Helius shows the list of DT practitioners the user logged in follows. Once the user has selected one or more users on the favorite users’ list and clicked on the “Next” button, Helius shows the DT techniques the selected users have used (v) and their experiences using DT techniques (vi). In the experience’s screen, Helius shows the DT practitioner’s name, the evaluation date, the technique’s rating, the experience by itself, the area to follow/unfollow the DT practitioner who has evaluated the technique, as well as a “+info” button, which shows detailed information about the project the technique was part of.

Figure 7.4–(vii) shows the button to filter the experiences deposited for a specific technique. Figure 7.5 presents the filters for the experiences. Helius allows filtering DT practitioners’ experiences of using DT techniques by rating – a star-like bar rating (i), and sorts the experiences by more recent (date when the technique was evaluated) or best rating (how the technique was rated by DT practitioners) (ii).

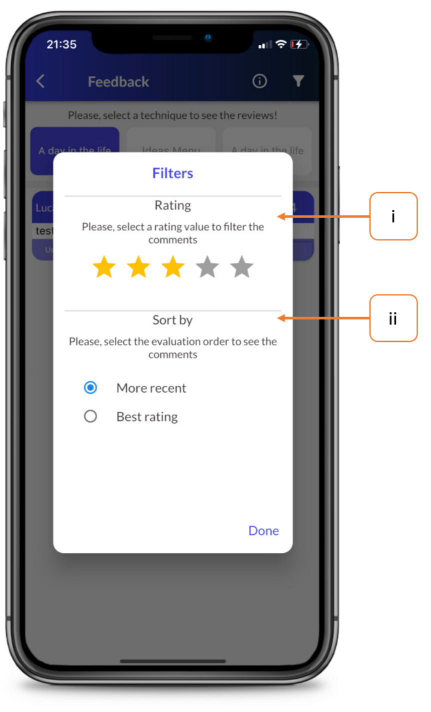


Figure 7.5: Filters for DT techniques experiences collection in Helius

7.2 Decision-making and techniques recommendation mechanisms

In Iterations 1 and 2 of the DSR method, we showed the activities that we performed to propose and start the implementation of Helius. However, since the core feature of Helius is the DT techniques recommendation feature itself, this section details the DT techniques recommendation in Helius.

To implement the recommendation mechanisms in Helius, we first looked for how DT practitioners make decisions for selecting DT techniques in their routine working on software development. However, the literature lacks studies focusing on how DT facilitators make decisions for selecting DT techniques for use in software development. There is no reference to what decision-making elements, strategies and criteria DT facilitators use to select DT techniques [195]. As we mentioned in Chapter 2, decision-making is the process of selecting an option from a set of available options based on alternatives or actions, considering a set of criteria or strategies [267]. Decisions can be based on a simple criterion or on a complex set of strategies and criteria.

Therefore, we conducted an interview-based study with DT facilitators to collect how they select DT techniques. We based our interview study on Ground Theory (GT). As a result, we proposed a descriptive DT techniques decision-making model. The model served as subside to the generation of the recommendation mechanisms implemented in Helius to recommend DT techniques (see Section 7.2).

We posed the following research question: "*How do DT facilitators make decisions for selecting DT techniques in software development?*". We performed an interview-based study, aiming to investigate i. how professionals make decisions for selecting DT techniques and ii. which are the decision strategies and criteria the professionals use for selecting DT techniques.

7.2.1 Interview-based study for data collection

Literature reporting on the use of DT in software development indicates that since DT is a thought-provoking approach, teams need to deal with the lack of problem understanding [251], the preconceiving of problems, and the preconceiving of solutions (outdated ideas) [64]. There are authors who point out that the selection of DT techniques is a key success activity in the use of DT in software development. For instance, Chasanidou *et al.* (2015) [48] argue that the deployment of appropriate methods is a success factor of DT, while Carlgren *et al.* (2016) [43] mention that the use of the right techniques allows collecting the right solution's requirement.

In this context, professionals who facilitate DT workshops (DT facilitators for simplification)[2] – responsible for planning and conducting DT activities– play an important role in achieving the success and to reach higher effectiveness on the use of DT in software development. DT facilitators are responsible for selecting appropriate techniques of DT to be applied in software development. But, since there are more than 80 DT techniques, as identified in [195], this selection might involve a span of contextual elements, turning it into a complex decision-making problem.

Study Design

We used Grounded Theory as a research method to model the DT facilitators' decision-making for selecting DT techniques in software development. GT supports researchers to investigate topics lacking theory [112], what is the scenario of DM involved in selecting DT techniques in software development, which still needs to be explored.

GT is a research method implemented in SE to formulate theories taking into account the human and the social aspects of software development [247, 112]. Glaser and Strauss proposed GT in 1967 [248, 93]. Despite having different GT approaches (Classic, Straussian, Constructivist), GT supports the generation of theory by fostering iterative data collection and constant data comparison in incremental levels of abstraction [47].

In this study, we followed the guidelines for using GT proposed by Stol *et al.* (2016) [247]. The guideline contains 10 key elements to be considered when using GT as a research method: limit exposure to literature, treat everything as data, immediate and continuous data analysis, theoretical sampling, theoretical sensitivity, coding procedures, memoing, memo sorting, cohesive theory and theoretical saturation. We used the Straussian GT version in this study since we set up an upfront research question (RQ): "*How do DT facilitators make decisions for selecting DT techniques in software development?*". We derived the RQ from a literature mapping we conducted [195].

Figure 7.6 illustrates the design of our study. We performed 3 activities: (A) Design and Planning, (B) Interview Conduction and Data Analysis, and (C) Theory Building.

A) Design and Planning

A.1) Interview Design

We conducted semi-structured interviews with DT facilitators. We started by preparing a script with questions that we generated based on the results of previous exploratory studies [195, 211]. The questionnaire contained 14 questions (13 open and

[2]There is no single definition for who organizes and moderates DT sessions. The professionals define themselves as DT coaches, DT moderators, DT facilitators and so on. In [211] the authors show that DT facilitators are the most used term for identifying those professionals who select DT techniques for conducting DT workshops/activities.

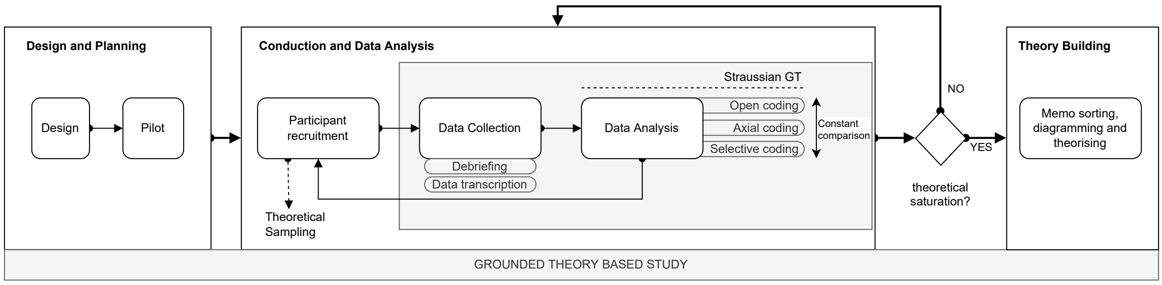


Figure 7.6: Interview study design

1 closed questions), followed by demographic questions[3]. A senior researcher in SE reviewed both the questionnaire and the design of this study.

We planned a slot of 30 minutes for each interview. We prepared a presentation with the research purpose, aiming to perform a warm-up activity and to provide the interviewee with a comfortable space to answer the questions.

A.2) Pilot study

Before kicking off the interviews, we conducted a pilot study. We interviewed a Ph.D. candidate in Computer Science who has more than 7 years of experience in the software development industry. She has been working as a DT facilitator for over 5 years. As a result of the pilot study, we adjusted the questions and increased the timebox which we allocated for each interview to 45 minutes. We also updated our interview script and dropped the pilot study out of the interview set.

B) Interview Conduction and Data Analysis

By using GT, we followed the concept of Theoretical Sampling (TS) [92, 247]. TS is defined as "*process of data collection for generating theory whereby the analyst jointly collects, codes, and analyzes his data and decides what data to collect next and where to find them, in order to develop his theory as it emerges*" [247]. We considered the concept of Theoretical Saturation, which indicates that new data collected will not contribute with new elements to the theory being grounded. To achieve Theoretical Saturation, we collected data iteratively, performing collection and analysis activities in different cycles. After we consider that we had reached out to Theoretical Saturation (Iteration 3), we conducted a new iteration interviewing professionals from other contexts to verify our results.

B.1) Participants recruitment

Before recruiting participants, we submitted our project to the Ethics Committee Board. Once approved, we defined the target audience as being professionals who manage, conduct, moderate, or facilitate DT workshops or projects.

[3]The questionnaire is available in https://encurtador.com.br/cwCKW

Next to the definition of the target audience, we searched for participants in our networking and on LinkedIn[4]. LinkedIn is a social network that brings together professionals from different fields, including software development. We sent an invitation by e-mail to 25 professionals from 20 software companies. Sixteen professionals voluntarily accepted to participate in the interviews (participants for simplification).

Table 7.2 shows the participants' profiles. The profile information includes each Participant's Identification ("ID"), Role, Experience in the IT industry in years ("Exp in IT'), Formation in DT courses ("Formation in DT"), Experience in DT in Years ("Experience in DT"), Years working at the current company ("Years in Company"), Number of employees of the company ("Number of Employees") and Company Site ("Company"). We identify the participants as P, followed by an ordinary number to keep their anonymity. Each participant agreed to the Informed Consent Form, which guarantees not only the possibility to withdraw from the interview at any time, but also the confidentiality and anonymity of data. We used Authentique tool[5] to sign the consent form digitally.

B.2) Data Collection

Figure 7.7 shows how we conducted our interview study and analyzed the data using GT. We performed 4 iterations between data collection and data analysis. In Iteration 1, we conducted 5 interviews, in Iteration 2 we conducted 4 interviews, in Iteration 3 we conducted 3 interviews, and lastly, in Iteration 4 we conducted 4 interviews. We interviewed DT facilitators working in Brazil (Iterations 1 to 3) and from Germany and the

Table 7.2: Participants' Demographic Data

	ID	Role	Exp in IT (yrs)	Background in DT	Exp. in DT	Years in Company	Company Type	Number of Employees
Iter. 1	P1	Facilitator/Moderator	30	School of DT	12	5	National	51-100
	P2	Agile Coach	19	DT open courses	7	7	Multinational	>1000
	P3	Agile Coach	5	In-company Training	2.5	2.5	Multinational	>1000
	P4	Facilitator	5	School of DT	2.3	1	National	51-100
	P5	Facilitator/Design Thinker	23	School of DT	4	4	National	1-10
Iter. 2	P6	Business Designer	3	School of DT	2.5	2.5	National	11-50
	P7	Product Owner	6	School of Design	7	2	Multinational	11-50
	P8	Support Analyst	8	In-company Training	5	8	Multinational	>1000
	P9	Solution Specialist	2	School of DT	4	1	Multinational	51-100
Iter. 3	P10	IT Analyst/DT coach	6	In-company Training	6	6	Multinational	>1000
	P11	Digital Product Manager	5	School of DT	10	3	Multinational	>1000
	P12	DT Analyst	5	School of DT	5	5	National	301-1000
Iter. 4	P13	Director of Customer Experience	25	In-company Training	25	25	Multinational	>1000
	P14	Designer	12	In-company Training	12	11	Multinational	>1000
	P15	User eXperience Designer	8	Design (graduation)	10	5	Multinational	>1000
	P16	DT Coach	23	In-company Training	23	4	Multinational	>1000

[4] Available at: http://www.linkedin.com
[5] Available at: https://www.authentique.com.br

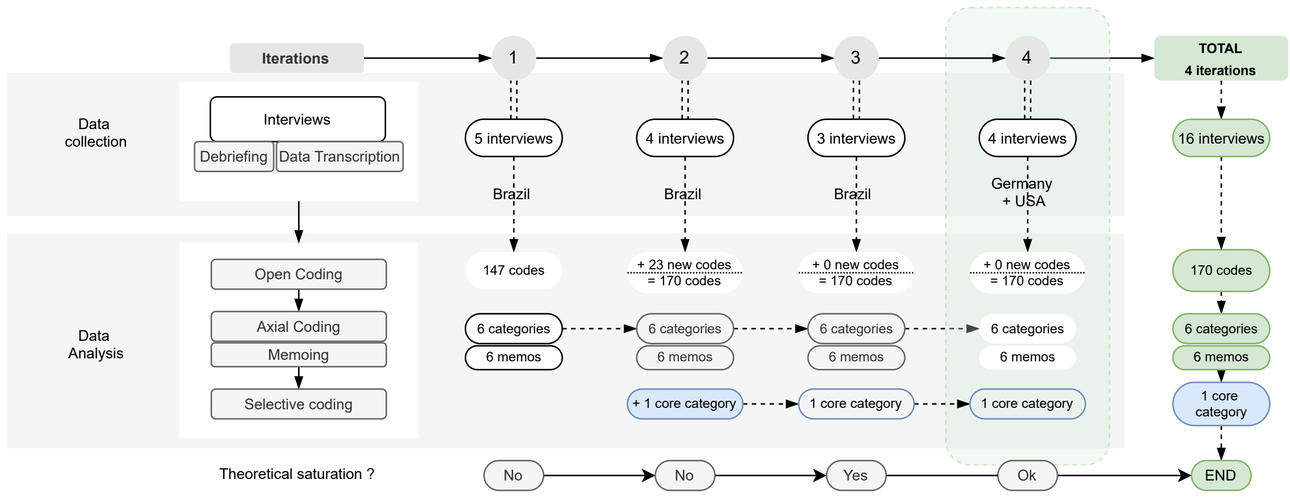


Figure 7.7: Interview study conduction and data analysis

USA (Iteration 4)[6]. All interviews were conducted via Zoom platform[7] in 2022 and 2023. We recorded audio and video, which was allowed by the participants. The interviews took 38 minutes on average.

Once conducting the interviews, we initially asked the participants if they were responsible for the selection of DT techniques to use for software development (e.g., conducting DT workshops). Next, we asked how they make decisions on which techniques to use, the decision strategies and what resources they use for collecting data about techniques to decide which to use. We also asked follow-up questions not initially included in the interview script in order to explore points the participants included in their answers. Finally, we asked them demographic questions[8].

B.3) Data Analysis

We started the data analysis activity right after each interview ended up. For each interview, we created a debriefing document to register the key points we perceived during the interviews. Debriefing documents help to record the nuances and the insights of the researcher during a data collection procedure [9]. We based our interviews' debriefings on Pereira *et al.* (2018) [202]. Next, we transcribed the interview's audio into text, starting the coding process. Since we followed the Straussian version of GT, we coded data in 3 steps: 1) Open Coding, 2) Axial Coding, and 3) Selective Coding.

1. *Open Coding*: the process of generating codes that represent text excerpts, which contain key points in the transcripts referring to the defined topic [112, 247]. We

[6] Even though we did not add new codes in Iteration 3, we performed a new iteration with professionals from other contexts to verify the results we obtained in the previous three iterations.
[7] Available at: http://zoom.us
[8] Additional data was collected during the Doctoral exchanging program at the Hasso-Plattner Institute at the Potsdam University, Potsdam, Germany.
[9] Example of a debriefing document: https://encurtador.com.br/KOX12

started the Open Coding right at the end of each interview (a GT procedure). We analyzed each interview transcript, line by line and sentence by sentence. We used Atlas.ti (academic web version)[10] to assign codes to the transcripts.

During the data analysis, we practiced the constant comparison method [92]. We compared the codes assigned to the interview being analyzed with the codes that we assigned to other interviews previously analyzed. We also analyzed the codes by looking at the interviews' debriefings. As a result, in Iteration 1 we generated 147 codes and in Iteration 2 more 23 codes. In Iterations 3 and 4 we did not add any new code during the Open Coding step[11] (see Figure 7.7).

2. *Axial Coding*: based on the codes generated in the Open Coding activity, we started to perform Axial Coding. Axial Coding is the next level of code abstraction in Straussian GT. Stol *et al.* (2016) [247] points out that Axial Coding is the data analysis activity that focuses on identifying the relationship of the categories given by the codes obtained in Open Coding. It aims to link categories and converge data in theory.

   We followed the example of Kroeger, Davidson and Cook (2013) [138] to look for relationships among the codes to generate categories. The authors created an affinity diagram to relate codes generated in the Open Coding activity and obtain the categories by clustering similar codes. In Iteration 1, we generated 7 categories of codes. Then, we started to write memos for each of the categories we identified. Each memo contains snippets to help us to generate a descriptive decision-making model which represents the selection of Design Thinking techniques. This process is called memoing [112]. In iteration 2, we refined the 7 categories and the content of each memo[12].

3. *Selective Coding*: after concluding the Axial Coding for the 9 interviews (Iteration 1 plus Iteration 2), we searched for the core category obtained by Axial Coding. This activity refers to Selective Coding, which refers to the identification of a core category to which the other categories are related to. In Iteration 3 we re-analyzed the core category based on the new data collected, while in iteration 4, we verified if the new data changed the categories.

C) Theory Building

Finally, we analyzed the memos that we generated in the Axial Coding and performed a memo sorting activity to obtain the conceptual relationship between them and to generate a theory. Based on the core category and by analyzing the memos that we

[10]Available at: http://my.atlasti.com
[11]The codes generated in Open Coding step can be found at: https://www.encurtador.com.br/vzBNW
[12]We created a network of the codes generated in this step. Available at https://encurtador.com.br/eRU57

generated in each category, we created relationships between the categories and wrote snippets that represent each relationship. One of the categories emerged as the core category since it was related to the others. Thus, we draw a descriptive decision-making model of the DT technique selection in software development.

Study Results

This section presents the results of this study. We clustered them into 5 groups: 1) Elements of the decision-making, 2) Influence of the DT techniques evaluation over the decision-making, 3) Influence of the facilitator's experience in DT over the decision-making, 4) Difficulty for selecting DT techniques, and 5) Sources and resources used for collecting information to support the decision-making of DT techniques.

Group 1: Elements and strategies of DM for selecting Design Thinking techniques

Table 7.3 shows the DM elements that DT facilitators use for selecting DT techniques in software development.

P1, P2, P4, P7, P8, P9, P10, P11, P12, P13, P14, P15 and P16 mentioned they consider the goal that they want to achieve for selecting a DT technique related to the phase that they are in the process. P2, P3, P8, P10, P12, P14 and P16 mentioned that they analyze the time available to apply/use a technique as a DM element for selecting DT techniques. P1, P4, P6, P8, P13 and P16 assume that the challenge to be solved is a key element to decide which DT technique to use. P1, P4, P7, P11, P13, P14 and P16 consider that the moderator's knowledge (DT expertise) impacts over the decision of the DT techniques. P1, P3, P6, P7, P11, P12, P14, P15 and P16 advocate that the selection of DT techniques relies on the organization's maturity in DT or the stakeholder's experience in DT. For instance, P1 indicates that teams where the members already know each other collaborate with the moderator's role in a DT session and if the team has experience in using DT techniques, change the way the moderator selects the techniques.

We also asked the participants about the decision-making strategies for selecting DT techniques. Decision-making strategies involve the criteria the individual uses to decide [267]. The authors proposed a DM strategies taxonomy. We followed such taxonomy

Table 7.3: DM elements for selecting DT techniques

Codes	P#
Technique selection by goal and phase	P1, P2, P4, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16
Technique selection based on time available	P2, P3, P8, P10, P12, P14, P16
Technique selection by challenge	P1, P4, P6, P8, P13, P16
Technique selection based on the moderator's knowledge	P1, P4, P7, P11, P13, P14, P16
Technique selection by the knowledge of stakeholders and the organization's maturity level	P1, P3, P6, P7, P11, P12, P14, P15, P16

and adapted it to the context of DT technique selection. Thus, in our interview study, the participants should indicate on a 5-point Likert Scale (Never, Rarely, Sometimes, Often, Always) the frequency they use to select DT techniques based on the DM strategies. We used a Likert scale based on an instrument for collecting DM proposed by Scott and Bruce [233]. Table 7.4 shows the statements we inserted into the interview script about the DM strategies for selecting DT techniques.

Table 7.4: Decision-making strategies adapted to the selection of DT techniques

Category	**Strategy**	**Criteria**	**DT techniques context**
Intuitive	Arbitrary	Based on the most easy or familiar choice	I select DT techniques by considering those that will give me the least effort to decide or those that I have previously selected
	Preference	Based on propensity, hobby, tendency, expectation	I select DT techniques based on trends in techniques that have been used or on expectations from techniques that I know of
	Common sense	Based on axioms and judgment	I select DT techniques based on other users' evaluations of techniques that are good alternatives to use
Empirical	Trial and error	Based on exhaustive trial	I select DT techniques based on several attempts to use techniques in search of the best alternative, and in some of these attempts I may not have been successful, which led me to look for other techniques
	Experiment	Based on experiment results	I select DT techniques based on the use of techniques, considering those that were successful
	Experience	Based on existing knowledge	I select DT techniques based on experience I have gained using the techniques and/or on experience of others who have also used such techniques
	Consultant	Based on professional consultation	I select DT techniques by consulting with other more experienced professionals (directly or indirectly) who use DT in software development
	Estimation	Based on rough evaluation	I select DT techniques from a rough evaluation of the result, even if I have not used them previously
Heuristic	Principles	Based on scientific theories	I select DT techniques based on scientific theories that reference the techniques
	Ethics	Based on philosophical judgment and belief	I select DT techniques based on philosophical judgments and beliefs that I have
	Representative	Based on common rules of thumb	I select DT techniques from the characteristics that represent those techniques
	Availability	Based on limited information or local maximum	I select DT techniques based on those that most quickly come to mind
	Anchoring	Based on presumption or bias and their justification	I select DT techniques based on a threshold analysis for what I need, and adapt as decision needs present themselves
Rational	Minimum Cost	Based on minimizing energy, time, money	I select DT techniques that show the lowest cost of resources, time, people, among others (without looking at the benefit)
	Maximum cost	Based on maximing gain usability, functionality, reliability, quality, dependability	I select DT techniques that are shown to generate a greater benefit (quality, meeting needs, usability, integration of participants)
	Maximum utility	Based on cost-benefit ratio	I select DT techniques analyzing the techniques that show a good cost-benefit
	Interactive events	Based on automata	I select DT techniques considering the events that can occur from this decision and following a line of thought
	Games	Based on conflict	I select DT techniques based on the theory of games, that is, from the analysis of gains and losses with the decisions that I will take, or by resolving conflicts (disagreements) with others
	Decision-grids	Based on a series of choices in a decision grid	I select DT techniques by making filters based on the alternatives they have, like, "If I select X, then I can select Y, ..."

Figure 7.8 shows the DM strategies for selecting DT techniques. The 3 most cited DM strategies for selecting DT techniques in software development are Decision-grids (Rational), and Experience and Experiment (both Empirical). Decision grids refer to decisions that are "*based on a series of choices in a decision grid*" [267]. Experiment and Experience refer to decisions "*based on experiment results*" and decisions "*based on existing knowledge*", respectively [267]. It is also important to mention that the criteria Minimum cost was indicated was the less considered criterion for DT facilitators when they select DT techniques.

Group 2: Influence of the DT techniques evaluation over the DM for selecting DT techniques

We asked the participants if they evaluated the DT techniques after using them and if the evaluation influences their decision-making (see Table 7.5). P8 and P14 indicated that they do not evaluate DT techniques after using them. On the other hand, all the other DT facilitators indicated that they somehow evaluate the techniques and consider such evaluation as an element that supports the decision-making of what techniques to use in future DT sessions/workshops.

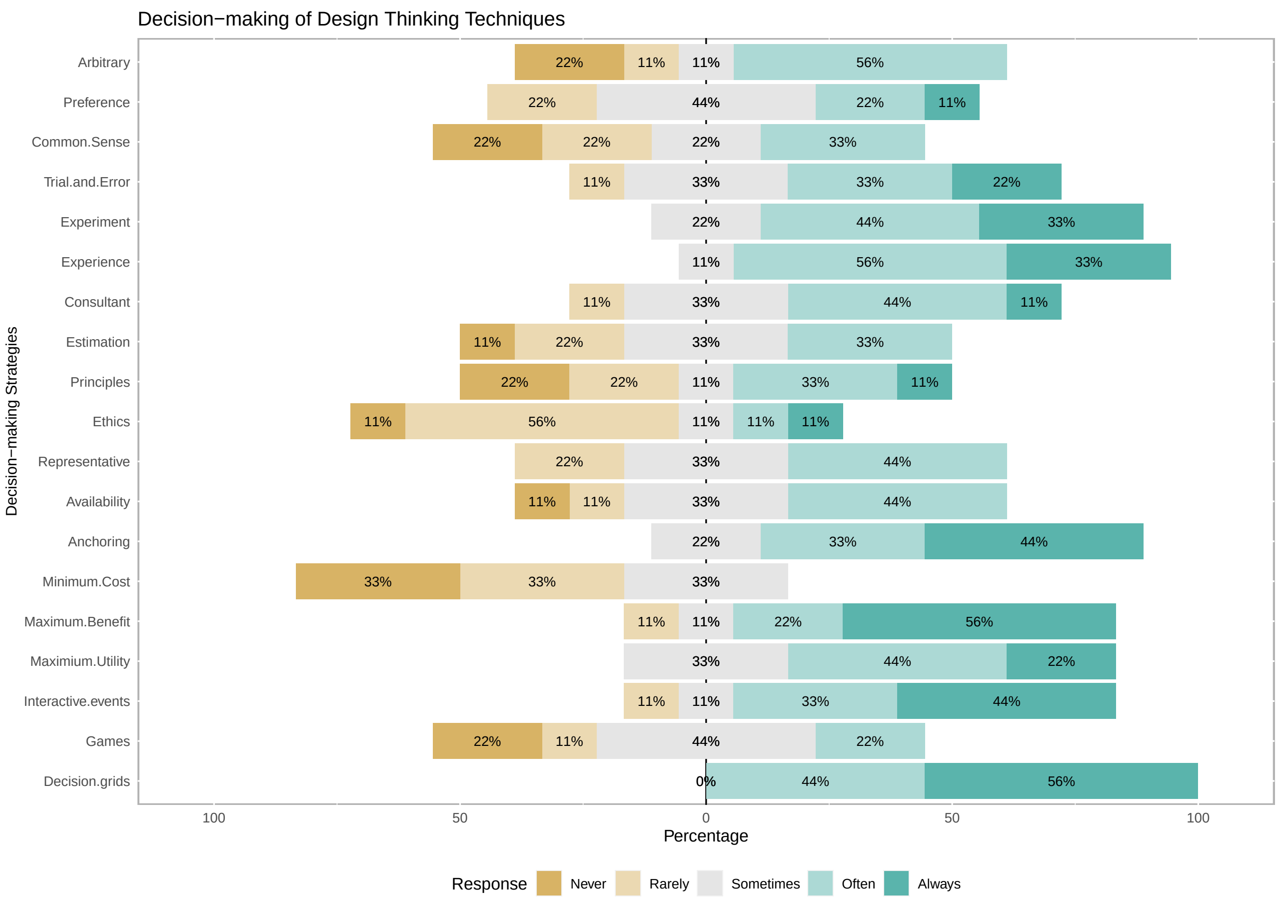


Figure 7.8: DM strategies for selecting DT techniques

Table 7.5: DT techniques evaluation

Codes	Sub-codes	P#
Explicit feedback collection with participants	Questionnaire/Satisfaction form	P1, P2, P3, P13, P15
Moderator analysis of the participants' perception	Experience to the participants	P1, P10, P13, P14, P15, P16
	Participant's understanding on using techniques	P1, P3, P6, P7, P10
	Participant's engagement	P1, P4, P5, P11, P12, P15
DT techniques results comparison	Application time	P2, P14
	Technique effectiveness (reaching the goal)	P1, P2, P7, P9, P13, P14, P15

Thus, we asked the participants what instruments they use to evaluate DT techniques and when they perform that evaluation. The participants indicated both evaluations per technique or per DT session/workshop (see Table 7.5).

P1, P2, P3, P13, and P15 mentioned they collect explicit feedback with the participants of the DT session using a questionnaire or a satisfaction form.

The interviewees also mentioned that they analyze the participant's perception on using DT techniques as an evaluation instrument. In that cases, they do not collect explicit data, but they use metrics such as the experience of the participants on using a technique (P1, P10, P10, P13, P14, P15, P16); the participants' understanding of using a technique (P1, P3, P6, P7, P10); the participants' engagement (P1, P4, P5, P11, P12, P15).

P1, P2, P7, P9, P13, P14, P15 cited that they compare the results obtained by applying DT techniques as an evaluation instrument, including the time spent to apply each technique (P2 and P11) and the technique effectiveness, which means if the predefined goal was reached (P1, P2, P7, P9, P13, P15).

Group 3: Experience in Design Thinking over the DM for selecting Design Thinking techniques

Table 7.6 presents the participants' experience in DT and the impact on the DM of selecting DT techniques.

P1, P2, P3, P4, P5, P6, P7, P8, P10, P11, P13, P14, P15, P16 mentioned that the experience in using Design Thinking is relevant to decrease the difficulty for selecting DT

Table 7.6: Experiences in DT and decision-making

Code Group	Codes	P#
Facilitator's experiences in conducting sessions	Decreasing the difficulty in technique selection	P1, P2, P4, P5, P6, P7, P8, P10, P11, P13, P14, P15, P16
	Making decisions together with other professionals	P8, P10, P11, P14
	Allocating time to understand the technique	P7, P13, P16
	Knowing the participants' profile	P1, P2, P3, P4, P5, P8, P9, P10, P12, P16
	Knowing the availability of the participants as a selection criterion	P2, P9, P10
	Understanding the maturity of the team/organization	P1, P3, P6
Participants experience in DT	Know what will be done as a DT activity	P1, P6, P16

techniques. P8, P10, P11 and P14 mentioned that experience in DT also opens rooms for making decisions about the selection of DT with other DT facilitators. P14 highlights that "the more experience you have and the more methods you know because I think a good and experienced Design Thinking Coach is highly connected with the amount of methods you know... and how secure you are". P7, P13 and P16 mentioned that after they gained experience in DT, they started to allocate time to better understand/ to experiment a technique before applying it.

Knowing the participants' profile (P1, P2, P3, P4, P8, P9, P10, P12 and P16) and their availability (P2, P9 and P10) for participating in a DT session before starting the session were elements mentioned by the interviewees. P1, P3 and P6 indicated that they consider the maturity of the team as an element of DM. P1, P6 and P16 mentioned that is relevant to consider the experience of the participants (i.e., what they know about the DT techniques).

Group 4: DT facilitator's difficulty for selecting DT techniques

We asked the participants to indicate on a scale of 1 (easier) to 10 (harder) what is the perceived difficulty of selecting DT techniques. Table 7.7 shows the participants' experience in DT (in years), their current difficulty level for selecting DT techniques and their status after having to experience in DT.

Table 7.7: Participant's difficulty for selecting DT techniques

P#	**Experience in DT (years)**	**Difficulty level (scale 1 - 10)**	**Status getting experience in DT**
P1	12	5	↖
P2	7	8	→
P3	2,5	9	→
P4	2,3	3	↖
P5	7	3	↖
P6	2,5	7	→
P7	7	4	↖
P8	5	7	↙
P9	4	5	↖
P10	3	7	→
P11	10	2	→
P12	5	7	→
P13	25	2	→
P14	12	easy*	→
P15	8	7	↙
P16	23	easy*	→

↖ Difficulty decreased after getting experience in DT.

→ Difficulty kept similar even after getting experience in DT.

↙ Difficulty decreased after getting experience in DT

* The participant did not indicate a numeric level of difficult

P1, P2, P4, P6, P7, P9, P10, P12 and P15 pointed 5 or higher as a perceived level of difficulty in selecting DT techniques. P1, P4, P7, and P9 mentioned that before having experience, the level of difficulty was higher. To the others, the level of difficulty did not change. P4 appears as the most decreasing value (8 to 3). P10, P13, P16 mentioned that they consider the difficulty in selecting the DT techniques as low since they consider themselves very experienced DT facilitators.

Table 7.8 shows the reasons that the participants indicated that turn the selection of DT techniques into a challenge.

P2, P7, P9, P10, P11, P13 indicated the time they have available for using the techniques or according to the context of the workshop modify the difficulty level of using a technique. On the other hand, P1, P8, P11, P12, P13 and P15 pointed out that the higher level of difficulty for selecting DT techniques comes from the lack of participants' knowledge in using DT techniques (including they as moderators). For instance, P13 indicated that when she started using DT it was harder to decide which techniques to use because they didn't know which methods or approaches were a better fit based on the stage they were in their design process and they didn't know what activities were good for a particular outcome and goal of the workshop.

P1, P2, P3, P5 and P12 associated difficulty in selecting DT techniques to the required changes to a virtual environment due to the COVID-19 restrictions. On the other hand, P16 mentioned that virtual mode does not make it harder or easier to decide what techniques to use, it's just a different way of working and it depends on the audience.

P2 pointed out that she considers it harder to select DT techniques when she has to work on a new product than when she has to work on a product improvement (i.e., when she already knows the current solution). For P14, it is more difficult for selecting DT techniques when it is out of her comfort zone. P8 mentioned that she started to conduct more complex workshops as she gained more experience and, therefore, as more complex the DT session is, the more difficult is to select the DT techniques. P4 indicates that in her case the difficulty in selecting DT techniques comes from the lack of exchange of experiences with fellow facilitators. On the other side, P14 indicated that when she started conducting design thinking workshops, it was easier for her to select DT techniques because she was not the expert and she could ask for help, but now it's different because she is considered the expert and she has to back up other people.

Table 7.8: Difficulty reasons when selecting DT techniques

Codes	**P#**
Time and workshop context	P2, P7, P9, P10, P11, P13
Participants' misunderstanding or lack of knowledge (including facilitators)	P1, P8, P12, P13, P15
Virtual format	P1, P2, P3, P5, P12
Solution type (product improvement is easier than creating new products)	P2, P8, P14
DT session complexity	P8
Lack of exchange of experiences of fellow facilitators	P4

Next, we show the sources and resources professionals use to look for information about DT techniques.

Group 5: Resources used by facilitators to support the selection of DT techniques

We also asked the participants what sources they use to collect data about DT techniques. Table 7.9 summarizes the sources of DT techniques' information.

Most professionals who we interviewed mentioned it was common for them to gather DT techniques information from books related to DT (P14, P16), especially before having experience in DT (P1, P3, P4, P7, P9 P11). P4 and P7 also highlighted they search for DT techniques in books from other subjects, such as Agile and Lean, in order to expand their toolbox. P6, P9, P11, P12, P13, P14 and P15 mentioned DT toolkits (compilation of DT techniques) as sources of DT techniques. P9 highlights that the company where she works created its own DT toolkit and she uses it as the main DT techniques information source. P13 mentioned they have templates built out for both in-person activities and we have templates and resources and those same resources built out for virtual.

On the other hand, P2 and P8 mentioned that they collect DT techniques in DT training activities. P8 also mentioned that her company offers a specialized DT formation to their employees. P2, P3, P5, P6, P10, P14 and P16 consider digital media such as online groups, LinkedIn profiles and groups, and forums as sources of DT techniques. P1, P2, P4, P6, P8, P11, P12, P13, P14, P15, P16 mentioned DT models (set of working spaces) as sources of DT techniques. Double Diamond and Human-Centered Innovation Approach were the DT models cited by the interviewees. In addition, P1, P4, P6, P7, P8, P10, P12, P14, P15, and P16 also mentioned they consider the experience of other DT practitioners to gather information about how to use other DT techniques.

Then, we made follow-up questions to identify if they ask fellow DT practitioners about the experiences they had in using DT techniques. The participants mentioned they consult other participants, creating communities of practice around DT in software development. P1, P4, P5 and P10 agreed that communities of DT techniques experiences are necessary resources to support DT practitioners in selecting DT techniques. P5 and P12 mentioned that these communities serve as sources of DT techniques. P1, P7, P9 and P10 confirmed the communities are composed of DT practitioners. Lastly, P1, P5, P6, P9,

Table 7.9: Source of DT techniques

Codes	**P#**
Books of DT when less experienced	P1, P3, P4, P7, P9, P11, P14, P16
Book of other topics (Agile, Lean)	P4, P7
DT toolkit/templates	P6, P9, P11, P12, P13, P14, P15
DT training	P2, P8
Digital media (LinkedIn, Groups)	P2, P3, P5, P6, P10, P14, P16
DT model (working spaces processes)	P1, P2, P4, P6, P8, P11, P12, P13, P14, P15, P16
DT practitioners	P1, P4, P6, P7, P8, P10, P12, P14, P15, P16

P10 and P12 mentioned that communities are important since they allow to exchange of experiences among the members. On the other hand, P16 mentioned that since she is an expert in DT, she does not collect information from outside of her company.

Discussion

Investigating how DT facilitators select DT techniques might boost the integration of DT into software development activities. Having subsides to know how to select DT techniques gives support to newcomers in DT and to companies that are starting to use that approach for generating innovative software solutions. It also might aid experienced DT practitioners to integrate DT into their daily SE activities by indicating how they might select DT techniques and infuse them into their work routine. Next, we discuss our findings about the decision-making behind the selection of DT techniques in software development and lastly, we present a decision-making model of the selection of DT techniques.

In this context, we present our findings about decision-making for selecting DT techniques in software development.

The decision-making of DT techniques is composed of decision-making elements, and it is **context-driven**.

Decision-making connects a creative design mindset to traditional business thinking based on rational problem-solving [258]. When selecting DT techniques in software development, DT facilitators consider 5 contextual elements: time, goal, challenge, phase in the process and session format (virtual or in-person) (see Figure 7.9). The facilitators use to collect those elements before starting a DT workshop, registering them in documents called “briefings”.

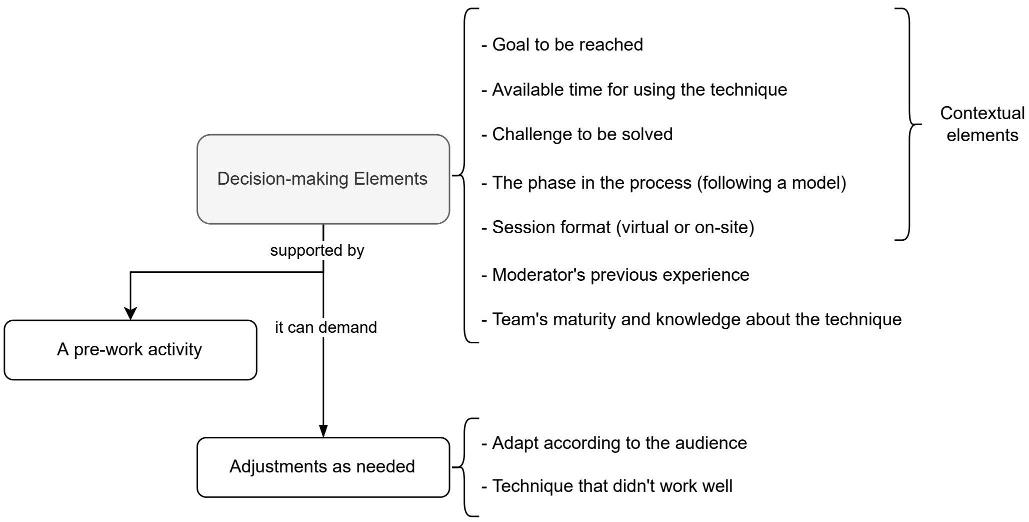


Figure 7.9: DT techniques decision-making elements

Facilitating DT sessions, workshops or projects is an activity that requires the facilitators to be able to work against the clock. Parizi *et al.* (2022) [195] also highlighted time pressure as an attention point when using DT in software development. De Paula, Amancio and Flores (2020) [64] also mention time as an element to pay attention to when selecting DT techniques. Souza *et al.* (2020) [245] cite the goal to be reached as a strategy to select the techniques to be used and Kasper *et al.* (2019) [104] indicates that the selection of DT techniques is based on the challenge to be solved.

The facilitator's knowledge of DT helps to work on the definition of the rational elements for selecting DT techniques. The team's maturity in DT also aids the selection of DT techniques. Therefore, the DT facilitator needs to profile the team and select DT techniques that attend such profile, adjusting the techniques wherever necessary (e.g., to increase the participants' engagement).

Experiment, experience and decision grids are the most used decision-making strategies for selecting DT techniques. Therefore, DT facilitators use to exercise the DT mindset that is open to experimentation and fails often and early [35]. Moreover, they consider their experiences by using techniques as subside to select the techniques for the next workshop, as also indicated in Literature [167, 64, 136]. In addition, the selection of DT techniques is supported by decision-grids, where the decisions are made based on a series of choices [267]. Decision grids are associated with DT models. DT models allow associating techniques to the working spaces (steps for simplification) and aid the selection of DT techniques since they encapsulate the goal to be achieved [195].

> The decision-making of DT techniques is supported by techniques evaluation.

Tello (2019) defines the evaluation of alternatives as part of the decision-making process [254]. In the software development context, DT facilitators evaluate DT techniques by collecting feedback from the participants. The feedback might be collected after the technique is used or after the workshop is completed (see Figure 7.10). In addition, the DT facilitator's perception of the participants' experience, understanding, and engagement are implicit feedback of each technique that supports the selection of that technique in future workshops. A technique is seen as good if the goal was reached or if the time required was feasible. A workshop was effective if it supported building the product backlog. Dobrigkeit and De Paula (2019) [68] also advocate that sharing results from DT activities and ensuring an effect on the final product is fundamental.

> The decision-making of Design Thinking techniques relies on DT practitioners and participants' experiences.

Experience in using DT supports the decrease in the difficulty of DT practitioners in selecting DT techniques (Figure 7.11). DT facilitators make decisions by allocating time

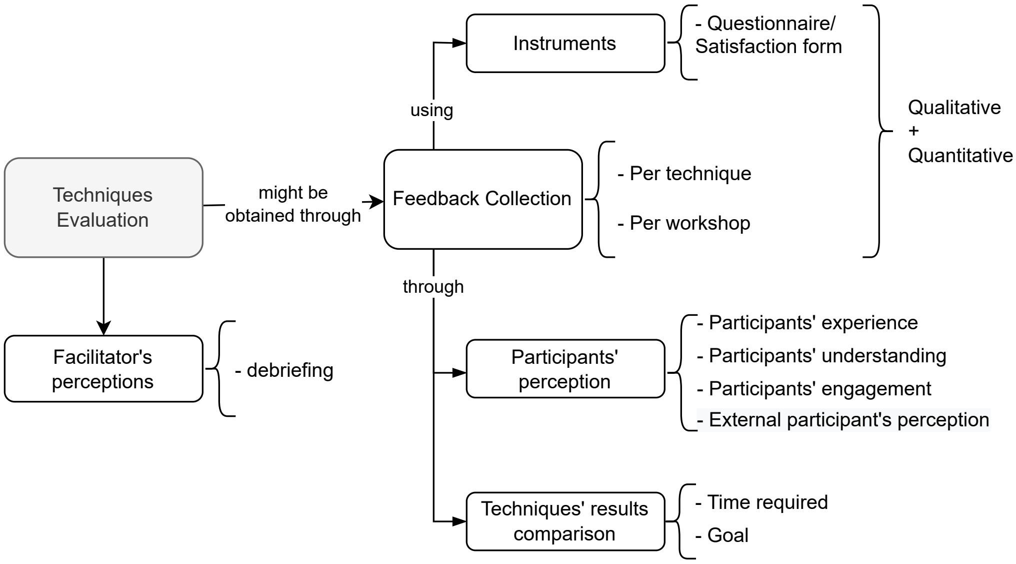


Figure 7.10: DT techniques evaluation

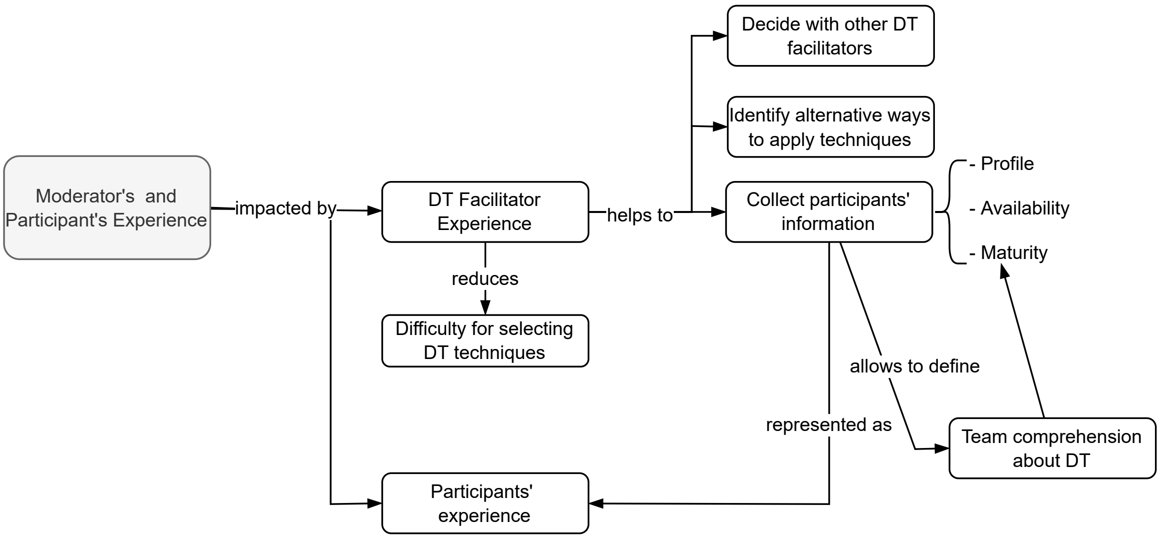


Figure 7.11: Participants' experience in the decision-making of DT techniques

to understanding a technique and profiling the participants. DT facilitators are aware that DT techniques are time-dependent and that the techniques also require a pre-work activity to well-establish the set of techniques to be used. Thus, before selecting DT techniques, it is necessary to investigate the stakeholders in detail, take the time to know their needs, and choose the appropriate innovation techniques for ensuring the production of solutions meeting the user's expectations [64].

Dobrigkeit and De Paula (2019) [68] argue that the professionals' experience in DT impacts how they understand the benefits of DT in software development. Less experienced professionals in DT use to see the perspective of DT as a technique for a particular goal, while the more experienced ones understand that DT can change the whole organization's mindset.

Less experienced DT facilitators perceive the DM of DT techniques as a hard task.

The literature points out that lack of employee commitment, collaboration and knowledge make the decision of what techniques to use in an endeavor [195]. Figure 7.12 shows elements that DT facilitators consider difficult to select DT techniques.

DT facilitators associate the product type as a factor that impacts the difficulty of selecting DT techniques. Using DT for proposing new products is considered harder than working on improving a known solution. This definition takes into account that when improving an already known product, the user's needs are already known too. The session format also collaborates with the DT techniques decision-making endeavor. DT facilitators indicated that the transition from in-person to remote (virtual) DT sessions increased the difficulty of selecting what DT techniques to use and also how to use them.

DT facilitators consider the availability of time (time destined to run a workshop and the techniques the facilitator has chosen), the participants' knowledge of DT (and about the techniques), and the accessibility required to provide the participants' experience when running a DT workshop as decision factors. Therefore, the facilitator has to know what techniques to use to cope with some participants' disabilities if it is the case.

Combining different techniques also makes it hard to select DT techniques. DT facilitators pointed out that combining techniques is challenging since they do not know how the participants will attend to that combination. This challenge fosters DT facilitators

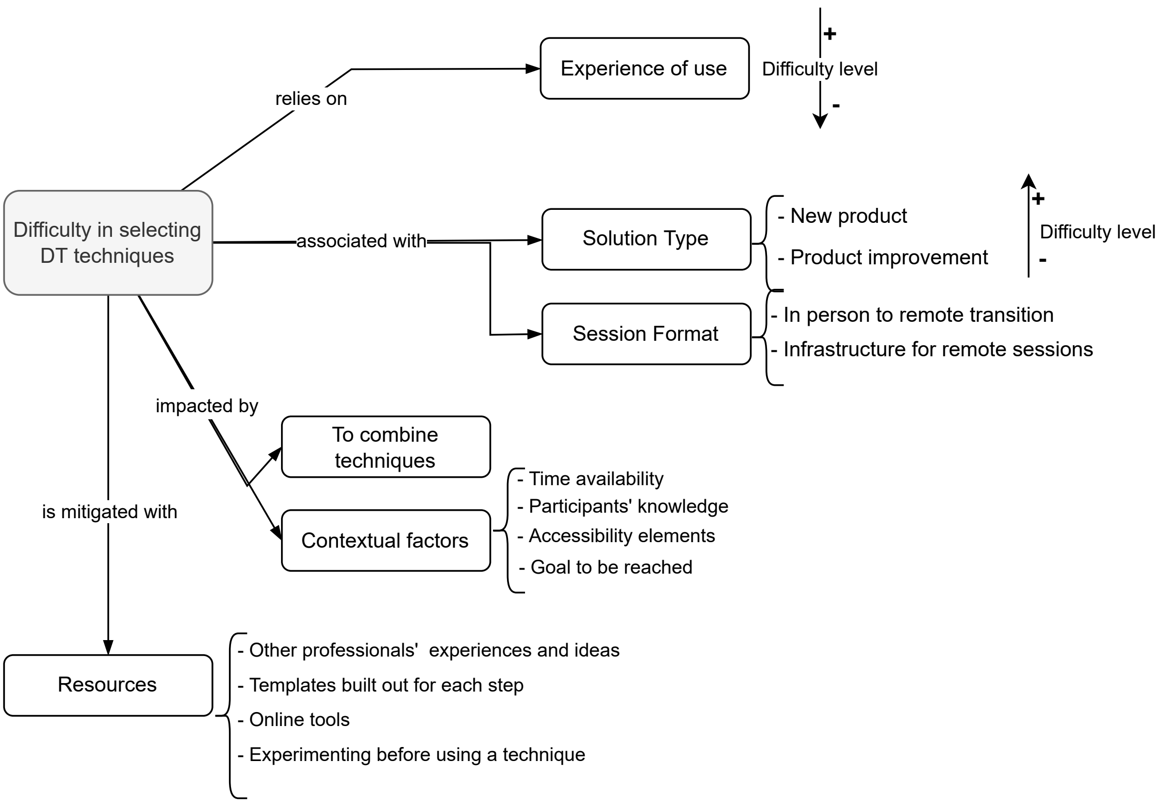


Figure 7.12: Difficulty for selecting DT techniques

to explore alternative resources that provide information about DT techniques, including other professionals' experiences, ideas, and trial and error. Online social media and digital DT toolkits are tools that provide mechanisms to find DT experiences.

In this context, Souza *et al.* (2020) [245] presented DTA4RE, a tool for recommending DT techniques to support RE. Through a form-based algorithm, the tool asks the users about the scenario they have and exposes the applicable DT techniques. DTA4RE is a web-based tool recommending techniques according to three working spaces of DT: inspiration, ideation, and prototyping.

> The decision-making of DT techniques is based on different sources of techniques and powered by a community of DT practitioners

DT facilitators aim to be not behind the time. They often look for new techniques and work against the clock to get more information about applying DT techniques. Such information can be gathered by practicing DT or by consulting other DT facilitators. Books, DT toolkits, and training are also sources of information about DT techniques (Figure 7.13). However, given the dynamic nature of DT, DT facilitators also use books from other subjects, such as Agile and Lean Inception. It means that DT in software development is seen as an approach that can be infused into SE activities [64].

DT facilitators search for data about DT techniques in their social networks or consult DT experts. Establishing a community of DT facilitators is a way of sharing experiences and collaborating to build collective knowledge (Figure 7.14). Community of practice (CoPs) is also applied in other SE subjects. For instance, Kniberg and Ivasson (2012) [135] proposed the Spotify model as a CoP for scaling management and collaboration in self-organized teams using Agile. In the context of DT, a CoP support experience exchange among similar professionals reduces the difficulty of selecting DT techniques.

Box 7.1 presents a Decision-making statement, which defines a theory behind the selection of DT techniques in the context of software development. Therefore, taking into account the decision-making elements, strategies, topics and discussions we presented

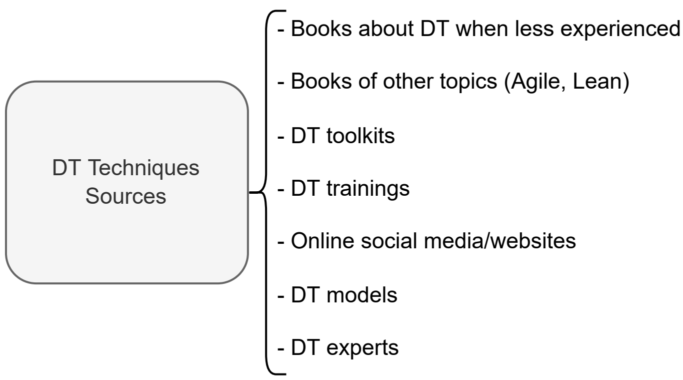


Figure 7.13: Source of DT techniques

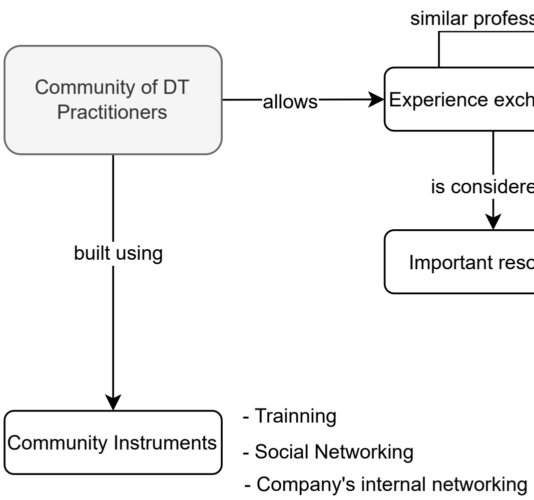


Figure 7.14: Community as a resource for selecting DT techniques

Box 7.1: Decision-making theory statement

Decision-making statement

The DT facilitators' decision-making for selecting Design Thinking techniques in software development is **context-driven**.

The selection of DT techniques

- is composed of decision-making elements
- is aided by techniques evaluation
- relies on participants' experiences
- is perceived as a difficult task
- is based on different sources of techniques
- is powered by a community of DT practitioners

in this study, we can define that the decision-making of DT techniques by DT facilitators in software development is context-driven. Figure 7.15 compiles a descriptive model of decision-making of the DT techniques selection in software development.

Study limitations

Our interview-based study for modeling the decision-making behind the selection of DT techniques contains the following threats to validity:

- *Interview sample*: We interviewed 16 DT facilitators in total and the information that they provided only represents their point of view and not necessarily the organization's point of view. Therefore, our results can not be generalized (External Validity). To mitigate this threat, we interviewed professionals from 10 different software development companies to get a more comprehensive view of decision-making.

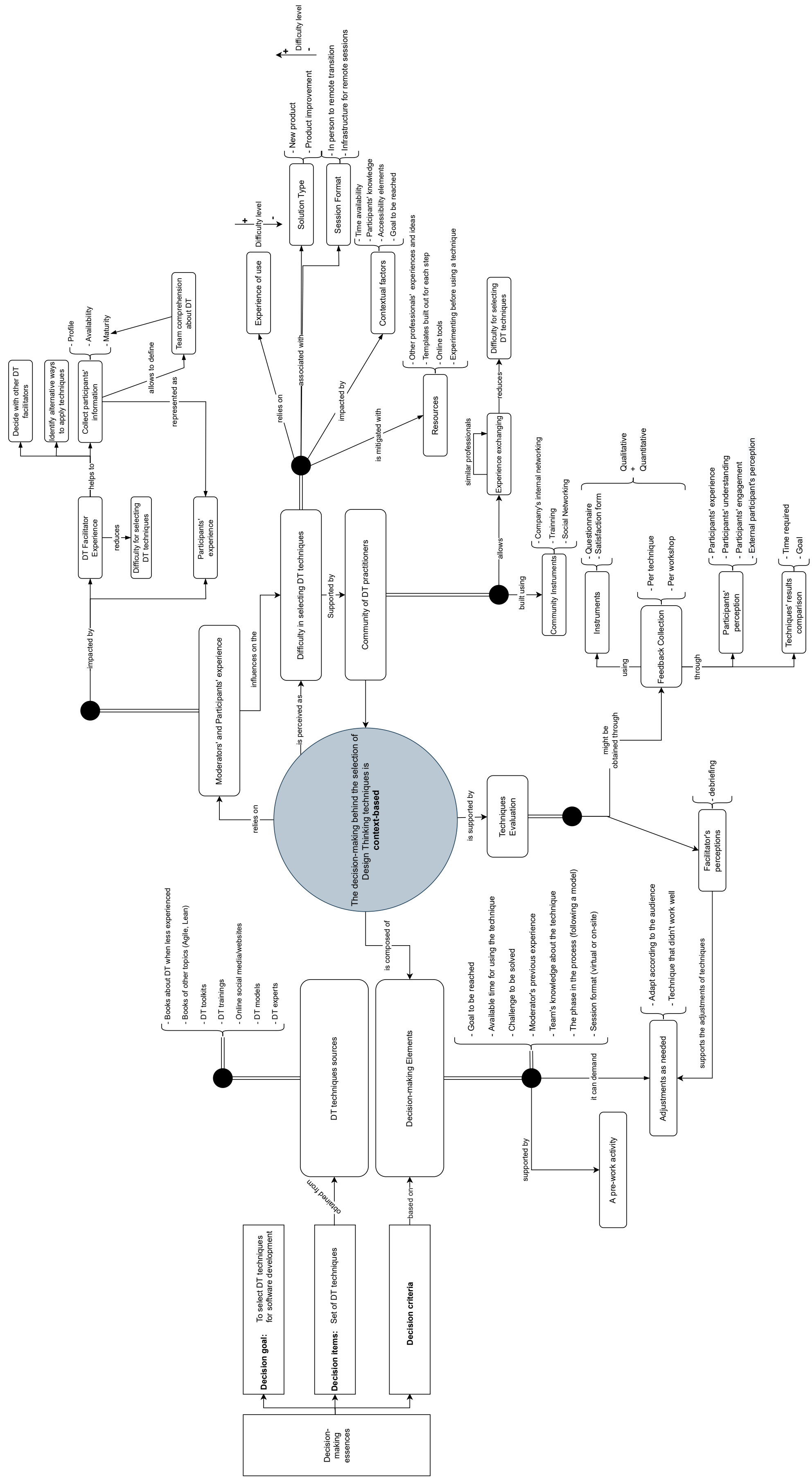


Figure 7.15: Abstract representation of the DT techniques decision-making

- *Data Analysis:* As we based our study on Grounded Theory, coding activities (open coding, axial coding, and selective coding) represent our interpretation of the data we collected through the interviews. To mitigate an interpretation bias, we analyzed the codes in meetings to discuss the results. We also practiced the constant comparison method proposed in GT to do multiple analyses of the codes in each method's iteration;

- *Interview script:* We created a script for conducting our semi-structured interviews. We might consider that we should include other questions in our script. To mitigate this threat (Internal Validity), a senior researcher in qualitative research evaluated and approved the interview script. We also conducted a pilot study to ensure the consistency of the script. In addition, the interviews were audio and video recorded, allowing us to double-check the data transcripts.

Study conclusions

We conducted an interview-based study to investigate the decision-making of DT technique selection in software development. We interviewed DT facilitators to identify how they decide what techniques to use in software development and to generate a theory through a descriptive decision-making model, abstracting the decision-making of DT techniques in software development.

By interviewing 16 DT facilitators, from small to large companies, with different backgrounds and years of experience in DT, our findings indicate that the time to conduct a technique, the challenge to be solved and the goal to be reached drive the DT techniques decision-making of DT facilitators in software development. We also presented the strategies they use for selecting DT techniques, the evaluation of DT techniques and their role in the decision, what happens when DT facilitators got experienced in DT, the difficulty of selecting DT techniques and what resources they use for supporting the decision-making on the selection of DT techniques.

7.2.2 Recommendation Mechanisms based on the decision-making model

Taking into account the decision-making model of the selection of DT techniques in software development, we started to design a set of DT techniques recommendation mechanisms to be implemented in Helius, corresponding to the Feature F01 - DT techniques recommendations (see Table 6.1 in Section 6.1).

Our goal of proposing and using a decision-making model as a starting point to generate the recommendation mechanisms was to approximate the recommendations provided by Helius to the way the DT facilitators select DT techniques in their work routines. We kept in mind the following decision-making elements that we collected with DT facilitators in our interview-based study (Section 7.2.1):

- The decision-making behind the selection of Design Thinking techniques is context-driven, and it involves the following decision-making elements (DMEs):

DME_1: Context definition as a pre-work activity;

DME_2: Experience exchange between professionals;

DME_3: Facilitator's previous experience in using DT;

DME_4: Goal to be achieve and challenge to be solved;

DME_5: Availability of resources and restrictions (e.g., time to execute a technique, participant's knowledge, etc.).

Therefore, the remainder of this section describes the mechanisms that make Helius a collaborative filtering-based recommendation system, taking into account the experience of IT professionals with DT techniques usage expressed in the form of technique ratings. Here, the terms user and active user refer to IT professionals who require DT techniques recommendations for use in software development.

A recommendation mechanism (RM) is a feature provided by Helius that recommends DT techniques based on certain criteria. The set of recommendation mechanisms composes a recommendation module, which is capable of providing intelligence to Helius recommending DT techniques. The results of our previous exploratory studies indicate that the recommendation module by itself makes Helius an innovative solution if compared to the other DT techniques and decision-support tools.

The proposed recommendation module and Helius by itself aim to further the state-of-practice on the collaboration with IT professionals in the decision-making of DT techniques' selection. By being collaborative, Helius uses the experiences of its users on using DT techniques to recommend techniques to other users.

In Helius, the DT technique recommendations work as follows: once the user has used DT techniques, she will be able to evaluate the techniques that she used, providing a rating from her experience on the use of that technique. The set of evaluations made by users creates a techniques' rating dataset, feeding the recommendation system.

Helius' recommendation mechanisms are associated with recommendation strategies. As mentioned in Section 2, there are 2 types of recommendation strategies: personalized and non-personalized, as follows:

- *Non-personalized DT techniques recommendation mechanisms:* recommendation mechanisms which provide non-personalized recommendations offering similar recommendations for all users [25]. In general, non-personalized RS use statistical methods such as Mean Score to generate recommendations, showing the Top-N items with better-resulting scores;

- *Personalized DT techniques recommendation mechanisms:* a collaborative recommendation mechanism provides the recommendation of items to the active user (the user who requires the recommendation) based on the behavior of similar users. In this context, behavior can be understood as the feedback provided by a user for an item in form of ratings (based on a 5-stars scale). These ratings are utilized as data input to provide recommendations for new items.

Figure 7.16 shows the recommendations mechanisms that we designed to implement in Helius. We proposed a total of 20 different mechanisms to recommend DT techniques. We developed independent mechanisms capable of being linked to the RS. Helius uses information about its users (techniques usage and evaluations (or ratings)) in order to process and compute the DT techniques recommendations. Therefore, there are 3 methods that are sources of information to recommend DT techniques: i) Helius users: all users registered in Helius, ii) Active user: user logged in and who is asking for a recommendation in Helius, and iii) Similar users: users that are considered similar to the active user. Helius determines a similarity measure using the KNN algorithm (see Section 7).

Figure 7.17 illustrates the links between the decision-making mechanisms that we designed for Helius recommending DT techniques and the decision-making elements that we collected with DT facilitators.

- Personalized recommendations mechanisms:

RM_1 *Techniques used in projects with a similar context:* it recommends DT techniques that were used by other professionals in projects that have one or more similar contextual items. Table 7.10 presents the contextual values. We collected the contextual variable through the Systematic Mapping Study [116, 245, 64, 71, 48, 104, 68] (see Section 4.1) and through our interview-based study (see Section 7.2.1). This mechanism is part of the *Context definition (DME_1)* decision-making element;

RM_2: *Techniques included to projects for a similar context:* it recommends DT techniques that belong to projects that have one or more similar contextual items. This mechanism is different from the RM_1 since the recommended techniques could not be used yet. This mechanism is part of the *Context definition (DME_1)* decision-making element;

RM_3: *Techniques used in projects with a similar domain:* it recommends DT techniques that were used by other professionals in projects for a similar domain. In Helius, the user might indicate a domain the project is proposed to find for

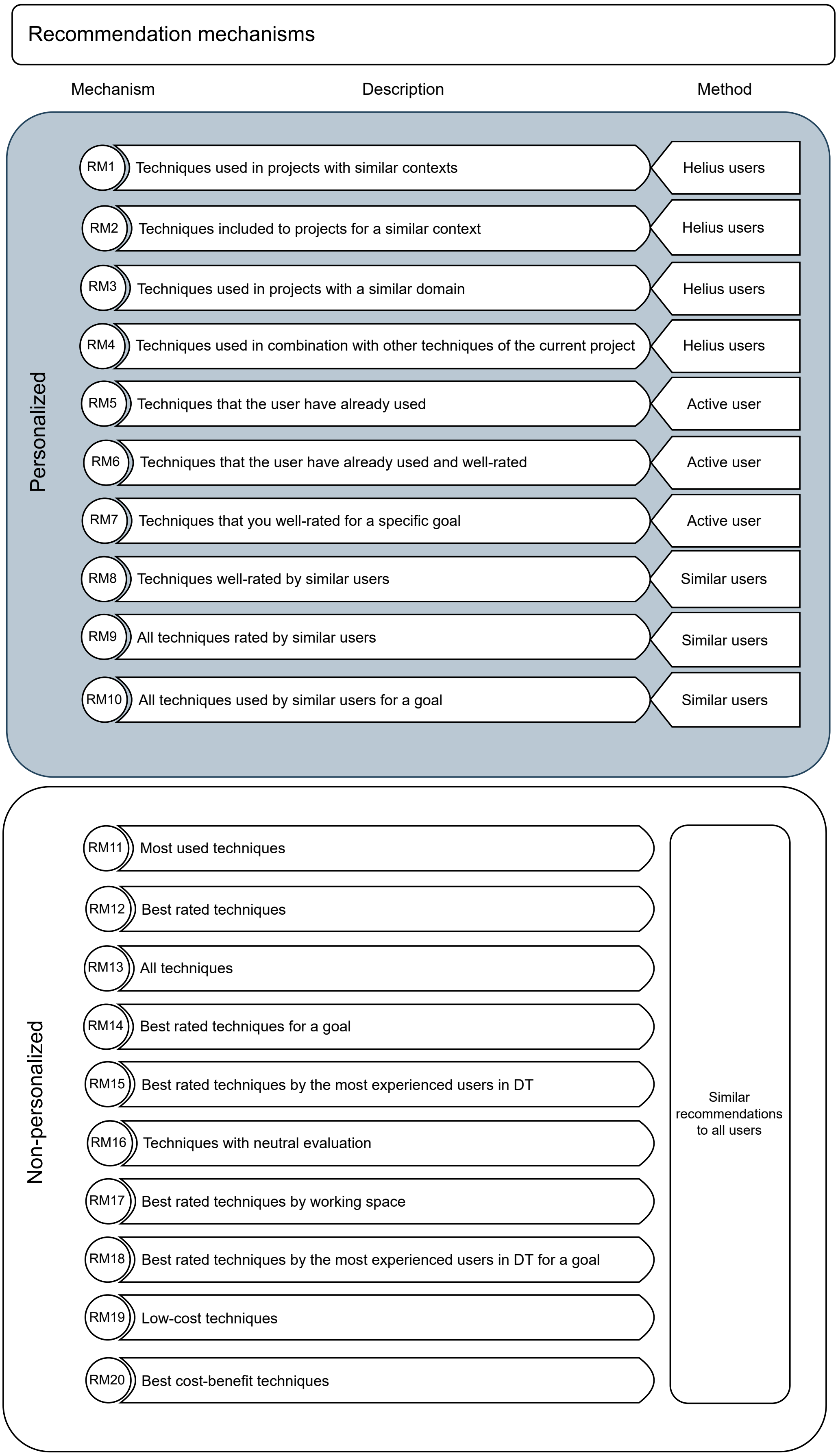


Figure 7.16: Recommendation mechanisms designed to be implemented in Helius

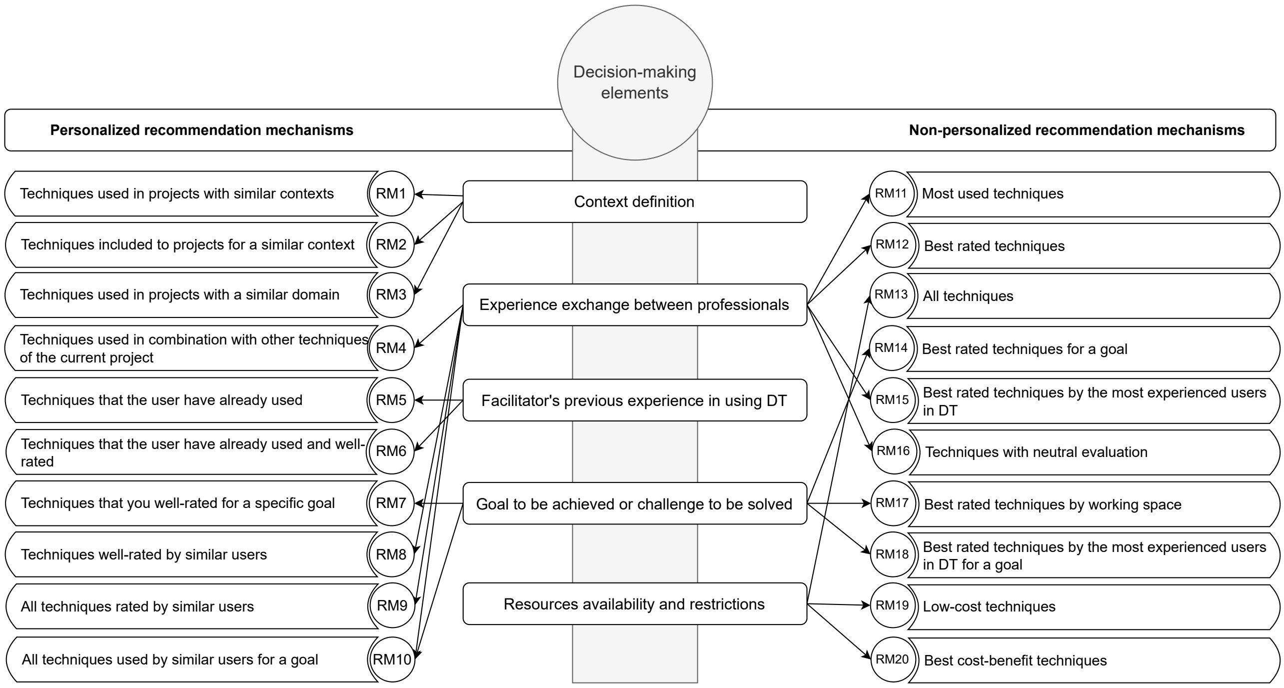


Figure 7.17: Recommendation mechanisms versus DT technique decision-making

Table 7.10: Context options available in Helius

Description	Options
Contact with customer/user	No, Yes
Place to perform the technique	Organization, Client,
Number of participants	Individual (1) , group (2 to 5) , > 5
Participants already know each other	No, Yes
Participants are familiar with DT	Low , Medium, High
Availability of materials	Low, Medium, High
Availability of space to conduct the workshop	No, Yes
Higher management participation	No, Yes
Days available for conducting the workshop	1 to 3, 3 to 5, $>$5
Clarity of objectives	Low, Medium, High
Clarity of problem	Low, medium, High,
Facilitator’s expertise (years)	$<$2 , 2 to 5 , $>$5

a solution. Examples of domains are Game development, embedded systems, cloud systems, and e-health systems. This mechanism is part of the *Context definition (DME_1)* decision-making element;

RM_4: *Techniques used in combination with other techniques of the current project:* it recommends techniques that were used in combination with the techniques present in the current project. RM_4 represents the following statement: *“Who used technique X also used techniques Y and Z”*. This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_5: *Techniques that the user has already used:* it recommends all the techniques that the active user has already used in Helius. This mechanism is part of the *Facilitator’s previous experience in using DT (DME_3)* decision-making element;

RM_6: *Techniques that the user has already used and well-rated:* it recommends all the techniques that the active user has already used in Helius, sorted by the rating she has attributed to the used techniques. This mechanism is part of the *Facilitator's previous experience in using DT (DME_3)* decision-making element;

RM_7: *Techniques that you well-rated for a specific goal:* it recommends all the techniques that the active user has already used in Helius, filtered for a specific goal and sorted by the rating she has attributed to the used techniques. In this mechanism, the user can indicate a goal to filter the techniques. This mechanism is part of the *goal to be achieved or challenge to be solved (DME_4)* decision-making element;

RM_8: *Techniques well-rated by similar users:* it recommends all the techniques that similar users to the active user have used in Helius, sorted by the rating they attributed to the used techniques. Similar users are computed by a neighborhood algorithm based on the techniques of evaluation and rating. This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_9: *All techniques rated by similar users:* it recommends all the techniques rated by similar users to the active user have used in Helius. Similar users are computed by a neighborhood algorithm based on the techniques rating. This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_{10}: *All techniques used by similar users for a goal:* it recommends all the techniques that similar users have used for a goal to the active user have used in Helius. Similar users are computed by a neighborhood algorithm based on the techniques' ratings. In this mechanism, the active user can filter the techniques by a goal. This mechanism is part of the *Experience exchange between professionals (DME_2)* and *goal to be achieved or challenge to be solved (DME_4)* decision-making elements.

- Non-personalized recommendations mechanisms:

RM_{11}: *Most used techniques:* it recommends the most used DT techniques, sorted by use. This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_{12}: *Best rated techniques:* it recommends the best rated DT techniques in Helius, sorted by rating. This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_{13}: *All techniques:* it recommends all the techniques used by the users in Helius, sorted by name. This mechanism is part of the *resources availability and restrictions (DME_5)* decision-making element;

RM_{14}: *Best rated techniques for a goal:* it recommends the techniques for a specific goal, informed by the user and sorted by use. This mechanism is part of the *goal to be achieved or challenge to be solved (DME_4)* decision-making element;

RM_{15}: *Best rated techniques by the most experienced users in DT:* it recommends the best rated DT techniques for those users that have the higher experience in using DT. Most experienced users in DT means to sort the users by experience in DT (information given when creating the user account – "Start DT in" field). This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_{16}: *Techniques with neutral ratings:* it recommends the techniques that have received a mean rating value as 3 (from 1 to 5). This mechanism is part of the *Experience exchange between professionals (DME_2)* decision-making element;

RM_{17}: *Best rated techniques by working spaces:* it recommends the best rated DT techniques according to the working spaces selected by the user. Working spaces represent the DT as a model, as we stated in Section 2. This mechanism is part of the *goal to be achieved or challenge to be solved (DME_4)* decision-making element;

RM_{18}: *Best rated techniques by the most experienced DT professionals for a goal:* it recommends the techniques that the most experienced users have used and rated for a specific goal. The user must be able to indicate a goal to be reached. This mechanism is part of the *goal to be achieved or challenge to be solved (DME_4)* decision-making element;

RM_{19}: *Low-cost techniques:* it recommends the techniques the users evaluated and indicated as low-cost. Low-cost is also represented as a 5-star range (1 low and 5 high) and involves workshop time required (duration), number of participants involved, time to prepare and execute the techniques, and resources required to use a technique. This mechanism is part of the *resources availability and restrictions (DME_5)* decision-making element;

RM_{20}: *Best cost-benefit techniques:* it recommends the techniques which have the best cost-benefit ratio. The benefit is the mean rating the users have attributed to a technique, while the cost is the mean cost attributed by the users to a technique. This mechanism is part of the *resources availability and restrictions (DME_5)* decision-making element.

7.3 Helius Evolution and Empirical Evaluation

This section describes Iteration 3 in the DSR-based research method. Our goal was to improve Helius by implementing the recommendation mechanisms on it to make Helius a DT techniques recommendation system (Section 7.3.1). We also present in this section the empirical evaluation of Helius through a case study with DT practitioners in the software industry (Section 7.3.2).

7.3.1 Implementation of the DT techniques recommendation mechanisms in Helius

After designing the recommendation mechanisms to provide DT techniques recommendations and starting Iteration 3 in the Design Approach in the DSR framework, we implemented the recommendation mechanisms and integrated them into Helius. As we mentioned in Section 7.2, we proposed 20 mechanisms to recommend DT techniques considering both non-personalized and personalized recommendations approaches.

Figure 7.18 abstracts how Helius works to recommend DT techniques. For instance, let's consider the following recommendation mechanisms i) best-rated DT techniques (non-personalized approach) and ii) best-rated DT techniques for similar users (personalized approach). We show below how these mechanisms work in Helius.

- Non-personalized DT techniques recommendation mechanism: *Best evaluated DT techniques* - Helius collects data from users based on explicit feedback (the user rating of the DT techniques she has used in software development). Helius queries a rating database for recommending DT techniques. During the Preprocessing phase, Helius computes the Mean Score of each technique using the ratings provided by the users. A user-techniques-rating matrix is created. In the Recommendation phase, Helius obtain the Top-N best-evaluated techniques by applying a sorting algorithm on the user-techniques-rating matrix (mean score attribute) and it recommends the Top-N best-evaluated techniques.

- Personalized DT techniques recommendation mechanism – *Best rated DT techniques by similar users:* By being a collaborative recommendation system, Helius provides recommendations of items to the active user (a user who requires the recommendation) based on the taste of similar users. In this context, taste represents the feedback provided by a user for an item in the form of ratings (based on a 5-stars scale). These ratings are utilized as data input to provide recommendations for new items. In the Data collection phase, Helius collects explicit feedback from users through the ratings of DT techniques they have used in software development. In this phase, Helius queries the users, techniques, and rating database. Next, during the Pre-processing phase, Helius computes the user-item matrix, a matrix *nxm* where *n* represents the users and *m* represents the items (DT techniques). The value in the matrix represents the rating from the user *u* to the technique *i*.

Helius recommends techniques based on the ratings of similar users from the active user. Helius uses a neighborhood method (KNN) to look for similar users to the active user. It computes the neighbor users (the most similar) to the active user and uses the neighbors' ratings for providing recommendations [163]. Once similar users (neighbors) have been found, Helius moves on and looks for the techniques best rated by the neighbors to the active user. The mean score rating for the techniques is calculated through the user-item matrix. Then, using a sorting algorithm, Helius recommended DT techniques to the active user.

To implement the recommendation mechanisms in Helius, we developed a recommendation module as an API. We used the framework FastAPI, which is based on Python (see Figure 6.10)[13]. For instance, Figure 7.19 shows the non-personalized recommendation mechanism "Most used", which recommends to the users the most used DT techniques. The recommendation module API returns a list of JSON objects, each one representing a technique. The techniques are sorted by use. Appendix B shows the JSON that represents a DT technique returned as a recommendation in Helius and the list of DT techniques we initially registered in the Helius database.

On the other hand, aiming to provide personalized recommendations, Helius take into account the user profile to recommend DT techniques through personalized recommendation mechanisms. Figure 7.20 shows examples of Helius' recommendation mechanisms that return a personalized recommendation based on the user's profile. Helius looks for similar users and then provides recommendations based on their tastes. Similar users are the users that made similar evaluations of DT techniques, which means other users who evaluate DT techniques with similar ratings. To do so, Helius' recommendation module uses the K-nearest neighbor (KNN) algorithm.

We used KNN as the algorithm for the personalized recommendation mechanisms since it is the most cited machine learning algorithm for collaborative recommendation systems, as we figured out through a Tertiary Literature Review that we conducted to identify the appropriate algorithm for our recommendation system (see Appendix D).

Next, we show the user's journey in Helius to access the recommendations mechanisms that recommend the DT techniques to support IT professionals in using DT in software development.

[13]The full documentation of the DT techniques recommendation mechanism is available in https://helius-dt-recommendation-api.herokuapp.com/docs#/.

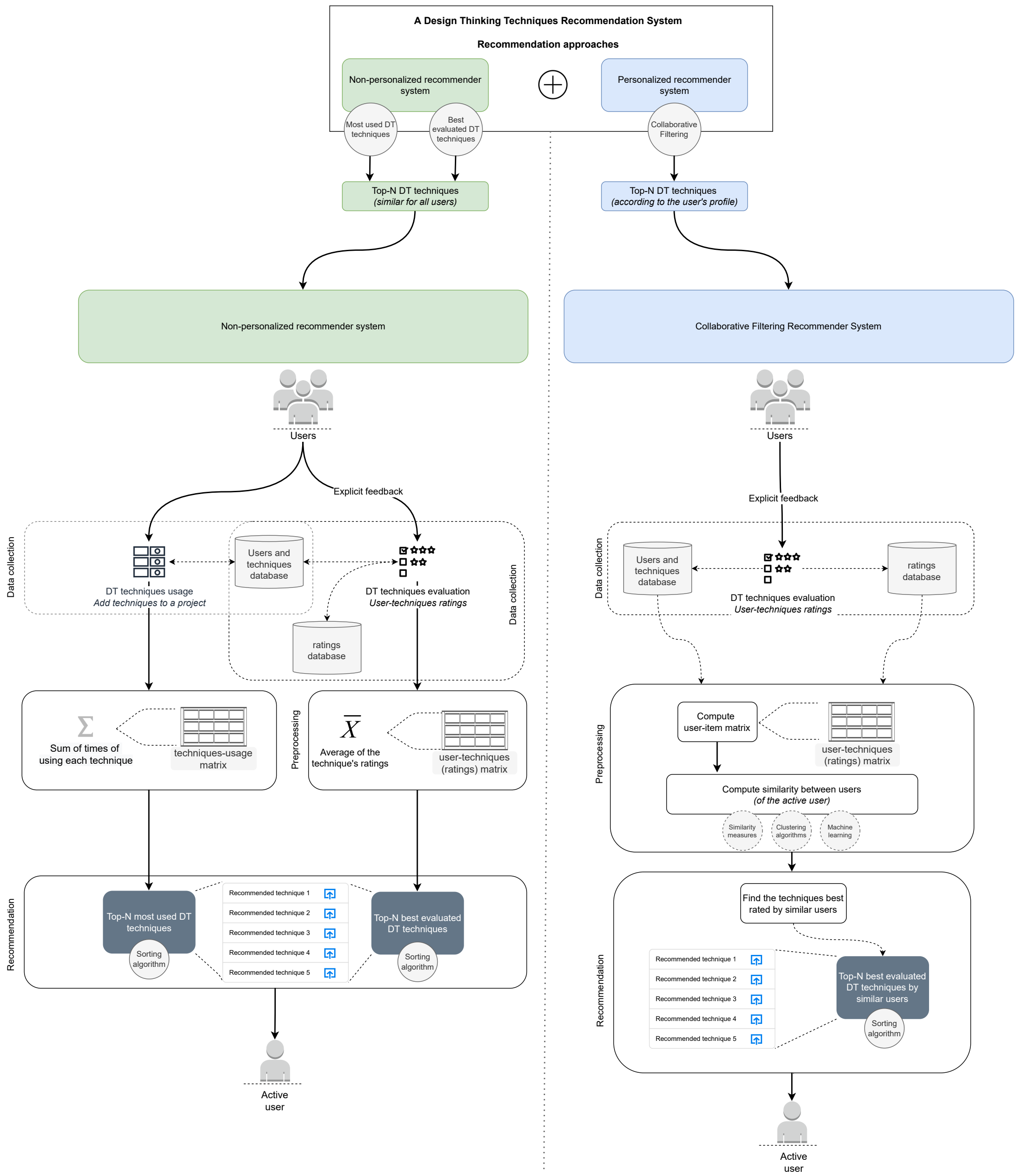


Figure 7.18: Abstract representation of the recommendation mechanisms in Helius

A) Accessing a summary of the DT techniques recommendations

Figure 7.21 represents the user journey to access a summary of the DT techniques recommendations. Item (i) represents the home screen in Helius where the user can access the summary of the recommendations by clicking on the "Recommendations" button. Once the user has clicked on the "Recommendations" button, Helius shows the screen of the recommendations summary. Items (ii) and (iii) show examples of non-personalized and personalized recommendation mechanisms, respectively. Item (iv) shows the button for the user to access the 20 mechanisms we implemented in Helius (as we show in Figure 7.22). In item (v) the user can access a history of the recommendations

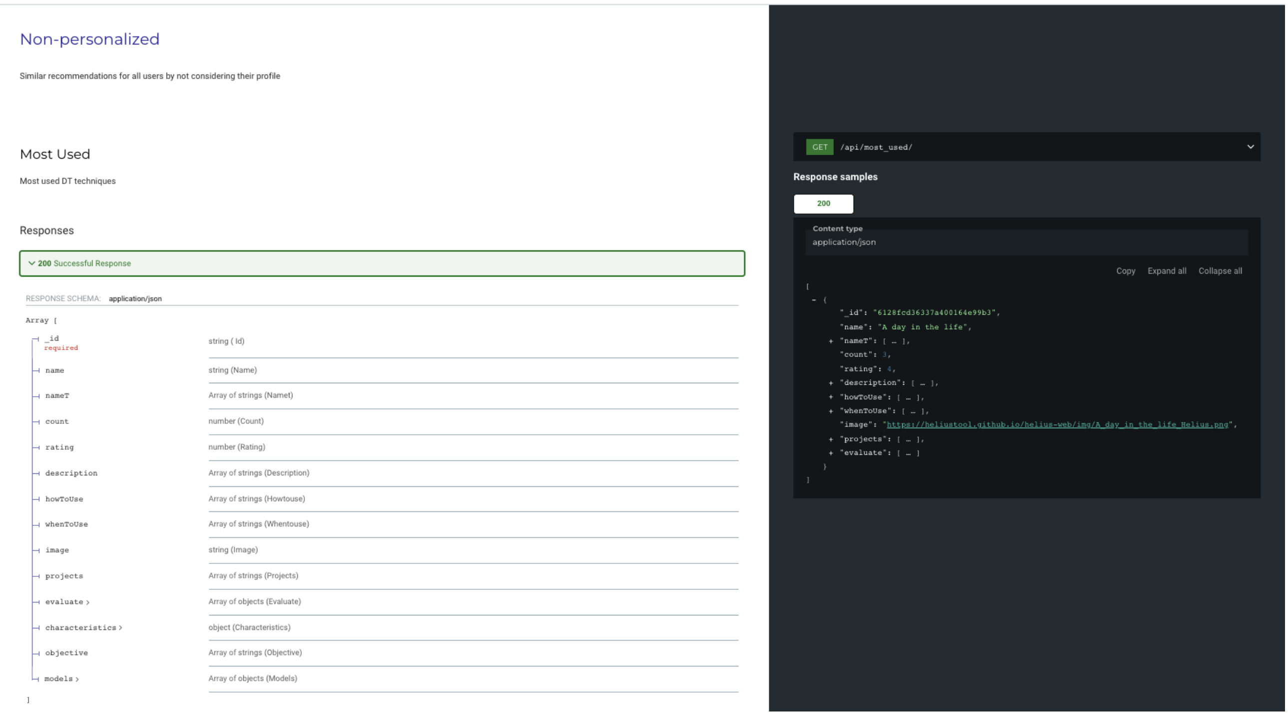


Figure 7.19: “Most used” recommendation mechanism implementation in Helius

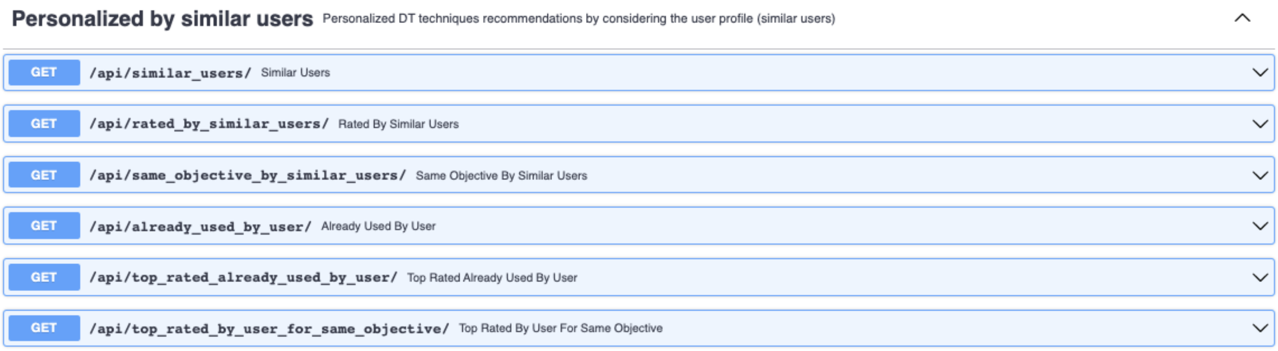


Figure 7.20: Overview of the personalized recommendation mechanisms

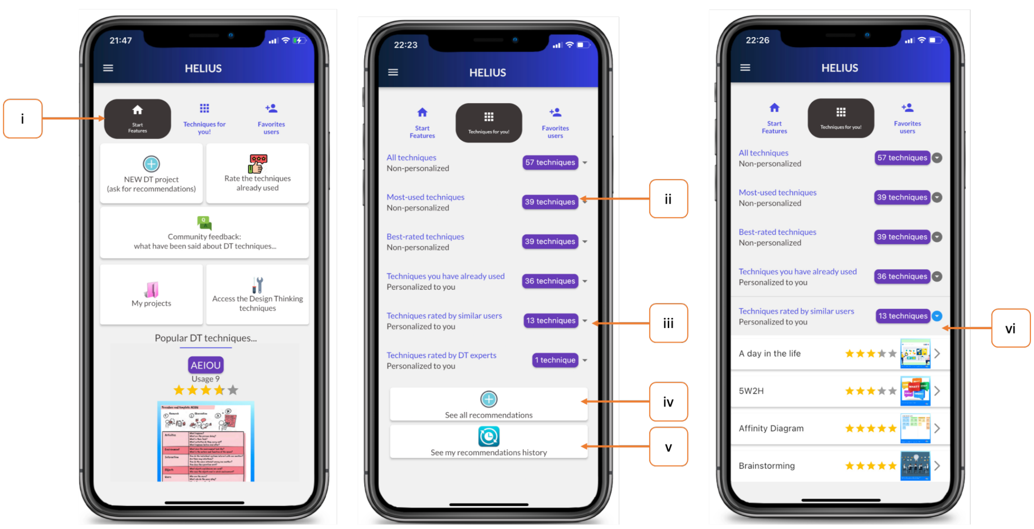


Figure 7.21: User journey in Helius for accessing the recommendations summary

the user has received in Helius. Item (vi) shows the list of techniques and their associated ratings attributed by the users.

Figure 7.22 shows the DT techniques recommendation mechanisms, organized in Non-personalized (i) and Personalized (ii). Once the user selects one recommendation mechanism, Helius shows Figure 7.22-(iii). For instance, in Figure 7.22, the user has selected the mechanism "Most used techniques" (i) and 12 techniques were recommended (iii). In item (iv), the user might apply filters to the list of DT techniques that Helius recommended. The filters include the suggested working spaces, among others.

B) Requiring DT techniques when creating a project

Figure 7.23 represents the user journey to require DT techniques recommendation in Helius. This user journey considers feature F1 – DT techniques recommendations (see Table 6.1). It allows the user to ask Helius for DT techniques recommendations and to add the recommended techniques to a project. In Helius' main screen (a), the user must click on Create a DT techniques portfolio (i). Then, Helius shows the "New Project" screen (b). Among other fields, after the user has indicated the project domain in the project creation screen (ii), the project context (iii), and added DT techniques to the project on her own (iv) if she clicks on the "Recommend techniques to me" button, Helius shows the screen of DT recommendation techniques (Figure 7.23-(c))[14]. In Figure 7.23-(c), since the user has indicated the domain, context and added techniques on her own, Helius shows 4 recommendation mechanisms (from vii to x). These mechanisms refer to the mechanisms RM_1 to RM_4 that we introduced in Section 7.2.2, respectively.

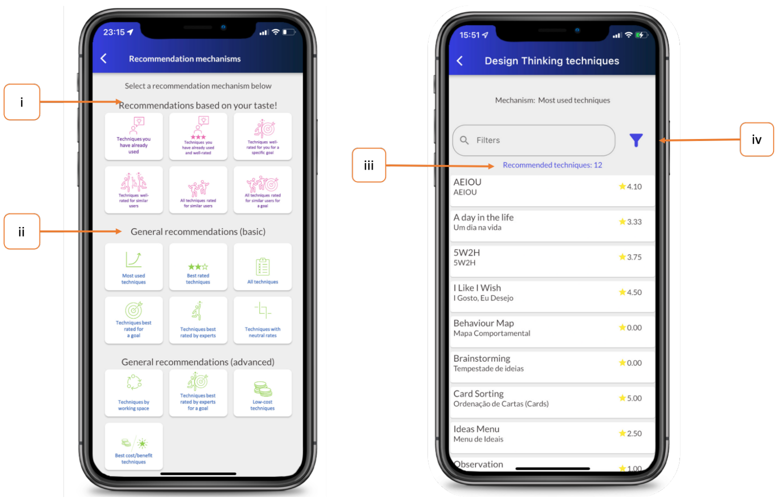


Figure 7.22: All recommendation mechanisms provided in Helius

[14]These fields are required to create a project in Helius. However, since the recommendations aim to support the professionals in discovering DT techniques using different recommendation mechanisms, the user is able to require DT techniques recommendations before concluding the project's creation. In addition, Helius just show the recommendation mechanisms according to the fields the user has selected. For instance, if the user has not indicated the project context and asks for a recommendation, Helius will not show the mechanisms of DT techniques recommendations related to the context.

Figure 7.23-(d) shows an example of DT techniques recommendation considering the user has clicked on mechanism "Techniques used combined to a technique" (item ix) in Figure Figure 7.23-(c). The screen is composed by the following items:

xi - The technique's name in which techniques used in combination were found in Helius;

xii - The list of techniques that were combined with the analyzed technique. For instance, Ideas Menu, Insight Cards and so on were already combined to A Day in the life in projects. Helius sorts the list of techniques by amount of combined usage;

xiii - Represents a button where the user might use the list of techniques that were used in combination with a specific technique;

xiv - Shows the number of combination a technique was used with other technique. For instance, in projects where users have used A day in the life, Ideas menu was also used 6 times;

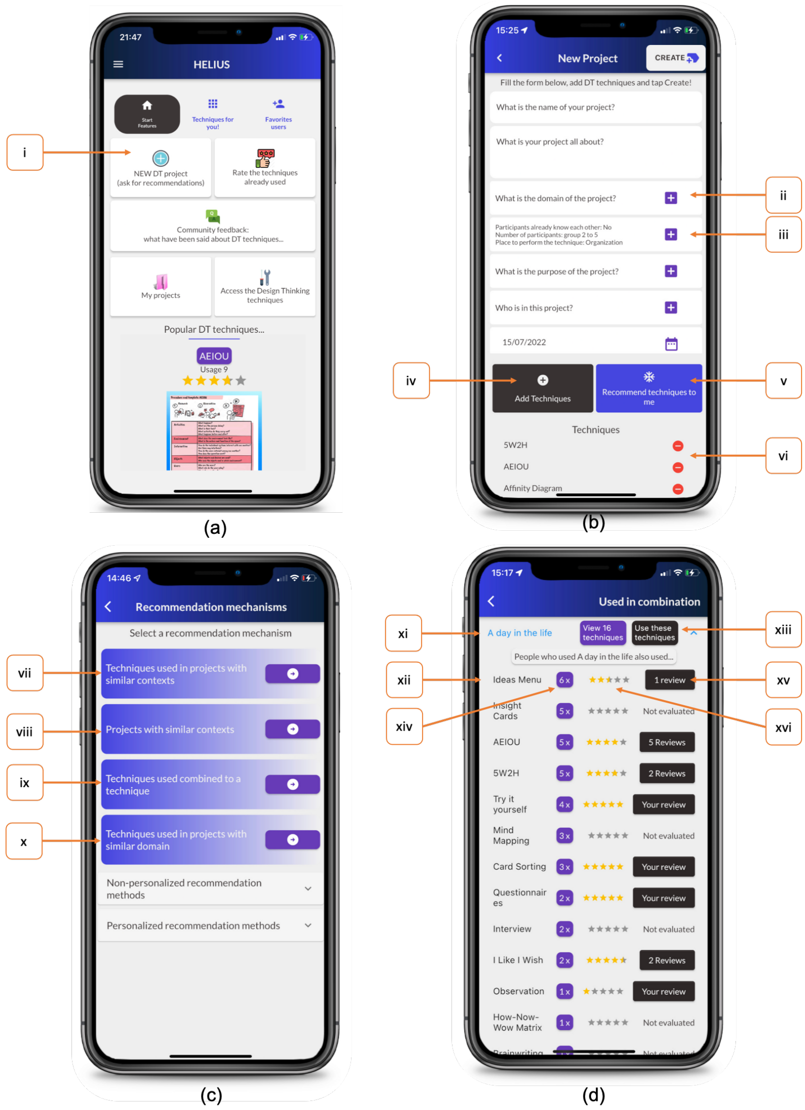


Figure 7.23: User journey for asking for DT techniques recommendation in Helius

xv - Represents a button where the user might visualize the reviews the users have attributed to the technique;

xvi - Represents the mean rating the techniques has received by Helius users.

Next we detail an empirical study that we conducted to evaluate Helius, its recommendations and its contributions to the decision-making involved on the selection of DT techniques for software development.

7.3.2 Empirical study for evaluating Helius

This section presents an empirical study that we conducted to validate Helius. Our goal is to evaluate if Helius supports DT practitioners to select DT techniques to use DT in software development projects.

In summary, we initially introduced Helius to DT practitioners and asked them to use the recommendation system. So, we collected data from the participants to evaluate how Helius worked on the DT techniques recommendation and decision-making of which techniques to use.

This study follows the guidelines for empirical studies proposed by Wohlin *et al.* (2012) [271] and Travassos (2002) [257]. Figure 7.24 shows the study's steps, including (1) Scoping, (2) Planning, (3) Execution, (4) Data analysis, and Results presentation.

Scoping

To define the scope of our study, we characterized the study's context, goal, research questions, and measures using the Goal-Question-Measure (GQM) template [19]. Table 7.11 shows the GQM template.

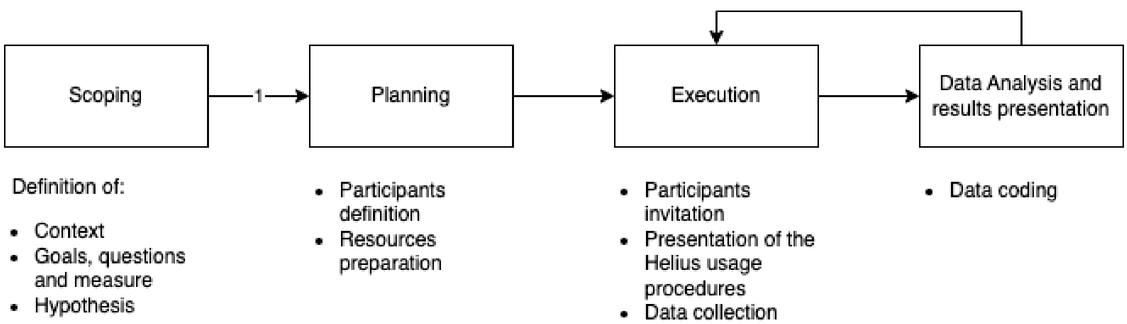


Figure 7.24: Activities of the empirical evaluation of Helius

Table 7.11: GQM template for evaluating Helius

Item	Description
To analyze	the use of Helius, the DT techniques recommendation system
with the purpose of	supporting the decision-making for selecting DT techniques,
with respect to	select the DT techniques through the collaboration of DT practitioners
under the perspective of	DT practitioners using DT in software development
in the context of	software development

Study's context

- *Participants and scenario* - DT practitioners who conduct DT sessions, workshops, or projects, or use DT techniques in a software development context;
- *Goal* - Investigating how Helius supports the decision-making of DT practitioners on the selection of DT techniques;
- *Research Questions*:
  - RQ1 – How do the DT practitioners perceive the DT techniques recommendations provided by Helius?
  - RQ2 – How has Helius impacted the decision-making behind the selection of DT techniques?
- *Measure*
  - Ability to use Helius:
    - Ability to create a portfolio of DT techniques;
    - Ability to use DT techniques recommended by Helius in real projects;
    - Ability to collect DT techniques experiences through the use of the DT techniques community of practice;
  - Ability to make decisions for selecting DT techniques based on recommendations provided by Helius.

We also used the System Usability Scale (SUS) method [34] to assess the usability scale of Helius (described next).

Planning

As a second activity, we prepared the materials and resources to conduct the study. In addition, we defined the participants' selection strategy. We outlined the study's procedures, including the presentation of the Ethics Committee Approval document, the Participant's Consent Form and the set of steps to be executed by each participant in order to participate in the study properly.

Participants' definition

For defining the participants, we followed the framework proposed by Rainer and Wohlin (2022) [220], referring to the selection of credible participants for studies in the Software Engineering field. The authors introduce the concept of Key Informants as *"the principal of a key informant is that the informant can provide more reliable, valid and relevant information than a sample of participants. In other words, a key informant can be a more credible participant than a sample"*.

Table 7.12 shows the 5 characteristics that define a participant as a Key Informant, including Role in the community, knowledge, willingness, communicability, and impartiality. Table 7.13 presents a classification of practitioners, including Performers, Observers and Advisors. Rainer and Wohlin (2022) [220] argue that the difference between these roles reflects the different degrees of credibility of the information collected.

In our study, we consider as Key informants those professionals who conduct DT sessions in software development or aim to use DT techniques in their activities for software development or who use DT techniques in their software development activities.

Resources preparation

- Ethics Committee Approval and Consent form submission: before conducting our study, we submitted it to the Ethical Committee board. As part of the Ethics Commit-

Table 7.12: Characteristics of Key Informants

#	Description
C1	Role in community. Their professional role in their peer community should expose them to the kind of information being sought by the researcher.
C2	Knowledge. In addition to having access to the information desired, the informant should have absorbed the information meaningfully.
C3	Willingness. The informant should be willing to communicate their knowledge to the interviewer and to cooperate as fully as possible.
C4	Communicability. They should be able to communicate their knowledge in a manner that is intelligible to the interviewer.
C5	Impartiality. The key informant should be objective and unbiased. Any relevant biases should be known by the interviewer, e.g., the key informant declares a bias or the interviewer can determine this from other sources.

Table 7.13: Classification of Key Informants

Classification	Description
Performer	who is experiencing the situation (e.g., a DT practitioner conducting DT sessions in software development scenario)
Observer	is an observer but not a performer and is located in the situation (e.g., a programmer who observes the DT practitioner making decisions of the DT techniques for conducting a DT session)
Advisor	has experience in a wide range of situations. Consultant, for example, may have experience but is not in the situation (not an observer) and is not the one who executes the situation (performer)

tee analysis, we generated a Consent Form to guarantee the voluntary participation and the correct data collection and manipulation[15].

- Instructions for using Helius: to support the participants in using Helius and subside their activities in the study, we generated a set of instructions (tutorial). The tutorial gives directions on how to use Helius for supporting the selection of DT techniques[16].

Execution

In the third activity, we present details about the execution of our empirical study. We show how we invited and selected the participants (Key Informants), a Helius presentation and the procedures for collecting and analyzing data. We conducted this activity in 2 steps (see Figure 7.25).

Step 1 – Interview-based study

In the first step of our empirical study, we evaluated Helius using interviews as the data collection instrument. Our goal was to initially present Helius to DT practitioners, stimulating them to use the DT techniques recommendation tool in their activities, asking for recommendations and collaborating with the DT techniques community of practices in Helius. Then, after a period of time when they could use Helius, we interviewed them back to collect data about their experiences in using our recommendation system.

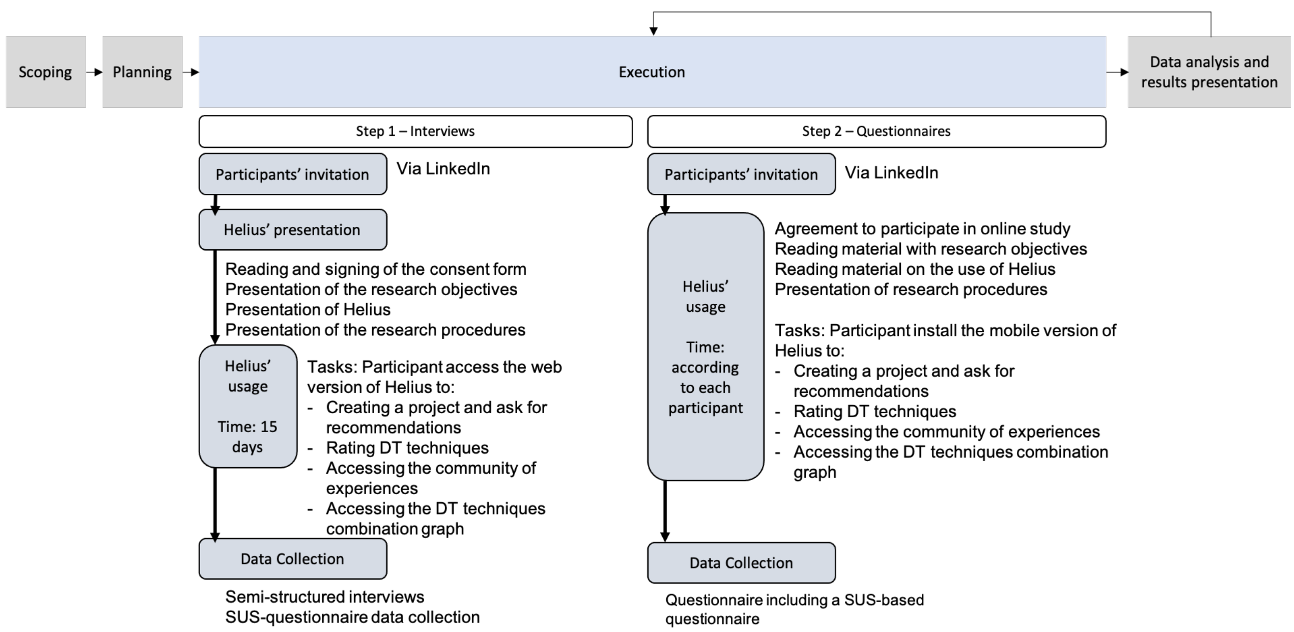


Figure 7.25: Steps for executing the evaluation of Helius

[15]Both documents are available in the following repository https://github.com/rafaelparizi/PHD_repository
[16]The tutorial is available in our research repository and can be accessed at https://github.com/rafaelparizi/PHD_repository

- *Participants' invitation:* In Step 1, we invited Design Thinking practitioners from different software companies to use Helius as a tool for selecting DT techniques for moderating DT workshops in projects in the context of software development. We sent invitations based on the LinkedIn platform. The selection of these participants was not randomly since we consider these participants as key informants, classified as Performers in the context of this study according to the participants' selection proposed by Rainer *et al.* (2022)[220].

  We invited 8 participants to participate in Step 1. Four participants agreed to participate. Figure 7.26 shows the participants of Step 1 and their profiling. After each interview, we applied some suggestions provided by the participants (indicated by the gear icon in Figure 7.26.

- *Helius' presentation:* After each participant has accepted to participate in our evaluation study, we scheduled a kick-off meeting to introduce Helius and our study. We used the Zoom platform to host the kick-off meetings, which took 25 minutes long in average. We conducted the kick-off meetings as follows:

  1. In each meeting, we presented the Consent Form, previously submitted and approved by the Ethical Committee. We showed the participants the study's ethical aspects, allowing them to withdraw if they wanted to.
  2. Once the participant had agreed to the consent form and signed it up, we quickly presented our research goals and the methodology to provide the participant with an overview of the study that she would participate.
  3. Next, we presented Helius. We showed up the Helius features and user flows to perform the tasks that we wanted to evaluate.
  4. Finally, before concluding the kick-off meeting, we presented the study's procedures, including the tasks the participant should perform when using Helius. We also scheduled a follow-up meeting to collect data in terms of the experience of using Helius for deciding which DT techniques to use.

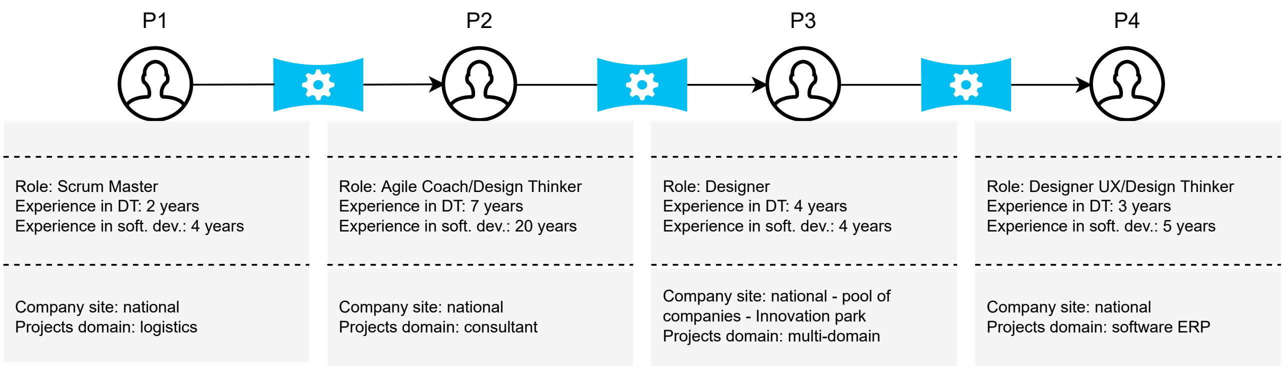


Figure 7.26: Empirical study – Participants of the Step 1

Appendix E presents the detailed procedures that we prepared for the participants using Helius during the study (in Portuguese). We also prepared a 24 minutes long video presentation about how to use Helius[17].

In summary, each participant should:

(a) Access the Web version of Helius;
(b) Create a user account;
(c) Create a project including DT techniques;
(d) Add DT techniques to a project by asking for the DT techniques; recommendations through the recommendation mechanisms provided by Helius;
(e) Rate some of the DT techniques that she used in her project;
(f) Consult the DT techniques community of experiences to see what other professionals were saying about the DT techniques;
(g) Access the DT techniques combination graph to understand what DT techniques were used at that time.

- *Helius' usage:* After participating in the kick-off meeting and indicating how much time it would be necessary to use Helius, each participant could use the recommendation system to execute the tasks that we have proposed to them. On average, the participants required 15 days to use Helius in their working space and come back with relevant information for our empirical study.
- *Data collection:* During the follow-up meetings, we collected data about the use of Helius. The meetings occurred from October to November 2022. We prepared in advance a script for conducting semi-structured interviews[18]. It is worth mentioning that the Data Analysis activity (we show next in this section) was started just after we finish each interview and its results served as a source to implement improvements in Helius as they were pointed out by the participants.

After we finished collecting data from 4 Key Informants from the software industry about the use of Helius, we decided to conduct the second step in our empirical study for evaluating Helius. Next, we show the details of how we conduct Step 2.

Step 2 – Questionnaire-based study

In step 2, we submitted and published an improved version of Helius on Google Play[19] and Apple Store[20]. Therefore, by publishing Helius in mobile application stores we could easily invite other professionals to participate in our study.

[17]The video presentation is available at https://www.youtube.com/watch?v=oAd5rqcrsHs&t=925s and https://youtu.be/M6Ziwx3uTwQ
[18]The script of the interviews for evaluating Helius might be found at https://github.com/rafaelparizi/PHD_repository/wiki#step-1--interview-based-study
[19]http://www.play.google.com
[20]https://www.apple.com/app-store/

- *Participants' invitation:* We sent invitations to DT practitioners who have the characteristics of a key informant for our study's context. We used LinkedIn social media for identifying the participants. Then, we sent them an e-mail introducing our research, Helius itself and the procedures for participating in the study. We filtered professionals who indicate to be Design Thinker/DT facilitators and work in software companies.

- *Helius' usage:* In step 2, the participants should perform the following tasks:

  1. Read a document presenting Helius' features and our research goals;
  2. Read the tasks to be done when using Helius;
  3. Read the participants' agreement document to participate in the study;
  4. Download and install Helius from one of the mobile application stores (Google Play or Apple Store).
  5. Create a user account;
  6. Create a project including DT techniques;
  7. Add DT techniques to a project by asking for the DT techniques; recommendations through the recommendation mechanisms provided by Helius;
  8. Rate some of the DT techniques that she used in her project;
  9. Consult the DT techniques community of experiences to see what other professionals were saying about the DT techniques

- *Data collection:* In this step we collected data using an online questionnaire. After using Helius, the participants should answer questions regarding the experience they had by using the DT techniques recommendation system. The questionnaire contained the same questions that we prepared as our interview script in Step 1[21].

Next, we present the data analysis and the results for both steps 1 and 2 that we conducted in this empirical study.

Data Analysis and Results Presentation

Step 1 - Interviews

After concluding each interview, we transcribed the audio into text. We used Atlas.ti to conduct the data analysis, proceeding with open coding analysis [247]. Table 7.14 shows the participants and their use of Helius during the empirical study, while Figure 7.27 shows an example of a project created in Helius by P2. For instance, P2 used the techniques Observation and Stakeholder Map.

[21] The questionnaire is available at https://encurtador.com.br/muJW1

Table 7.14: Helius usage by the study's participants

	Features			
P#	**Project created**	**Project's domain**	**Techniques' rating**	**Users' experiences (Community of practice)**
P1	1	Hotel management system	9 (Affinity diagram, Brainstorming, Card Sorting, Empathy map, Feedback Grid, Insight Cards, Interview, Personas, Paper prototyping)	yes
P2	1	ERP to control devices in hospitals and schools	3 (A day in the life, Ideas Menu, Insight Cards)	yes
P3	1	Mobile Application for testing Helius	5 (Interview, Affinity Diagram, Try it yourself, Brainstorming, Empathy map)	yes
P4	2	Simulating system Testing project	8 (Observation, Stakeholder Map, 5w2H, Exploratory research, Card Sorting, Empathy map, A day in the life, I like I wish) 5 (Personas, Interview, Empathy Map, Insight Cards, Feedback Grid)	yes

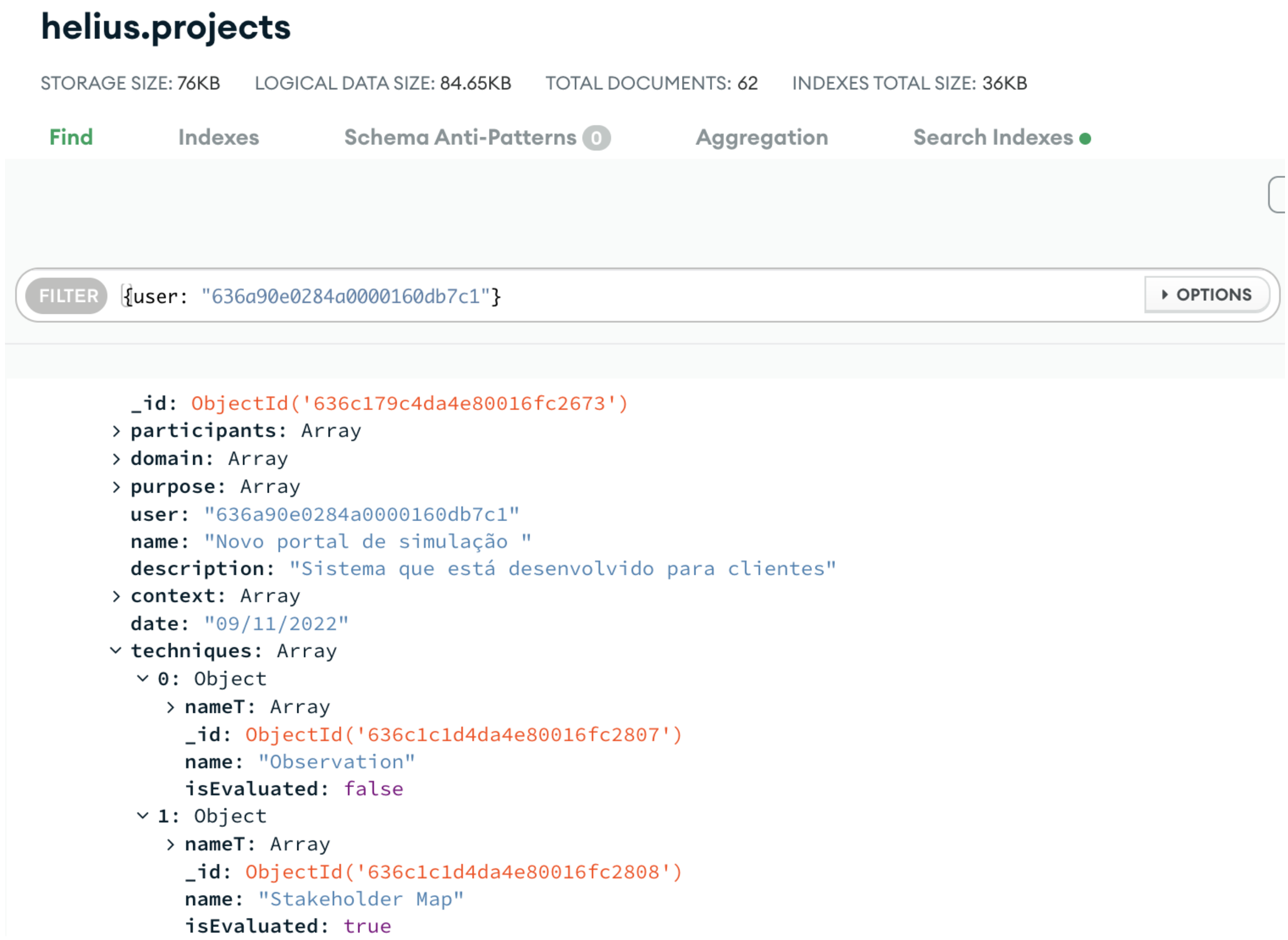


Figure 7.27: Example of project creation in Helius

Next, we show the results of our empirical study grouped according to the 2 main research questions that we defined using the GQM template.

RQ1- How the DT practitioners perceived the recommendations provided by Helius?

We initially asked the participants to indicate in a 5-item Likert scale [7] how relevant they considered the recommendations that Helius provided to them (Table 7.15). The scale varies from Not at all important (1) to Important (5).

P1 indicated that Helius recommended techniques differently from those she had already used, making her to considering the recommendations as partially relevant. She mentioned that she expected to receive recommendations for techniques that she has already used before. She said: –"*I created a project based on the experience that I have already had. So I expected Helius to recommend the [techniques] that I had already used and it did not recommend the ones that I we had used... so I found it relevant because I got to know some that I didn't know, because we used a scope x about 10 techniques at most, so as he has many more techniques it increases the range of possibilities and even of knowledge, so I found it quite relevant.*" - P1;

P2 argued that –"*I think the tool is very useful, the idea is extremely good, it will help a lot from beginners to advanced and the Feature Ideas are also very good...your system is simple, the idea is excellent, [the features] are great.*" - P2;

P2 complemented that –"*if I had a very difficult project with ideation to extract from people their needs and create the solution... we already had the idea of the solution there and we needed to get solutions and understand all the ideas for the solution. If I had done these 10 ideation activities and had your application and there was someone recommending a tool that I have not used yet or that suddenly the person would say oh I recommend... ah I used it, but I used it differently... I used it this way and it worked! Yes, the recommendations are super important.*" - P2;

P3 mentioned that the recommendations were relevant. She indicated that she selected recommendations for a specific goal: –"*I selected them to know the user... to know who the user is, it is not some specific term.*" - P3;

P3 also argued that Helius can be used both for companies and for professionals who are starting out, such as students: –"*I found that it is very focused on just consult-*

Table 7.15: Participant's perceptions of Helius' recommendations

Participants	Likert-item	Recommendations' importance
P1	1 2 3 **4** 5	Somewhat important
P2	1 2 3 4 **5**	Important
P3	1 2 3 4 **5**	Important
P4	1 2 3 **4** 5	Somewhat important

1 2 3 4 5 - Not at all important
1 **2** 3 4 5 - Somewhat unimportant
1 2 **3** 4 5 - Neutral
1 2 3 **4** 5 - Somewhat important
1 2 3 4 **5** - Important

ing a technique. For example, I am in a certain phase and it is very good, not only for professionals I can imagine even in universities, for example, where you need to develop a project and you don't know how to direct yourself... you don't know which direction to take... and then you open Helius, see the tools, see in which phase they are used, and this makes it much easier to develop... even in small projects or large projects." - P3;

P4 indicated that the recommendations were partially relevant since she considers that it depends on the project: –"*I think it depends on the project, but it is with I think that based on the objectives there that I put makes sense, but I don't know if I would use all the techniques.*" - P4;

We also asked the participants if they would use in new projects the techniques which Helius recommended to them. All the participants informed that they would use the techniques which Helius recommends.

P1 pointed out that although she might not know the recommended techniques, Helius is able to support her in gathering information about how to use the techniques: –"*I believe so, but I would have to learn these techniques in this case, but as it says there [in Helius], it already explains, I would try to use the techniques...*" - P1;

Then we made a follow-up question to P1 based on the answer about the use of the recommended techniques: –"*Would Helius be able to help you learn about these new techniques?*" - Researcher; She answered –"*yes, yes... as much as I don't know it, it would help me to use different things, right, to test different things in projects.*" - P1;

P3 mentioned that she used the techniques that Helius recommended to her: –"*I used what it [Helius] recommended. I used Persona, Scenarios and Empathy Mapping.*" - P3; She also mentioned that the information provided by Helius for each technique helped her to find the details to use the techniques. –"*There were some techniques that I think that just by reading what I did I didn't understand very well that's why I didn't use all the techniques you recommended, but then I saw that there were references and then I opened the references I saw that I didn't stop for others but I saw that there was an explanation. I think the references helped to see I even think that the application is not the same, some gave an example anyway and it helped a lot.*" - P3;

P4 suggested to Helius to allow the user to access the information of the technique as a technique is recommended. –"*I have a suggestion because why does it recommend techniques [...] when I go to select I can't see the description of the technique.. so I think that's the description that I would like to have what I'm going to need to do that technique... I'm going to need paper. I'm going to need to be in a room with people, so even to help with that choice, I think it would be interesting to show at that moment.*" - P4; After interviewing P4, we implemented her suggestions in Helius. We included a button where the user might open the techniques' details, as Helius recommended the technique.

Next, we asked the participants which techniques recommended by Helius they already knew to evaluate if Helius supported them in knowing new Design Thinking techniques. P1, P2 and P3 considered that Helius supported them in knowing new DT techniques. P1 indicated that –"*Many [techniques] I didn't know... some I didn't know at all... I didn't even know them by name. I didn't read all [techniques] but reading a little bit of some that I opened, I found it interesting how they work. P1 also indicated that Helius aided in increasing her DT techniques portfolio.*" - P1;

P3 also suggested that when accessing a DT technique, Helius could show the projects that the techniques were used, to help to understand how to use the technique. –"*I think maybe if when you make recommendations for techniques you show the projects that I have used if some of these techniques would make it easier for me to understand how it was used, you know... in which project, what type of project it is used that has some I'm like: I think this AEIOU that I don't understand or was it "a day in the life" it was one of those 2 that I looked at I didn't understand how I could use it, you know only after I I went to look at these references and then I don't know, I don't know if when you go to research a technique to make a technique recommendation, show if it's people who use techniques from that recommendation.*" - P3;

P4 mentioned that she did not know all the techniques that Helius recommended to her. She reinforced that Helius could provide more information about a technique when it is recommended: –"*I think I didn't know all of them because some have names that were not familiar to me. There were some techniques I already knew. Maybe in the description, I could have seen there that it was actually another technique.*" - P4;

On the other hand, P2 mentioned that she already knew all the techniques recommended by Helius. She has more than 7 years of experience in using Design Thinking. She mentioned that her experience in using DT supports her to use the techniques, but also indicated that the techniques' evaluation provided in Helius is essential to see if some details might be added in how to use a DT technique: –"*when you realize it is a very used tool one of the most used tools you go there and take a look and sometimes you've been doing it for so long that maybe sometimes you forget a little detail that can help in a next DT session that sometimes you say wow it's the true look that's just the general rule, I already did it differently from here and during the time you already changed it and then it's nice to have these evaluations and also the tools to be able to consult, right?*" - P4;

RQ2 - How has Helius impacted the decision-making behind selecting DT techniques?

We asked the participants if Helius supported them in making decisions about which DT techniques to use. All participants understood that Helius contributes to the decision-making of the selection of DT techniques.

For example, P1 mentioned that she knew few techniques and that Helius helped her to know new techniques and to make decisions: –"*Let me say that I only knew about*

20% of what was there, so there's a lot, It has a lot of interesting material. P1 also argued that: I believe so [it helped in making the decision] because Helius is based on recommendations, so sometimes we may be using things that are no longer used so much or that for my particular context other techniques are more appropriate than no... I don't know yet, but if applied, it might work better than the ones I already know, so yes [it supports my decision-making]!" - P1;

P2 argued that even though she has a large experience in DT, Helius would collaborate with her decision-making: –"*Yes, I think that in software development that is our focus here... yes, definitely [it would help decision-making]*" - P2;

P3 mentioned that Helius contributes to overcoming the barrier of using techniques with professionals unaware of how to use them. This is a known challenge in the literature, as we reported in [195]. P3 indicated: –"*Definitely using... it's researching techniques on Helius and then finding one that I think fits in a certain phase that I'm in a project and... it's applying, right, applying by calling people; I'll explain in more detail details how it works and how to apply it...*" - P3;

P3 added up the following about the use of DT techniques: –"*For example, in my experience, it is usually the designer who brings this type of tool, not some programmer. The designer who does this action, like setting up a meeting with everyone, explaining how he uses it and making people use it, you know? so that's what I did. They did not know any technique despite already knowing the term design thinking. I showed them and explained what these techniques are and that many of them can sometimes seem a little abstract, especially these techniques that are to get to know the user. It's not that you have to think about the scenario, but these are things that we use to sometimes know how to communicate with the user... So, if our user is a 20-year-old person, we can use a language younger is not.... all this because we identified it in the techniques... through the techniques, we identified that a person in this age group speaks that way, right? OP has a very good empathy map, I know: what he wants to see, what he [user] hears... this helps in the development of systems in these criteria, you know, to increase the user experience.... that's why in this more abstract view programmers sometimes have a barrier that you have to break down little by little, educating and talking and then when they see that it works, especially when it works, they start to use it.*" - P3;.

We then asked about which elements of the recommendation system helped the participants in their decision-making for selecting DT techniques.

P1 cited the following about the decision-making supported by Helius: –"*There is a sequence of parameters that you place according to what you want to achieve. For example, there is [create project], what is your team like, what is your objective, and also what do you hope to achieve with that project, if you want to know more about requirements and you want to know more about problems which are some that we tag there. So I believe that there influences a lot on the type of recommendation. So it's this functionality* *link where we have to say what we expect and the platform gives us some suggestions related to that.*" - P1;

P2 considers that it is important for the technique evaluation form to contain structured fields, which helps the user to fill in relevant information about the use of techniques: –"*The form is very good. No, for example, only you liked it, did you use it exactly as it is here, did you make any changes? If you made any modifications, what was it and what result did you have... because then those who read it correctly, then with more information, get richer, right? the system... begins to place a form that the person has to fill out.*" - P2;

P3 indicated as decision-making elements the projects' information of other users and the access to the information of the experiences of other users: –"*What I think that adds value to know the project that was used... it gave a bad experience... so look what it was the project that had this bad experience, you know? project of the user who used this tool... then I could maybe have a better idea of how it was used...*" - P3;

P4 pointed out the following about Helius' support for decision-making: –"*You might see the techniques he used [another user]... you will see which projects he created and the recommendations then... Helius allows you to do several things: create a project and ask for a recommendation of techniques it allows evaluating the techniques and thus feeding the recommendation algorithms. It allows you to search for a community of experiences; you don't want to see the comments of others and thus analyze the techniques to decide which ones to use.*" - P4;

Next, we present the results of Step 2 of our empirical study. We collected data from DT practitioners of Helius' usage through an online questionnaire.

Step 2 - Questionnaire

In this step of the empirical study, 12 professionals who use DT in their activities related to software development answered the questionnaire (P5 to P16). Table 7.16 shows the demographic data of the participants, including the participants' background, current roles, experience in years using DT and in software development, the company domain and the company site. To keep the participants' anonymity, we identify the participants using the letter P followed by an ordinary number sequentially to Step 1.

Regarding the recommendations of DT techniques provided by Helius, the participants answered the following:

RQ1- How the DT practitioners perceived the recommendations provided by Helius?

Table 7.17 shows the participants' perceptions about the recommendations provided by Helius. Half of them considered the recommendations partially relevant while

Table 7.16: Participant's demographic data (Questionnaire)

P#	Background	Current role	Experience in DT (Yrs)	Experience in Software Develop-ment (Yrs)	Company domain	Company site
P5	Computer Science	Lead Design Thinking / Cloud Support Specialist	4 - 5	4 - 5	Software house	Multinational
P6	Design	UX Designer/Design Thinker	2 - 3	4 - 5	Personal loan solution development	National
P7	Computer Science	Developer/Lecturer	2 - 3	4 - 5	Software for education	Multinational
P8	Design	UX Designer	0 - 1	0 - 1	Software house	National
P9	Management	Product Designer	2 - 3	2 - 3	Software house	National
P10	Management	Agile Manager	4 - 5	+7	Software house	Multinational
P11	Design	Head of Design	+7	4 - 5	Software house	National
P12	Computer Science	Software Engineer	2 - 3	4 - 5	Health systems	National
P13	Computer Science	Product Manager	+7	+7	Tech & innovation	Multinational
P14	Computer Science	Data scientist/UX analist	4 - 5	+7	Data Dashboards	National
P15	Design	UX designer	2 - 3	2 - 3	B2B solutions	National
P16	Design	Team Leader	2 - 3	4 - 5	Bank systems	Multinational

Table 7.17: Participant's perceptions of the Helius' recommendations

Participants	Likert-item	Recommendations' importance
P5	1 2 3 **4** 5	Somewhat important
P6	1 2 3 **4** 5	Somewhat important
P7	1 2 3 **4** 5	Somewhat important
P8	1 2 3 4 **5**	Important
P9	1 2 3 4 **5**	Important
P10	1 2 3 4 **5**	Important
P11	1 2 **3** 4 5	Neutral
P12	1 2 3 **4** 5	Somewhat important
P13	1 **2** 3 4 5	Somewhat unimportant
P14	1 2 3 **4** 5	Somewhat important
P15	1 2 3 **4** 5	Somewhat important
P16	1 2 3 **4** 5	Somewhat important

1 2 3 4 5 - Not at all important
1 **2** 3 4 5 - Somewhat unimportant
1 2 **3** 4 5 - Neutral
1 2 3 **4** 5 - Somewhat important
1 2 3 4 **5** - Important

the other half considered them relevant. For instance, P5 indicated that Helius is capable of recommending DT techniques:

–"*The system is capable of making good suggestions for techniques and can help less experienced professionals choose the most appropriate tools for each case.*" - P5;

P7 mentioned that the recommendations were useful to the context of her project and that she could learn more techniques after receiving recommendations of techniques by Helius: –"*I enjoyed receiving techniques targeted to my application context, and I enjoyed learning more about each of the techniques.*" - P7;

P10 confirmed that since there are a lot of DT techniques, having recommendations of which ones to use might support their activities: –"*We have a lot of techniques and a lot of diversity in the teams, products and it is always good to learn and know more about the applicability of the techniques in the scenarios we are in.*" - P10;

P9 mentioned the recommendations support making decisions: –"*Based on the recommendations we were able to make more assertive and faster decisions on what to apply at that moment in the project*" - P9;. P14 also mentioned that she would use the techniques suggested by Helius: –"*The techniques that Helius recommended to me when were techniques I didn't know all about. I would use the techniques.*" - P14;

P6 also considered the recommendations of techniques as relevant. However, she mentioned that she would like to be able to access more information about the recommended techniques: –"*The recommendations were useful for my project. But Helius could bring the technical information, so I know how to select.*" - P6;

The participants also answered if they already knew the DT techniques recommended by Helius. They mentioned that they had some experience in using some DT techniques but indicated that Helius supported them in learning new techniques.

P6 mentioned that although she knows some techniques, she could learn new techniques based on the information provided in Helius: –"*There are techniques that I knew under another name, but there is information about them.*" - P6;

P7, P8 and P10, P14, informed they learned new techniques using Helius: –"*I got to know several new techniques.*" - P7;

–"*I learned about some techniques that I had not studied before.*" - P8;

–"*It is recommending me new techniques.*" - P10;

P9 mentioned that the recommendations gave to her elements to have new ideas to apply the methods: –"*With the descriptions already gave me ideas to apply some methods with the team.*" - P9; P15 mentioned Helius supports the use of DT: –"*I understand that this application [Helius] shows the techniques that people are using, right! I thought that the techniques could collaborate with my activities using DT.*" - P15;

P16 mentioned that she would recommend Helius to her workmates to support them in learning more about DT techniques: –"*I found it very interesting mainly to direct new people in this universe. For example, I already have more excellent knowledge on the subject, but I thought of recommending it to the new Product Owners (POs) in my organization that would help them a lot in the processes and many of them do not have this broad knowledge of tools*" - P16;

RQ2 - How has Helius impacted the decision-making behind selecting DT techniques?

Table 7.18 presents the participants' perceptions in terms of the support of Helius to the decision-making for selecting the DT techniques. We collected data considering the

4 features that Helius provides in terms of supporting the selection of DT techniques to use in software development, as we posed in Table 6.1 - Helius' features.

Excepting for P10, the other participants (from P5 to P9) indicated that Helius' features provided support to the selection of DT techniques. Looking at the demographic data about the participants (Table 7.16), we figured out that P10 informed us to have higher experience in software development (+7 years) and in Design Thinking (4 – 5 years). Therefore, based on her experience, she might have considered the Helius features with low importance. Literature reports that professionals with higher experience in DT already have their own Design Thinking toolbox [195].

Figure 7.28 shows the results of the participants' perceptions about the decision-making elements provided by Helius for selecting DT techniques clustered by each element (Techniques information, Techniques recommendation, Community of practice, Recommendation graph). Thus, it is possible to observe that the participants mostly considered the features as important or very important, validating Helius as a resource capable of supporting the selection of DT techniques in software development.

Next, we delve into the recommendation mechanisms provided by Helius. We asked the participants to indicate which mechanisms they have accessed and how frequently they would use each recommendation mechanism in their projects. It is important to mention that the answers represent that the participant did not necessarily use the recommendation mechanism, but she intended to use it in a new project.

Table 7.18: Helius' recommendations importance to the decision-making

Q#	Techniques' information	Techniques' recommendation	Community of practices	Recommendation' graph
P5	5	5	5	5
P6	4	4	4	3
P7	5	5	4	5
P8	4	5	5	4
P9	5	4	5	4
P10	2	2	2	2
P11	4	4	5	3
P12	5	3	3	4
P13	2	4	5	4
P14	4	4	4	2
P15	5	4	4	3
P16	5	5	5	3

1 - Not at all important
2 - Low important
3 - Neutral
4 - Important
5 - Very important

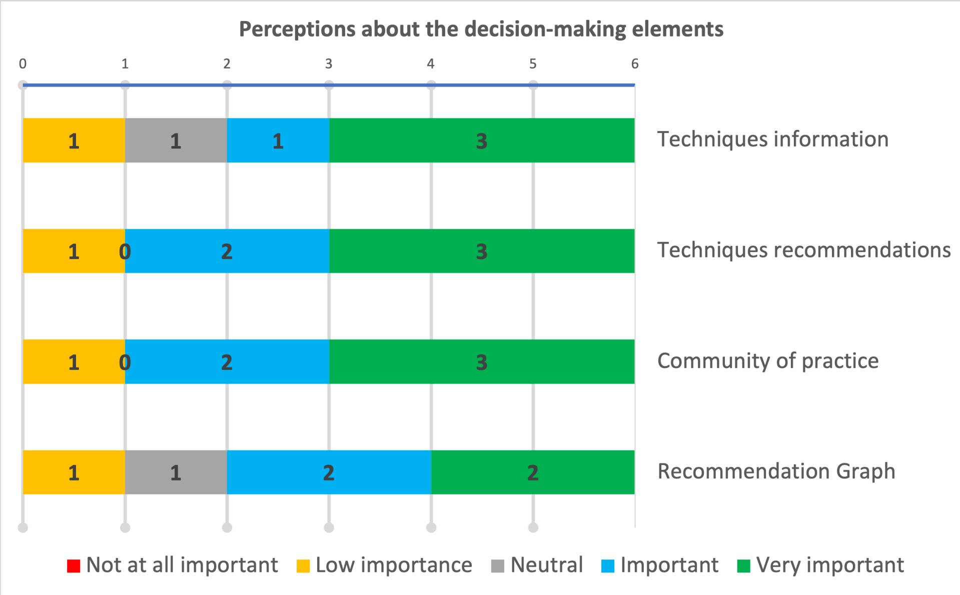


Figure 7.28: Participants' perception of the decision-making elements

Figure 7.29 clusters the data into each frequency scale of use (Never, Rarely, Occasionally, Frequently, Very Frequently). The results show that most of the users have the intention of using the recommendation mechanisms provided by Helius to select DT techniques. Table E.2 in Appendix E shows the raw data of the recommendation mechanisms indicated by the study participants.

The results also show that the recommendations mechanisms "Most used DT techniques" and "Best rated DT techniques" were highlighted as the most frequently used. Both mechanisms belong to the set of Non-personalized recommendation mechanisms, which recommend the same techniques for all users but indicate a tendency of what DT techniques are being used and best evaluated.

7.3.3 System Usability Scale of Helius

Martins *et al.* (2015) [166] argue that the SUS method has been largely used once it provides a measure of customer satisfaction. To calculate the System Usability Scale of Helius, we compiled the answers from all of the participants (from Step 1 to Step 2) to a SUS questionnaire. SUS is composed of 10 usability statements scored on a 5-point Likert scale of the strength of agreement. Its final score range from 0 to 100, where higher scores indicate better usability levels.

Table 7.19 shows the statements of the SUS method. To calculate the SUS scale, Brooke (1996) [34] suggests the following equation (Equation 7.1):

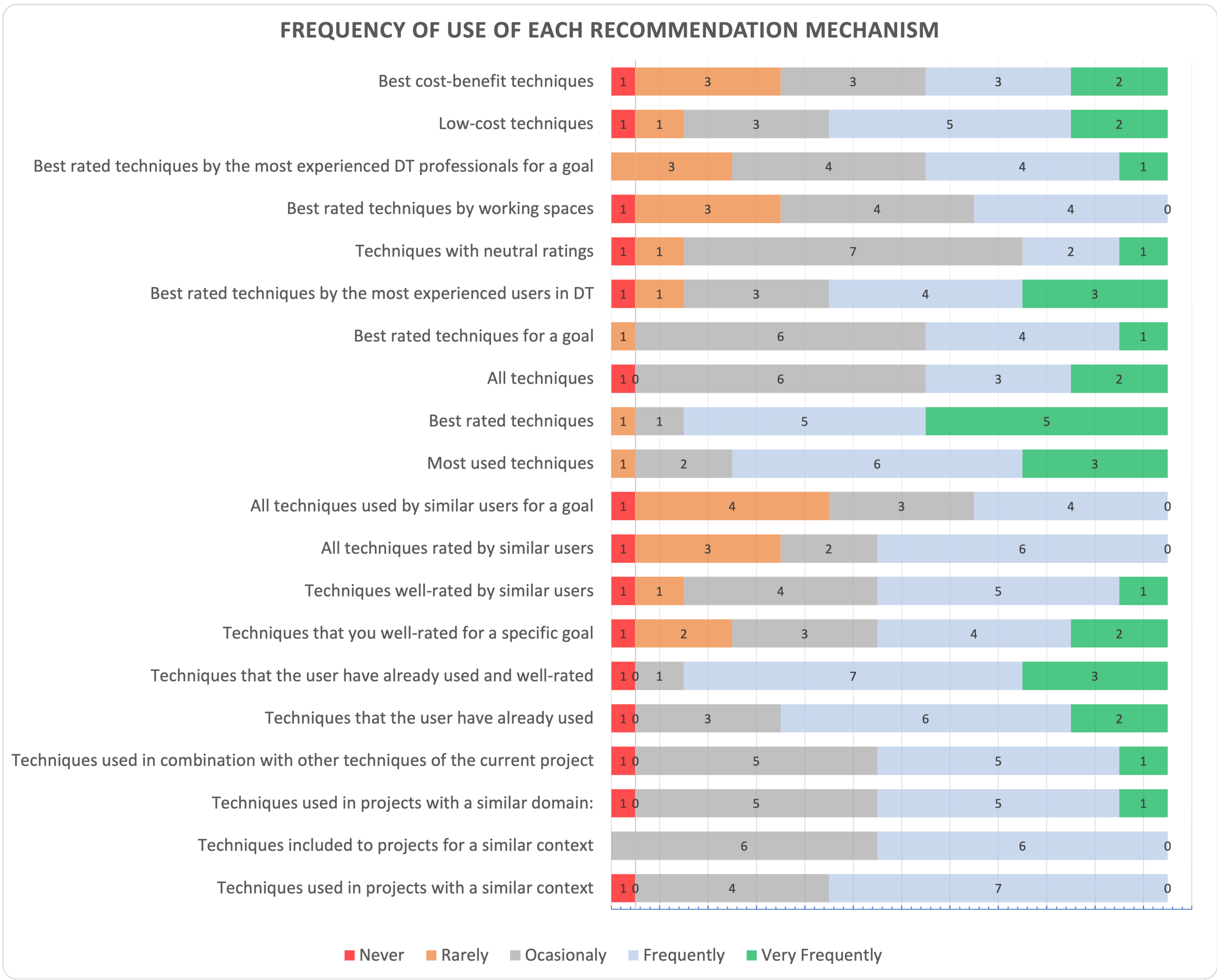


Figure 7.29: Participants' frequency of use of each recommendation mechanism

Table 7.19: SUS statements to evaluate Helius' Usability

Item	Statement
q1	I think that I would like to use this system frequently
q2	I found the system unnecessarily complex
q3	I thought the system was easy to use
q4	I think that I would need the support of a technical person to be able to use this system
q5	I found the various functions in this system were well integrated
q6	I thought there was too much inconsistency in this system
q7	I would imagine that most people would learn to use this system very quickly
q8	I found the system very cumbersome/awkward to use
q9	I felt very confident using the system
q10	I needed to learn a lot of things before I could get going with this system

$$
\begin{aligned}
X &= \text{Sum of the points for all odd-numbered questions} - 5 \\
Y &= 25 - \text{Sum of the points for all even-numbered questions} \\
\text{SUS SCORE} &= (X + Y) * 2.5
\end{aligned}
\tag{7.1}
$$

Table 7.20 shows the SUS scores and grades indicating the results obtained when applying the method.

Table 7.20: SUS Score and grades [34]

SUS Score	Grade	Adjective rating
$>$ 80	A	Excellent
68 – 80	B	Good
68	C	Okay
51 – 60	D	Poor
$<$ 51	E	Awful

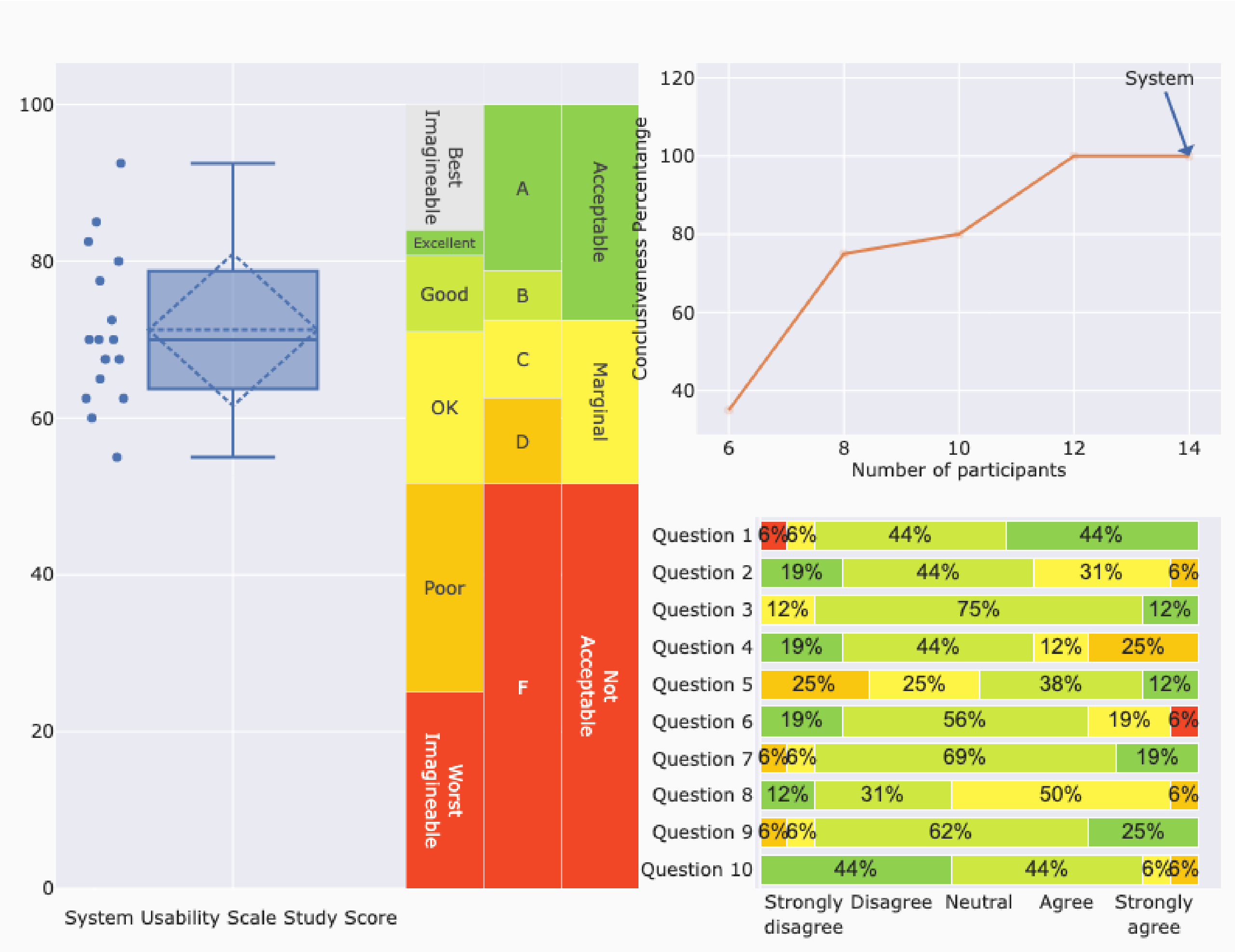


Figure 7.30: SUS score of Helius calculated using the tool [23]

Blattgerste *et al.* (2022) [24] proposed a web-based analysis toolkit for the System Usability Scale. The tool [23] helps to calculate the SUS score, the median value, the distribution of answers, etc. Figure 7.30 shows the SUS scale for Helius. The overall SUS score was 71.75, which is considered “Good”. In addition, the results show that our study reached 100% of the conclusiveness percentage by collecting data from more than 14 participants (top-right graph).

Blattgerste *et al.* (2022) [24] argues that the conclusiveness graph “*determines which sample sizes are needed for different usability questionnaires to be conclusive, [...] arguing that with a sample size of 8, a SUS study is already 75%, with 10 80% and with 12 or more 100% conclusive.*”

However, although the users had a good experience using Helius, they also pointed out improvements to the system's usability. Examples of suggestions to improve Helius indicated by the participants include:

- –"*Helius could improve the user interface and interaction. I suggest improving the interface flows.*" - P6;
- –"*Make available the techniques and the methods in PDF format so that we can download them and ask to confirm the registration via email.*" - P9;
- –"*Provide different user levels, from beginner to intermediate and advanced, so that the tool could already have some different ways of showing the selection of techniques.*" - P6;

Finally, looking back at the measurement variables that we defined using the GQM model, we assume that the users were able to achieve what we defined as goals when using Helius (Table 7.21). Thus, as a final result, we assume that Helius supports professionals in selecting DT techniques for use in software development.

7.4 Research Solution Outcomes: Framing the Solution's Constructs

In this chapter, we presented and discussed the activities that we conducted to propose and validate a Design Thinking techniques recommendation tool. As we posed in Chapter 3, our main goal is to support IT professionals who wish to decide and select DT techniques for use in software development activities. We conducted studies followed by activities based on the DSR research methodology.

Figure 7.31 shows our DSR-based methodology, highlighting that we attended to the Rigor, Relevance and Novelty cycles, as well as the Technological Rule proposed. After understanding and defining a research problem (Problem understanding approach), we performed 3 iterations in the DSR framework to propose a solution (Design approach) and to conduct empirical evaluations (Validation approach).

Throughout this Doctoral research, we could contribute not only to the practice by developing Helius as a technological artifact but also to theory by modeling the decision-making of professionals who conduct DT sessions, as well as by characterizing both the

Table 7.21: Analysis of the Measure variables from the GQM model

Measure variable	Evaluation
Ability to create a portfolio of DT techniques	Ok
Ability to use DT techniques recommended by Helius in real projects	Ok
Ability to collect DT techniques experiences through the use of the DT techniques community of practice	Ok
Ability to make decisions for selecting DT techniques based on recommendations provided by Helius	Ok

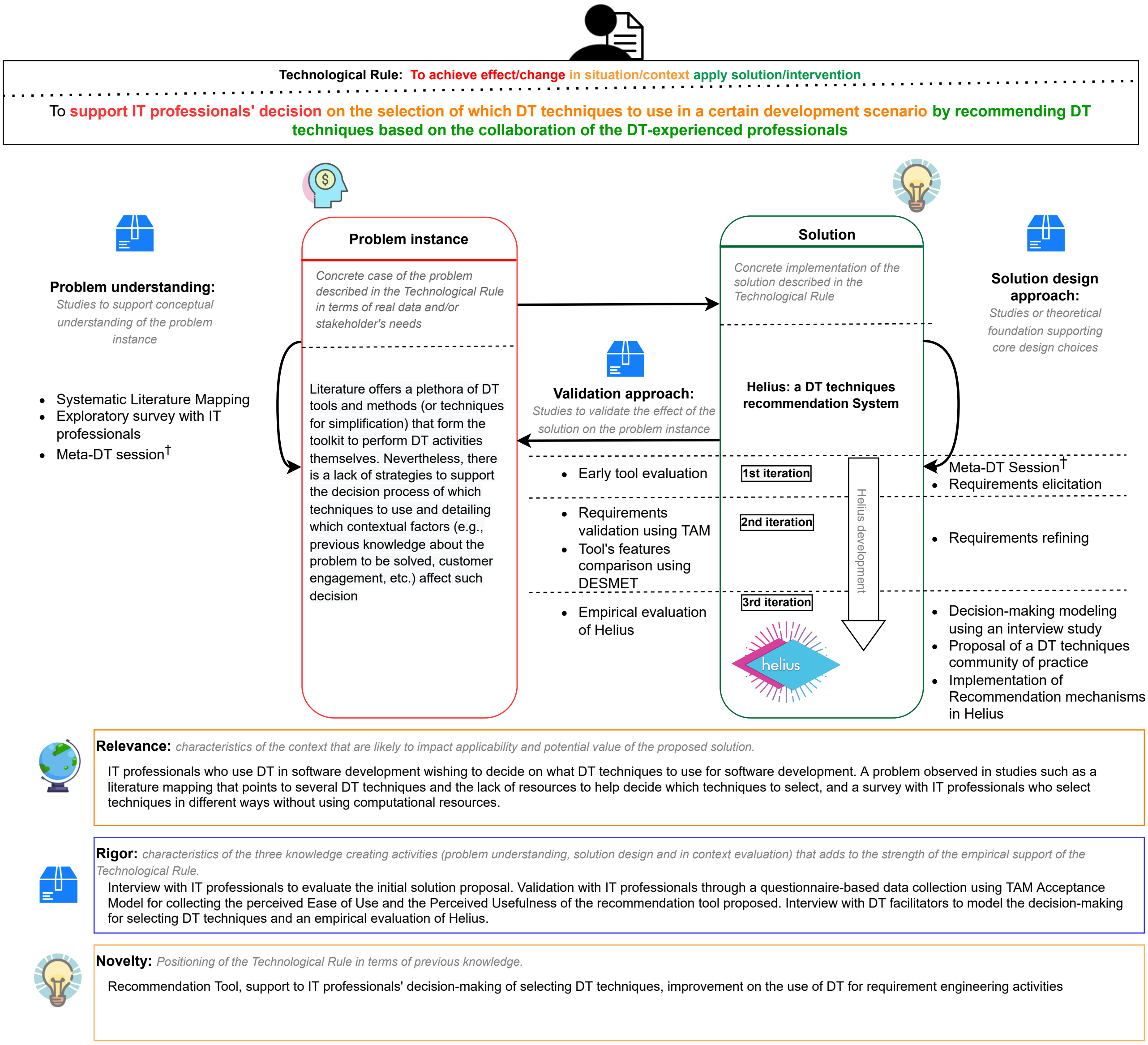


Figure 7.31: DSR method followed in this thesis (Runeson *et al.*, 2020) [228]

state-of-the-art and the state-of-the-practice in terms of DT use in software development (systematic literature mapping and survey, respectively).

This doctoral thesis sheds light on the importance of Design Thinking for software development. It also shows the potential of computational resources to help professionals use DT techniques to understand user needs and develop desirable and feasible solutions. As a result, we updated our Technological Rule to the following:

> To support IT professionals' decision-making to select which DT techniques to use in a specific development scenario by recommending DT techniques based on the experience of DT-experienced professionals.

Thus, considering the outcomes that we obtained from the studies that we carried out, we can frame the following solutions' constructs:

- Theoretical:

  – DT practitioners use a span of DT models and techniques in software development;
  – Although the use of DT brings benefits, DT practitioners face challenges in using such a problem-solving approach. Then, DT is not a silver bullet;
  – The decision of which techniques to use is a difficult task. It is made based on the context of the project to be developed.
  – The personalized and non-personalized recommendation mechanisms provide support to the selection of DT techniques

- Practical:

  Helius is a resource that supports the decision-making in the selection of Design Thinking techniques for use in software development.

8. FINAL CONSIDERATIONS

8.1 Thesis Overview

This thesis presents empirical and qualitative research in the field of Software Engineering. We followed the Design Science Research method, based on the template proposed by Runeson *et al.* (2020) [228]. We performed the following activities:

- Exploratory studies to identify the status quo and collect a research problem:
  - A systematic mapping of the literature to find out DT models, DT techniques, integration approaches with requirements engineering, and what challenges they face regarding the use of DT in the context of software development;
  - An exploratory survey with software development professionals to empirically identify which DT techniques they use, and the challenges and difficulties they face when using DT techniques in the context of software development.
- Problem definition and solution proposition:
  - Taking into consideration the scientific rigor stimulated by the use of DSR and seeking to achieve novelty and relevance to the research, we performed a DT meta-session. We used DT itself to solve a problem of using DT in software development. Through the collaboration of professionals from industry and academia, a recommendation system was proposed to contribute to the selection of DT techniques;
  - Next, we iterated over the DSR framework, conducting activities of solution design and constant validation. We evaluated the different stages and versions of the proposed technique recommendation system.
- Characterization of Helius decision-making, implementation and validation of the DT techniques recommendation system:
  - Next, we perceived that it was needed to conduct a study for understanding the DT moderators' decision-making behind the selection of DT techniques to subsidize the implementation of different DT technique recommendation mechanisms in Helius. As a result, we implemented personalized and non-personalized recommendation mechanisms. We also implemented a community of practice in Helius following the Spotify model of scalability for agile teams. This feature allows the users to know experiences from other professionals, turning Helius into a collaborative recommendation system;

– Finally, we validated the proposed recommendation system with professionals from the software development industry, where we found that Helius provides resources that support the selection of DT techniques for use in software development.

8.2 Contributions: implications for research and practice

This thesis brings the following contributions to research and practice in the Software Engineering field:

- Characterization of the state-of-the-art on the use o DT in software development, showing the cited DT models, how DT has been integrated into software development, what the DT techniques used and in what DT working spaces they are used, the attention points to be aware when using DT, the challenges and benefits of its use in software development, and the strategies that the professionals use to select the DT techniques taking into account the eyes of the literature (see Section 4.1);
- Characterization of the state-of-the-practice on the use of DT techniques in software development, showing what DT models and techniques the Brazilian DT practitioners use, how difficult they consider using DT and what the challenges to using DT in software development (see Section 4.2);
- Modeling of the DT facilitators' decision-making for selecting DT techniques to conduct DT workshops, exploring the collaborative environment of DT, and fostering the creative mindset of the stakeholders (see Section 7.2.1);
- Proposition, specification and implementation of 20 recommendation mechanisms to support the selection of DT techniques in software development (see Section 7.2.2);
- Proposition, implementation, and validation of a DT techniques recommendation system, composed of a specialized recommendation module capable of recommending DT techniques for both newcomers and expert DT practitioners who want to select DT techniques to use in the context of software development (see Section 7.2.2).

The solution developed in this thesis assists the decision-making of the selection of DT techniques in the following way: it helps practitioners search for information on available Design Thinking (DT) techniques for use in software development, provides access to a community of practice of DT techniques experiences and a representation of the combination of DT techniques used in real projects. Thus, taking into account the decision-making process, Helius helps to know the decision alternatives, the analysis of the use of techniques represented by the experiences of use and the evaluation of the decision via evaluation of the techniques.

Additionally, this thesis provides a set of 20 specialized recommendation mechanisms that can be used to evaluate other sources of information on the techniques to select. These mechanisms take into account various factors, such as the workshop objectives, participant profiles, and problem characteristics being addressed, to help professionals identify the possible decision alternatives and choose the appropriate technique for each situation.

In summary, the tool developed in this thesis helps professionals search for information on available DT techniques, identify possible decision alternatives, and evaluate other sources of information to select the most appropriate technique for each situation. This can significantly improve the decision-making process in software development and increase the effectiveness of using DT in software projects.

In addition, we consider that this thesis has the following implications:

- For researchers, we believe that our work contributes by synthesizing and advancing what is known about the use of DT in software development through a reproducible research process. Furthermore, our results show that there is room for further research on the topic, indicating that it is still possible to explore unaddressed questions such as the impact of the recommendations in projects which use DT or the challenges faced by newcomers in using the techniques. The challenges represent open opportunities for new research on the use of DT in software development.
- For practitioners, our research provides information on the strategies that can be used to integrate DT into software development, helping them to guide the DT process through different DT models and working spaces, and also supporting teams to know what DT techniques are available, what criteria are considered for the selection of techniques, and what are the challenges that could be faced when applying DT in software development. Our work also delivers a collaborative recommendation system that supports both novice and experts DT practitioners to use of DT techniques in software development. Therefore, we believe that our research contributes to the state of the practice by assisting DT practitioners in software development activities.

8.3 Limitations

This section clarifies the threats inherent to our studies that are qualitative by nature, and it also shows our actions aiming to mitigate them.

- *Problem understanding and solution definition using DT:* the use of DT as an approach to understanding the problem and defining a solution is an activity that explores the diversity of ideas and creativity. The solutions proposed in a DT session consider the participants' worldview. Although we have invited professionals who have used DT in software development, the proposed solution represents the ideas of those who participated in the session. Therefore, the proposed solution could not represent the needs of all DT practitioners. To mitigate this threat, we based our research on the DSR method and through 3 iterations, we empirically evaluated the proposed solution. In total, 209 professionals participated in different activities of our research.

- *Defined RE Artifacts:* The DT session's moderator introduced only 2 Personas (Joano and Sindi) to represent stereotypes of potential users. However, each persona aims to group users who face the challenge of selecting DT techniques from different perspectives. Literature has shown that Personas serve to connect users with similar profiles. We also generated other RE artifacts with the elicitation and refining of requirements (Blueprint, User Journeys, and Prototypes). They do not represent the complete set of existing artifacts that we could use for requirements elicitation and specification (e.g., scenarios, storyboards, etc.). However, we were able to collect and specify the requirements, moving forward in proposing a DT techniques recommendation system.

- *Feedback from professionals:* We collected data from different professionals on 3 different activities. The feedback given by professionals represents the individual point of view for our proposal and might not be generalized. To mitigate this, we interviewed 5 professionals in the early evaluation step, and we sent the invitation to 80 professionals for the tool's requirement validation step. Seven professionals accepted to participate in the requirements validation. Thus, although a more significant number of participants allows a more expressive capture of professionals' needs, our sample also represents professionals with different needs. Another threat to validity regarding the professionals' feedback is the introduction of our research goals and the tool's features in the Tool's requirements validation step. Although we have sent a pre-recorded video to introduce the goals and features, which may have biased the feedback given by the participants, we mitigated this threat by not showing the participants how they should use the features proposed by the tool.

- *Results:* The results represent our interpretation from the feedback collected in the Validation activity (early tool evaluation in Iteration 1, tool requirements validation in Iteration 2, and Helius empirical evaluation in Iteration 3). To mitigate the interpretation bias, we held meetings among the authors to discuss the artifacts produced and to analyze the feedback captured by professionals. In addition, we invited and collected data from different professionals in each iteration, spreading the target audience of our empirical studies.

8.4 Future Work

Although we have reached results throughout this thesis that contribute the theory and practice in the SE field, we still consider that there are open opportunities to be performed as future work:

- A confirmatory survey to verify with DT practitioners the decision-making elements for selecting DT techniques that we collected through the interview-based study (See Section 7.2);
- A longitudinal case study to observe the use of DT techniques recommendation system with IT teams developing software;
- New iterations in the DSR-based method (design space and validation approach space) to refine the recommendation mechanisms based on the results collected to our empirical evaluation study (See Section 7.3.2).

Next, we detail the three suggested studies as future work.

8.4.1 Survey on the decision-making of the selection of DT techniques

Linaker *et al.* (2015) [153] defines surveys as studies-in-the-large that support the researcher to discover insights into an area that is to some degree unknown. Thus, we suggest the conduction of a survey to assess with a large spam of DT practitioners the decision-making for selecting DT techniques that we pointed out in the thesis.

Next, we suggest some examples of questions to be part of the survey:

- Considering the decision-making elements for selecting DT techniques, which of them do you use?
- What frequency do you usually use the decision-making element that you mentioned before for selecting DT techniques?
- Is there any other decision-making element that you consider as relevant when selecting DT techniques?

To conduct the suggested survey, the guidelines provided by Kitchenham and Pfleeger's (2008) [132] can be used since they support the conduction of personal opinion surveys in software engineering research.

8.4.2 A longitudinal confirmatory case study on the use of Helius

Another study that we suggest as future work is a longitudinal confirmatory case study [77]. Easterbrook (2008) argues that confirmatory case studies are used to test existing theories, observing a phenomenon empirically in its real-life context [272]. The goal of this case study would be to observe DT practitioners after using Helius as a recommendation system for a long period of time (more than 5 months) to evaluate if the professionals' way of choosing DT techniques was affected by Helius.

In order to conduct the study, we suggest using interviews and ethnography as data collection instruments [77]. The results of this study might provide subsidies to the next suggestion that we make as future work, as we show in Section 8.4.3.

8.4.3 New iterations in the DSR-based method for refining Helius

We also suggest as future work the conduction of new iterations in the DSR-based research design to refine Helius, and its recommendation mechanisms and also to validate it with a larger number of professionals. For instance, we describe below some elements to be considered to refine Helius:

1. Discovering and framing new recommendation mechanisms in Helius. Since Helius is a recommendation system, new recommendation mechanisms might support even more DT practitioners in selecting DT techniques in software development. For instance, it could be added mechanisms recommending techniques based on the time needed to execute a technique, or even in regards to the DT session participants' profile (not covered by the current version of Helius). In addition, Helius could implement trust elements for recommending DT techniques based on Trust-aware Recommender Systems [58];

2. Evaluating the Helius' recommendation using recommendations' evaluation metrics such as Recall and Precision [79]. Recall indicates the number of relevant items that the RSs recommended from all possible items, while Precision means how relevant the recommended items were to the user. In addition, metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) indicate the RSs' capability of recommending the relevant items in the future [174];

3. Evaluating and applying alternative measures of items' similarity, such as Cosine Coefficient [79, 173]. We did choose the Pearson Correlation Coefficient to compute the users' similarity in Helius. However, it would be nice to use other similarity measures to compare the results and to be able to choose the most appropriate one.

4. Implementing new features taking into account the suggestions presented by the participants during the empirical study to evaluate Helius, such as:
   - To allow the user to add new DT techniques;
   - To add new filters to the graph of techniques combination;
   - To export the DT techniques to PDF files;
   - To update the techniques based on new concepts;
   - To notify the users about updates in the techniques ratings and experiences.
5. Proposing new versions of Helius for integrating it into different platforms, such as bots for forums-based systems.

REFERENCES

[1] Adikari, S.; Keighran, H.; Sarbazhosseini, H. “Embed Design Thinking in Co-Design for Rapid Innovation of Design Solutions”. In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 3–14.

[2] Adikari, S.; McDonald, C.; Campbell, J. “Reframed Contexts: Design Thinking for Agile User Experience Design”. In: Proceedings of the International Conference of Design, User Experience, and Usability, Las Vegas, USA: Springer, 2013, pp. 3–12.

[3] Aggarwal, C. “Recommender systems”. Heidelberg, Germany: Springer, 2016, 1 ed., vol. 1, 498p.

[4] Ahmad, R.; Chyi, W. A.; Sarlan, A.; Kasbon, R. “Guiding Novice Database Developers in Database Schema Creation”. In: Proceedings of the Conference on e-Learning, e-Management and e-Services, Melbourne, Australia: IEEE, 2014, pp. 64–49.

[5] Ahmed, B.; Dannhauser, T.; Philip, N. “A Lean Design Thinking Methodology (LDTM) for Machine Learning and Modern Data Projects”. In: Proceedings of the Computer Science and Electronic Engineering, Colchester, UK: IEEE, 2018, pp. 11–14.

[6] Al-Bashiri, H.; Abdulgabber, M. A.; Romli, A.; Hujainah, F. “Collaborative Filtering Recommender System: Overview and Challenges”, *Advanced Science Letters*, vol. 23–9, Sep 2017, pp. 9045–9049.

[7] Albaum, G. “The Likert Scale Revisited”, *Market Research Society Journal*, vol. 39–2, Feb 1997, pp. 1–21.

[8] Alhazmi, A.; Huang, S. “Integrating Design Thinking into Scrum Framework in the Context of Requirements Engineering Management”. In: Proceedings of the International Conference on Computer Science and Software Engineering, Beijing, China: ACM, 2020, pp. 33–45.

[9] Almeida, F. V.; Canedo, E. D.; da Costa, R. P. “Definition of Indicators in the Execution of Educational Projects with Design Thinking Using the Systematic Literature Review”. In: Proceedings of the Frontiers in Education Conference, Covington, USA: IEEE, 2019, pp. 1–9.

[10] Almohsen, K. A.; Al-Jobori, H. “Recommender Systems in Light of Big Data”, *International Journal of Electrical and Computer Engineering*, vol. 5–6, dec 2015, pp. 1553–1563.

[11] Amin, S. A.; Philips, J.; Tabrizi, N. “Current Trends in Collaborative Filtering Recommendation Systems”. In: Proceedings of the Services Conference Federation, San Diego, USA: Springer, 2019, pp. 46–60.

[12] Araújo, C. M. M. S.; Santos, I. M.; Canedo, E. D.; de Araújo, A. P. F. “Design Thinking Versus Design Sprint: A Comparative Study”. In: Proceedings of the International Conference on Design, User Experience, and Usability, Orlando, USA: Springer, 2019, pp. 291–306.

[13] Araújo, R.; Anjos, E.; Silva, D. R. “Trends in the Use of Design Thinking for Embedded Systems”. In: Proceedings of the International Conference on Computational Science and Its Applications, Banff, Canada: IEEE, 2015, pp. 82–86.

[14] Arbieto-Batallanos, C.; Villanueva-Montoya, L.; Chavez-Ponce, D.; Alfonte-Zapana, R.; Córdova-Martínez, M. “Mobile Application Based on Design Thinking for Teaching Kinematics”. In: Proceedings of the International Congress on Educational and Technology in Sciences, Arequipa, Peru: CEUR-WS, 2019, pp. 257–266.

[15] Asante, G. “Effective Design Methodologies”, *Design Management Review*, vol. 29–2, Jun 2018, pp. 10–15.

[16] Avalos, M.; Larios, V. M.; Salazar, P.; Maciel, R. “Hackathons, Semesterathons, and Summerathons as Vehicles to Develop Smart City Local Talent that via Their Innovations promote Synergy Between Industry, Academia, Government and Citizens”. In: Proceedings of the International Smart Cities Conference, Wuxi, China: IEEE, 2017, pp. 1–6.

[17] Azab, A.; Mostafa, N.; Park, J. “Ontimecargo: a Smart Transportation System Development in Logistics Management by a Design Thinking Approach”. In: Proceedings of the Pacific Asia Conference on Information Systems, Chiayi, Taiwan: AIS, 2016, pp. 1–16.

[18] Bargh, M. S.; Choenni, S. “Towards Applying Design-Thinking for Designing Privacy-Protecting Information Systems”. In: Proceedings of the International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, Los Angeles, USA: IEEE, 2019, pp. 196–202.

[19] Basili, V. R. “Goal Question Metric Paradigm”, *Encyclopedia of Software Engineering*, vol. 1, Jan 1994, pp. 528–532.

[20] Berger, A. “Design Thinking for Search User Interface Design”, *Central Europe Workshop Proceedings*, vol. 763, Jan 2011, pp. 1–4.

[21] Bhareti, K.; Perera, S.; Jamal, S.; Pallege, M. H.; Akash, V.; Wiieweera, S. "A Literature Review of Recommendation Systems". In: Proceedings of the International Conference for Innovation in Technology, Banglore, India: IEEE, 2020, pp. 1–7.

[22] Bhuvanya, R.; Kavitha, M. "Recommendation System: A Systematic Overview on Methods, Issues and Solutions", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 09–05, 1 2020, pp. 8851–8859.

[23] Blattgerste, J.; Behrends, J.; Pfeiffer, T. "A Web-Based Analysis Toolkit for the System Usability Scale". Source: https://github.com/jblattgerste/sus-analysis-toolkit, Last accessed on november, 2022.

[24] Blattgerste, J.; Behrends, J.; Pfeiffer, T. "A Web-Based Analysis Toolkit for the System Usability Scale". In: Proceedings of the International Conference on PErvasive Technologies Related to Assistive Environments, Corfu, Greece: ACM, 2022, pp. 1–10.

[25] Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. "Recommender Systems Survey", *Knowledge-Based Systems*, vol. 46–1, 10 2013, pp. 109–132.

[26] Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. "Recommender Systems Survey", *Knowledge-based Systems*, vol. 46–1, Jul 2013, pp. 109–132.

[27] Bock, A. "The Concepts of Decision Making: an Analysis of Classical Approaches and Avenues for the Field of Enterprise Modeling". In: Working Conference on The Practice of Enterprise Modeling, Valencia, Spain: Springer, 2015, pp. 306–321.

[28] Bordin, S.; De Angeli, A. "Communication Breakdowns in the Integration of User-centred Design and Agile Development". Heidelberg, Germany: Springer, 2016, 1 ed., chap. 6, pp. 137–161.

[29] Bouraga, S.; Jureta, I.; Faulkner, S.; Herssens, C. "Knowledge-based Recommendation Systems: a Survey", *International Journal of Intelligent Information Technologies*, vol. 10–2, Jul 2014, pp. 1–19.

[30] Bourque, P.; Richard, E. "Guide to the Software Engineering Body of Knowledge, Version 3.0", *IEEE Computer Society*, vol. 3, 2 2014, pp. 335.

[31] Brad, S.; Brad, E.; Homorodean, D. "CALDET: A TRIZ-Driven Integrated Software Development Methodology". In: Proceedings of the International TRIZ Future Conference, Marrakesh, Morocco: Springer, 2019, pp. 400–416.

[32] Braz, R. d. S.; Merlin, J. R.; Freitas Guilhermino Trindade, D.; Eduardo Ribeiro, C.; Sgarbi, E. M.; Junior, F. d. S. "Design Thinking and Scrum in Software Requirements

Elicitation: A Case Study". In: Proceedings of the International Conference on Human-Computer Interaction, Orlando, USA: Springer, 2019, pp. 179–194.

[33] Brenner, W.; Uebernickel, F.; Abrell, T. "Design Thinking as Mindset, Process, and Toolbox". Heidelberg, Germany: Springer, 2016, 1 ed., chap. 1, pp. 3–21.

[34] Brooke, J. "SUS-A Quick and Dirty Usability Scale", *Usability Evaluation in Industry*, vol. 189–194, Jan 1996, pp. 4–7.

[35] Brown, T. "Design Thinking", *Harvard Business Review*, vol. 86, Jul 2008, pp. 84–95.

[36] Brown, T. "Design Thinking: uma Metodologia Poderosa para Decretar o Fim das Velhas Ideias". Porto Alegre, Brasil: Alta Books, 2018, 2 ed., vol. 1, 272p.

[37] Brown, T.; Wyatt, J. "Design Thinking for Social Innovation", *Development Outreach*, vol. 12–1, Jan 2010, pp. 29–43.

[38] Budiarto, A.; Kacamarga, M. F.; Suparyanto, T.; Purnamasari, S.; Caraka, R. E.; Muljo, H. H.; Pardamean, B. "SMARTD Web-Based Monitoring and Evaluation System". In: Proceedings of the Indonesian Association for Pattern Recognition International Conference, Jakarta, Indonesia: IEEE, 2018, pp. 172–176.

[39] Burchardt, C.; Maisch, B. "Advanced Agile Approaches to Improve Engineering Activities", *Procedia Manufacturing*, vol. 25–1, May 2018, pp. 202–212.

[40] Canedo, E. D.; Parente da Costa, R. "The Use of Design Thinking in Agile Software Requirements Survey: A Case Study". In: Proceedings of the International Conference of Design, User Experience, and Usability, Las Vegas, USA: Springer, 2018, pp. 642–657.

[41] Canedo, E. D.; Pergentino, A. C. D. S.; Calazans, A. T. S.; Almeida, F. V.; Costa, P. H. T.; Lima, F. "Design Thinking Use in Agile Software Projects: Software Developers' Perception". In: Proceedings of the International Conference on Enterprise Information Systems, Online: Scitepress, 2020, pp. 217–224.

[42] Carell, A.; Lauenroth, K.; Platz, D. "Using Design Thinking for Requirements Engineering in the Context of Digitalization and Digital Transformation: A Motivation and an Experience Report". Heidelberg, Germany: Springer, 2018, chap. 7, pp. 107–120.

[43] Carlgren, L.; Rauth, I.; Elmquist, M. "Framing Design Thinking: The Concept in Idea and Enactment", *Creativity and Innovation Management*, vol. 25–1, Mar 2016, pp. 38–57.

[44] Carroll, N.; Richardson, I. “Aligning Healthcare Innovation and Software Requirements Through Design Thinking”. In: Proceedings of the International Workshop on Software Engineering in Healthcare Systems, Austin, USA: ACM, 2016, pp. 1–7.

[45] Chai-Arayalert, S.; Puttinaovarat, S. “Designing Mangrove Ecology Self-Learning Application Based on a Micro-Learning Approach”, *International Journal of Emerging Technologies in Learning*, vol. 15–11, Nov 2020, pp. 29–41.

[46] Challiol, C.; Borrelli, F. M.; Mendiburu, F. I.; Servat, C. M. R.; Goin Plexevi, F.; Orellano, D. H.; Gomez-Torres, E.; Gordillo, S. E. “Design Thinking’s Resources for in-situ Co-Design of Mobile Games”. In: Proceedings of the International Conference on Information Systems and Computer Science, Quito, Ecuador: IEEE, 2019, pp. 339–345.

[47] Charmaz, K. “Constructing Grounded Theory”. Los Angeles, USA: Sage, 2014, 2 ed., vol. 1, 416p.

[48] Chasanidou, D.; Gasparini, A. A.; Lee, E. “Design Thinking Methods and Tools for Innovation”. In: Proceedings of the Design, User Experience, and Usability: Design Discourse, Los Angeles, USA: Springer, 2015, pp. 12–23.

[49] Cho, J.; Wash, R. “How Potential New Members Approach an Online Community”, *Computer Supported Cooperative Work (CSCW)*, vol. 30–1, Jan 2021, pp. 35–77.

[50] Chongwatpol, J. “Operationalizing Design Thinking in Business Intelligence and Analytics Projects”, *Decision Sciences Journal of Innovative Education*, vol. 18–3, Mar 2020, pp. 409–434.

[51] Corral, L.; Fronza, I. “Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation”. In: Proceedings of the Special Interest Group on Conference on Information Technology Education, Fort Lauderdale, USA: ACM, 2018, pp. 26–31.

[52] Council, D. “The Design Process: What is the Double Diamond?” Source: designcouncil.org.uk/, Last accessed on january 2020.

[53] Coutinho, E. F.; Gomes, G. A. M.; José, M. A. “Applying Design Thinking in Disciplines of Systems Development”. In: Proceedings of the Euro American Conference on Telematics and Information Systems, Cartagena, Colombia: IEEE, 2016, pp. 1–8.

[54] Cremonesi, P.; Garzotto, F.; Negro, S.; Papadopoulos, A. V.; Turrin, R. “Looking for ”good” Recommendations: A Comparative Evaluation of Recommender Systems”. In: Proceedings of the International Conference on Human-Computer Interaction, Orlando, USA: Springer, 2011, pp. 152–168.

[55] Cruzes, D. S.; Dybå, T. "Synthesizing Evidence in Software Engineering Research". In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, Bolzano, Italy: ACM, 2010, pp. 1–10.

[56] Daniëls, N. E. M.; Hochstenbach, L. M. J.; van Bokhoven, M. A.; Beurskens, A. J. H. M.; Delespaul, P. A. E. G. "Implementing Experience Sampling Technology for Functional Analysis in Family Medicine – A Design Thinking Approach", *Frontiers in Psychology*, vol. 10–1, Dec 2019, pp. 1–15.

[57] Darrin, M. A. G.; Devereux, W. S. "The Agile Manifesto, Design Thinking and Systems Engineering". In: Proceedings of the International Systems Conference, Quebec, Canada: IEEE, 2017, pp. 1–5.

[58] Da'u, A.; Salim, N. "Recommendation System Based on Deep Learning Methods: a Systematic Review and New Directions", *Artificial Intelligence Review*, vol. 53–4, Aug 2020, pp. 2709–2748.

[59] Davis, F. D. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, vol. 13–3, Sep 1989, pp. 319–340.

[60] De, S. "A Novel Perspective to Threat Modelling using Design Thinking and Agile Principles". In: Proceedings of the International Conference on Parallel, Distributed and Grid Computing, Waknaghat, India: IEEE, 2020, pp. 31–35.

[61] De, S.; Vijayakumaran, V. "A Brief Study on Enhancing Quality of Enterprise Applications using Design Thinking", *International Journal of Education and Management Engineering*, vol. 9–5, Sep 2019, pp. 1–26.

[62] de Carvalho Souza, C. L.; Silva, C. "An Experimental Study of the Use of Design Thinking as a Requirements Elicitation Approach for Mobile Learning Environments", *CLEI Electronic Journal*, vol. 18–1, Apr 2015, pp. 1–6.

[63] De Paula, D. F.; Araújo, C. C. "Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment". In: Proceedings of the International Conference on Human-Computer Interaction, Toronto, Canada: Springer, 2016, pp. 30–34.

[64] De Paula, T. R.; Amancio, T. S.; Flores, J. A. N. "Design Thinking in Industry", *IEEE Software*, vol. 37–2, Jul 2020, pp. 49–51.

[65] De Souza, A. F. B.; Ferreira, B. M.; Conte, T. "Applying Design Thinking in Software Engineering: A Systematic Mapping". In: Proceedings of the Ibero-American Conference on Software Engineering, Buenos Aires, Argentina: Curran Associates, 2017, pp. 1–14.

[66] Diaz Intal, G. L.; Senoro, D.; Palaoag, T. “User Experience Design for Disaster Management Mobile Application Using Design Thinking Approach”. In: Proceedings of the International Conference on Software and E-Business, Osaka, Japan: ACM, 2020, pp. 7–13.

[67] Dobrigkeit, F.; de Paula, D. “The Best of Three Worlds-The Creation of InnoDev a Software Development Approach that Integrates Design Thinking, Scrum and Lean Startup”. In: Proceedings of the International Conference on Engineering Design, Vancouver, Canada: Design Society, 2017, pp. 319–328.

[68] Dobrigkeit, F.; de Paula, D. “Design Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering”. In: Proceedings of the European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia: ACM, 2019, pp. 1059–1069.

[69] Dobrigkeit, F.; de Paula, D.; Carroll, N. “InnoDev Workshop: A One Day Introduction to Combining Design Thinking, Lean Startup and Agile Software Development”. In: Proceedings of the Conference on Software Engineering Education and Training, Munich, Germany: IEEE, 2020, pp. 1–10.

[70] Dobrigkeit, F.; de Paula, D.; Uflacker, M. “InnoDev: a Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup”. Heidelberg, Germany: Springer, 2019, chap. 10, pp. 199–227.

[71] Dobrigkeit, F.; Pajak, P.; de Paula, D.; Uflacker, M. “DT@IT Toolbox: Design Thinking Tools to Support Everyday Software Development”. Springer, 2020, chap. 12, pp. 201–227.

[72] Dobrigkeit, F.; Wilson, M.; Nicolai, C. “Adding Scrum-Style Project Management to an Advanced Design Thinking Class”. In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 1–12.

[73] Docherty, C. “Perspectives on Design Thinking for Social Innovation”, *Design Journal*, vol. 20–6, Oct 2017, pp. 719–724.

[74] d.School Stanford. “Design Thinking - Bootcamp Bootleg Compilation”. Source: https://dschool.stanford.edu/resources/the-bootcamp-bootleg, Last accessed on september 2021.

[75] Dunne, D.; Martin, R. “Design Thinking and How it Will Change Management Education: An Interview and Discussion”, *Academy of Management Learning & Education*, vol. 5, Nov 2006, pp. 512–523.

[76] Díaz, P.; Aedo, I.; Cubas, J. “CoDICE: Balancing Software Engineering and Creativity in the Co-design of Digital Encounters with Cultural Heritage”. In: Proceedings of the Workshop on Advanced Visual Interfaces, Como, Italy: ACM, 2014, pp. 253–256.

[77] Easterbrook, S. “Empirical Research Methods for Software Engineering”. Atlanta, USA: IEEE/ACM, 2007, chap. 11, pp. 574.

[78] El-Sharkawy, S.; Schmid, K. “A Heuristic Approach for Supporting Product Innovation in Requirements Engineering: A Controlled Experiment”. In: Proceedings of the International Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany: Springer, 2011, pp. 78–93.

[79] Fayyaz, Z.; Ebrahimian, M.; Nawara, D.; Ibrahim, A.; Kashef, R. “Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities”, *Applied Sciences*, vol. 10–21, Nov 2020, pp. 1–20.

[80] Felderer, M.; Travassos, G. H. “The Evolution of Empirical Methods in Software Engineering”. Heidelberg, Germany: Springer, 2020, chap. 1, pp. 1–24.

[81] Felizardo, K. R.; Mendes, E.; Kalinowski, M.; Souza, É. F.; Vijaykumar, N. L. “Using Forward Snowballing to Update Systematic Reviews in Software Engineering”. In: Proceedings of the International Symposium on Empirical Software Engineering and Measurement, Ciudad Real, Spain: ACM/IEEE, 2016, pp. 1–6.

[82] Ferreira, B.; Conte, T.; Barbosa, S. D. J. “Eliciting Requirements Using Personas and Empathy Map to Enhance the User Experience”. In: Proceedings of the Brazilian Symposium on Software Engineering, Belo Horizonte, Brazil: IEEE, 2015, pp. 80–89.

[83] Ferreira, B.; Silva, W.; Barbosa, S. D.; Conte, T. “Technique for Representing Requirements using Personas: a Controlled Experiment”, *IET Software*, vol. 12, Jun 2018, pp. 280–290.

[84] Figma. “Figma prototyping Tool”. Last checked on Nov 10, 2022, Source: https://www.figma.com, Last accessed on november 2022.

[85] Freitas, R.; Peres, S.; Fantinato, M.; Steinbeck, R.; Araújo, U.; et al.. “Experimenting with Design Thinking in Requirements Refinement for a Learning Management System”. In: Proceedings of the Brazilian Symposium on Information Systems, João Pessoa, Brazil: SBC, 2013, pp. 182–193.

[86] Fuks, H.; Raposo, A.; Gerosa, M. A.; Pimental, M.; Lucena, C. J. “The 3c collaboration model”. Rio De Janeiro, Brasil: IGI Global, 2008, pp. 637–644.

[87] Gama, K.; Alencar, B.; Calegario, F.; Neves, A.; Alessio, P. “A Hackathon Methodology for Undergraduate Course Projects”. In: Proceedings of the Frontiers in Education Conference, San Jose, USA: IEEE, 2018, pp. 1–9.

[88] Gama, K.; Castor, F.; Alessio, P.; Neves, A.; Araújo, C.; Soares-Neto, F.; Oliveira, H. “Combining Challenge-Based Learning and Design Thinking to Teach Mobile App Development”. In: Proceedings of the Frontiers in Education Conference, San Jose, USA: IEEE, 2018, pp. 1–5.

[89] Gamble, M. T. “Can Metamodels Link Development to Design Intent?” In: Proceedings of the International Workshop on Bringing Architectural Design Thinking into Developers’ Daily Activities, Austin, USA: ACM, 2016, pp. 14–17.

[90] Gasparic, M.; Janes, A. “What Recommendation Systems for Software Engineering Recommend: A Systematic Literature Review”, *Journal of Systems and Software*, vol. 113–1, Mar 2016, pp. 101–113.

[91] Genzorova, T.; Corejova, T.; Stalmasekova, N. “How Digital Transformation can Influence Business Model, Case Study for Transport Industry”, *Transportation Research Procedia*, vol. 40–1, May 2019, pp. 1053 – 1058.

[92] Glaser, B. G. “Theoretical Sensitivity: Advances in the Methodology of Grounded Theory”. Mill Valley, CA: Sociology Press, 1978, 1 ed., vol. 1, 164p.

[93] Glaser, B. G. “Doing Grounded Theory: Issues and Discussions”. Mill Valley, CA: Sociology Press, 1998, 1 ed., vol. 1, 254p.

[94] Glomann, L. “Introducing ‘Human-Centered Agile Workflow’(HCAW)–An Agile Conception and Development Process Model”. In: Proceedings of the International Conference on Applied Human Factors and Ergonomics, Los Angeles, USA: Springer, 2017, pp. 646–655.

[95] Gmach, I.; Melek, G.; Sidhom, S.; Khrifish, L. “Towards an Approach of Trust-based Recommendation System”. In: Proceedings of the International Conference on Information Systems and Economic Intelligence, Tunis, Tunisia: IEEE, 2015, pp. 150–157.

[96] González, C. S. G.; Fleitas, Y. d. C. B. “Promoting Creativity and Innovative Thinking in Software Engineering Teaching: a Case Study”. In: Proceedings of the IEEE Global Engineering Education Conference, Tallinn, Estonia: IEEE, 2015, pp. 989–995.

[97] González, C. S. G.; González, E. G.; Cruz, V. M.; Saavedra, J. S. “Integrating the Design Thinking into the UCD’s Methodology”. In: Proceedings of the Engineering Education Conference, Madrid, Spain: IEEE, 2010, pp. 1477–1480.

[98] Goodspeed, R.; Riseng, C.; Wehrly, K.; Yin, W.; Mason, L.; Schoenfeldt, B. “Applying Design Thinking Methods to Ecosystem Management Tools: Creating the Great Lakes Aquatic Habitat Explorer”, *Marine Policy*, vol. 69–1, Jul 16, pp. 134–145.

[99] Goudsouzian, L. K.; Riola, P.; Ruggles, K.; Gupta, P.; Mondoux, M. A. “Integrating Cell and Molecular Biology Concepts: Comparing Learning Gains and Self-efficacy in Corresponding Live and Virtual Undergraduate Laboratory Experiences”, *Biochemistry and Molecular Biology Education*, vol. 46–4, Jul 2018, pp. 361–372.

[100] Guinan, P. J.; Parise, S.; Langowitz, N. “Creating an Innovative Digital Project Team: Levers to Enable Digital Transformation”, *Business Horizons*, vol. 62–6, Nov 2019, pp. 717–727.

[101] Gupta, S.; Dave, M. “An Overview of Recommendation System: Methods and Techniques”. In: Proceedings of the International Conference on Advancements in Computing and Management, Jaipur, India: Springer, 2020, pp. 231–237.

[102] Gürdür, D.; Törngren, M. “Visual Analytics for Cyber-physical Systems Development: Blending Design Thinking and Systems Thinking”. In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 1–15.

[103] Gurusamy, K.; Srinivasaraghavan, N.; Adikari, S. “An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation”. In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 34–42.

[104] Harriet, K.; Monika, K.; Verena, P. “InnoDeck: Card based Innovation Support - A Modular Human-Centered Approach to Facilitate Innovation Workshops”. In: Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Vienna, Austria: SciTePress, 2019, pp. 83–91.

[105] Hasso-Plattner-Institute, D.-S. “Hasso-Plattner-Institute: School of Design Thinking”. Source: https://hpi.de/en/school-of-design-thinking, Last accessed on february 2020.

[106] Hastie, R. “Problems for Judgment and Decision Making”, *Annual review of psychology*, vol. 52–1, Feb 2001, pp. 653–683.

[107] Hehn, J.; Mendez, D.; Uebernickel, F.; Brenner, W.; Broy, M. “On Integrating Design Thinking for Human-Centered Requirements Engineering”, *IEEE Software*, vol. 37–2, Jul 2020, pp. 25–31.

[108] Hehn, J.; Uebernickel, F. “The Use of Design Thinking for Requirements Engineering: An Ongoing Case Study in the Field of Innovative Software-Intensive Systems”. In: Proceedings of the International Requirements Engineering Conference, Banff, Canada: IEEE, 2018, pp. 400–405.

[109] Hehn, J.; Uebernickel, F.; Stöckli, E.; Brenner, W. “Designing Human-Centric Information Systems: Towards an Understanding of Challenges in Specifying Requirements within Design Thinking Projects”. In: Proceedings Of The Multikonferenz Wirtschaftsinformatik, Luneburg, Germany: Leuphana University of Lüneburg, 2018, pp. 1–12.

[110] Hevner, A. R. “A Three Cycle View of Design Science Research”, *Scandinavian Journal of Information Systems*, vol. 19–2, Jan 2007, pp. 87–92.

[111] Hiremath, M.; Sathiyam, V. “Fast Train to DT: a Practical Guide to Coach Design Thinking in Software Industry”. In: Proceedings of the International Federation for Information Processing, Cape Town, South Africa: Springer, 2013, pp. 780–787.

[112] Hoda, R.; Noble, J.; Marshall, S. “Developing a Grounded Theory to Explain the Practices of Self-organizing Agile Teams”, *Empirical Software Engineering*, vol. 17–6, Apr 2012, pp. 609–639.

[113] Hug, N. “Surprise: A Python Library for Recommender Systems”, *The Journal of Open Source Software*, vol. 5–52, Aug 2020, pp. 1–3.

[114] Husaria, A.; Guerreiro, S. “Requirement Engineering and the Role of Design Thinking”. In: Proceedings of the International Conference on Enterprise Information Systems, Online: Scitepress, 2020, pp. 353–359.

[115] IDEO. “IDEO Design Kit”. Source: https://designkit.org/methods, Last accessed on february 2020.

[116] Ignácio, R. C.; Benitti, F. B. V. “Improving the Selection of Requirements Elicitation Techniques: a Faceted Guide”. In: Proceedings of the Workshop on Requirements Engineering, Virtual conference: PUC-RIO, 2020, pp. 80–93.

[117] Isinkaye, F.; Folajimi, Y.; Ojokoh, B. “Recommendation Systems: Principles, Methods and Evaluation”, *Egyptian Informatics Journal*, vol. 16–3, Nov 2015, pp. 261–273.

[118] Jariha, P.; Jain, S. K. “A State-of-the-art Recommender Systems: An Overview on Concepts, Methodology and Challenges”. In: Proceedings of the International Conference on Inventive Communication and Computational Technologies, Coimbatore, India: IEEE, 2018, pp. 1769–1774.

[119] Jensen, M. B.; Lozano, F.; Steinert, M. “The Origins of Design Thinking and the Relevance in Software Innovations”. In: Proceedings of the International Conference on Product-Focused Software Process Improvement, Trondheim, Norway: Springer, 2016, pp. 675–678.

[120] Johansson-Sköldberg, U.; Woodilla, J.; Çetinkaya, M. “Design Thinking: Past, Present and Possible Futures”, *Creativity and Innovation Management*, vol. 22–2, Mar 2013, pp. 121–146.

[121] Jolak, R.; Wortmann, A.; Liebel, G.; Umuhoza, E.; Chaudron, M. R. V. “The Design Thinking of Co-Located vs. Distributed Software Developers: Distance Strikes Again!” In: Proceedings of the International Conference on Global Software Engineering, Seoul, Republic of Korea: ACM, 2020, pp. 106—-116.

[122] Júnior, G. G.; do Nascimento, R. L. S. “Desenvolvendo um Objeto de Aprendizagem Virtual para o Ensino da Programação Web Mobile com o Design Thinking”. In: Anais do Congresso Regional sobre Tecnologias na Educação, Natal, Brasil: CEUR-WS, 2016, pp. 357–368.

[123] Kah-Hoe; Wang, S. M. “Design Thinking for Usability Evaluation of Cloud Platform Service”. In: Proceedings of the International Conference on Applied System Invention, Chiba, Japan: IEEE, 2014, pp. 247–250.

[124] Kahan, E.; Genero, M.; Oliveros, A. “Challenges in Requirement Engineering: Could Design Thinking Help?” In: Proceedings of the International Conference on the Quality of Information and Communications Technology, Ciudad Real, Spain: Springer, 2019, pp. 79–86.

[125] Kasunic, M. “Designing an Effective Survey”. London, UK: Carnegie Mellon University, 2018, 1 ed., vol. 1, 143p.

[126] Kawano, A.; Motoyama, Y.; Aoyama, M. “A LX (Learner EXperience)-Based Evaluation Method of the Education and Training Programs for Professional Software Engineers”. In: Proceedings of the International Conference on Information and Education Technology, Aizu-Wakamatsu, Japan: ACM, 2019, pp. 151–159.

[127] Keighran, H.; Adikari, S. “Developing High-Performing Teams: A Design Thinking Led Approach”. In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 53–64.

[128] Khoali, M.; Tali, A.; Laaziz, Y. “Advanced Recommendation Systems Through Deep Learning”. In: Proceedings of the International Conference on Networking, Information Systems & Security, Marrakech, Morocco: ACM, 2020, pp. 1–8.

[129] Kitchenham, B.; Charters, S. “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, *Technical Report - School of Computer Science and Mathematics Keele University*, vol. 2–1, Jul 2007, pp. 512–523.

[130] Kitchenham, B.; Pretorius, R.; Budgen, D.; Brereton, O. P.; Turner, M.; Niazi, M.; Linkman, S. “Systematic Literature Reviews in Software Engineering–A Tertiary Study”, *Information and software technology*, vol. 52–8, Aug 2010, pp. 792–805.

[131] Kitchenham, B. A. “Evaluating Software Engineering Methods and Tool part 1”, *ACM SIGSOFT Software Engineering Notes*, vol. 21–1, jan 1996, pp. 11–14.

[132] Kitchenham, B. A.; Pfleeger, S. L. “Personal Opinion Surveys”, *Guide to Advanced Empirical Software Engineering*, vol. 1–1, Jul 2008, pp. 63–92.

[133] Kitchenham, B. A.; Pfleeger, S. L.; Pickard, L. M.; Jones, P. W.; Hoaglin, D. C.; El Emam, K.; Rosenberg, J. “Preliminary Guidelines for Empirical Research in Software Engineering”, *Trans. on Software Engineering*, vol. 28, Aug 2002, pp. 721–734.

[134] Kloeckner, A. P.; Libânio, C. d. S.; Ribeiro, J. L. D. “Design Thinking Methods and Techniques in Design Education”. In: Proceedings of the International Conference on Engineering and Product Design Education, Building Community, Oslo, Norway: Design Society, 2017, pp. 537–542.

[135] Kniberg, H.; Ivarsson, A. “Scaling Agile Spotify with Tribes, Squads, Chapters & Guilds”, *Spotify Whitepaper*, vol. 12–1, Oct 2012, pp. 1–8.

[136] Kongot, A.; Pattanaik, M. “Empowering Project Managers in Enterprises - A Design Thinking Approach to Manage Commercial Projects”. In: Proceedings of the Conference on Human Computer Interactions, Mumbai, India: Springer, 2017, pp. 189–197.

[137] Konstan, J. A.; Riedl, J. “Recommender Systems: From Algorithms to User Experience”, *User Modeling and User-adapted Interact*, vol. 22–1, Mar 2012, pp. 101–123.

[138] Kroeger, T. A.; Davidson, N. J.; Cook, S. C. “Understanding the Characteristics of Quality for Software Engineering Processes: A Grounded Theory Investigation”, *Information and Software Technology*, vol. 56–2, Feb 2014, pp. 252–271.

[139] Kumar, N.; Sneha, Y.; Mungara, J.; Prasad, S. R. “A Survey on Data Mining Methods Available for Recommendation System”. In: Proceedings of the International Conference on Computational Systems and Information Technology for Sustainable Solution, Bengaluru, India: IEEE, 2017, pp. 1–6.

[140] Kuula, S.; Haapasalo, H.; Kosonen, J. “Three Phases of Transforming a Project-Based IT Company Into a Lean and Design-Led Digital Service Provider”, *IEEE Software*, vol. 37–2, Jul 2020, pp. 41–48.

[141] Landis, J. R.; Koch, G. G. “The Measurement of Observer Agreement for Categorical Data”, *Biometrics*, vol. 33–1, Fev 1977, pp. 159–174.

[142] Larsen, L. J. “Juicing the Game Design Process: Towards a Content-centric Framework for Understanding and Teaching Game Design in Higher Education”, *Educational Media International*, vol. 55–3, Sep 2018, pp. 231–254.

[143] Lee, W. S.; Sohn, S. Y. “Discovering Emerging Business Ideas Based on Crowdfunded Software Projects”, *Decision Support Systems*, vol. 116–1, Jan 2019, pp. 102 – 113.

[144] Lehto, M. R.; Nah, F. F.-H.; Yi, J. S. “Decision-Making Models, Decision Support, and Problem Solving”. New Jersey, USA: John Wiley & Sons, 2012.

[145] Levy, M. “Promoting the Elicitation of Usability and Accessibility Requirements in Design Thinking: Using a Designed Object as a Boundary Object”. In: Proceedings of the International Requirements Engineering Conference, Lisbon, Portugal: IEEE, 2017, pp. 156–159.

[146] Levy, M. “Educating for Empathy in Software Engineering Course”. In: Proceedings of the International Conference on Requirements Engineering, Utrecht, Holland: CEUR-WS, 2018, pp. 1–9.

[147] Levy, M.; Hadar, I. “The Importance of Empathy for Analyzing Privacy Requirements”. In: Proceedings of the International Workshop on Evolving Security & Privacy Requirements Engineering, Banff, Canada: IEEE, 2018, pp. 9–13.

[148] Levy, M.; Huli, C. “Design Thinking in a Nutshell for Eliciting Requirements of a Business Process: A Case Study of a Design Thinking Workshop”. In: Proceedings of the International Requirements Engineering Conference, Jeju Island, South Korea: IEEE, 2019, pp. 351–356.

[149] Lewrick, M.; Link, P.; Leifer, L. “The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods”. New Jersey, USA: John Wiley & Sons, 2020, 1 ed., vol. 1, 316p.

[150] Liang, T.-P. “Recommendation Systems for Decision Support: An editorial introduction”, *Decision Support Systems*, vol. 45–3, Jun 2008, pp. 385–386.

[151] Liedtka, J. “Perspective: Linking Design Thinking With Innovation Outcomes through Cognitive Bias Reduction”, *Journal of Product Innovation Management*, vol. 32–6, Mar 2015, pp. 925–938.

[152] Liikkanen, L. A.; Kilpiö, H.; Svan, L.; Hiltunen, M. “Lean UX: the Next Generation of User-Centered Agile Development?” In: Proceedings of the Nordic Conference on Human-Computer Interaction, Helsinki, Finland: ACM, 2014, pp. 1095—-1100.

[153] Linaker, J.; Sulaman, S. M.; Höst, M.; de Mello, R. M. “Guidelines for Conducting Surveys in Software Engineering”, *Journal of Lund University*, vol. 1–1, Mar 2015, pp. 1–63.

[154] Lindberg, T.; Meinel, C.; Wagner, R. “Design Thinking: A Fruitful Concept for IT Development?” Heidelberg, Germany: Springer, 2011, chap. 3, pp. 3–18.

[155] Liu, C.; Li, L.; Yao, X.; Tang, L. “A Survey of Recommendation Algorithms Based on Knowledge Graph Embedding”. In: Proceedings of the International Conference on Computer Science and Educational Informatization, Kunming, China: IEEE, 2019, pp. 168–171.

[156] Lucena, P.; Braz, A.; Chicoria, A.; Tizzei, L. “IBM Design Thinking Software Development Framework”. In: Proceedings of the Brazilian Workshop on Agile Methods, Curitiba, Brazil: Springer, 2016, pp. 98–109.

[157] Luchs, M. G. “A Brief Introduction to Design Thinking”. New Jersey, USA: John Wiley & Sons, 2015, 1 ed., vol. 1, chap. 1, pp. 1–12.

[158] Luedeke, T. F.; Köhler, C.; Conrad, J.; Grashiller, M.; Sailer, A.; Vielhaber, M.; et al.. “CPM/PDD in the Context of Design Thinking and Agile Development of Cyber-Physical Systems”. In: Proceedings of the NordDesign, Linköping, Sweden: Design Society, 2018, pp. 1–10.

[159] Luma, I. “Innovating for People: Handbook of Human-Centered Design Methods”. New York, USA: LUMA Institute, LLC, 2012, 1 ed., vol. 1, 89p.

[160] Magare, A.; Lamin, M.; Chakrabarti, P. “Inherent Mapping Analysis of Agile Development Methodology Through Design Thinking”. In: Proceedings of the International Conference on Data Science and Intelligent Applications, Gujarat, India: Springer, 2020, pp. 527–534.

[161] Magare, A.; Lamin, M.; Chakrabarti, P. “Inherent Mapping Analysis of Agile Development Methodology Through Design Thinking”. In: Proceedings of International Conference on Data Science and Intelligent Applications, Gujarat, India: Springer, 2020, pp. 527–534.

[162] Mahe, N.; Adams, B.; Marsan, J.; Templier, M.; Bissonnette, S. “Migrating a Software Factory to Design Thinking: Paying Attention to People and Mind-Sets”, *IEEE Software*, vol. 37–2, Jul 2020, pp. 32–40.

[163] Mali, M.; Mishra, D. S.; Vijayalaxmi, M. “Multifaceted Recommender Systems Methods: A Review”, *Journal of Statistics and Management Systems*, vol. 23–2, Apr 2020, pp. 349–361.

[164] Malins, J.; Maciver, F. “From the Real to the Virtual: Developing Improved Software Using Design Thinking”. Heidelberg, Germany: Springer, 2016, 1 ed., vol. 1, chap. 16, pp. 37–349.

[165] Marshall, C.; Brereton, P.; Kitchenham, B. “Tools to Support Systematic Reviews in Software Engineering”. In: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, New York, USA: ACM, 2014, pp. 1–10.

[166] Martins, A. I.; Rosa, A. F.; Queirós, A.; Silva, A.; Rocha, N. P. “European Portuguese Validation of the System Usability Scale (SUS)”, *Procedia Computer Science*, vol. 67–1, Mar 2015, pp. 293–300.

[167] Martins, H. F.; Carvalho de Oliveira Junior, A.; Canedo, E. D.; Kosloski, R. D. A.; Paldês, R. A.; Oliveira, E. C. “Design Thinking: Challenges for Software Requirements Elicitation”, *Information*, vol. 10–12, Nov 2019, pp. 27.

[168] Matz, A.; Germanakos, P. “Increasing the Quality of Use Case Definition Through a Design Thinking Collaborative Method and an Alternative Hybrid Documentation Style”. In: Proceedings of the International Conference on Learning and Collaboration Technologies, Toronto, Canada: Springer, 2016, pp. 48–59.

[169] McMahon, E. “From Product Development to Innovation”. In: Proceedings of the International Conference of the American Society for Engineering Management, Virginia Beach, USA: ASEM, 2014, pp. 1–11.

[170] Meireles, M.; Souza, A.; Conte, T.; Maldonado, J. “Organizing the Design Thinking Toolbox: Supporting the Requirements Elicitation Decision Making”. In: Proceedings of the Brazilian Symposium on Software Engineering, Joinville, Brazil: ACM, 2021, pp. 285–290.

[171] Miranda, L.; Viterbo, J.; Bernardini, F. “Towards the Use of Clustering Algorithms in Recommender Systems”. In: Proceedings of the Americas Conference on Information Systems, Online conference: AIS, 2020, pp. 1–10.

[172] Mohamed, M. H.; Khafagy, M. H.; Ibrahim, M. H. “Recommender Systems Challenges and Solutions Survey”. In: Proceedings of the International Conference on Innovative Trends in Computer Engineering, Aswan, Egypt: IEEE, 2019, pp. 149–155.

[173] Mohan Kubendrian, N. P. “Recommendation System: A Literature Survey”, *International Journal of Innovative Technology and Exploring Engineering*, vol. 08–07, Jul 2019, pp. 2153–2161.

[174] Monti, D.; Rizzo, G.; Morisio, M. “A Systematic Literature Review of Multicriteria Recommender Systems”, *Artificial Intelligence Review*, vol. 54–6, Jun 2021, pp. 427–468.

[175] Mosconi, G.; Korn, M.; Reuter, C.; Tolmie, P.; Teli, M.; Pipek, V. “From Facebook to the Neighbourhood: Infrastructuring of Hybrid Community Engagement”, *Computer Supported Cooperative Work*, vol. 26–4, Jun 2017, pp. 959–1003.

[176] Mucherino, A.; Papajorgji, P. J.; Pardalos, P. M. “K-nearest Neighbor Classification”. In: *Data Mining in Agriculture*, 1 ed., Heidelberg, Germany: Springer, 2009, vol. 1, pp. 83–106.

[177] Mullins, C. “Responsive, Mobile App, Mobile First: Untangling the UX Design Web in Practical Experience”. In: Proceedings of the International Conference on the Design of Communication, Limerick, Ireland: ACM, 2015, pp. 1–6.

[178] Murugesan, L. K.; Hoda, R.; Salcic, Z. “Identifying Design Features Using Combination of Requirements Elicitation Techniques”. In: Proceedings of the International Workshop on Design and Innovation in Software Engineering, Buenos Aires, Argentina: IEEE, 2017, pp. 6–12.

[179] Mustafa, N.; Ibrahim, A. O.; Ahmed, A.; Abdullah, A. “Collaborative Filtering: Techniques and Applications”. In: Proceedings of the International Conference on Communication, Control, Computing and Electronics Engineering, Khartoum, Sudan: IEEE, 2017, pp. 1–6.

[180] Mutuku, L. N.; Colaco, J. “Increasing Kenyan Open Data Consumption: A Design Thinking Approach”. In: Proceedings of the International Conference on Theory and Practice of Electronic Governance, Albany, USA: ACM, 2012, pp. 18–21.

[181] Nagarnaik, P.; Thomas, A. “Survey on Recommendation System Methods”. In: Proceedings of the International Conference on Electronics and Communication Systems, Cairo, Egypt: IEEE, 2015, pp. 1603–1608.

[182] Nalepa, G.; Fontana, R. M.; Reinehr, S.; Malucelli, A. “Using Agile Approaches to Drive Software Process Improvement Initiatives”. In: Proceedings of the European Conference on Software Process Improvement, Edinburgh, UK: Springer, 2019, pp. 495–506.

[183] Nedeltcheva, G.; Shoikova, E. “Innovation through Design Thinking, User Experience and Agile: Towards Cooperation Framework”. In: Proceedings of the International Conference Telecom, Sofia, Bulgaria: NSTC, 2018, pp. 42–49.

[184] Nedeltcheva, G. N.; Shoikova, E. “Coupling Design Thinking, User Experience Design and Agile: Towards Cooperation Framework”. In: Proceedings of the International Conference on Big Data and Internet of Thing, London, England: ACM, 2017, pp. 225–229.

[185] Newman, P.; Ferrario, M. A.; Simm, W.; Forshaw, S.; Friday, A.; Whittle, J. “The Role of Design Thinking and Physical Prototyping in Social Software Engineering”. In: Proceedings of the International Conference on Software Engineering, Florence, Italy: IEEE, 2015, pp. 487–496.

[186] Ning, X.; Karypis, G. “Recent Advances in Recommender Systems and Future Directions”. In: Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, Tezpur, India: Springer, 2015, pp. 3–9.

[187] Oliveira, E.; Conte, T.; Cristo, M.; Valentim, N. “Influence Factors in Software Productivity—A Tertiary Literature Review”, *International Journal of Software Engineering and Knowledge Engineering*, vol. 28–12, Nov 2018, pp. 1795–1810.

[188] Ong, K.; Haw, S.-C.; Ng, K.-W. “Deep Learning Based-Recommendation System: An Overview on Models, Datasets, Evaluation Metrics, and Future Trends”. In: Proceedings of the International Conference on Computational Intelligence and Intelligent Systems, Bangkok, Thailand: ACM, 2019, pp. 6–11.

[189] Ortiz-Rojas, M.; Maya, R.; Jimenez, A.; Hilliger, I.; Chiluiza, K. “A Step by Step Methodology for Software Design of a Learning Analytics Tool in Latin America: A Case Study in Ecuador”. In: Proceedings of the Latin American Conference on Learning Technologies, San Jose Del Cabo, Mexico: IEEE, 2019, pp. 116–122.

[190] Ozsoy, M. G.; Polat, F. “Trust Based Recommendation Systems”. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Niagara Falls, Canada: ACM/IEEE, 2013, pp. 1267–1274.

[191] Palacin Silva, M.; Khakurel, J.; Happonen, A.; Hynninen, T.; Porras, J. “Infusing Design Thinking Into a Software Engineering Capstone Course”. In: Proceedings of the Conference on Software Engineering Education and Training, Savannah, United States: IEEE, 2017, pp. 212–221.

[192] Palomares, I.; Kovalchuk, S. V. “Multi-view Data Approaches in Recommender Systems: an Overview”, *Procedia Computer Science*, vol. 119–1, Nov 2017, pp. 30–41.

[193] Parizi, R.; Couto, I.; Hanauer, L.; Marczak, S.; Conte, T. “Helius: On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals’ Collaboration.” In: Proceedings of the Workshop on Requirements Engineering, Virtual Conference: PUC-RIO, 2020, pp. 80–93.

[194] Parizi, R.; Moreira da Silva, M.; de Souza Couto, I.; Pavin Trindade, K.; Plautz, M.; Marczak, S.; Conte, T.; Candello, H. “Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool”. In: Proceedings of the Ibero-American Conference-American on Software-American Engineering, Curitiba, Brazil: Curran Associates, 2020, pp. 14.

[195] Parizi, R.; Prestes, M.; Marczak, S.; Conte, T. “How has Design Thinking being Used and Integrated into Software Development Activities? A Systematic Mapping”, *Journal of Systems and Software*, vol. 187, 5 2022, pp. 1–27.

[196] Park, D. H.; Kim, H. K.; Choi, I. Y.; Kim, J. K. “A Literature Review and Classification of Recommender Systems Research”, *Expert Systems with Applications*, vol. 39–1, Sep 2012, pp. 10059–10072.

[197] Park, J.; Mostafa, N. A.; Han, H.-J. “StoryWeb: A Storytelling-based Knowledge-sharing Application Among Multiple Stakeholders”, *Creativity and Innovation Management*, vol. 29–2, Apr 2020, pp. 224–236.

[198] Pendse, A.; Amre, H. “Software 4.0: ’How’ of Building ’Next-Gen’ Systems”. In: Proceedings of the Innovations in Software Engineering Conference, Hyderabad, India: ACM, 2018, pp. 1–12.

[199] Penzenstadler, B. “When Does Design Help Thinking, and When Does Design Thinking Help?”, *IEEE Software*, vol. 37–2, Jul 2020, pp. 6–9.

[200] Pereira, J.; Russo, R. “Design Thinking Integrated in Agile Software Development: A Systematic Literature Review”, *Computer Science*, vol. 138–1, Oct 2018, pp. 775–782.

[201] Pereira, L.; Parizi, R.; Prestes, M.; Marczak, S.; Conte, T. “Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering”. In: Proceedings of the Annual ACM Symposium on Applied Computing, Virtual Event, Republic of Korea: ACM, 2021, pp. 1338–1345.

[202] Pereira, L.; Sharp, H.; de Souza, C.; Oliveira, G.; Marczak, S.; Bastos, R. “Behavior-Driven Development Benefits and Challenges: Reports from an Industrial Study”. In: Proc. of the Int’l Conference on Agile Software Development, Porto, Portugal: ACM, 2018, pp. 1–4.

[203] Pessotto, V. “Cognitive biases in Design Thinking processes: the Active Learning Lab case study”, Master’s Thesis, Ca’ Foscari University of Venice, Venice, Italy, 2017, 112p.

[204] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. “Systematic Mapping Studies in Software Engineering”. In: Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy: ACM, 2008, pp. 68–77.

[205] Petersen, K.; Vakkalanka, S.; Kuzniarz, L. “Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update”, *Information and Software Technology*, vol. 64–1, Aug 2015, pp. 1–18.

[206] Pfleeger, S. L.; Kitchenham, B. A. “Principles of Survey Research: Part 1: Turning Lemons into Lemonade”, *Software Engineering Notes*, vol. 26–6, Nov 2001, pp. 16–18.

[207] Pham, Y. D.; Fucci, D.; Maalej, W. “A First Implementation of a Design Thinking Workshop during a Mobile App Development Course Project”. In: Proceedings of the International Workshop on Software Engineering Education for Millennials, Gothenburg, Sweden: IEEE, 2018, pp. 1–8.

[208] Piras, L.; Dellagiacoma, D.; Perini, A.; Susi, A.; Giorgini, P.; Mylopoulos, J. “Design Thinking and Acceptance Requirements for Designing Gamified Software”. In: Proceedings of the International Conference on Research Challenges in Information Science, Brussels, Belgium: IEEE, 2019, pp. 1–12.

[209] Poriya, A.; Bhagat, T.; Patel, N.; Sharma, R. “Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems”, *International Journal of Applied Information Systems*, vol. 6–9, Mar 2014, pp. 22–27.

[210] Prasad, W. R.; Perera, G.; Padmini, K. J.; Bandara, H. D. “Adopting Design Thinking Practices to Satisfy Customer Expectations in Agile Practices: A Case from Sri Lankan Software Development Industry”. In: Proceedings of the Moratuwa Engineering Research Conference, Moratuwa, Sri Lanka: IEEE, 2018, pp. 471–476.

[211] Prestes, M.; Parizi, R.; Marczak, S.; Conte, T. “On the Use of Design Thinking: A Survey of the Brazilian Agile Software Development Community”. In: Proceedings of the International Conference on Agile Software Development, Copenhagen, Denmark: Springer, 2020, pp. 73–86.

[212] Prilla, M.; Blunk, O.; Chounta, I.-A. “How Does Collaborative Reflection Unfold in Online Communities? An Analysis of Two Data Sets”, *Computer Supported Cooperative Work*, vol. 29–6, Aug 2020, pp. 697–741.

[213] Przybilla, L.; Schreieck, M.; Klinker, K.; Pflügler, C.; Wiesche, M.; Krcmar, H. “Combining Design Thinking and Agile Development to Master Highly Innovative IT Projects”, *Proceedings of the Project Management and Procedure Models conference*, vol. 1–1, Jan 2018, pp. 113–124.

[214] Puad, N. H. M.; Rahim, N. A.; Firdaus, K. H. A.; Sayedi, N.; Mohadis, H. M. “Designing a Persuasive Application for Behaviour Change with Children”, *Journal of Marketing and Information Systems*, vol. 1–3, Feb 2019, pp. 1–7.

[215] Pusca, D.; Northwood, D. O. “Design Thinking and its Application to Problem Solving”, *Global Journal of Engineering Education*, vol. 20–1, Jan 2018, pp. 1–3.

[216] Péraire, C. “Dual-Track Agile in Software Engineering Education”. In: Proceedings of the International Conference on Software Engineering, Montreal, Canada: ACM/IEEE, 2019, pp. 38–49.

[217] Pérez, I. J.; Cabrerizo, F. J.; Herrera-Viedma, E. “A Mobile Decision Support System for Dynamic Group Decision-Making Problems”, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40–6, Nov 2010, pp. 1244–1256.

[218] Quant-UX. “QuantUX Prototyping Tool”. Source: https://www.quant-ux.com, Last accessed on november, 2021.

[219] Queiros, L. M.; Da Silveira, D. S.; da Silva Correia-Neto, J.; Vilar, G. “LODPRO: Learning Objects Development Process”, *Journal of the Brazilian Computer Society*, vol. 22–3, Jul 2016, pp. 1–3.

[220] Rainer, A.; Wohlin, C. “Recruiting Credible Participants for Field Studies in Software Engineering Research”, *Information and Software Technology*, vol. 151–1, Nov 2022, pp. 1–15.

[221] Rajashekharaiah, K. M. M.; Pawar, M.; Patil, M. S.; Kulenavar, N.; Joshi, G. “Design Thinking Framework to Enhance Object Oriented Design and Problem Analysis Skill in Java Programming Laboratory: An Experience”. In: Proceedings of the International Conference on MOOCs, Innovation and Technology in Education, Madurai, India: IEEE, 2016, pp. 200–205.

[222] Rangel, A.; Camerer, C.; Montague, P. R. “A Framework for Studying the Neurobiology of Value-based Decision Making”, *Nature reviews neuroscience*, vol. 9–7, Jun 2008, pp. 545–556.

[223] Reddy, P. D.; Iyer, S.; Sasikumar, M. “FATHOM: TEL Environment to Develop Divergent and Convergent Thinking Skills in Software Design”. In: Proceedings

of the International Conference on Advanced Learning Technologies, Timisoara, Romania: IEEE, 2017, pp. 414–418.

[224] Rhinow, H.; Meinel, C. "Design Thinking: Expectations from a Management Perspective". Heidelberg, Germany: Springer, 2014, 1 ed., vol. 1, chap. 3, pp. 239–252.

[225] Ricci, F.; Rokach, L.; Shapira, B. "Introduction to Recommender Systems Handbook". In: *Recommender Systems Handbook*, 1 ed., Heidelberg, Germany: Springer, 2011, vol. 1, pp. 1–35.

[226] Robillard, M.; Walker, R.; Zimmermann, T. "Recommendation Systems for Software Engineering", *IEEE Software*, vol. 27–4, Dec 2010, pp. 80–86.

[227] Rösel, A. "Are We Ready for Disruptive Improvement?" Heidelberg, Germany: Springer, 2016, 1 ed., vol. 1, chap. 5, pp. 77–91.

[228] Runeson, P.; Engström, E.; Storey, M.-A. "The Design Science Paradigm as a Frame for Empirical Software Engineering". Heidelberg, Germany: Springer, 2020, 1 ed., vol. 1, chap. 5, pp. 127–147.

[229] Runeson, P.; Höst, M. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering", *Empirical Software Engineering*, vol. 14–2, Apr 2009, pp. 131–164.

[230] Sandino, D.; Matey, L. M.; Vélez, G. "Design Thinking Methodology for the Design of Interactive Real-time Applications". In: Proceedings of the International Conference of Design, User Experience, and Usability, Las Vegas, USA: Springer, 2013, pp. 583–592.

[231] Santos, H. R. M.; Alves, C. "A2BP: A Method for Ambidextrous Analysis of Business Process". In: Proceedings of the International Conference on Enterprise Information Systems, Porto, Portugal: Scitepress, 2017, pp. 12.

[232] Sarbazhosseini, H.; Adikari, S.; Keighran, H. "Design Thinking Framework for Project Portfolio Management". In: Proceedings of the International Conference of Design, User Experience, and Usability, Toronto, Canada: Springer, 2016, pp. 1–8.

[233] Scott, S. G.; Bruce, R. A. "Decision-making Style: The Development and Assessment of a New Measure", *Educational and psychological measurement*, vol. 55–5, 10 1995, pp. 818–831.

[234] Sedano, T.; Ralph, P.; Péraire, C. "The Product Backlog". In: Proceedings of the International Conference on Software Engineering, Montreal, Canada: ACM/IEEE, 2019, pp. 200–211.

[235] Seidel, V. P.; Fixson, S. K. “Adopting Design Thinking in Novice Multidisciplinary Teams: The Application and Limits of Design Methods and Reflexive Practices”, *Journal of Product Innovation and Management*, vol. 30–1, Oct 2013, pp. 19–33.

[236] Senapathi, M.; Drury-Grogan, M. L. “Systems Thinking Approach to Implementing Kanban: A Case Study”, *Journal of Software: Evolution and Process*, vol. 33–4, Oct 2021, pp. 22–32.

[237] Senft, B.; Rittmeier, F.; Fischer, H.; Oberthür, S. “A Value-Centered Approach for Unique and Novel Software Applications”. In: Proceedings of the International Conference on Human-Computer Interaction, Orlando, USA: Springer, 2019, pp. 366–384.

[238] Shah, K.; Salunke, A.; Dongare, S.; Antala, K. “Recommender Systems: An Overview of Different Approaches to Recommendations”. In: Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems, Coimbatore, India: IEEE, 2017, pp. 1–4.

[239] Sharma, R.; Rani, S.; Tanwar, S. “Machine Learning Algorithms for building Recommender Systems”. In: Proceedings of the International Conference on Intelligent Computing and Control Systems, Madurai, India: IEEE, 2019, pp. 785–790.

[240] Shefali Gupta, M. D. “A Recommendation System: Trends and Future”, *International Journal of Engineering and Advanced Technology*, vol. 08–0, Jul 2019, pp. 1361–1364.

[241] Smite, D.; Moe, N. B.; Levinta, G.; Floryan, M. “Spotify Guilds: How to Succeed with Knowledge Sharing in Large-scale Agile Organizations”, *IEEE Software*, vol. 36–2, Feb 2019, pp. 51–57.

[242] Snowden, D. J.; Boone, M. “A Leader’s Framework for Decision Making”, *Harvard Business Review*, vol. 1, Nov 2007, pp. 1–8.

[243] Sohaib, O.; Solanki, H.; Dhaliwa, N.; Hussain, W.; Asif, M. “Integrating Design Thinking into Extreme Programming”, *Journal of Ambient Intelligence and Humanized Computing*, vol. 10–6, Jun 2019, pp. 2485–2492.

[244] Sosnin, P. “Question-answer Analysis in Design Thinking at the Conceptual Stage of Developing a System with a Software”. In: Proceedings of the International Conference on Information, Intelligence, Systems Applications, Larnaca, Cyprus: IEEE, 2017, pp. 1–6.

[245] Souza, A.; Ferreira, B.; Valentim, N.; Correa, L.; Marczak, S.; Conte, T. "Supporting the Teaching of Design Thinking Techniques for Requirements Elicitation Through a Recommendation Tool", *IET Software*, vol. 14, Dec 2020, pp. 693–701.

[246] Souza, C. L. d. C.; Silva, C. T. "Use of Design Thinking in Requirements Elicitation of Mobile Learning in Virtual Environments". In: Proceedings of the Workshop on Requirements Engineering, Pucón, Chile: PUC-Rio, 2014, pp. 14.

[247] Stol, K.-J.; Ralph, P.; Fitzgerald, B. "Grounded Theory in Software Engineering Research: a Critical Review and Guidelines". In: Proceedings of the International Conference on Software Engineering, Austin, USA: IEEE, 2016, pp. 120–131.

[248] Strauss, A.; Corbin, J. "Discovery of Grounded Theory". New Jersey, USA: Aldine Transaction, 1967, vol. 1, 284p.

[249] Suzianti, A.; Wulandari, A. D.; Yusuf, A. H.; Belahakki, A.; Monika, F. "Design Thinking Approach for Mobile Application Design of Disaster Mitigation Management". In: Proceedings of the Asia Pacific Information Technology Conference, Bali Island, Indonesia: ACM, 2020, pp. 29–33.

[250] Taghavi, M.; Bentahar, J.; Bakhtiyari, K.; Hanachi, C. "New Insights Towards Developing Recommender Systems", *Computer Journal*, vol. 61–1, Jun 2018, pp. 319–348.

[251] Tannian, M. F. "Embracing Quality with Design Thinking", *The Future of Software Quality Assurance*, vol. 1–1, Mar 2020, pp. 161–174.

[252] Tarnowska, K. A.; Ras, Z. W.; Jastreboff, P. J. "Recommender Solutions Overview". In: *Decision Support System for Diagnosis and Treatment of Hearing Disorders*, 1 ed., Heidelberg, Germany: Springer, 2017, vol. 1, pp. 17–33.

[253] Tellioğlu, H. "Models as Bridges from Design Thinking to Engineering". In: Proceedings of the International Conference on Interfaces and Human Computer Interaction, Madeira, Portugal: IADIS, 2016, pp. 1–4.

[254] Tello, E. A.; Castañón-Puga, M.; Gaxiola Pacheco, C.; Evans, R. D. "Applied Decision Making in Design Innovation Management", *Studies in Systems, Decision and Control*, vol. 209–1, May 2019, pp. 95–114.

[255] Thorat, P. B.; Goudar, R.; Barve, S. "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System", *International Journal of Computer Applications*, vol. 110–4, Jan 2015, pp. 31–36.

[256] Tonetto, L. M.; Kalil, L. L.; Melo, W. V.; Schneider, D. D. G.; Stein, L. M. "O Papel das Heurísticas no Julgamento e na Tomada de Decisão sob Incerteza", *Estudos de Psicologia*, vol. 23–2, Jun 2006, pp. 181–189.

[257] Travassos, G. H.; Gurov, D.; Amaral, E. "Introdução à Engenharia de Software Experimental". Rio de Janeiro, Brasil: UFRJ, 2002, 1 ed., vol. 9, 52p.

[258] Tschimmel, K. "Design Thinking as an Effective Toolkit for Innovation". In: Proceedings of the International Society for Professional Innovation Management, Barcelona, Spain: ISPIM, 2012, pp. 1–20.

[259] Unnikrishnan, R.; Hebert, M. "Measures of similarity". In: Proceedings of the Workshop on Applications of Computer Vision, Washington D.C., USA: IEEE, 2005, pp. 1–10.

[260] Valentim, N.; Silva, W.; Conte, T. "The Students' Perspectives on Applying Design Thinking for the Design of Mobile Applications". In: Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training Track, Buenos Aires, Argentina: IEEE, 2017, pp. 1–10.

[261] Vetterli, C.; Brenner, W.; Uebernickel, F.; Petrie, C. "From Palaces to Yurts: Why Requirements Engineering Needs Design Thinking", *Internet Computing*, vol. 17, Mar 2013, pp. 91–94.

[262] Vianna, M. "Design Thinking: Business Inovation". Rio de Janeiro, Brasil: MJV Technology & Innovation, 2012, 1 ed., vol. 1, 195p.

[263] Viera, A. J.; Garrett, J. M.; et al.. "Understanding Interobserver Agreement: the Kappa Statistic", *Family Medicine*, vol. 37–1, Jun 2005, pp. 360–363.

[264] Wadikar, D.; Kumari, N.; Bhat, R.; Shirodkar, V. "Book Recommendation Platform using Deep Learning Book Recommendation Platform using Deep Learning", *International Research Journal of Engineering and Technology*, vol. 07–1, Jul 2020, pp. 6764–6770.

[265] Waidelich, L.; Richter, A.; Kölmel, B.; Bulander, R. "Design Thinking Process Model Review. A Systematic Literature Review of Current Design Thinking Models in Practice". In: Proceedings of the International Conference on Engineering, Technology and Innovation, Stuttgart, Germany: IEEE, 2018, pp. 1–9.

[266] Walker, R. J. "Recent Advances in Recommendation Systems for Software Engineering". In: Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Amsterdam, Netherlands: Springer, 2013, pp. 372–381.

[267] Wang, Y.; Ruhe, G. "The Cognitive Process of Decision Making", *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 1–2, Aug 2007, pp. 73–85.

[268] Wieringa, R. “Design Science Methodology for Information Systems and Software Engineering”. Heidelberg, Germany: Springer, 2014, 1 ed., *1*, vol. 1, 332p.

[269] Wieringa, R.; Maiden, N.; Mead, N.; Rolland, C. “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion”, *Requirements Engineering*, vol. 11–1, Mar 2006, pp. 102–107.

[270] Wohlin, C.; Höst, M.; Henningsson, K. “Empirical Research Methods in Software Engineering”, *Empirical Methods and Studies in Software Engineering: Experiences from ESERNET*, vol. 2765–1, Jan 2003, pp. 7–23.

[271] Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M. C.; Regnell, B.; Wesslén, A. “Experimentation in Software Engineering”. Heidelberg, Germany: Springer, 2012, 1 ed., *1*, vol. 1, 236p.

[272] Yin, R. K. “Case Study Research and Applications: Design and Methods”. Thousand Oaks, USA: Sage, 2018, 5 ed., vol. 1, 350p.

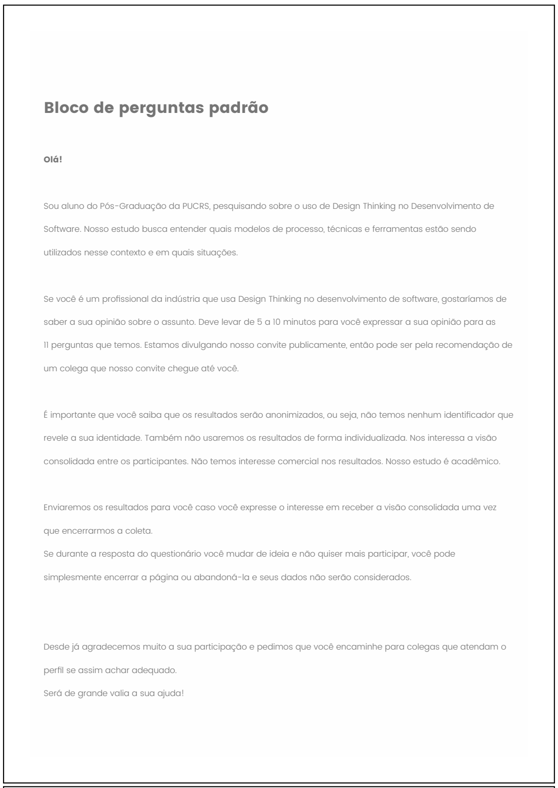
[273] Zachary, W.; Wherry, R.; Glenn, F.; Hopson, J. “Decision Situations, Decision Processes, and Decision Functions: Towards a Theory-based Framework for Decision-aid Design”. In: Proceedings of the Conference on Human Factors in Computing Systems, Gaithersburg, USA: ACM, 1982, pp. 355–358.

[274] Zeebaree, M.; Aqel, M. “A Comparison Study Between Intelligent Decision Support Systems and Decision Support Systems”, *The ISC International Journal of Information Security*, vol. 11–3, Jan 2019, pp. 187–194.

[275] Zhiqin, J.; Jian, Z. “Research Progress and Development Trend Evolution of Recommendation System”. In: Proceedings of the International Symposium on Computational Intelligence and Design, Hangzhou, China: IEEE, 2019, pp. 98–103.

APPENDIX A – DESIGN THINKING IN SOFTWARE DEVELOPMENT: QUESTIONNAIRE

This appendix shows the questionnaire used for collecting data from DT practitioners in the Survey study (Section 4.2).



Bloco de perguntas padrão

Olá!

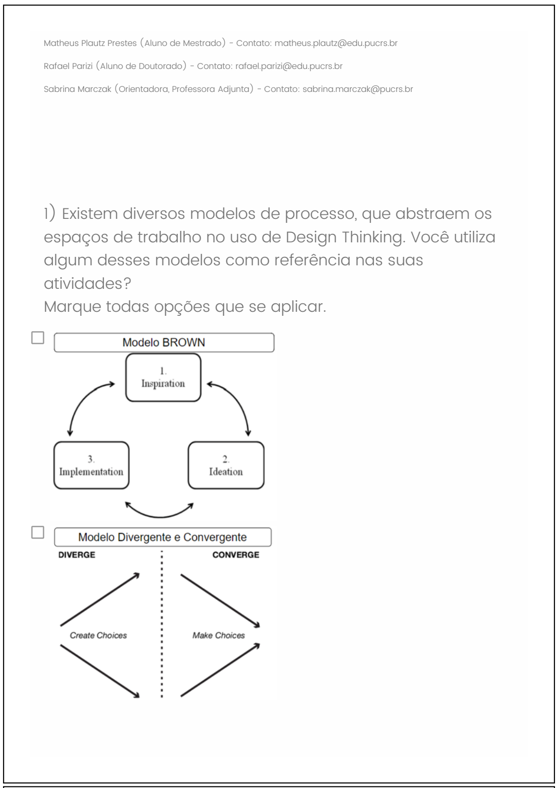
Sou aluno do Pós-Graduação da PUCRS, pesquisando sobre o uso de Design Thinking no Desenvolvimento de Software. Nosso estudo busca entender quais modelos de processo, técnicas e ferramentas estão sendo utilizados nesse contexto e em quais situações.

Se você é um profissional da indústria que usa Design Thinking no desenvolvimento de software, gostaríamos de saber a sua opinião sobre o assunto. Deve levar de 5 a 10 minutos para você expressar a sua opinião para as 11 perguntas que temos. Estamos divulgando nosso convite publicamente, então pode ser pela recomendação de um colega que nosso convite chegue até você.

É importante que você saiba que os resultados serão anonimizados, ou seja, não temos nenhum identificador que revele a sua identidade. Também não usaremos os resultados de forma individualizada. Nos interessa a visão consolidada entre os participantes. Não temos interesse comercial nos resultados. Nosso estudo é acadêmico.

Enviaremos os resultados para você caso você expresse o interesse em receber a visão consolidada uma vez que encerrarmos a coleta.
Se durante a resposta do questionário você mudar de ideia e não quiser mais participar, você pode simplesmente encerrar a página ou abandoná-la e seus dados não serão considerados.

Desde já agradecemos muito a sua participação e pedimos que você encaminhe para colegas que atendam o perfil se assim achar adequado.
Será de grande valia a sua ajuda!

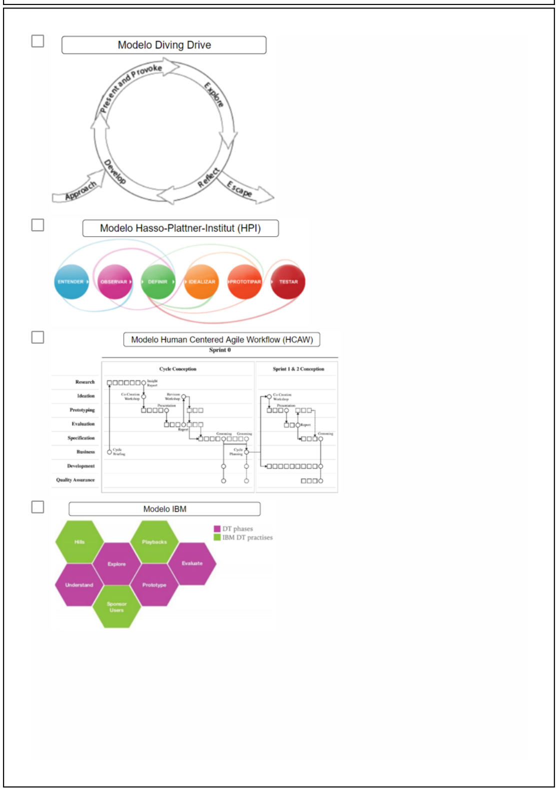


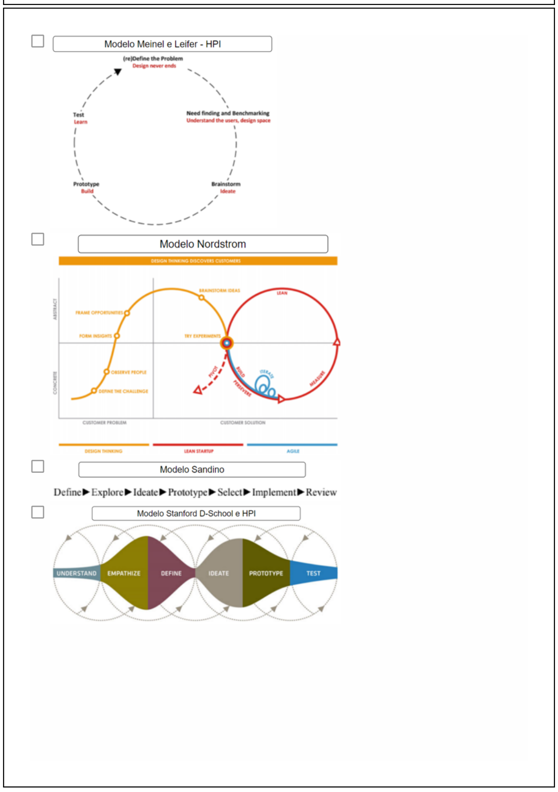
Matheus Plautz Prestes (Aluno de Mestrado) - Contato: matheus.plautz@edu.pucrs.br

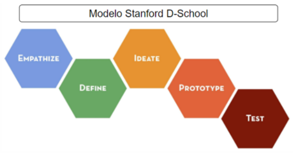
Rafael Parizi (Aluno de Doutorado) - Contato: rafael.parizi@edu.pucrs.br

Sabrina Marczak (Orientadora, Professora Adjunta) - Contato: sabrina.marczak@pucrs.br

1) Existem diversos modelos de processo, que abstraem os espaços de trabalho no uso de Design Thinking. Você utiliza algum desses modelos como referência nas suas atividades?
Marque todas opções que se aplicar.





☐ 

☐ Outro?

☐ Não sei informar.

2) Diversas técnicas podem ser usadas para apoiar o uso de Design Thinking. Quais técnicas você costuma utilizar?
Marque todas as opções que se aplicar.

- ☐ 5w2h
- ☐ A day in the life
- ☐ Behaviour Archaeology
- ☐ Blueprint
- ☐ BodyStorming
- ☐ Brainstorming
- ☐ Brainwriting
- ☐ Business Model Canvas (BMC)
- ☐ Card Sort
- ☐ Character Profiles
- ☐ Co-creation Workshops
- ☐ Cognitive and Behaviour Map
- ☐ Conceptual mind maps
- ☐ Costumer Journey Maps
- ☐ Desk Research
- ☐ Empathy Maps
- ☐ Error Analysis
- ☐ Extreme User interviews
- ☐ fishbowl
- ☐ Five Whys?
- ☐ Fly on the wall
- ☐ How-Might-We-Question
- ☐ I like I wish
- ☐ Insight cards
- ☐ Interview
- ☐ Living Labs
- ☐ Mind Mapping
- ☐ Mockups
- ☐ Narration
- ☐ Observation
- ☐ Personal Inventory
- ☐ Personas
- ☐ Positioning Matrix
- ☐ Rapid Ethnography
- ☐ Role-Playing
- ☐ Scenarios Mapping
- ☐ Shadowing
- ☐ Social Network Mapping
- ☐ Stakeholder Mapping
- ☐ Storyboard
- ☐ Storytelling
- ☐ Surveys Questionnaires
- ☐ Try it yourself
- ☐ Feedback grid
- ☐ User story
- ☐ World Café
- ☐ Não utilizo esses técnicas
- ☐ Outras?

3) Como você geralmente decide quais técnicas utilizar?
Marque todas as opções que se aplicar.

- ☐ Baseado na minha experiência prévia
- ☐ Por indicação de um colega
- ☐ Recomendação pela minha empresa
- ☐ Escolho as técnicas de acordo com a espaço/etapa do DT, onde cada espaço/etapa possui suas próprias técnicas
- ☐ Quando a técnica se enquadra na minha necessidade
- ☐ Já tenho o meu catálogo de técnicas que sempre utilizo
- ☐ Depende muito do contexto que vou utilizar
- ☐ Geralmente preciso estudar as técnicas pois nunca sei qual é a melhor para o momento
- ☐ Outro motivo?

4) Em uma escala de 0 (Nenhuma dificuldade) a 10 (Dificuldade extrema), qual o grau de dificuldade que você sente em decidir quais técnicas utilizar em uma determinada situação?

0 1 2 3 4 5 6 7 8 9 10

Grau de Dificuldade

5) Você utiliza algum software (ou sistema computacional, como preferir denominar) para apoiar o uso das técnicas de Design Thinking?
Marque todas as opções que se aplicar.

- ☐ Smaply (Personas)
- ☐ Circle (Stakeholder Map)
- ☐ Touchpoint Dashboard (Customer Journey)
- ☐ Creately (Service Blueprint)
- ☐ Strategyzer (Business Model Innovation)
- ☐ Axure RP (Rapid Prototype)
- ☐ Outras?

Bloco 2

6) Para qual fim você usa Design Thinking no desenvolvimento de software?
Marque todas as opções que se aplicar.

- ☐ Ganhar a empatia dos usuários
- ☐ Para o desenvolvimento de jogos

- [ ] Entender e especificar os requisitos
- [ ] Explorar e entender o problema
- [ ] Para gerar ideias e soluções
- [ ] Para gerenciar projetos
- [ ] Para criar ideias inovadoras
- [ ] Aproximar a equipe de desenvolvimento com o cliente
- [ ] Melhorar a satisfação do cliente
- [ ] Para reduzir as incertezas
- [ ] Fácil relacionamento com os métodos ágeis
- [ ] Validação do software
- [ ] Outro?

7) Quais são os cenários de uso comuns onde você usa o Design Thinking?
Marque todas as opções que se aplicar.

- [ ] Com equipes multidisciplinares
- [ ] Criação de produtos/softwares inovadores
- [ ] Dentro de um processo desenvolvimento de software diariamente/semanalmente, acompanhado por toda equipe (do cliente até o desenvolvedor
- [ ] Inovação como um todo, desde o processo de desenvolvimento até o software
- [ ] Criar cocriação entre os participantes do projeto
- [ ] Mudanças e melhorias no desenvolvimento de software
- [ ] Utilizado em parceria com os métodos ágeis (Lean, Scrum)

- [ ] Outro?

8) Na sua experiência de uso de Design Thinking no desenvolvimento de software, o que você apontaria como benefícios ou pontos positivos trazidos pela adoção da abordagem?

9) E quais seriam as dificuldades ou os pontos negativos?

Bloco 3

10) Qual a sua experiência, em anos, nos seguintes critérios?

	Menos de 1 ano de experiência	De 1 a 3 anos	De 4 a 7 anos	Acima de 8 anos
No uso de Design Thinking no desenvolvimento de software	○	○	○	○

APPENDIX B – DESIGN THINKING TECHNIQUES IN HELIUS

This appendix summarizes the DT techniques that we identified in Literature (see Section 4.1) in 3 tables, as follows:

- Table B.1 shows the list of Design Thinking techniques stored in Helius' database.
- Table B.2 shows the Design Thinking techniques classified according to the DT model proposed by Brown (2018) [36]. The working spaces are Inspiration, Ideation and Implementation.
- Table B.3 shows the Design Thinking techniques classified by goal.

Next, Figure B.1 shows a JSON object representing a DT technique, while Figure B.2 represents a graphical version of the template of a DT technique in Helius[1].

Table B.1: Design Thinking Techniques registered in Helius

Problem space	**Solution space**
A Day In The Life	Acceptance Test
AEIOU	Bodystorming
Affinity Diagram	Brainstorming
As-Is Scenario Map	Brainwriting
Behaviour Archaeology	Cost Benefit Matrix
Behaviour Map	Conceptual Mind Mapping (Cognitive)
Blueprint	Crazy Eight's
Bodystorming	Dot Voting
Brainstorming	Feedback Grid
Business Model Canvas	Fishbowl (Fishbone)
Card Sorting	Ideas Menu
Conceptual Mind Mapping (Cognitive)	I Like I Wish
Customer Journey Map	Insight Cards
Desk Research	Interview
Empathy Map	Mind Mapping
Ethnography	How - Now - Wow Matrix
Exploratory Research	Observation
Fishbowl (Fishbone)	Pitch Presentation
Five Human Factors	Positioning Matrix
Five Why's	Paper Prototype
Fly On The Wall	Questionnaires
Focus Group	Role Playing
Generative Sessions	Sailboat (Retrospective)
How Might We Questions	Service Walkthrough
Interview	Storyboarding
Observation	Storytelling
Personas	Survey
Questionnaires	Try It Yourself
Shadowing	World Cafe
SIPOC Review	Yes, But/ Yes And Then... Game
Social Network Mapping	5w2h
Stakeholder Map	
Survey	
Trend Matrix	

[1] We used the tool https://jsoncrack.com/editor to generate the graph.

Table B.2: DT techniques according to 3 working spaces by Brown

Techniques classification - Brown		
Inspiration	**Ideation**	**Implementation**
A Day In The Life	Bodystorming	Acceptance Test
AEIOU	Brainstorming	Cost Benefit Matrix
Affinity Diagram	Brainwriting	Feedback Grid
Behaviour Archaeology	Crazy Eight's	Fishbowl (Fishbone)
Behaviour Map	Dot Voting	I Like I Wish
Blueprint	Ideas Menu	Observation
Bodystorming	How - Now - Wow Matrix	Pitch Presentation
Brainstorming	Positioning Matrix	Paper Prototype
Business Model Canvas	Role Playing	Questionnaires
Card Sorting	Try It Yourself	Sailboat (Retrospective)
Conceptual Mind Mapping (Cognitive)	World Cafe	Service Walkthrough
Customer Journey Map	Yes, But/ Yes And Then... Game	Storyboarding
Desk Research	Conceptual Mind Mapping	Storytelling
Empathy Map		Survey
Ethnography		Interview
Exploratory Research		
Fishbowl (Fishbone)		
Five Human Factors		
Five Why's		
Fly On The Wall		
Focus Group		
Generative Sessions		
How Might We Questions		
Mind Mapping		
Observation		
Personas		
Questionnaires		
Shadowing		
SIPOC Review		
Social Network Mapping		
Stakeholder Map		
Survey		
Trend Matrix		
Interview		
As-Is Scenario Map		
Blueprint		
5w2h		
Insight Cards		

Table B.3: Design Thinking techniques classified by goal

Goals	Techniques
Inspiration	
Discovering the project stakeholders	Observation
Knowing more about users (NO interaction, user's workplace)	Behavior Archaeology, Fly on the wall, AEIOU, Five human factors, Observation, Shadowing, Social network mapping
Knowing more about users (WITH interaction, user's workplace)	Interview, Ethnography, Questionnaire, A day in the life, Five whys, How might we
Knowing more about users (WITH interaction, everywhere)	Interview, Exploratory research, Questionnaire, Five whys, Generative sessions, How might we
Discovering the users	Exploratory research, Questionnaire, Social Network mapping
Discovering problem details or information	Desk research, Focus group, Observation
Understanding the users-organization relationship	User Journey Map, 5W2H, Empathy Map
Organizing the stakeholders' information	Motivation Map, Stakeholder map
Representing the user's Behavior graphically	Behavior Map
Summarizing the user's behavior (motivations, expectations, needs)	Personas, As-is scenario map
Comprehending the user's interactions with similar solutions	Contact point map
Representing cognitive structures and patterns	Cognitive mapping
Recording the data collected in a structured way to better understand the data	Insight cards
Organizing the data collected to communicate them to the team	Conceptual mapping
Clustering the insights gathered with the data collected, making relations by affinity, similarity, and priority	Affinity diagram, Card sorting, Trend matrix
Modeling the service as it is offered by a company	Service Blueprint, SIPOC Review
Modeling the relationship between the business strategies and value to identify potential solutions	Business model canvas
Ideation	
Discussing and speaking various ideas based on the information collected	Brainstorming, Fishbowl, World cafe, Yes, but/ Yes, and then..
Discussing and writing various ideas based on the information collected	Brainwriting
Discussing and drawing various ideas based on the information collected	Group sketching (or sketching), Crazy eights
Discussing and simulating various ideas based on the information collected	Bodystorming, Role playing
Making associations between the ideas generated	Mental mapping, Ideas menu, how-now-wow matrix, Conceptual mind mapping
Experimenting other solutions to gather new ideas	Try it yourself
Implementation	
Validating ideas	Observation, Acceptance test, Feedback grid, Cost-benefit matrix, Fishbowl, I Like, I wish, Questionnaire, Sailboat, Service Walk-through, Interview
Presenting ideas in detail	Storyboard, prototyping
Communicating the ideas to the users	Storytelling, Pitch Presentation

modelo.json 〉 No Selection


```json
{
    "nameT": [
        "",
        ""
    ],
    "description": [
        "",
        ""
    ],
    "howToUse": [
        "",
        ""
    ],
    "whenToUse": [
        "",
        ""
    ],
    "name": "",
    "projects": [],
    "evaluate": [],
    "characteristics": {
        "materials": [
            "",
            ""
        ],
        "duration": [
            "",
            ""
        ],
        "location": [
            "",
            ""
        ],
        "participants": [
            "",
            ""
        ],
        "classification": [
            "",
            ""
        ],
        "reference": [
            ""]
    },
    "models": [
        {
            "name": ["Brown", "Brown"],
            "etapas": ["", ""],
            "label": ["", ""]
        },
        {
            "name": ["Geral", "General"],
            "etapas": ["Espaço do problema"],
            "label": ["Problem space"]
        }
    ],
    "rating": 0,
    "count": 0,
    "image": ""
}
```


Figure B.1: Template of a DT technique in the JSON format

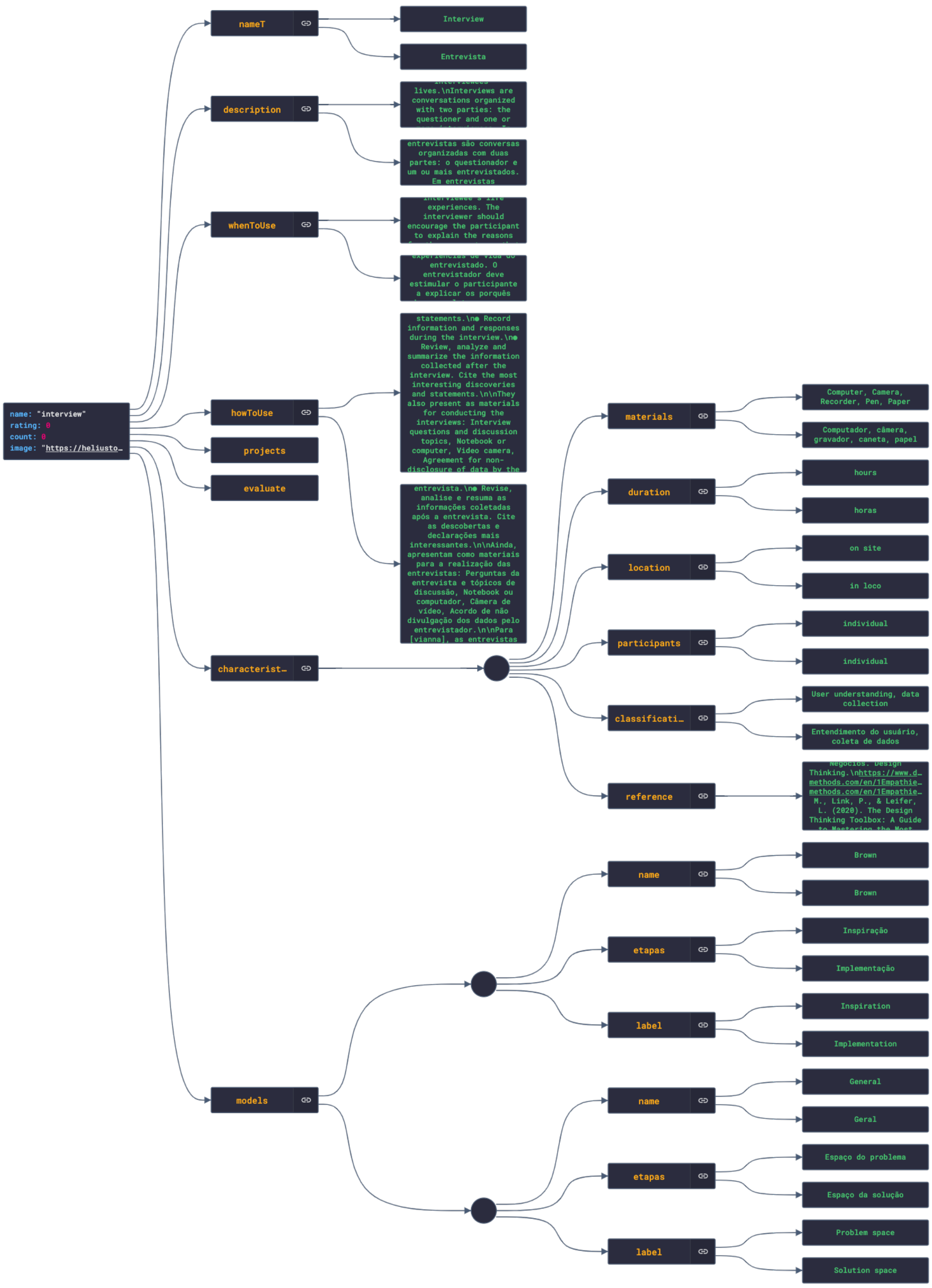


Figure B.2: A DT technique graphically represented

APPENDIX C – DECISION-MAKING OF DESIGN THINKING TECHNIQUES: PROTOCOL OF AN INTERVIEW-BASED STUDY

This appendix illustrates the questionnaire for collecting data from DT facilitators in the interview-based study (in Portuguese)[1].

Instrumento	Roteiro semiestruturado
Formato	• Roteiro semiestruturado para entrevista presencial ou via videoconferência. • Com duração prevista de 30-60 minutos, incluindo a leitura do TCLE e introdução do estudo que está sendo realizado.
Objetivo	• Identificar como os profissionais do desenvolvimento de software decidem quanto à seleção de técnicas de *Design Thinking* (DT), quais os critérios e estratégias de decisão utilizadas para a seleção das técnicas de DT.
Participantes	• Critério de inclusão: profissionais de Tecnologia da Informação (TI) que participam de equipes de desenvolvimento de software e que sejam responsáveis por atividades que envolvam o *Design Thinking* no desenvolvimento de software. • Critério de exclusão: menores de idade e/ou profissionais que não sejam responsáveis selecionar técnicas de DT para desenvolvimento de software
Dados Demográficos	Perfil do participante • Nome (lembrar confidencialidade): • Formação acadêmica (nível e curso): • Cargo atual: • Anos de experiência na indústria de tecnologia: • Anos de trabalho nesta empresa: • Contexto dos projetos que você costuma trabalhar: • Anos de experiência você tem com *Design Thinking* no desenvolvimento de software: • Formação/capacitação sobre *Design Thinking* voltado para o desenvolvimento de software (por exemplo: algum curso oferecido internamente por sua empresa, cursos independentes ofertados em escolas especializadas de formação): Perfil da empresa • Nome da empresa (lembrar confidencialidade): • Ano de início das atividades da empresa: • Número de colaboradores da empresa: ○ ( ) 1-10 pessoas ○ ( ) 11-50 pessoas ○ ( ) 51-100 pessoas ○ ( ) 101-300 pessoas ○ ( ) 301 – 1000 pessoas ○ ( ) mais de 1000 pessoas • Modelo de negócio da empresa/ramo de atividades: • Local dos clientes da empresa: ○ ( ) dentro do estado da sede ○ ( ) mais de um estado brasileiro ○ ( ) em mais de um país

Questões

Questões sobre experiência com técnicas de DT e o processo de seleção das técnicas

Experiência com técnicas

1. Qual foi (é) o seu papel durante a sessão (ões) de DT?
   - ( ) facilitador de DT
   - ( ) participante
   - ( ) outro. Qual? _____________
2. Você já foi responsável pela seleção e organização das técnicas a serem aplicadas em uma sessão de DT?
   - ( ) sim
   - ( ) não
3. Quais técnicas você costuma utilizar?

Elementos de tomada de decisão:

4. Quais elementos você considera ao realizar a seleção de técnicas associadas ao DT?
5. Você segue algum modelo de processo de DT? Se sim, o quão importante este modelo é para a seleção das técnicas?
6. Quais as fontes de consulta você utiliza para descobrir técnicas? Essas fontes auxiliam você a decidir sobre quais técnicas usar?
7. Você avalia as técnicas após utilizá-las? Como?
8. Essa avaliação colabora para a tomada de decisão sobre as técnicas? De que forma?

Estratégias e critérios de seleção Decision-making taxonomy – Wang e Ruhe (2007)

9. Wang e Ruhe (2007) estudaram o processo cognitivo de tomada de decisão e propuseram uma taxonomia de tomada de decisão, com categorias de decisão, estratégias e critérios. Portanto, tomando como base um questionário de coleta de tomada de decisão proposto por Scott e Bruce (1995), adaptado para a taxonomia mais recente, abaixo temos uma lista de estratégias e critérios de decisão. Usando a escala tipo-Likert de 5 elementos de frequência (Nunca, raramente, às vezes, frequentemente, sempre), você pode informar como é a sua tomada de decisão de técnicas frente a estas estratégias.

Critério	Descrição	Likert
Arbitrário	Seleciona técnicas considerando aquelas que lhe trarão menor esforço para decidir ou ainda aquelas que você já tenha selecionado previamente?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Preferência	Seleciona técnicas com base em tendências de uso ou a partir da expectativa de resultado gerado pela técnica?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Senso comum	Seleciona técnicas com base em conhecimento geral de que tais técnicas são boas alternativas para utilizar?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Tentativa e erro	Seleciona técnicas a partir de tentativa e erro, ou seja, vai testando técnicas e verificando se funcionam ou não?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Experimento	Seleciona técnicas a partir da experimentação, registrando as que proporcionaram sucesso?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Experiência	Seleciona técnicas a partir da sua experiência de uso e/ou em experiência de outros profissionais que também usaram as técnicas?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Consultoria	Seleciona técnicas consultando (direta ou indiretamente) outros profissionais que usem DT?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Estimativa	Seleciona técnicas a partir de uma avaliação aproximada do resultado, mesmo não tendo utilizado-as previamente?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Princípios	Seleciona técnicas a partir da consulta a teorias científicas que façam referências às técnicas?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Ética	Seleciona técnicas considerando julgamentos filosóficos e crenças que você tenha?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Representatividade	Seleciona técnicas a partir das características que representam tais técnicas?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Disponibilidade	Seleciona técnicas com base naquelas que mais rapidamente vêm à sua mente?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Âncora	Seleciona técnicas a partir uma análise inicial do que você precisa e adapta outras técnicas conforme necessidades surgem?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Custo mínimo	Seleciona técnicas a partir da estimativa de custo para utilização da mesma, visando utilizar as de menor custo?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Máximo benefício	Seleciona técnicas a partir da estimativa de benefício que a técnica pode trazer, visando utilizar as de maior benefício?	1 – Nunca 2 – Raramente 3 – As vezes
		4 – Frequentemente 5 - Sempre
Máxima utilidade	Seleciona técnicas considerando a relação custo-benefício estimada a partir da utilização da técnica?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Eventos interativos	Seleciona técnicas considerando os eventos que podem ocorrer a partir da utilização de tais técnicas?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Jogos	Seleciona técnicas fazendo análise de ganhos e perdas a partir da utilização de tais técnicas, ou pela necessidade de resolução de conflitos (discordâncias) entre os participantes?	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre
Matriz de decisão	Seleciona técnicas de DT a partir da análise da inter-relação entre técnicas, como algo do tipo: "Se eu seleciono a técnica X, depois poderei selecionar a técnica Y, ..."	1 – Nunca 2 – Raramente 3 – As vezes 4 – Frequentemente 5 - Sempre

10. Você considera que conforme você foi ganhando experiência com uso de DT, diferentes estratégias de tomada de decisão foram usadas?
    - Sim ( ) _ Por que?
    - Não ( )

Dificuldade para tomada de decisão

11. Em uma escala de 1 a 10, sendo 1 fácil e 10 difícil, como você classifica a dificuldade decidir quais técnicas selecionar?
12. O que te levou a atribuir esse grau de dificuldade para selecionar técnicas?

Recursos de apoio à tomada de decisão:

13. Você utiliza algum recurso computacional de apoio à tomada de decisão para seleção de técnicas? Se sim, qual é o recurso?
14. E como este recurso colabora com a tomada de decisão?

Questões de fechamento

Mais alguma observação sobre o tema, que gostaria de relatar?

[1]The full research protocol can be accessed online at: https://github.com/rafaelparizi/PHD_repository/blob/main/README.md.

APPENDIX D – HELIUS' RECOMMENDATIONS ALGORITHM

This appendix describes a tertiary literature review that we conducted to identify the algorithm to implement our DT techniques recommendation module, and the design and implementation of the DT techniques recommendation module in Helius. The module is responsible for recommending DT techniques according to different recommendation mechanisms. We used the K-nearest Neighbor (KNN) algorithm for identifying similar users and for recommending DT techniques based on their tastes to a Helius user who is requesting DT technique recommendations.

The remaining of this appendix is structured as follows: Section D shows a tertiary literature review on recommendation algorithms in order to identify the algorithms to support the recommendation of DT techniques. Section D introduces the KNN algorithm. Next, Section D shows how we designed a recommendation module capable of receiving recommendation requisitions and of returning DT techniques that are recommended, and how we implemented the recommendation module in Python in order to make it available for recommending DT techniques in Helius.

Tertiary Literature Review on Recommendation Algorithms

Aiming to enhance our understanding of recommendation systems, what are the recommendation approaches, and the algorithms being used, we conducted a tertiary review of the literature. Our tertiary review followed the method proposed in [130].

Figure D.1 shows the research design and the respective steps we conducted in this tertiary review.

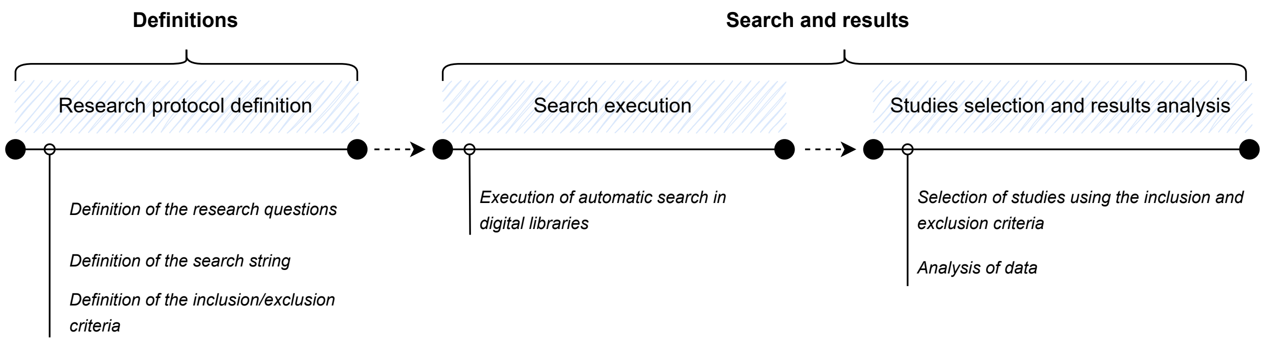


Figure D.1: Tertiary study design

Study Design

In this step, we worked on the definition of the research questions, the search string for searching for publications in digital libraries and the definition of inclusion/exclusion criteria to select relevant publications.

Definition of the research questions

Our goal is to identify what are the recommendation approaches cited in Literature and what algorithms support them in order to implement our DT techniques recommendation module. Therefore, in this study we posed 2 Research Questions:

- *RQ1: What are the existing recommendation approaches?*
- *RQ2: Which algorithms support the recommendation approaches?*

Definition of the search string

In order to search for publications related to the topic, we used the following digital libraries: ACM Digital Library, IEEE Xplore Digital Library and Scopus. We consider that for a tertiary study these digital databases represent relevant sources of studies related to Recommendation Systems.

Table D.1 shows the terms that we used for generating a search string. We followed Gasparic *et al*. (2016) [90] to define terms with ID 1 and 2, and we followed Oliveira *et al*. (2018) [187] to define the terms with ID 3 to 12. Then, we compiled the terms using the Boolean operators *OR* and *AND*, resulting into the following search string:

("recommendation systems" OR "recommendation system" OR "recommender systems" OR "recommender system") AND ("literature review" OR "overview" OR "literature" OR "meta-analysis" OR "past studies" OR "in-depth survey" OR "subject matter expert" OR "analysis of research" OR "empirical body of knowledge" OR "overview of existing research" OR "body of published research")

Definition of the inclusion/exclusion criteria

Table D.2 presents the inclusion and exclusion criteria for studies that we defined for this tertiary review. These criteria serve to select papers that make it possible

Table D.1: Search string terms for the tertiary study

Id	term
1	*recommendation systems*, *recommendation system*
2	*recommender systems*, *recommender system*
3	*literature review*
4	*literature*
5	*meta-analysis*
6	*past studies*
7	*in-depth survey*
8	*subject matter expert*
9	*analysis of research*
10	*empirical body of knowledge*
11	*overview of existing research*
12	*body of published research*
A	*{1 OR 2 }*
B	*{ 3 OR 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11 OR 12 }*
C	*{A AND B}*

Table D.2: Inclusion and exclusion criteria for the tertiary study

Type	ID	Description
Inclusion	IC1	SR-related articles that are literature reviews, or mappings, or algorithms from the literature
	EC1	Publications that are not RS-related or that report SR studies for specific areas for computing
	EC2	Publications that report on primary studies
Exclusion	EC3	Publications that are duplicates
	EC4	Publications not available for download
	EC5	Publications not written in English
	EC6	Publications not *peer-reviewed*
	EC7	Secondary studies not illustrating primary studies reviewed

to answer the defined research questions. In this tertiary review, we sought to select secondary studies, *surveys* and *maps of the literature* related to SRs, comparative studies on the algorithms used and assessment methods (Inclusion Criterion 1 - CI1).

In contrast, exclusion criteria were added to disregard unavailable or duplicate studies (CE4, CE3), and studies presented in other languages, restricting to English only (CE5). Primary studies were also disregarded (CE2), with the intention of only considering secondary studies that consolidated the primary studies (CE7). In addition, studies that were not peer-reviewed (CE6), and those that reported on SR for specific areas other than computing, were not considered.

Execution of the search

Next, we run the search string in the digital libraries. Table D.3 presents the results of the searches. We discarded unavailable and duplicate studies resulting in a total of 1,633 publications. Figure D.2 shows the process for selecting publications that we performed in this study.

In this step, we performed the publication selection process in two iterations, reading the title, abstract, and keywords of the publications returned by the digital libraries. Two researchers read the articles. We applied the inclusion and exclusion criteria.

In Iteration 1, we discarded 1,258 studies, which were not related to RSs, presented research on a specific area, or represented primary studies (CE1, CE2). In a second iteration, we reanalyzed the remaining publications and discarded another 301 according to the exclusion criteria.

Following the publication selection process, we performed a full read-through of the 74 studies with a high potential of being relevant for answering the research questions. Of these, we discarded 44 as they did not represent literature mappings, literature reviews, *surveys*, or even studies discussing Recommendation Systems or represented primary studies (CI1, CE7). As a result, we selected 30 studies, as presented in Table D.4.

Table D.3: Search execution results

Step	**IEEE**	**ACM**	**Scopus**	**Total**
Initial search	339	329	1561	2,229
Not available	-	-	-	24
Duplicated	-	-	-	572
Available	318	177	1138	1,633

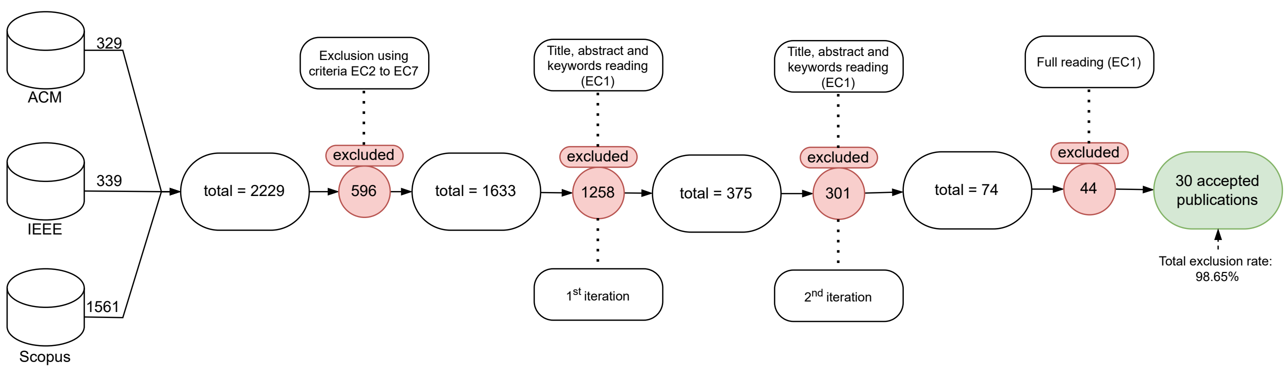


Figure D.2: Selection process of secondary studies

Table D.4: Publications selected in the tertiary study

ID	Title	Authors	Year	Ref.
1	A systematic literature review of multicriteria recommender systems	Monti, Diego and Rizzo, Giuseppe and Morisio, Maurizio	2021	[174]
2	A Literature Review of Recommendation Systems	Bhareti, Kushan and Perera, Shevon and Jamal, Shehan and Pallege, Manul Hiyare and Akash, Vishma and Wiieweera, Sihan	2020	[21]
3	Recommendation systems: Algorithms, challenges, metrics, and business opportunities	Fayyaz, Zeshan and Ebrahimian, Mahsa and Nawara, Dina and Ibrahim, Ahmed and Kashef, Rasha	2020	[79]
4	Recommendation system based on deep learning methods: a systematic review and new directions	Da'u, Aminu and Salim, Naomie	2020	[58]
5	Recommendation system: A systematic overview on methods, issues and solutions	Bhuvanya, R and Kavitha, M	2020	[22]
6	Towards the use of clustering algorithms in recommender systems	Miranda, Leandro and Viterbo, José and Bernardini, Flávia	2020	[171]
7	Advanced Recommendation Systems Through Deep Learning	Khoali, Mohamed and Tali, Abdelhak and Laaziz, Yassin	2020	[128]
8	Machine Learning Algorithms for building Recommender Systems	Sharma, Richa and Rani, Shalli and Tanwar, Sarvesh	2019	[239]
9	Research Progress and Development Trend Evolution of Recommendation System	Zhiqin, Jia and Jian, Zhang	2019	[275]
10	A recommendation system: Trends and future	Shefali Gupta, Meenu Dave	2019	[240]
11	Recommendation system: A literature survey	Mohan Kubendrian, Nishal Pradhan	2019	[173]
12	A survey on data mining techniques in recommender systems	Kumar, Niranjan and Sneha, YS and Mungara, Jitendranath and Prasad, SG Raghavendra	2017	[139]
13	Current trends in collaborative filtering recommendation systems	Amin, Sana Abida and Philips, James and Tabrizi, Nasseh	2019	[11]
14	Deep Learning Based-Recommendation System: An Overview on Models, Datasets, Evaluation Metrics, and Future Trends	Ong, Kyle and Haw, Su-Cheng and Ng, Kok-Why	2019	[188]
15	A state-of-the-art Recommender Systems: An overview on Concepts, Methodology and Challenges	Jariha, Priyanka and Jain, Sanjay Kumar	2018	[118]
16	Recommender systems: An overview of different approaches to recommendations	Shah, Kunal and Salunke, Akshaykumar and Dongare, Saurabh and Antala, Kisandas	2017	[238]
17	A Survey on Data Mining Methods Available for Recommendation System	Liu, Chan and Li, Lun and Yao, Xiaolu and Tang, Lin	2019	[155]
18	Collaborative filtering: Techniques and applications	Mustafa, Najdt and Ibrahim, Ashraf Osman and Ahmed, Ali and Abdullah, Afnizanfaizal	2017	[179]
19	Collaborative filtering recommender system: Overview and challenges	Al-Bashiri, Hael and Abdulgabber, Mansoor Abdullateef and Romli, Awanis and Hujainah, Fadhl	2017	[6]
20	Multi-View Data approaches in Recommender Systems: An Overview: (Invited Paper)	Palomares, Ivan and Kovalchuk, Sergey V	2017	[192]
21	Recommender solutions overview	Tarnowska, Katarzyna A and Ras, Zbigniew W and Jastreboff, Pawel J	2017	[252]
22	Towards an approach of trust-based recommendation system	Gmach, Imen and Melek, Ghenima and Sidhom, Sahbi and Khrifish, Lofi	2015	[95]
23	Recommender systems in light of big data	Almohsen, Khadija A and Al-Jobori, Huda	2015	[10]
24	Recent advances in recommender systems and future directions	Ning, Xia and Karypis, George	2015	[186]
25	Knowledge-based recommendation systems:A survey	Bouraga, Sarah and Jureta, Ivan and Faulkner, Stéphane and Herssens, Caroline	2014	[29]
26	Trust based recommendation systems	Ozsoy, Makbule Gulcin and Polat, Faruk	2013	[190]
27	Recent advances in recommendation systems for software engineering	Walker, Robert J	2013	[266]
28	Recommender systems survey	Bobadilla, Jesús and Ortega, Fernando and Hernando, Antonio and Gutiérrez, Abraham	2013	[26]
29	A literature review and classification of recommender systems research	Park, Deuk Hee and Kim, Hyea Kyeong and Choi, Il Young and Kim, Jae Kyeong	2012	[196]
30	Looking for "good" recommendations: A comparative evaluation of recommender systems	Cremonesi, Paolo and Garzotto, Franca and Negro, Sara and Papadopoulos, Alessandro Vittorio and Turrin, Roberto	2011	[54]

Results analysis

Once we completed the selection of publications through the inclusion and exclusion criteria, we conducted a full reading of the articles in order to identify answers to the research questions. Next, we present the answers to the RQ that we posed to this study:

Q1: What are the existing recommendation approaches?

In question Q1, we aimed to identify the approaches pointed out in the literature for the development of a Recommendation System. Our goal was to learn about which recommendation approaches would be able to recommend DT techniques in Helius. Table D.5 presents the approaches identified from the reviewed studies:

The 2 most cited approaches used in RSs are Collaborative Filtering and Content-based recommendations. Next, we present details about both approaches.

- Collaborative Filtering: it recommends items taking into account the ratings given by users for the items [21]. Collaborative filtering recommender systems calculate similarities among users in the system, recommending items based on their similar patterns [11]. In this way, the system then searches for users who have similar item evaluation behavior to the active user, to explore which items to suggest.

  Monti *et al.* (2021) [174] indicates that collaborative filtering is considered the most popular recommendation technique, as described in the literature. Table D.5 shows that the collaborative filtering approach is the most mentioned in the selected studies in this literature review. Collaborative recommendation systems are widely used by *online* services such as Amazon and Netflix [6].

  To recommend items, systems based on collaborative filtering follow a process that, involves the following set of activities [11]:

Table D.5: Recommendation approaches cited in Literature

Approach	**Ref.**
Collaborative filtering	[174][21][79][58][22][171][128][239][275][240] [173][139][11][188][118][238][155][179][6][192] [252][95][10][186][29][190][266][26][196][54]
Content-based	[174][21][79][58][22][171][128][239][275][240][173][139][11][188] [118][238][155][179][6][192][252] [95][10][186][29][190][266][26][196][54]
Knowledge-based	[174][21][58] [239][240][179][192][252] [29] [190] [266]
Demographic-based	[239][173][192][190] [26]
Ontology-based	[21]
Community-based	[174][239]
Utility-based	[79]
Hybrid approach	[174][21][79][58][22][171][128][239][275][240][173][139][11][188][118] [238][155][179][6][192][252] [95][190][266][26][196]

1. The system represents items and users as a two-dimensional matrix, where i and j correspond to users and items, respectively. Each rating given by a user to an item is the value R_ij. If an item has not been rated by a user, that item has a value of 0.
2. This matrix R is then used to predict the rating of item j provided by user *i* in order to recommend a list of N items that the user might like.

- *Content-based approach:* The content-based approach recommends items by considering the values of the item's attributes. The recommendation engines of this approach analyze the items that the user has previously viewed and use the characteristics of these items to recommend new similar items [21].

  In this approach, the users' rate items and the system uses such rates for looking for items with similar rates [6]. To calculate this similarity, correlation functions such as cosine similarity measure, or *Pearson correlation* can be used. Once the similarities between the items have been calculated, the system proceeds to use the relationship between the items the user has ranked with others in the database to determine which items are most suitable for the active user [173].

  Content-based recommendation systems learn to recommend items similar to those that the user has liked in the past. Thus, recommendations represent the feature similarity between items (rather than the similarity between users as in the collaborative approach) [252].

In addition to the collaborative filtering and content-based recommendation approaches, other cited approaches in the literature are:

- *Knowledge-based approach:* it uses domain knowledge in order to identify a user's preference [240]. Such an approach generates item recommendations from given domain knowledge. The user specifies his or her needs to the system which further compares these needs with its knowledge base and provides the most relevant suggestions, i.e. the items that agree with this knowledge base [239]. For example, recommendation systems for the medical domain use domain knowledge to recommend items [174].

- *Demographic-based approach:* it uses user demographic profiling for item recommendation [174], including user attributes such as age, gender, language, etc [79]. Sharma (2019) [239] states that e-commerce sites like Amazon, eBay, Flipkart, etc. are examples of environments using this approach.

- *Ontology-based approach*: it uses the structure of an ontology to support, for example, content-based systems for generating recommendations. An ontology has the capacity to store many data fields from different sources and takes on a database-like format. Therefore, data from different sources can be processed to make the recommendations [21].

- *Community-based approach*: it is a variation of collaborative filtering-based systems, which search for similar users; In this case, only users that are part of networks of the target user are searched to evaluate similarities [174, 239];
- *Utility-based approach*: it provides recommendations based on generating a utility model of each item for the user. The recommendation system searches for user utility functions and proceeds to recommend items that are of higher utility based on user utility [79].

There is also a hybrid approach that mixes different recommendation approaches, e.g., it combines two or more approaches to recommend items to the active user. Combining the approaches are employed to leverage the advantages and solve the challenges of each approach if applied in particular [95].

Techniques for hybridizing a recommendation system are:

- *Weighted Method* - the *score* of independent techniques is combined to create a single *score*;
- *Switching* - in case recommendation technique 1 fails, recommendation technique 2 is then used;
- *Mixed* - presentation of results of different recommendation techniques together;
- *Feature combination* - combines features of different algorithms;
- *Feature augmentation* - this technique uses a logical order of Recommender Systems in which targeted features from one system are used as input to the next recommender system;
- *Cascade* - the output of one recommender system is prioritized over another;
- *Meta-level* - first a recommender system is applied and its result is used as an input for the other technique.

As a result, we identified that the collaborative approach matches with the purpose of Helius in recommending DT techniques in software development.

Q2: Which algorithms support the recommendation approaches?

Table D.6 presents a list of algorithms identified from the publications that we selected. We identified 29 distinct algorithms for item recommendation. We ranked the algorithms according to the recommendation approaches we identified in Question Q1.

The *K-Nearest Neighbors* (*KNN*) algorithm was the most cited out of a total of 18 studies for the collaborative filtering approach.

The *KNN* is exploited to find neighbors among users, to present users who have similar preferences, or can exploit it among the items that are part of the base of the Recommender System. When used in a Recommendation System with a collaborative filtering approach, for example, *KNN* uses user ratings to determine neighboring users.

Table D.6: Recommendation algorithms cited in Literature

Algorithm	**Content-based**	**Collaborative Filtering**	**Hybrid**
KNN	[252][275]	[174][179][192][6][238][275][21][22][79] [139][173] [186][26][196]	[174][179]
Bayesian classifier	[6][252][79] [58][173]	[79][22][240] [173] [139] [11] [238] [179]	[26] [174] [173]
Clustering	[173]	[22] [240] [173] [139] [11] [118] [238] [179]	[26]
Pearson correlation		[192] [79] [11] [179] [26]	[179]
SVD			[179]
k-menoid		[171] [192]	
Decision Trees	[173] [252]	[139] [10]	
TF-IDF	[240][173] [252]		
SVC	[118]		
SVD	[238] [10]	[26]	
SVM		[238]	
Fuzzy Logic			[174] [26]
Regression Analysis			[174]
Matrix manipulation	[275]	[79] [240] [95] [10]	[174] [26]
Neural Networks	[58] [173]	[173] [139] [11]	[174] [21] [173] [26]
Genetic Algorithms			[174][173] [26]
Deep Learning			[21]
Random Forest			[21]
Spearman Correlation		[79]	
Heuristic Functions	[79]		
Cosine Similarity	[79]	[173] [179] [186] [26]	
k-Means		[79] [171] [139]	
Convolutional neural network	[58]		
Neural Attentive Item Similarity (NAIS)		[22]	
Sparse Linear Methods		[186]	
Factorized Item Similarity Method		[186]	
Feature-based factorized Bilinear Similarity Model		[186]	
User-specific Feature-based Similarity Models		[186]	
Euclidean similarity	[179]		

Studies in Literature point to *KNN* as the most widely used algorithm for implementing Recommendation Systems based on collaborative filtering approach [21, 174, 26]. Then, we decided to use that *KNN* algorithm in Helius, since Helius is a collaborative recommendation system. We present details about the KNN in Section D.

K-Nearest Neighbor KNN

KNN is a machine learning algorithm that can be used for classifying similar elements (nearest neighbors) in a set of elements [176]. K represents the number of clusters of neighbors the algorithm creates. In order to compute similar elements, different similarity measures can be used [173, 174, 239]. Pearson correlation coefficient and Euclidean distance are the 2 most cited similarity measures in literature [11, 79, 192].

We implemented the KNN algorithm using Python. We choose Python as the programming language to implement the Helius recommendation module since it is used for contexts such as data science, artificial intelligence, data mining, and so on. In order to implement KNN in Helius, we used the Surprise library [113][1]. Surprise is a library that provides the necessary resources for implementing recommendation systems that use explicit data ratings as the main data analysis variable[2].

As the first step in the implementation of the KNN algorithm, we coded the official sample available on the library website[3]. In the example, the KNN algorithm is used for recommending movies based on the MovieLens dataset. MovieLens is a movie dataset and it is the most cited dataset used for testing recommending systems [22, 58, 173, 174, 239]. Figure **??** shows the results of the movie recommendations using the surprise library implementation based on the sample code.

Recommendation Module Implementation using KNN

Recommendation Module Architecture

Then, after concluding the implementation of a movie recommendation system, we adapted the code to find similar users on the Helius database. In order to implement the KNN in Helius, we designed a recommendation architecture based on modules, as

[1] Available at: http://surpriselib.com/

[2] The Surprise Library provides a tested KNN implementation at: https://surprise.readthedocs.io/en/stable/similarities.html#surprise.similarities.pearson

[3] Available at: https://bit.ly/3x7NBz8

```
Estimating biases using als...
Computing the pearson_baseline similarity matrix...
Done computing similarity matrix.

The 10 nearest neighbors of Toy Story are:
Beauty and the Beast (1991)
Raiders of the Lost Ark (1981)
That Thing You Do! (1996)
Lion King, The (1994)
Craft, The (1996)
Liar Liar (1997)
Aladdin (1992)
Cool Hand Luke (1967)
Winnie the Pooh and the Blustery Day (1968)
Indiana Jones and the Last Crusade (1989)
```

Figure D.3: Example of the *KNN* algorithm using *Surprise* library

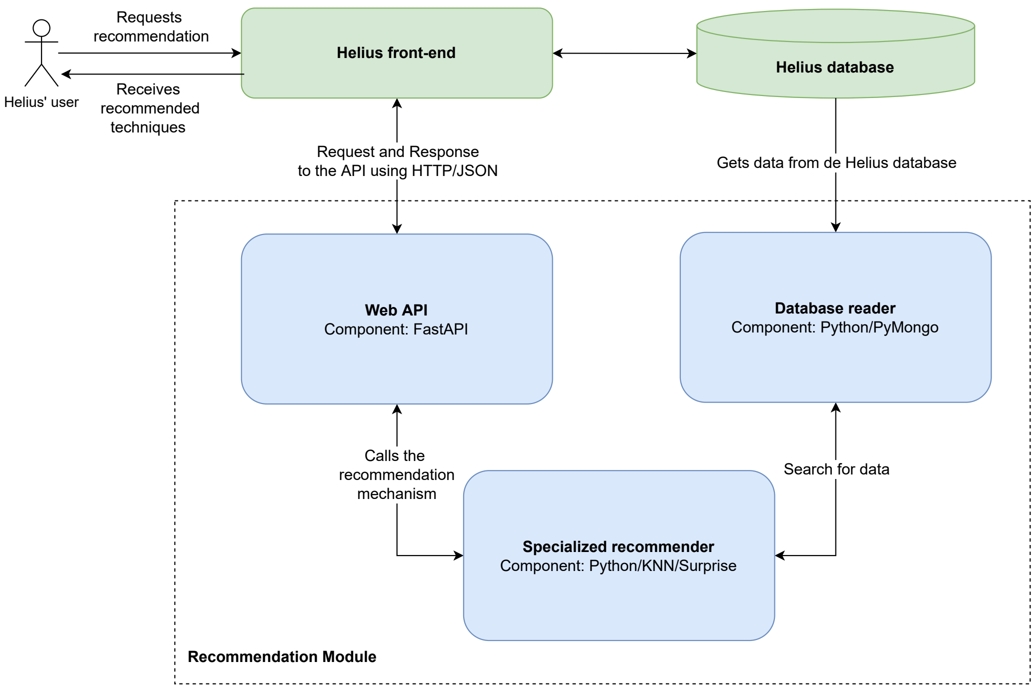


Figure D.4: Recommendation module architecture

exposed in Figure D.4. Such architecture is composed of the following components: Web API, Specialized Recommender and Database Reader[4].

The components of the recommendation module can be defined as follows:

- *Web API:* component that works as a web interface responsible for receiving and returning requisitions of DT techniques recommendations;
- *Specialized recommender:* component that implements the DT techniques recommendation mechanisms;

[4]The Helius' recommendation API is available at https://helius-dt-recommendation-api.herokuapp.com/docs#/

- *Database Reader:* component that connects the recommendation module to the Helius' database to read the data about DT techniques.

The Helius' recommendation module architecture needs that the user makes a request for DT techniques recommendation using the Helius app (front-end). Once a request has been done, the recommendation module makes a request to the Web API component. Then, the Web API components make a request to the Specialized Recommender module, calling the recommendation mechanism the user has selected in Helius. Thus, the Specialized Recommender module uses the Database reader for reading the data about DT techniques usage (ratings x users) in the Helius database.

Recommendation Module Implementation

As we mentioned in Section 7.2.2, Helius provides personalized and non-personalized DT techniques and recommendation mechanisms. The KNN algorithm is used in the personalized recommendation mechanisms set. Those mechanisms require the identification of similar users to recommend DT techniques based on their tastes (ratings for techniques they have already used). Therefore, we used KNN to identify the similarity between the Helius users.

In order to exemplify how a personalized recommendation works in Helius, Figure D.5 shows the process in the recommendation module for recommending a set of DT techniques based on the user profile (personalized). The Surprise library provides the mechanisms to implement the KNN algorithm.

Once user X has requested a recommendation of DT techniques and the Specialized Recommender components has been called by the Web API component, the Surprise library starts to work. It is responsible for identifying similar users to the user that has made the request. Surprise computes the similarity between users by creating a similarity matrix *nxn*, composed by the relation users x users. For each cell user x users, a similarity value is computed using the *Pearson correlation coefficient*. Equation D.1 shows the formula to compute the Pearson Correlation Coefficient. R represents the correlation coefficient and X and Y are two Helius users.

$$R = \frac{\sum_{i=1}^{n} x.y - (\frac{\sum_{i=1}^{n} x.\sum_{i=1}^{n} y}{n})}{\sqrt{(\sum_{i=1}^{n} x^2 - \frac{(\sum_{i=1}^{n} x)^2}{n}).(\sum_{i=1}^{n} y^2 - \frac{(\sum_{i=1}^{n} y)^2}{n})}} \tag{D.1}$$

Once all the similarities between Helius' users have been computed (R), the algorithm shows the list of similar users (user Y). These procedures are abstracted in the Surprise library. After finding similar users, the recommendation module filters the DT techniques used by those similar users. The next step is to filter the techniques according

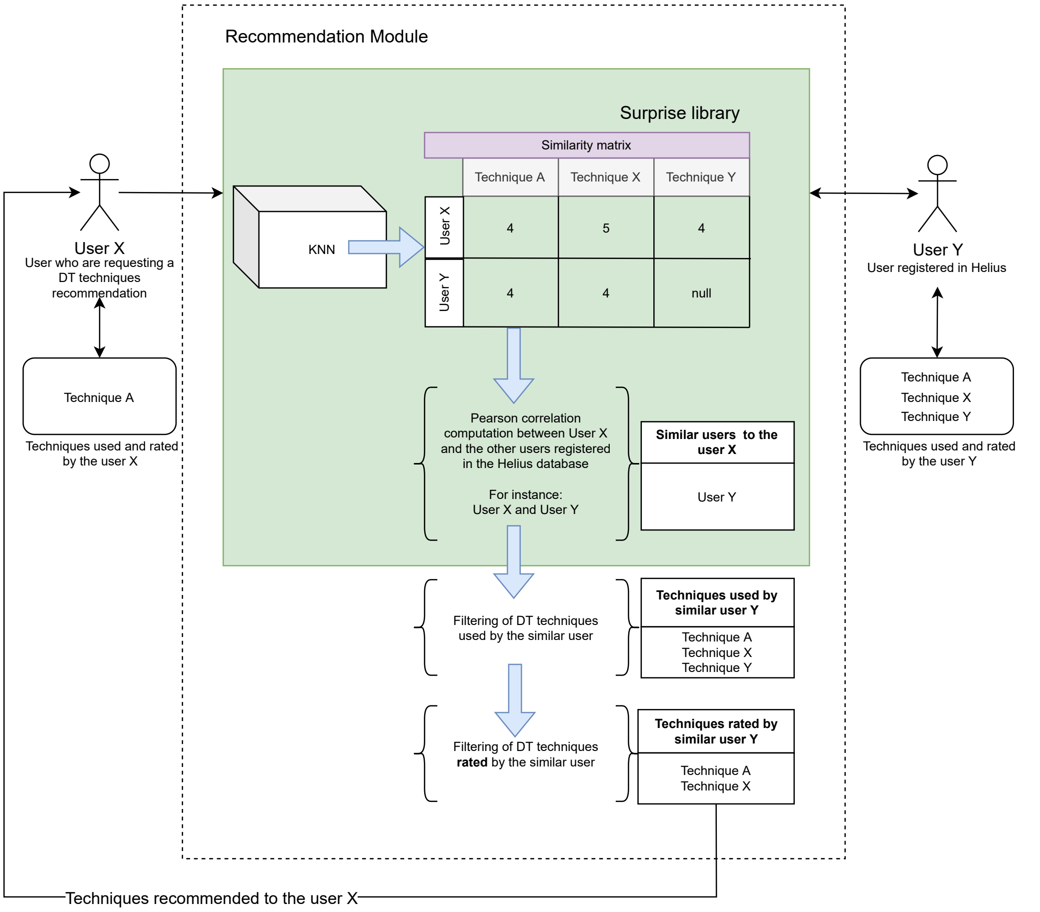


Figure D.5: Example of a recommendation process in Helius using KNN

to the recommendation mechanism selected by the user who requested the recommendation itself. For instance, in Figure D.5, considering the user has selected the mechanism “Rated by similar users”, which returns all the techniques that similar users have rated, the recommendation module returns just the techniques that were rated by similar users (e.g., technique Y was not evaluated by the user Y and, therefore, it have not been recommended to the user X).

Figure D.6 shows an example of the implementation of the “Rated for similar users” mechanism. This mechanism recommends the techniques that similar users have evaluated in Helius. The code is composed of 3 methods (starting with “def”):

- “fit_knn”: the method loads the data set of techniques ratings to the Surprise library. This method also uses a K value. K represents the number of near neighbors to be evaluated[5]. We also set up the similarity measure (Pearson Correlation Coefficient) and that the similarity must be computed between users (“user_based”).

- “find_similar_users”: the method looks for similar users. The method used the procedure get_neighbors for identifying the 5 most similar users (k = 5) according to

[5]We defined the K value as 5, since it is a value commonly used for K, as mentioned in literature.

```python
def fit_knn(self, ratings_df):
    reader = Reader(rating_scale=(1.0, 5.0))
    data = Dataset.load_from_df(ratings_df[['user_id', 'tech_name', 'rating']], reader)
    trainset = data.build_full_trainset()
    algo = KNNBaseline(k=5, sim_options={'name': 'pearson', 'user_based': True})
    algo.fit(trainset)
    return algo

def find_similar_users(self, algo, raw_user_id):
    if algo.trainset.knows_user(raw_user_id):
        raise ValueError(f"User {raw_user_id} is unknown")
    user_inner_id = algo.trainset.to_inner_uid(raw_user_id)
    neighbors = algo.get_neighbors(user_inner_id, k=5)
    neighbors_user_id = [algo.trainset.to_raw_uid(inner_id) for inner_id in neighbors]
    return neighbors_user_id

def get_rated_by_similar_users(self, user_id, numer_of_recommendations = None):
    print(f'User that wants a recommendation id = [{user_id}]')
    ratings_df = self.get_df_ratings()
    algo = self.fit_knn(ratings_df)
    neighbors_user_id = self.find_similar_users(algo, user_id)

    recommendations = []
    for user_id in neighbors_user_id:
        print(f"Similar Users id = [{user_id}]")
        ratings = ratings_df.loc[ratings_df['user_id'] == user_id]
        if numer_of_recommendations is None:
            list_tech_names = list(ratings['tech_name'])
            recommendations.extend(list_tech_names)
        else:
            top_n = ratings.sort_values('rating', ascending=False).head(numer_of_recommendations)
            list_tech_names = list(top_n['tech_name'])
            recommendations.extend(list_tech_names)

    recommendations = list(set(recommendations))
    print(f"Techniques used by similar users = [{recommendations}]")
    df_tech = self.get_df_techniques()
    df_tech = df_tech.loc[df_tech['name'].isin(recommendations)]
    print(f"Techniques recommended = [{list(df_tech['name'])}]")
    json_value = df_tech.to_json(orient="records", default_handler = str)
    return json_value
```

Figure D.6: Example of a recommendation mechanism using *Surprise* library

the id of the user that required the recommendations. The method returns a list of similar users (neighbors).

- “get_rated_by_similar_users”: the method uses the aforementioned methods to get all techniques rated by similar users. To do so, it starts by identifying similar users and looks for the techniques the identified users have rated. The set of rated techniques is returned to the user that has requested it. In addition, the method allows the user to indicate how many techniques she wants to receive as recommendations (“number_of_recommendation”). In the default implementation of Helius, we defined 5 as the number of DT techniques that the module recommends to the user. The method returns a JSON response that can be decoded in the Helius front-end.

In order to evaluate the implementation of the KNN algorithm, we performed offline tests [225]. Offline testing performs the separation of the data set under analysis into 2 groups. One of the groups is used to train the algorithm while the other group is used for the test itself. In this sense, we use offline testing as a way to evaluate the implementation we performed with the Surprise library.

We used the following procedure to perform offline testing:

1) We run the KNN algorithm by entering the id of a user present in the Helius database;

```
User that wants a recommendation id = [6115a3db3397360015fb9fd8]
Estimating biases using als...
Computing the pearson similarity matrix...
Done computing similarity matrix.
Similar Users id = [61250453ab505d00016d519ab]
Techniques used by similar users = [['World Café', 'Brainstorming', 'A day in the life', '5W2H']]
Techniques recommended = [['A day in the life', '5W2H', 'Brainstorming', 'World Café']]
```

Figure D.7: Example of a recommendation in Helius

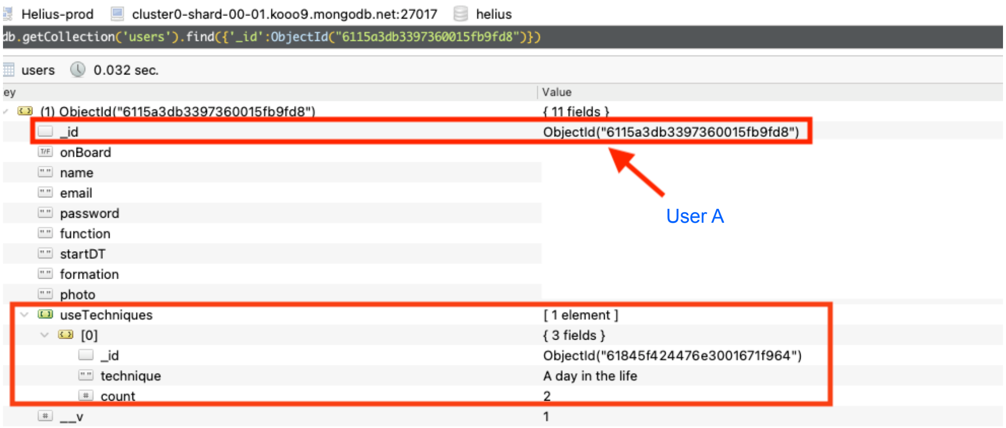


Figure D.8: User A rated technique “A day in the life”

2) We manually searched in the Helius database for other users who evaluated DT techniques in a similar way (same ratings);

3) We compared the results returned by the algorithm with those that we obtained using the manual search.

Figure D.7 shows the records used for the tests performed as part of procedure 1). We selected the user with id “6115a3db3397360015fb9fd8” in Helius, here named as user A for identification purposes. The id was then passed as a parameter to the method “get_rated_by_similar_users” in order to find similar users. Such similar users are identified by the Surprise library. With the result of the method’s execution, the similar user-id "61250453ab505d0016d519ab" was obtained, which for identification purposes is assigned the identifier B.

Next, as part of procedure 2), we performed a manual search of the Helius database for User A and User B. We observed that both users evaluated the same “A day in the life” technique, as shown in Figures D.8 and D.9, respectively. Therefore, we were able to confirm that both users are indeed similar (procedure 3).

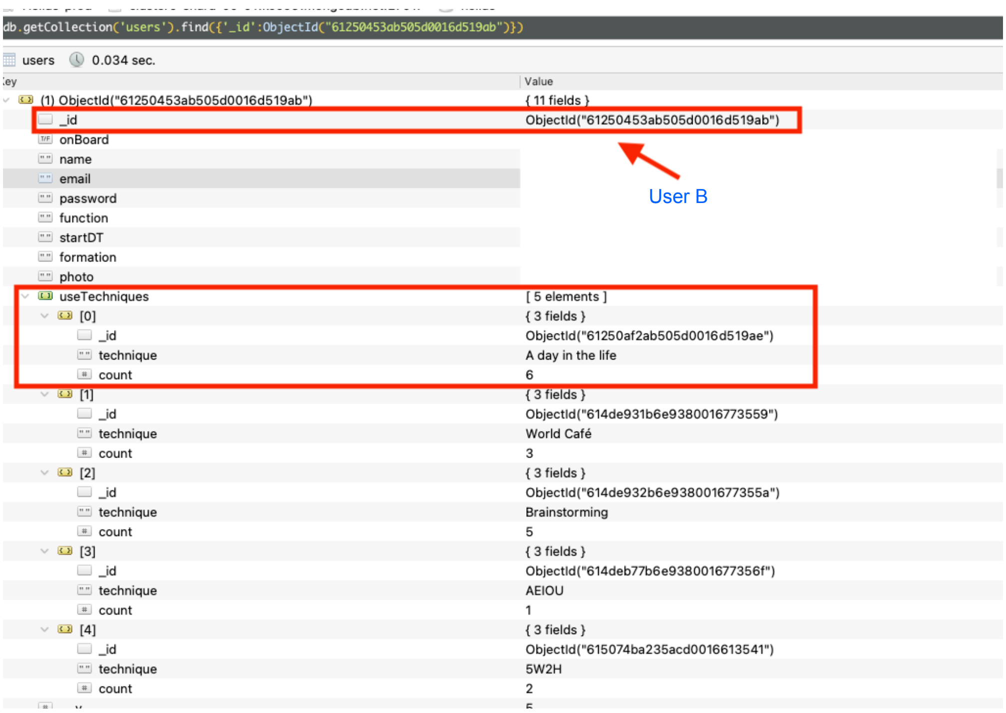


Figure D.9: Similarity between user B and user A

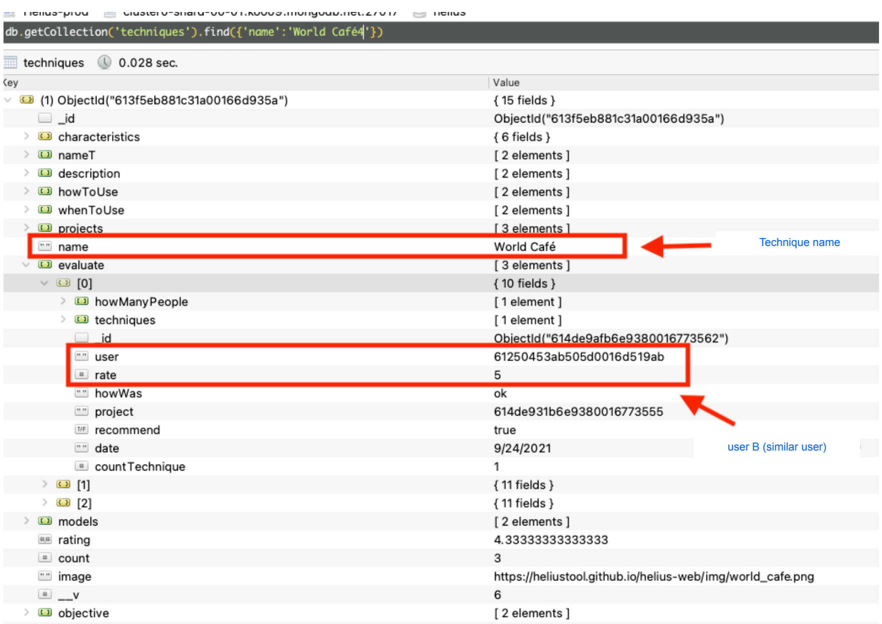


Figure D.10: User B rating for technique “World Cafe”

Once similar users are found (user B in the example), the method “get_rated_by_similar_users” method searches for the techniques used and evaluated by user B. Through a search of the Helius database, the method returned that the techniques used and evaluated by B are “World Cafe”, “5W2H”, among others, as shown in Figures D.10 and D.11, respectively.

Thus, like the "get_rated_by_similar_users" recommendation mechanism, the other mechanisms are made available by the recommendation module as endpoints in the Web API and can be accessed by the Helius front-end. Figure D.12 shows the recommendation mechanisms and their respective endpoints in the Web API.

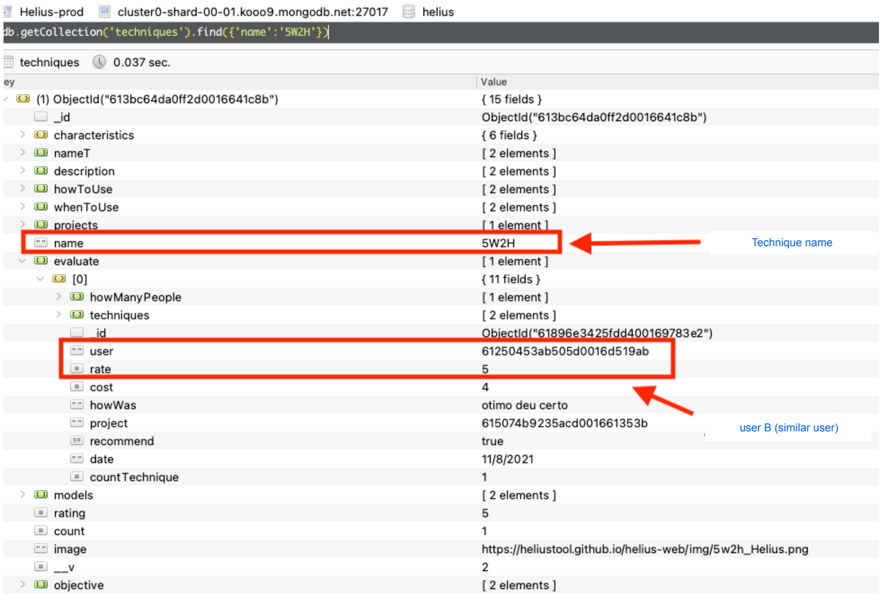


Figure D.11: User B rating for technique "5W2H"

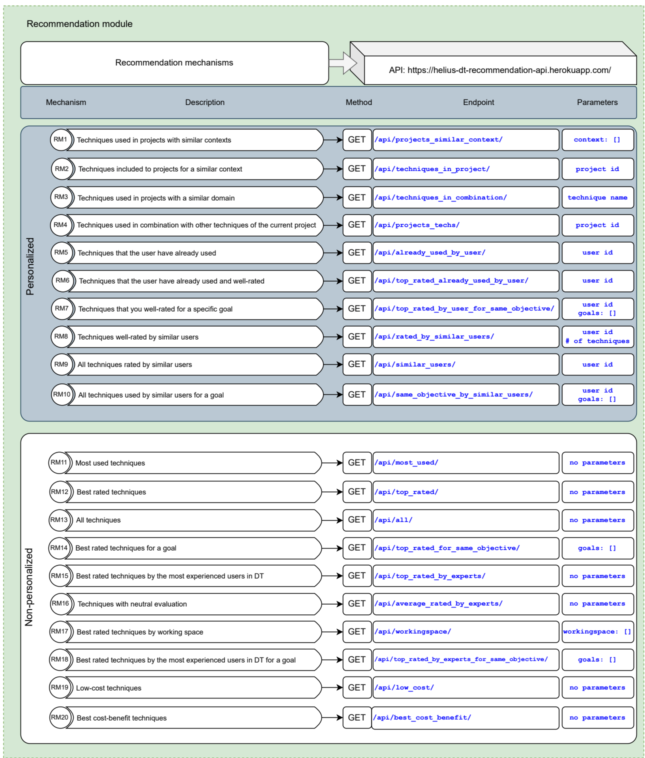
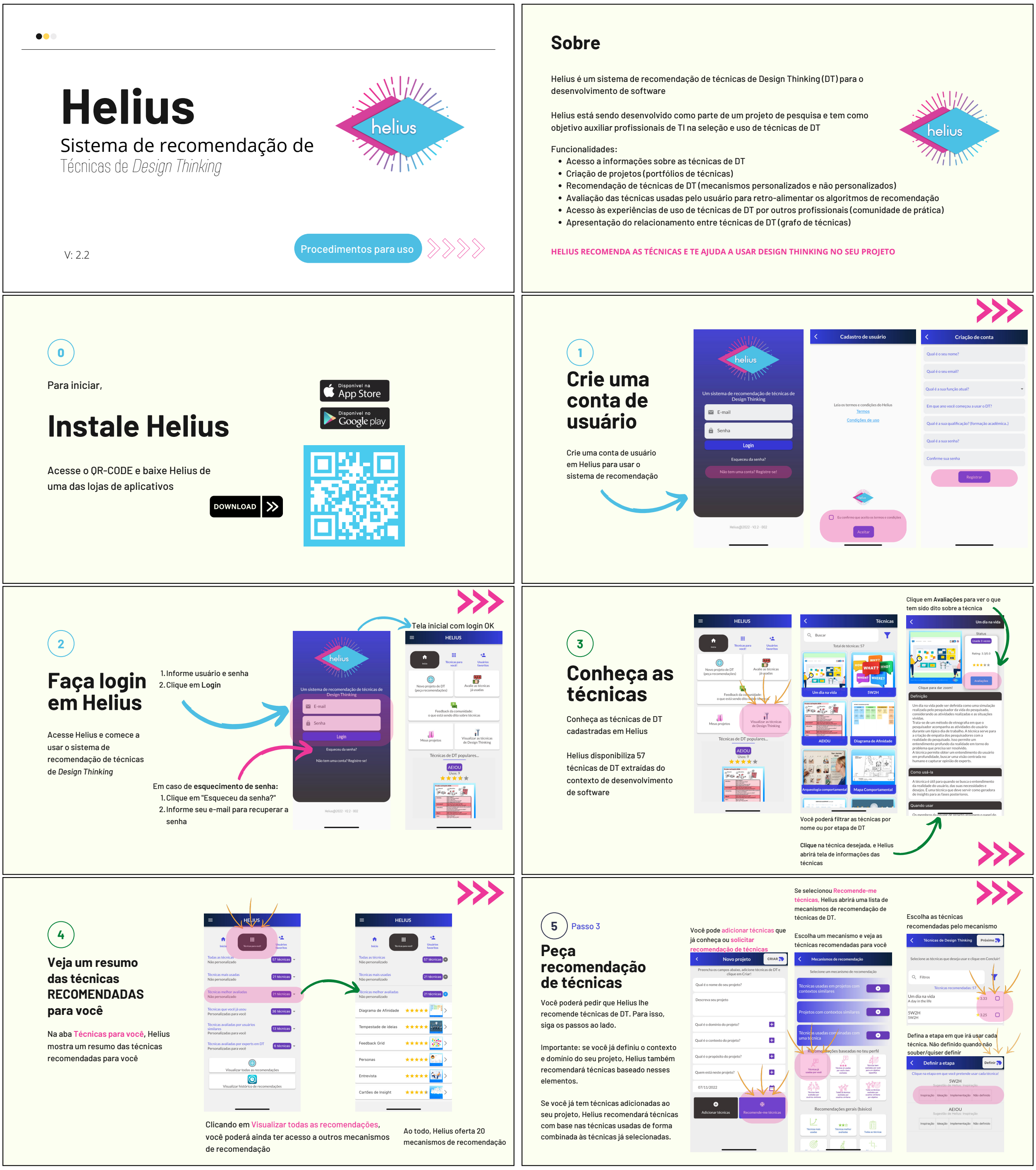


Figure D.12: Recommendation mechanisms endpoints in the Web Api

APPENDIX E – HELIUS' EMPIRICAL EVALUATION: PROCEDURES AND COLLECTED DATA

This chapter presents examples of the procedures that the participants should perform for using Helius in the empirical study (Portuguese version). It also present the raw data answered by the participants to a questionnaire the evaluate data (Table E.2) and the SUS data informed by all participants (Table E.1).



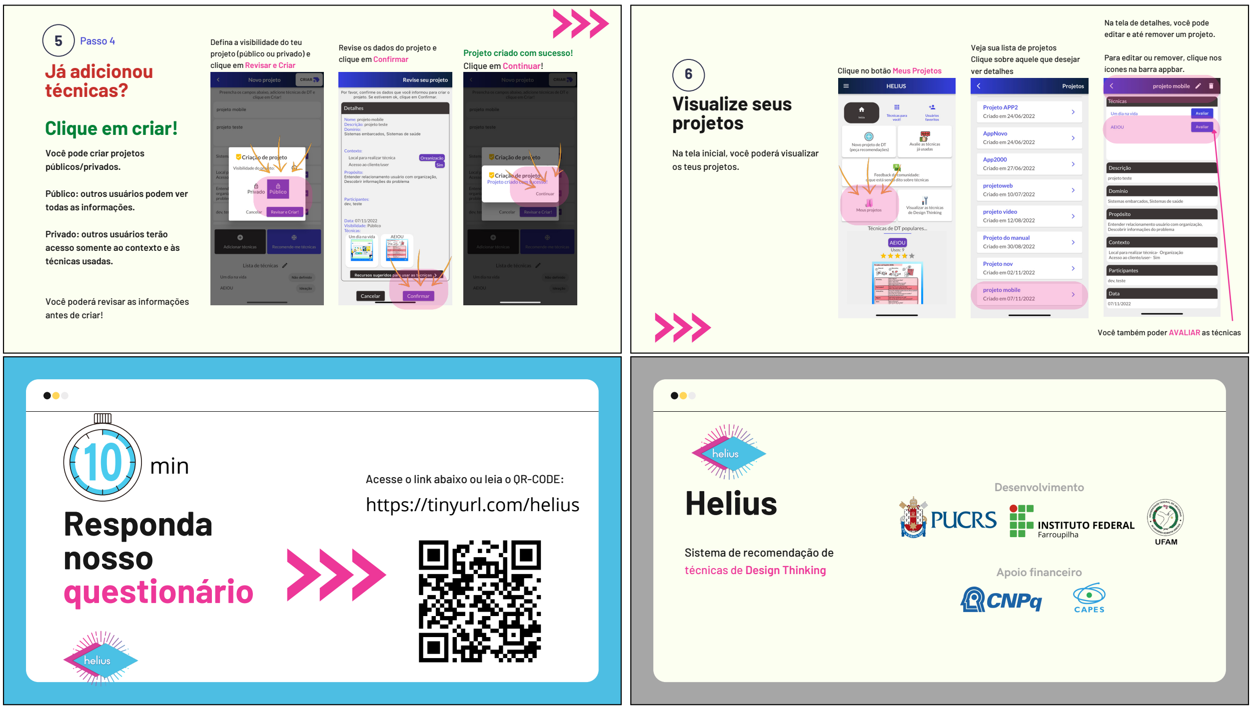


Table E.1: SUS raw data collected with participants in the empirical study

Participant	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10
SUS data collected with participants in Step 1										
P1	5	1	4	2	4	3	4	2	4	1
P2	4	4	4	4	4	1	2	3	4	4
P3	5	2	4	2	2	3	4	3	4	1
P4	5	2	4	2	2	5	4	3	5	1
SUS data collected with participants in Step 2										
P5	5	1	4	3	5	1	5	1	5	1
P6	4	3	4	4	3	2	4	3	4	2
P7	4	1	5	4	5	2	4	1	4	1
P8	5	2	3	2	2	2	4	3	2	2
P9	5	2	5	2	4	2	5	2	5	2
P10	4	3	4	3	4	2	3	3	3	3
P11	4	2	4	2	3	2	4	2	4	2
P12	3	3	3	1	3	2	4	3	5	1
P13	1	3	4	1	4	2	4	2	4	2
P14	4	3	4	4	4	2	4	3	4	2
P15	4	2	4	2	3	1	5	2	4	2
P16	5	2	4	1	2	3	4	4	4	1

q1	I think that I would like to use this system frequently
q2	I found the system unnecessarily complex
q3	I thought the system was easy to use
q4	I think that I would need the support of a technical person to be able to use this system
q5	I found the various functions in this system were well integrated.
q6	I thought there was too much inconsistency in this system
q7	I would imagine that most people would learn to use this system very quickly
q8	I found the system very cumbersome to use
q9	I felt very confident using the system
q10	I needed to learn a lot of things before I could get going with this system

Table E.2: Participants' frequency of use of each recommendation mechanism

Mechanisms	Participants' answers											
	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Techniques used in projects with a similar context	4	1	4	4	4	3	4	3	3	4	4	3
Techniques included to projects for a similar context	4	3	4	4	4	3	4	3	3	3	4	3
Techniques used in projects with a similar domain:	4	1	5	4	4	3	4	3	3	4	3	3
Techniques used in combination with other techniques of the current project	4	1	5	4	4	3	4	3	3	3	3	4
Techniques that the user have already used	5	1	4	4	4	3	5	3	3	4	4	4
Techniques that the user have already used and well-rated	5	1	4	5	4	4	5	3	4	4	4	4
Techniques that you well-rated for a specific goal	4	1	3	4	4	5	5	3	3	2	2	4
Techniques well-rated by similar users	4	1	2	4	4	3	5	3	4	3	3	4
All techniques rated by similar users	4	1	2	4	4	3	4	3	4	2	4	2
All techniques used by similar users for a goal	4	1	2	4	4	3	4	3	2	2	3	2
Most used techniques	5	3	4	4	4	4	4	3	2	5	5	4
Best rated techniques	4	4	5	5	4	4	5	3	2	5	5	4
All techniques	3	4	3	1	4	3	3	3	4	5	3	5
Best rated techniques for a goal	4	3	3	3	4	4	4	3	2	3	3	5
Best rated techniques by the most experienced users in DT	4	1	5	3	4	4	5	3	2	3	4	5
Techniques with neutral ratings	4	1	3	3	4	3	3	3	3	3	2	5
Best rated techniques by working spaces	4	1	3	4	4	3	4	3	3	2	2	2
Best rated techniques by the most experienced DT professionals for a goal	4	2	3	4	4	3	4	3	2	2	3	5
Low-cost techniques	4	1	4	5	4	3	4	3	4	2	3	5
Best cost-benefit techniques	4	1	5	2	4	3	4	3	2	3	2	5

1 - Not at all important
2 - Low important
3 - Neutral
4 - Important
5 - Very important

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br