

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**GERENCIAMENTO DE RISCOS EM PROJETOS DE
DESENVOLVIMENTO DE SOFTWARE COM SCRUM**

PAULO JACÓ RECH

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências da Computação, pelo Programa de Pós-Graduação em Ciências da Computação da Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

Porto Alegre

2013

Dados Internacionais de Catalogação na Publicação (CIP)

R296g Rech, Paulo Jacó
Gerenciamento de riscos em projetos de desenvolvimento de software com Scrum / Paulo Jacó Rech. – Porto Alegre, 2013.
95 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Rafael Prikladnicki.

1. Informática. 2. Engenharia de Software. 3. Administração de Projetos. I. Prikladnicki, Rafael. II. Título.

CDD 005.1

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Gerenciamento de Riscos em Projetos de Desenvolvimento de Software com SCRUM" apresentada por Paulo Jacó Rech como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Engenharia de Software e Banco de Dados, aprovada em 27/03/2013 pela Comissão Examinadora:

Prof. Dr. Rafael Prikladnicki -
Orientador

PPGCC/PUCRS

Prof. Dr. Ricardo Melo Bastos -

PPGCC/PUCRS

Prof. Dr. Marcelo Soares Pimenta -

UFRGS

Homologada em 01/10/2013, conforme Ata No. 17 pela Comissão Coordenadora.

Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P32- sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

DEDICATÓRIA

*Dedico este trabalho ao meu **Deus**, pelo chamado, capacitação e talentos confiados.
Não ando a procura de grandes coisas, de grandes obras ou realizações,
quero apenas ser encontrado fiel e retribuir mais este talento
a serviço do Reino Eterno ao qual eu pertença e sirvo.*

*À minha amada e linda esposa **Cacilda Rech**, que
com amor, carinho e dedicação soube entender
os momentos de privação, me incentivando
e sendo suporte nesta caminhada.*

*Aos meus amados filhos **Nathan** e **Nicolas**.
Sem os “cafunés” eu não teria conseguido.*

Vocês são o motivo deste esforço!

AGRADECIMENTOS

Acima de tudo sou grato a Deus por ter-me dado a vida, um chamado e a capacitação para exercê-lo. Retribuo a Ele também este talento.

À minha família. Minha esposa Cacilda por entender os momentos de privação e sempre me apoiar nesta caminhada. Aos meus filhos, Nathan e Nicolas, por serem meus filhos e por me distraírem nos momentos mais críticos, não deixando que me aprofundasse além do necessário.

Aos meus pais, Paulo Valter Rech e Miriam Nadir Hertz pelo apoio, conselhos e orações em todo tempo, mas principalmente pela amizade que muito prestigio. Agradeço aos meus irmãos Rubem Rech, que sempre, desde que nasceu foi meu melhor amigo, e Timóteo Rech, com quem tanto nos divertimos. Minha cunhada Daniele pelas dicas em um caminho já traçado. Também agradeço a minhas irmãs Cristina e Rebeca pelo carinho que sinto quanto estamos juntos.

Nossos companheiros que sempre demonstram o amor de Deus por mim e pela minha família: Fabrício e Camila Azevedo, Eduardo e Taís Barbosa, Samir e Luciane Machado. Vocês são pilares que Deus colocou para nos apoiarmos, com quem podemos sempre contar. Sabemos disso! Agradeço e louvo a Deus por vocês estarem por perto.

Agradeço à Igreja, minha família da fé, sempre presente a qual desejo servir com todos os talentos que Deus me confiar, inclusive este. Meus pastores, Márcio e Adriana Rocha, Vinícius e Amira Gonçalves, João Nelson e Sirlei Otto (estes últimos, colegas no serviço de capelania) os quais, todos eles, certamente estiveram orando por mim.

Esta conquista deve-se ao esforço desbravador de uma pessoa que tem o dom de ver sempre além. Obrigado, Prof. Dr. Rafael Prikkladnicki, por ter acreditado em meu potencial e assim, ter-me orientado neste caminho, investindo seu tempo e conhecimento. A contribuição de cada professor do Programa de Pós-Graduação da Faculdade de Informática da PUCRS foi basilar para a minha formação. Obrigado!

Agradeço aos meus colegas, amigos que fiz durante esta caminhada: Bernardo, Roni, Vanessa e Francine. Muito mais rimos do que choramos juntos.

Finalmente, mas com igual importância, agradeço ao Sicedi onde trabalho. Sempre tive todo o apoio necessário para conduzir esse estudo, mesmo durante o horário comercial em algumas vezes, contando com a colaboração de meus colegas Alexandre Silveira, Gustavo Costa, Vernei Cunico, André Cunha, Raquel Rosa e Remo Wilke. Obrigado! Vocês também *cooperaram* para esta conquista.

A todos vocês, MUITO OBRIGADO!

RESUMO

As empresas estão sempre em busca de vantagens competitivas, redução de custos, aumento de qualidade e produtividade. O desenvolvimento de software está inserido neste contexto, com contribuições das áreas de Engenharia de Software e o Gerenciamento de Projetos, visando produzir software com qualidade, menos desperdício e com a rapidez exigida pelo mercado atual. Para enfrentar este desafio, a indústria de desenvolvimento de software tem buscado novas maneiras de criar novos produtos. As abordagens adaptativas, com práticas que procuram ser mais flexíveis do que as abordagens prescritivas, muitas vezes consideradas pesadas e lentas, enfatizam a agilidade dos processos de desenvolvimento de software, buscando maior eficiência em situações onde mudanças são habituais.

O método ágil Scrum é uma das abordagens adaptativas mais conhecidas para o gerenciamento de projetos e define um conjunto de boas práticas aplicado através de ciclos iterativos e incrementais, com envolvimento e visibilidade constante do cliente, proporcionando entrega rápida e com valor para o negócio. Entretanto, o gerenciamento de riscos, prática muito relevante na condução de projetos, é tratado de forma implícita em projetos que utilizam abordagens adaptativas como o Scrum.

Desta forma, o objetivo deste trabalho é desenvolver um estudo empírico que visa identificar como os riscos mais comuns encontrados na literatura de gerenciamento de projetos de desenvolvimento de software são tratados no Scrum. Para o desenvolvimento desta pesquisa foram utilizados estudos secundários (revisão sistemática da literatura) e primários (estudo de campo). Esta pesquisa contribui para a teoria e para a prática de gerenciamento de projetos de software, especificamente na área de gerenciamento de risco e sua intersecção com o método ágil Scrum.

Palavras-chave: Scrum, gerenciamento de riscos, métodos ágeis, abordagens adaptativas.

ABSTRACT

Companies are always looking for competitive advantage, costs reduction, quality increasing and more productivity. Software development is part of this context, with contributions from the areas of Software Engineering and Project Management, aiming at producing software with quality, with less waste, and with the speed required in today's market. To meet this challenge, the software development industry has sought new ways to develop new products. The adaptive approaches, with practices that seek to be more flexible than prescriptive approaches, often considered cumbersome and slow, emphasize the agility of software development processes, seeking greater efficiency in situations where changes are common.

The Scrum framework is one of the most popular agile methods and it is considered an adaptive approach for project management. It defines a set of practices implemented through iterative and incremental cycles, with constant involvement and visibility of the customer, providing quick delivery and business value. However, risk management, which is a very relevant practice in conducting projects, is implicitly treated in projects that use adaptive approaches such as Scrum.

Thus, the aim of this work is to develop an empirical study that seeks to identify how the list of common risks found in the software project management literature is managed in Scrum. In order to develop this research we have used secondary (systematic literature review) and primary studies (field study). This research contributes to the theory and practice of software project management, specifically in the area of risk management and its intersection with the Scrum framework.

Keywords: Scrum, risk management, agile methods, adaptive approaches.

LISTA DE FIGURAS

FIGURA 1 – MODELO DE PROCESSO EM CASCATA.....	19
FIGURA 2 – PRIORIZAÇÃO DE CONCEITOS DE RELAÇÕES	20
FIGURA 3 – NOVO PAPEL DO GERENTE DE PROJETOS EM ABORDAGENS ADAPTATIVAS.....	21
FIGURA 4 – A ENGRENAGEM DO SCRUM	23
FIGURA 5 – GRÁFICO SPRINT BURNDOWN	24
FIGURA 6 – RESUMO DO GERENCIAMENTO DE RISCOS EM PROJETOS.....	26
FIGURA 7 – COMPARAÇÃO ENTRE O GUIA PMBOK E A FERRAMENTA RISKFREE	31
FIGURA 8 – ETAPAS DA PROPOSTA NO-RISK	32
FIGURA 9 – FASES DO CICLO DE VIDA DO SCRUM	34
FIGURA 10 – DESENHO DE PESQUISA	39
FIGURA 11 – SUBCONJUNTOS DE RISCOS COMUNS COM SCRUM.....	57
FIGURA 12 – FLUXO DE ENTREVISTAS DE ESTUDO DE CAMPO	59
FIGURA 13 – MAPA DE TRATAMENTO DE RISCOS COMUNS COM SCRUM	63
FIGURA 14 – CONTEXTO ORGANIZACIONAL DA GESTÃO DE PORTFÓLIO	70

LISTA DE TABELAS

TABELA 1 – MATURIDADE DAS ORGANIZAÇÕES EM GERENCIAMENTO DE PROJETOS.....	28
TABELA 2 – MAPEAMENTO DAS PRINCIPAIS OCORRÊNCIAS ENTRE GR E EVENTOS DO SCRUM ...	34
TABELA 3 – CLASSIFICAÇÃO DOS ESTUDOS ANALISADOS	35
TABELA 4 – ARTIGOS SELECIONADOS PARA LISTA DE RISCOS COMUNS.....	41
TABELA 5 – DIMENSÕES DE RISCOS COMUNS.....	42
TABELA 6 – RISCOS COMUNS EM PROJETOS DE SOFTWARE.....	42
TABELA 7 – FONTES DE PESQUISA, PALAVRAS DE BUSCA E RESULTADOS	47
TABELA 8 – REVISÃO SISTEMÁTICA - ESTUDOS PRIMÁRIOS EXCLUÍDOS	49
TABELA 9 – ESTUDOS RESULTANTES DA REVISÃO SISTEMÁTICA.....	51
TABELA 10 – PROCEDIMENTOS DO ESTUDO DE CAMPO.....	56
TABELA 11 – PERGUNTAS DE INFORMAÇÕES DEMOGRÁFICAS DO ESTUDO DE CAMPO.....	58
TABELA 12 – PLANILHA DE DADOS PRINCIPAIS DO ESTUDO DE CAMPO.....	60
TABELA 13 – INFORMAÇÕES DEMOGRÁFICAS DO ESTUDO DE CAMPO	62
TABELA 14 – TDRC - DIMENSÃO DE AMBIENTE ORGANIZACIONAL	65
TABELA 15 – TDRC - DIMENSÃO DE PLANEJAMENTO E CONTROLE	65
TABELA 16 – TDRC - DIMENSÃO DE COMPLEXIDADE	65
TABELA 17 – TDRC - DIMENSÃO DE REQUISITOS.....	65
TABELA 18 – TDRC - DIMENSÃO DE EQUIPE	65
TABELA 19 – TDRC - DIMENSÃO DE USUÁRIO.....	66
TABELA 20 – EFETIVIDADE DO SCRUM POR DIMENSÃO DE RISCO	66
TABELA 21 – DIMENSÃO DE AMBIENTE ORGANIZACIONAL X PORTFÓLIO – SCRUM OF SCRUMS.	71
TABELA 22 – DIMENSÃO DE COMPLEXIDADE DE RISCOS X SPRINT ZERO E P&D.....	73
TABELA 23 – DIMENSÃO DE EQUIPE DE RISCOS X BENEFÍCIOS DE PP	76

LISTA DE EQUAÇÕES

EQUAÇÃO 1 - TRATAMENTO DE RISCO COMUM (TRC).....	64
EQUAÇÃO 2 - TRATAMENTO DE DIMENSÃO DE RISCO COMUM (TDRC).....	64

LISTA DE ABREVIATURAS E SIGLAS

CMMI	CAPABILITY MATURITY MODEL INTEGRATION
DDS	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE
GERDDoS	GERENCIAMENTO DE RISCOS EM PROJETOS DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE
GP	GERENCIAMENTO DE PROJETOS
GR	GERENCIAMENTO DE RISCOS
MSF	MICROSOFT SOLUTIONS FRAMEWORK
PMBOK	PROJECT MANAGEMENT BODY OF KNOWLEDGE
PMI	PROJECT MANAGEMENT INSTITUTE
RUP	RATIONAL UNIFIED PROCESS
TI	TECNOLOGIA DA INFORMAÇÃO
XP	EXTREME PROGRAMMING

SUMÁRIO

LISTA DE FIGURAS	8
LISTA DE TABELAS	9
LISTA DE EQUAÇÕES	10
LISTA DE ABREVIATURAS E SIGLAS	11
1 INTRODUÇÃO	14
1.1 JUSTIFICATIVA	15
1.2 ORGANIZAÇÃO DO VOLUME	15
2 OBJETIVOS	16
2.1 OBJETIVO GERAL	16
2.2 OBJETIVOS ESPECÍFICOS	16
3 REFERENCIAL TEÓRICO	17
3.1 GERENCIAMENTO DE PROJETOS DE SOFTWARE	17
3.2 ABORDAGENS DE GERENCIAMENTO DE PROJETOS	18
3.2.1 ABORDAGENS PRESCRITIVAS	18
3.2.2 ABORDAGENS ADAPTATIVAS	19
3.3 O MÉTODO ÁGIL SCRUM	21
3.4 GERENCIAMENTO DE RISCOS	25
3.4.1 CENÁRIO HISTÓRICO	25
3.4.2 IMPORTÂNCIA DO GERENCIAMENTO DE RISCOS EM PROJETOS	27
3.4.3 GERENCIAMENTO DE RISCOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE	28
3.5 TRABALHOS RELACIONADOS	29
3.5.1 A FERRAMENTA RISKFREE	30
3.5.2 A PROPOSTA NO-RISK	32
3.5.3 RISK BACKLOG: UM MODELO DE GERENCIAMENTO DE RISCOS COM SCRUM	33
3.5.4 ANÁLISE CRÍTICA SOBRE OS ESTUDOS AVALIADOS	35
4 METODOLOGIA DE PESQUISA	36
4.1 ASPECTOS METODOLÓGICOS	36
4.1.1 PROPÓSITO – OBJETIVO AMPLO	36
4.1.2 DELINEAMENTO – PROCEDIMENTOS TÉCNICOS	37
4.2 DESENHO DE PESQUISA	39
5 LISTA DE RISCOS COMUNS EM PROJETOS DE SOFTWARE	41
5.1 SELEÇÃO DE ESTUDOS E DIMENSÕES DE RISCOS	41
5.2 LISTA DE RISCOS COMUNS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE	42
6 REVISÃO SISTEMÁTICA DA LITERATURA	44
6.1 PROCESSO DE REVISÃO SISTEMÁTICA DA LITERATURA	44
6.2 QUESTÃO DE PESQUISA	44

6.3	QUALIDADE E AMPLITUDE DA QUESTÃO	45
6.4	SELEÇÃO DE FONTES DE PESQUISA	46
6.5	SELEÇÃO DE ESTUDOS	47
6.6	PROCEDIMENTOS DE SELEÇÃO DE ESTUDOS	48
6.7	ESTUDOS PRIMÁRIOS EXCLUÍDOS	49
6.8	ESTUDOS SELECIONADOS	51
6.9	RESPOSTAS ÀS QUESTÕES DE PESQUISA	52
6.10	CONCLUSÃO DA REVISÃO SISTEMÁTICA DA LITERATURA DA ÁREA	54
7	ESTUDO DE CAMPO	55
7.1	OBJETIVO	55
7.2	UNIDADE DE ANÁLISE E CONFIDENCIALIDADE	55
7.3	PLANEJAMENTO DO ESTUDO DE CAMPO	56
7.4	RECURSOS UTILIZADOS	56
7.5	DIMENSÕES DO ESTUDO	57
7.6	COLETA DE DADOS	58
7.7	EXECUÇÃO E ANÁLISE DOS DADOS	61
7.8	CONCLUSÃO DO ESTUDO DE CAMPO	63
8	GERÊNCIA DE RISCOS EM PROJETOS COM SCRUM	68
8.1	SUGESTÃO DE EXTENSÃO DO SCRUM	68
8.1.1	DIMENSÃO DE AMBIENTE ORGANIZACIONAL – PORTFÓLIO - SCRUM OF SCRUMS	68
8.1.2	DIMENSÃO DE COMPLEXIDADE – SPRINT ZERO E PESQUISA E DESENVOLVIMENTO	72
8.1.3	DIMENSÃO DE EQUIPE – PROGRAMAÇÃO EM PAR	73
9	CONCLUSÕES	77
9.1	CONTRIBUIÇÃO	77
9.2	LIMITAÇÕES	78
9.3	TRABALHOS FUTUROS	78
	REFERÊNCIAS BIBLIOGRÁFICAS	79
	ANEXO I – FICHA DE ENTREVISTA	85
	ANEXO II – ROTEIRO DE ENTREVISTA	89
	ANEXO III – PLANILHA DE RESPOSTAS DIRETAS DAS ENTREVISTAS	90
	ANEXO IV – RESPOSTAS DISSERTATIVAS DAS ENTREVISTAS	91

1 INTRODUÇÃO

A evolução tecnológica e a globalização dos negócios têm demandado adaptações nos processos industriais em todo o mundo, estimulando as empresas na busca por vantagens competitivas, reduzindo custos, mantendo qualidade e aumentando a produtividade [AP08]. Esta adaptação é fundamental para a continuidade dos negócios, pois pode definir a sobrevivência de uma empresa [AAS+08]. O mercado de desenvolvimento de software está inserido neste contexto [AAS+08, AMV06, BOE06], proporcionando o crescimento das áreas relacionadas ao gerenciamento de projetos, conforme apontam estudos publicados pelo *Project Management Institute* (PMI), através de sua série *PMI-Today*, afirmando que a profissão de gerenciamento de projetos tem expandido fortemente nos últimos anos [VAR05, PMI11].

Neste cenário, a indústria de desenvolvimento de software tem utilizado abordagens adaptativas para gerenciar projetos e o ciclo de desenvolvimento de produtos [FOR05, WG10], em detrimento ao uso de abordagens prescritivas, muitas vezes consideradas pesadas e lentas [GOD08, DBR06]. As abordagens adaptativas, por sua vez, enfatizam a agilidade dos processos de desenvolvimento de software, buscando maior eficiência em situações onde mudanças de requisitos de projetos são mais habituais [CLC04, PFL09, PRE10].

Uma das abordagens adaptativas mais utilizadas para o gerenciamento de projetos de desenvolvimento de software é o método ágil Scrum [FC08], que define um conjunto de boas práticas de gerenciamento através de ciclos iterativos e incrementais, com envolvimento e visibilidade constante do cliente, proporcionando entrega rápida de valor [CLC04, SCH04, FC08]. Entretanto, o gerenciamento de riscos (GR), que constitui uma prática muito relevante no gerenciamento de projetos [DL03, SCH07, PMB08], é tratado de forma implícita em projetos que utilizam abordagens adaptativas [CTH08, COH12] como o Scrum.

Neste contexto, este trabalho propõe um estudo que visa identificar de que maneira o gerenciamento de riscos é tratado no Scrum. O objetivo é identificar os riscos mais comuns apontados e tratados no gerenciamento de projetos de desenvolvimento de software [BOE91, SLK+01, WKA04, HH07] e verificar a forma como equipes de projeto que utilizam o Scrum tratam esses riscos comuns.

1.1 JUSTIFICATIVA

Este trabalho se justifica dentro da necessidade de investigar novas técnicas para gerenciamento de riscos em projetos de desenvolvimento de software, no contexto de abordagens adaptativas, especificamente com o método ágil Scrum. Os projetos de desenvolvimento de software têm utilizado abordagens adaptativas ao invés de abordagens prescritivas [AP08, SCH07] proporcionando o surgimento de novas oportunidades de pesquisa, conforme apontam [HSW+08], onde pode haver contribuição significativa, tanto para a academia quanto para a indústria.

Desta forma, o estudo dos impactos destas mudanças no contexto de gerenciamento de projetos e suas áreas de conhecimento surge como uma oportunidade. Neste sentido, o gerenciamento de riscos será investigado, pela necessidade de acompanhar as mudanças de cenários e processos de gerenciamento de projetos em abordagens prescritivas e adaptativas [LDO03]. Para tal, serão realizados estudos sobre gerenciamento de riscos no contexto de projetos com equipes co-locadas e métodos ágeis, tendo o framework Scrum como referência [HSW+08, DM06].

Para contemplar um estudo científico que considere contribuir com o cenário exposto acima, a questão de pesquisa que norteia este estudo é: *Como o gerenciamento de riscos é tratado em projetos que utilizam o Scrum?*

1.2 ORGANIZAÇÃO DO VOLUME

Este trabalho está distribuído em 9 capítulos contendo também as referências bibliográficas e uma sessão de anexos. A introdução está descrita no Capítulo 1. Os objetivos gerais e específicos se encontram no Capítulo 2. No Capítulo 3 está descrito todo o referencial teórico utilizado como apoio inicial neste estudo, discriminando conceitos de engenharia de software, gerenciamento de projetos, gerenciamento de riscos, além de uma explanação sobre métodos ágeis e o Scrum, além de trabalhos relacionados.

No Capítulo 4 é possível identificar uma descrição da metodologia científica aplicada neste trabalho e a justificativa pela escolha do método utilizado. O processo de obtenção da lista de riscos comuns em projetos de desenvolvimento de software está descrito no Capítulo 5, enquanto que a revisão sistemática da literatura executada neste estudo está detalhada no Capítulo 6. O Capítulo 7 descreve o Estudo de Campo realizado. Finalmente, o Capítulo 8 traz uma proposta de extensão do Scrum, a partir dos resultados encontrados no estudo e o Capítulo 9 conclui o trabalho.

2 OBJETIVOS

Os objetivos deste trabalho foram definidos a partir de oportunidades identificadas na literatura, bem como na motivação do aluno em aprofundar seu conhecimento na sua área profissional, onde atua como gerente de projetos aplicando práticas ágeis tendo por base o Scrum. As seções 2.1 e 2.2 seguintes apresentam o objetivo geral e os específicos desta pesquisa, respectivamente.

2.1 OBJETIVO GERAL

O objetivo geral dessa pesquisa é identificar como o gerenciamento de riscos é tratado no método ágil Scrum, buscando verificar como os riscos comuns identificados no gerenciamento de projetos de desenvolvimento de software estão sendo tratados em projetos que utilizam o Scrum.

2.2 OBJETIVOS ESPECÍFICOS

Os seguintes objetivos específicos foram definidos para este trabalho:

- Aprofundar os estudos da base teórica;
- Identificar os riscos comuns em projetos de desenvolvimento de software;
- Realizar uma revisão sistemática da literatura para identificar estudos relacionados e métodos científicos adotados para estudos similares;
- Conduzir uma pesquisa de campo com o objetivo de colher as experiências de gerentes de projetos que utilizam Scrum;
- Analisar os dados coletados e identificar como os riscos comuns em projetos de desenvolvimento de software são tratados no Scrum;
- Identificar a necessidade de sugerir extensões para o Scrum.

3 REFERENCIAL TEÓRICO

Na Seção 3.1, os conceitos de gerenciamento de projetos são descritos, apontando sua origem e a importância no contexto das empresas. A Seção 3.2 descreve as abordagens de gerenciamento de projetos, destacando as principais diferenças entre abordagens prescritivas e adaptativas. O método ágil Scrum é descrito na Seção 3.3, incluindo os principais papéis e processos desta abordagem adaptativa. A Seção 3.4 apresenta a importância e os fatores relevantes ao gerenciamento de riscos em projetos de desenvolvimento de software. Os trabalhos relacionados são apresentados e analisados na Seção 3.5.

3.1 GERENCIAMENTO DE PROJETOS DE SOFTWARE

De acordo com Denis Alcides Rezende [REZ05], software é um subsistema de um sistema computacional. A sua utilização tem aumentado significativamente nos mais variados mercados e domínios de conhecimento, impulsionado pela alta competitividade das empresas e consequente busca de melhores resultados com menor custo, apoiadas na tecnologia para alcançarem tal objetivo [AP08].

O desenvolvimento de software como produto é sustentado pela Engenharia de Software, que segundo a IEEE é definida como sendo “*a aplicação sistemática, disciplinada e com abordagem quantitativa para o desenvolvimento, operação e manutenção de software*” [PRE10]. Desta forma, a Engenharia de Software constitui um tema importante que abrange os variados fatores da produção de software [SOM07], conduzida através de atividades definidas em processos e instanciadas em projetos de desenvolvimento de software.

O PMBOK, da sigla em inglês para *Corpo de Conhecimento em Gerenciamento de Projetos – Project Management Body of Knowledge*, desenvolvido a cada 4 anos pelo *PMI – Project Management Institute* com a contribuição dos associados profissionais em gerenciamento de projetos distribuídos em todo o mundo, define que projeto é um empreendimento único com início e fim claramente definidos, sendo conduzido por pessoas para que possa atingir seus objetivos, respeitando o custo, prazo e qualidade estipulados para determinar o sucesso do empreendimento [PMB08].

Projetos envolvem certo grau de incerteza, pois são únicos e inicialmente não conhecidos ou entendidos de forma incompleta. A execução de projetos é dividida em fases, com o objetivo de melhorar o controle e gerenciamento, minimizando os riscos

que possam ocorrer. As fases dos projetos são denominadas Ciclo de Vida do Projeto e conectam o início e o fim do projeto. Cada fase é caracterizada com o término e a aceitação de um conjunto de produtos, ou seja, um resultado mensurável e verificável do trabalho [OLI06, PMB08, PZB10].

O gerenciamento de projetos (GP) é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos [PMB08]. O GP é realizado através da aplicação e da integração dos processos de iniciação, planejamento, execução, monitoramento e controle, e encerramento. A responsabilidade por alcançar os objetivos do projeto incide sobre o gerente de projetos.

3.2 ABORDAGENS DE GERENCIAMENTO DE PROJETOS

Tradicionalmente, os projetos são gerenciados através de abordagens prescritivas, onde é exigido o cumprimento de etapas rígidas em processos que muitas vezes se tornam pesados e lentos [GOD08, DBR06]. No entanto, as mudanças do mercado de desenvolvimento de software sugerem também abordagens adaptativas para GP, que enfatizam a agilidade do processo de desenvolvimento de software ao qual o projeto está inserido, seguindo princípios que levam a uma condução mais eficiente, com características de gerenciamento adaptáveis às situações onde as mudanças de requisitos dos projetos são mais frequentes [CLC04, PFL09, PRE10].

Nas seções seguintes, serão detalhados os dois tipos de abordagens para GP de software: prescritivas e adaptativas.

3.2.1 ABORDAGENS PRESCRITIVAS

As abordagens prescritivas de gerenciamento de projetos buscam apoiar e dar consistência aos projetos por meio da definição de uma estrutura mais rígida de desenvolvimento, necessitando de um maior controle no cumprimento de todas as etapas definidas para o projeto [GOD08]. De Bortoli et al [DBR06] afirmam que estes processos são considerados pesados e lentos, quando se aplicam a projetos com curto prazo de desenvolvimento, sendo mais adequados nas situações em que os requisitos não mudam constantemente, ou seja, são estáveis e previsíveis.

Na definição de projetos conduzidos de maneira prescritiva, para que ocorra a definição da arquitetura do sistema sendo desenvolvido, é necessário que a análise de requisitos seja desenvolvida de maneira detalhada e profunda. Para dar sequência ao processo de desenvolvimento do projeto, é necessário que a definição da arquitetura

seja elaborada com detalhes precisos de todos os componentes do projeto. Por fim, após o desenvolvimento do projeto, deve ser conduzida uma sequência de testes completa, possibilitando que o software gerado seja entregue com qualidade ao cliente.

Considerando o modelo prescritivo para condução de um projeto de software, deve-se buscar o planejamento prévio como forma de evitar mudanças de escopo no decorrer do projeto em andamento.

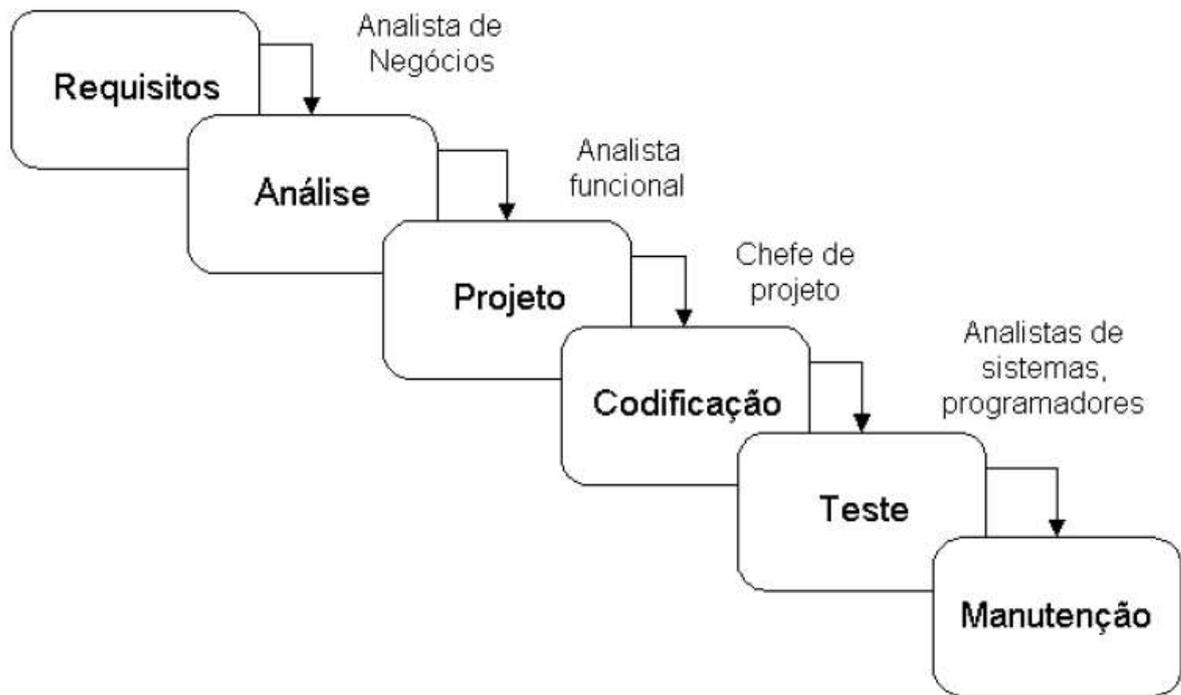


Figura 1 – Modelo de Processo em Cascata

O ciclo de vida cascata é o modelo prescritivo mais conhecido [PRE10]. Idealizado em 1970 por Winston W. Royce [ROY70] recebeu este nome em virtude da sequência em cascata de uma fase para outra. Possui um método linear e sequencial, onde cada fase deve ser concluída para que a seguinte seja iniciada, conforme demonstrado na Figura 1.

3.2.2 ABORDAGENS ADAPTATIVAS

As abordagens adaptativas representam métodos baseados na prática de gerenciamento e desenvolvimento efetivos de sistemas de software. Aplicam um conjunto de práticas guiadas por princípios e valores que podem ser aplicados por profissionais de software no dia a dia, onde a comunicação é um dos fatores de grande importância [CLC04]. Um dos exemplos mais conhecidos de abordagens adaptativas são os métodos ágeis.

O desenvolvimento ágil de software evoluiu a partir da metade de 1990 em decorrência da reação contra abordagens classificadas como pesadas, caracterizadas pela prescrição do planejamento e micro gerenciamento usando o modelo em cascata para desenvolvimento. O processo originou-se da visão de que o modelo em cascata era burocrático, lento e contraditório se comparado à forma usual com que os profissionais de desenvolvimento de software realizam seus projetos [CLC04].

No ano de 2001, entre os dias 11 e 13 de Fevereiro, 17 desenvolvedores experientes da comunidade se reuniram na cidade de *Snowbird*, estado de Utah nos Estados Unidos da América, para sintetizarem suas ideias, publicando o *Manifesto Ágil*, documento que reúne os princípios e valores desta abordagem de desenvolvimento. Mais tarde, algumas pessoas formaram a *Agile Alliance*, uma organização não lucrativa que promove o estudo, divulgação e evolução dos métodos ágeis [CLC04].

O Manifesto Ágil foi proposto levando em consideração a priorização dos conceitos básicos que regem qualquer interação entre pessoas, processos, produtos finais, contratos, documentação, etc. Definido o que seria mais importante nessas relações, chegou-se ao modelo exposto na Figura 2 abaixo [CP10].

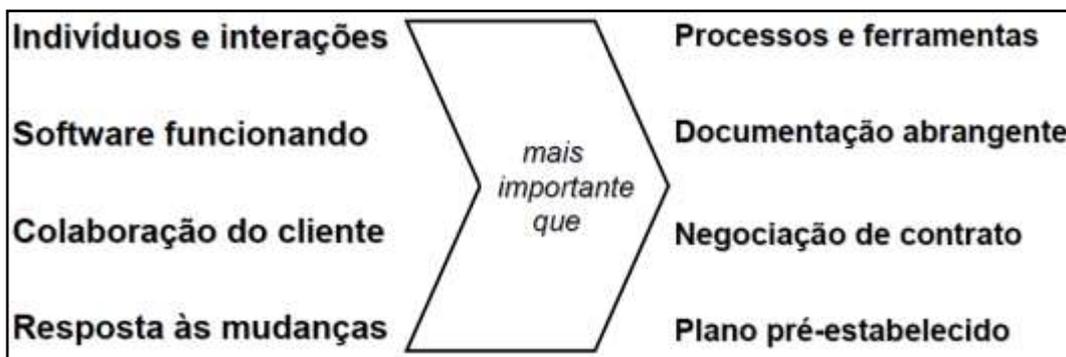


Figura 2 – Priorização de conceitos de relações

Este novo posicionamento exige uma nova postura dos gerentes de projetos que utilizam esta abordagem, considerando novos princípios para a condução dos trabalhos [CP10, SIL09]:

- Indivíduos e iterações são mais importantes que processos e ferramentas;
- Software funcionando é mais do que documentação abrangente;
- Colaboração do cliente é mais importante que negociação de contratos;
- Responder a mudanças é mais importante que seguir um plano.

Como resultado desta reflexão, se obtém um novo modelo onde o gerente do projeto é responsável pela integração das equipes envolvidas no projeto, exercendo um papel de facilitador das tarefas. Esta nova definição contrasta com as abordagens

tradicionais que apontam o gerente do projeto como o principal condutor das atividades do projeto. A Figura 3 demonstra as diferenças no envolvimento das equipes com o gerente do projeto (GP na Figura) e com as demais equipes [CP10, SIL09].

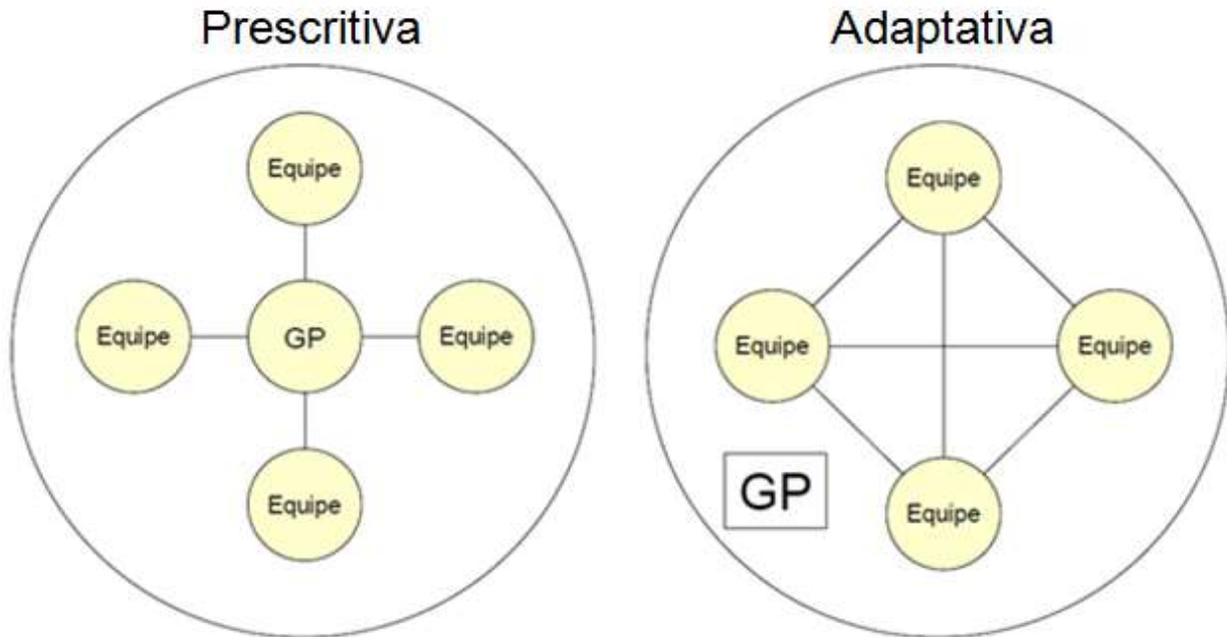


Figura 3 – Novo papel do Gerente de Projetos em abordagens adaptativas

De forma geral, métodos ágeis priorizam software funcionando como principal métrica de progresso. Tendo em vista a preferência por comunicação presencial, as abordagens adaptativas normalmente produzem menos documentação do que abordagens prescritivas [CP10].

Projetos com abordagens adaptativas tem maior aceitação à mudanças se comparados àqueles com abordagens prescritivas, como no modelo cascata onde a alteração de escopo implica em um processo mais custos de mudança [AMB08].

3.3 O MÉTODO ÁGIL SCRUM

O Scrum é um método ágil utilizado para gerenciar o desenvolvimento de um projeto através de práticas iterativas e incrementais. É composto por um conjunto de boas práticas de gerenciamento que aceita ajustes rápidos, além de acompanhamento e visibilidade constantes [FC08, PZB10, ETT12].

O desenvolvimento de software em projetos que utilizam o Scrum como método é dividido em ciclos chamados de *Sprint*, sendo que cada ciclo ocorre em intervalos entre 2 e 4 semanas. As equipes são pequenas e formadas por projetistas, programadores, engenheiros de software e gerentes de qualidade que desenvolvem suas tarefas a partir

dos requisitos ou *Product Backlog* [SCH04, ETT12]. Os papéis e responsabilidades do Scrum são:

- **Product Owner:** representa os interesses de todos os envolvidos no projeto. Define os fundamentos do projeto criando requisitos iniciais e gerais (*Product Backlog*), objetivos e planos de entregas. Prioriza os itens do *Product Backlog* em cada *Sprint*, compondo a *Sprint Backlog* e solicitando que sejam priorizadas as funcionalidades de maior importância para o negócio.
- **ScrumMaster:** garante que o método está sendo seguido. Deve repassar o conhecimento sobre Scrum a todos os envolvidos no projeto, ajustando o método de modo que seja melhor adequado à cultura da empresa. Deve garantir que todos sigam as regras e práticas do Scrum. É responsável por remover os impedimentos do projeto.
- **ScrumTeam:** Representa a equipe que é responsável pelo sucesso de cada *Sprint* e pelo sucesso do projeto como um todo. A equipe é responsável pelas funcionalidades do produto, devendo analisar, desenvolver e testar cada um dos itens das *Sprints*. Devem ser autogerenciáveis e multiespecialistas.

Conforme Schwaber [SCH04, PZB10] o Scrum possui ciclo de vida composto por três fases:

- **Pre-game:** Nesta fase se estabelece a visão do projeto e alocação de recursos para a sua execução. São criadas as versões iniciais do *Product Backlog*, o plano de instalação e a arquitetura de negócio. As equipes são formadas e mecanismos de comunicação e coordenação são definidos;
- **Game:** Esta fase é contemplada através das *Sprints* onde ocorre o desenvolvimento iterativo e incremental dos itens do *Product Backlog* priorizados no *Pre-game*;
- **Post-game:** Nesta fase ocorre a entrega do produto final ao cliente.

A definição das tarefas que irão compor o *Sprint* é realizada no início de cada ciclo durante a reunião inicial conhecida como *Planejamento de Sprint*. Cada profissional recebe a responsabilidade pelo desenvolvimento de um conjunto de atividades que tem o objetivo comum de atender aos requisitos selecionados pelo *Product Owner* para compor a *Sprint Backlog* [SCH04, PZB10, ETT12]. A engrenagem do Scrum pode ser observada na Figura 4.

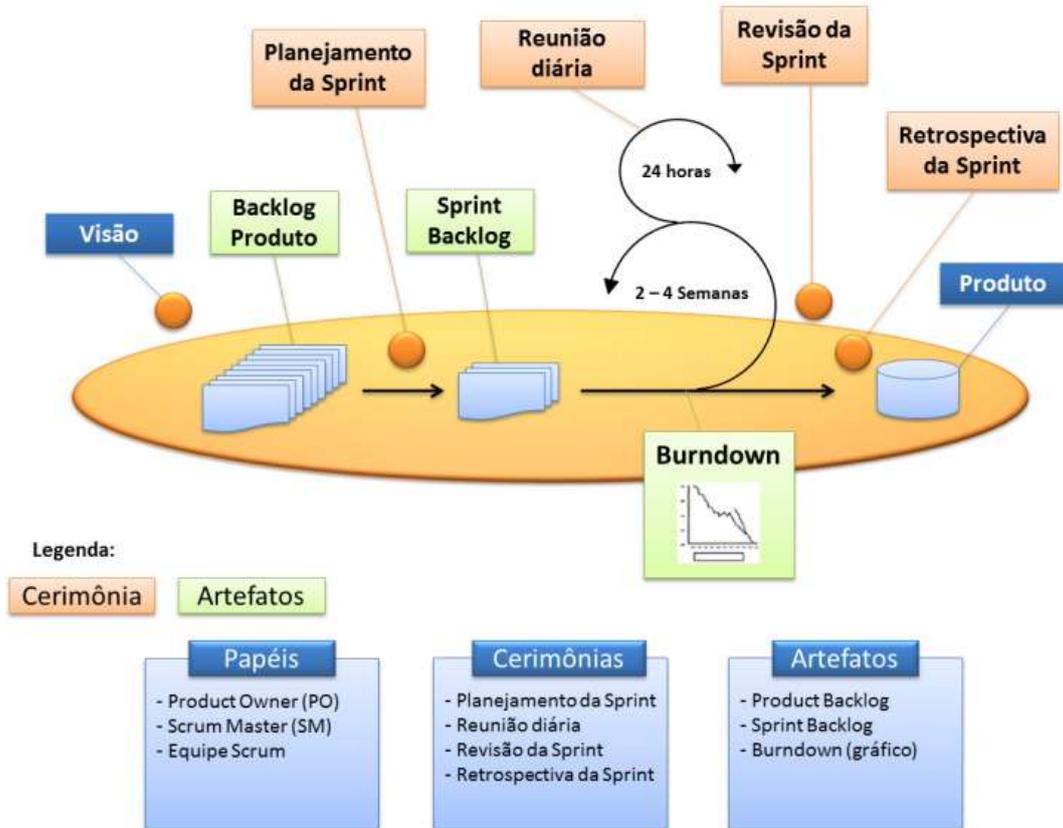


Figura 4 – A engrenagem do Scrum [ETT12]

Um dos princípios do método Scrum estabelece que sejam realizadas reuniões diárias de aproximadamente 15 minutos, denominadas *Daily Scrum Meeting* ou *Daily Standups* onde a equipe expõe ao *ScrumMaster* o que foi desenvolvido até o momento e o que será realizado em seguida, até a próxima reunião diária. Nestas reuniões a equipe também lista os fatores de impedimento e o progresso geral do desenvolvimento [SCH04, ETT12]. Todos os membros da equipe respondem questões sobre o andamento das atividades do projeto e reportam eventuais problemas que estão sendo enfrentados [PZB10, ETT12].

Após o término de cada *Sprint* é realizada uma reunião para avaliação do produto entregue, onde a equipe deve apresentar o que foi realizado. A *Revisão da Sprint* ou *Sprint Review* deve também validar a entrega do produto com as partes interessadas, garantindo a entrega de valor ao cliente com maior frequência. A equipe deve ainda discutir sobre o andamento das atividades realizadas durante a *Sprint*, analisar problemas enfrentados e como eles foram tratados. A reunião *Revisão da Sprint* possibilita a geração de informações de extremo valor para as próximas reuniões de *Planejamento de Sprint* ou *Sprint Planning*, ao iniciar novo ciclo de desenvolvimento [SCH04, ETT12].

A reunião denominada Retrospectiva da *Sprint* ou *Sprint Retrospective* é o passo seguinte e deve ocorrer antes da próxima reunião de *Planejamento da Sprint*. Durante essa retrospectiva o *ScrumMaster* encoraja a equipe a revisar seu processo de desenvolvimento de forma a torná-lo mais produtivo para a próxima *Sprint*. Neste momento deve ser verificado como ocorreu a interação entre as pessoas, processos e ferramentas durante a última *Sprint*. Ao final dessa etapa a equipe deve ter identificado melhorias factíveis que deverão ser incorporadas na próxima *Sprint* [SCH04, ETT12].

O monitoramento de projetos conduzidos com Scrum é realizado através de informações do *gráfico de Burndown* [FC08, NSC+12], cuja interpretação permite:

- Medir o progresso e velocidade da equipe;
- Apresentar a quantidade de trabalho restante;
- Analisar se a equipe é organizada;
- Avaliar riscos no projeto.

Para observar o andamento do desenvolvimento do projeto, o *gráfico de Burndown* é utilizado, ilustrando o trabalho restante em relação ao tempo. O eixo horizontal do gráfico exibe o tempo, enquanto que o eixo vertical mostra a quantidade de trabalho (pontos de história, horas de trabalho, dias de equipe) restante [NSC+12]. A Figura 5 demonstra um exemplo de *gráfico de Burndown*.

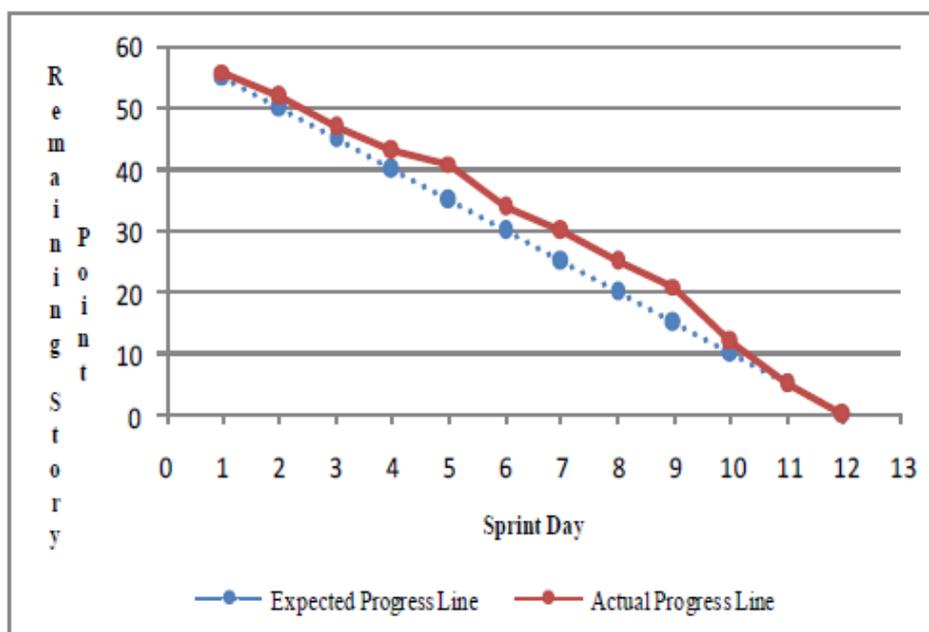


Figura 5 – Gráfico Sprint Burndown [NSC+12]

O gráfico de Burndown demonstra o progresso diário da *Sprint*, onde o trabalho restante deve reduzir a cada dia em um *Sprint*. O ideal é que a estimativa siga uma linha reta entre o ponto de esforço total do projeto, no eixo vertical, até a data definida para a

final da *Sprint*, no eixo horizontal. No entanto, em tempo real, alguns desvios da linha reta real podem existir devido a erros de estimativa, algum impedimento ou ainda a adição de novas exigências por parte do proprietário do produto.

3.4 GERENCIAMENTO DE RISCOS

Nesta seção serão abordados alguns dos principais aspectos relacionados ao Gerenciamento de Riscos (GR) em projetos de desenvolvimento de software. A importância do GR para o sucesso do projeto, o contexto histórico do tema, bem como a maneira como este assunto é tratado no âmbito das organizações também são pontos tratados nesta seção.

3.4.1 CENÁRIO HISTÓRICO

A palavra risco vem da antiga palavra italiana *risicare*, que significa ousar. Boehm foi quem primeiramente examinou o risco por meio do modelo espiral: análise de risco em cada iteração de desenvolvimento de software [BOE91]. DeMarco e Lister, por outro lado, definem o risco como um problema que ainda não ocorreu, enquanto que um problema é um risco que já se materializou [DL03]. Antes que aconteça, o risco é apenas uma abstração, algo que pode ou não afetar um projeto [PEA+08].

A definição da palavra risco no dicionário Aurélio da língua portuguesa [FER04] está associada a exposição ao perigo ou dano, com conotação negativa da palavra. No contexto de desenvolvimento de software, no entanto, a palavra risco está relacionada também a um sentido positivo, onde risco está associado a oportunidade. Considerando a característica dos projetos em produzir algo único, os ambientes onde estão sendo conduzidos tornam-se ambientes de incertezas [SCH07], sugerindo tanto riscos de ameaças quanto oportunidades [KNO07]. Como consequência destas incertezas presentes em todos os projetos, torna-se importante um GR adequado dos riscos oriundos deste ambiente [PMB08]. A Figura 6 apresenta uma visão geral dos processos de gerenciamento de riscos apontados como melhores práticas pelo Guia PMBOK [PMB08], onde os riscos devem ser tratados pelos processos de Planejamento do Gerenciamento de Riscos, Identificação dos Riscos, Análise Qualitativa e Quantitativa de Riscos, Planejamento de Respostas a Riscos, Monitoramento e Controle de Riscos.

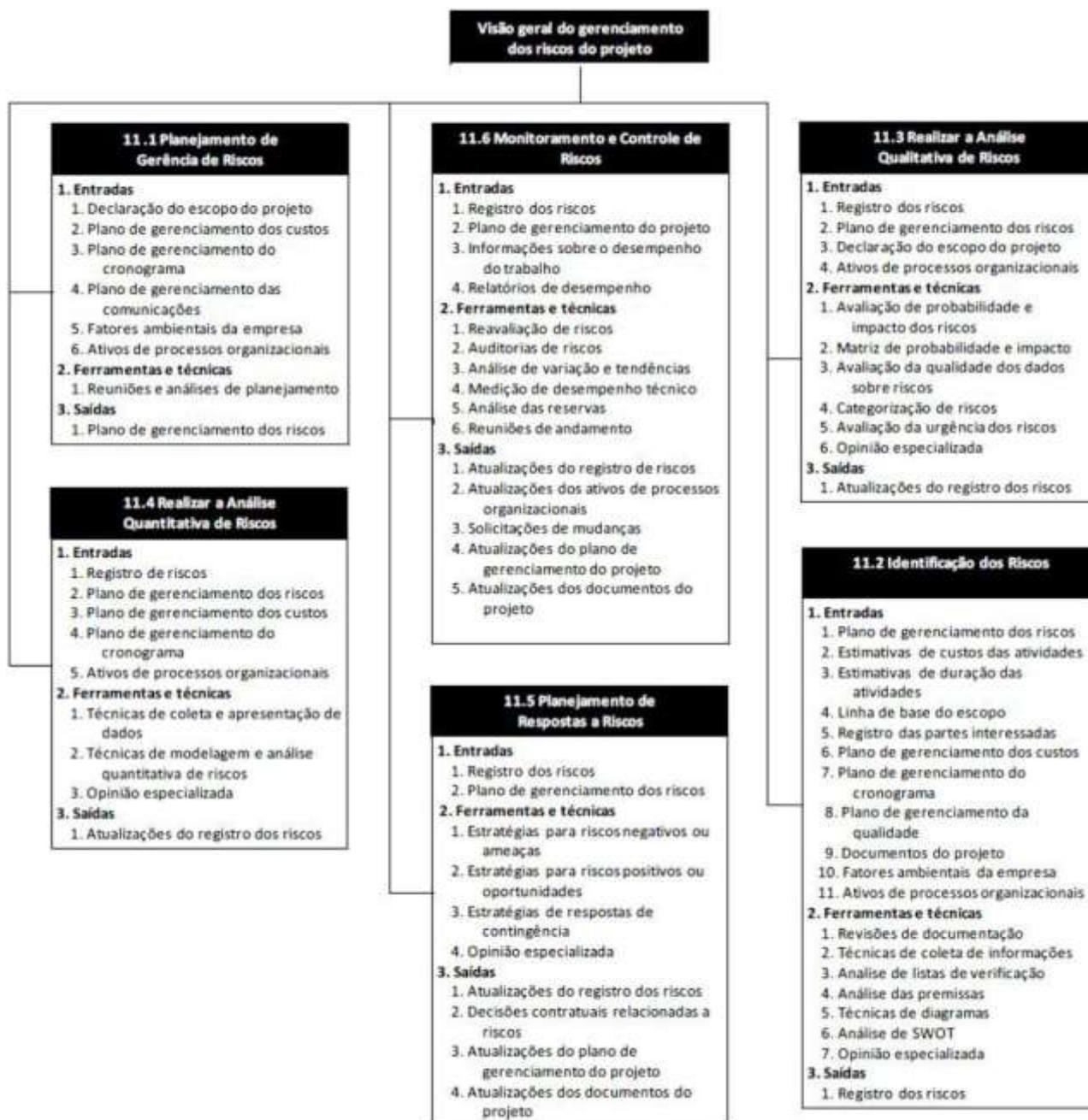


Figura 6 – Resumo do Gerenciamento de Riscos em Projetos [PMB08]

Estes processos interagem entre si com os processos das outras áreas de conhecimento, sendo que cada processo pode ocorrer pelo menos uma vez em uma ou mais fases do projeto, possibilitando que o GR seja uma atividade constante durante a condução do projeto, levando em consideração que o objetivo do GR em qualquer projeto é aumentar a probabilidade e o impacto dos eventos positivos e diminuir a probabilidade e o impacto dos eventos adversos ao projeto [PMB08].

3.4.2 IMPORTÂNCIA DO GERENCIAMENTO DE RISCOS EM PROJETOS

A primeira proposta para incluir o gerenciamento de riscos no ciclo de vida de desenvolvimento de software foi feita no final dos anos 80, quando Barry Boehm propôs uma abordagem adaptativa de desenvolvimento em espiral [BOE88, PRE10]. As principais características desta proposta são a iteratividade e o fato de ser dirigida por riscos, ocorrendo uma nova análise dos riscos do projeto a cada ciclo.

Mais tarde, em 1991, Barry Boehm publicou um artigo específico sobre GR em projetos de desenvolvimento de software [BOE91]. Ao longo de sua carreira, Boehm teve a chance de observar diversos gerentes de projetos em ação e com isso pode identificar características que diferenciam os mais eficientes dos menos eficientes. Como resultado destas observações, ele percebeu uma característica presente em todos os gerentes de projetos eficientes é que estes são ótimos gerentes de riscos [BOE91].

O artigo publicado por Boehm teve como objetivo tentar definir princípios e práticas baseados nesta característica comum aos gerentes de projetos observados. Desde então, o GR em projetos de desenvolvimento de software vem crescendo consideravelmente. No início da década de 90, as metodologias de gerenciamento de projetos costumavam deixar o GR em segundo plano [SEI11], normalmente dentro de alguma outra área de conhecimento. Atualmente, estas mesmas metodologias colocam o GR em posição de destaque, dedicando capítulos exclusivos para esta área de conhecimento. Foi assim com o Guia PMBOK, que em 1987 deu maior visibilidade ao GR dedicando uma área de conhecimento específica ao assunto [PMB08], e do CMMI, que ao evoluir do SW-CMM reuniu as práticas referentes ao GR, até então inclusas dentro de outras áreas chave de processo, em uma área de processo também específica para o assunto [SEI11].

Segundo DeMarco e Lister [DL03], organizações que preferem fugir dos riscos, mantendo o foco apenas naquilo em que são especialistas, estão perdendo espaço para a concorrência. Este fato se confirma, pois ao proceder desta maneira estas organizações estão deixando de aproveitar novas oportunidades. Conforme Kruchten [KRU03], em um ciclo de vida iterativo, muitas decisões são conduzidas pela análise dos riscos do projeto. Para tomar decisões efetivas é preciso ter uma compreensão muito clara dos riscos que ameaçam o projeto e de estratégias que possam reduzir o impacto destes riscos caso os mesmo venham a ocorrer.

Schwalbe [SCH07] diz que, apesar de frequentemente ser um fator decisivo para que o projeto seja bem sucedido, o GR é ainda um aspecto ignorado dentro do gerenciamento de projetos, ressaltando as seguintes vantagens da aplicação de GR:

- Auxiliar a seleção de projetos;
- Ajudar a determinar o escopo de projetos;
- Ajudar a desenvolver cronogramas e estimativas de custos realistas;
- Ajudar os interessados do projeto a entenderem a natureza do projeto;
- Envolver a equipe do projeto na definição de pontos fortes e fracos;
- Integrar as demais áreas de conhecimento no gerenciamento de projetos.

Assim como DeMarco e Lister [DL03], Schwalbe também ressalta que organizações não devem fugir dos riscos, pois podem estar deixando de aproveitar grandes oportunidades. Dado que todos os projetos envolvem riscos e oportunidades, a questão consisti em saber decidir quais projetos são vantajosos para a organização e como os riscos do projeto serão gerenciados ao longo do seu ciclo de vida [KNO07].

3.4.3 GERENCIAMENTO DE RISCOS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Segundo o PMBOK, um projeto está aplicando o GR se neste projeto é feito um planejamento dos riscos e oportunidades envolvidas através da identificação, análise, monitoramento e controle dos incidentes que possam ocorrer [PMB08].

Sabe-se que o GR é frequentemente ignorado dentro de projetos de software [SCH07]. Este fato foi comprovado nos resultados da pesquisa realizada por William Ibbs e Young H. Kwak com empresas de diversos segmentos que identificou a maturidade destas organizações em cada uma das áreas de conhecimento do gerenciamento de projetos [SCH07], conforme resultados apresentados na Tabela 1.

Tabela 1 – Maturidade das organizações em Gerenciamento de Projetos [SCH07]

Área de conhecimento	Engenharia / Construção	Telecomunicações	Sistemas de Informação	Indústria de alta tecnologia
Escopo	3.52	3.45	3.25	3.37
Tempo	3.55	3.41	3.03	3.50
Custo	3.74	3.22	3.20	3.97
Qualidade	2.91	3.22	2.88	3.26
Recursos Humanos	3.18	3.20	2.93	3.18
Comunicações	3.53	3.53	3.21	3.48
Risco	2.93	2.87	2.75	2.76
Aquisições	3.33	3.01	2.91	3.33
Legenda: 1 → Mais baixo nível de maturidade; 5 → Mais alto nível de maturidade				

Percebe-se que o segmento de sistemas de informação obteve o menor nível de maturidade entre todos os segmentos pesquisados na área de conhecimento de gerenciamento de riscos. Segundo o Guia PMBOK, para terem sucesso, as organizações devem estar comprometidas com o GR em todos os projetos [PMB08].

Organizações que adotam práticas de GR podem ter diferentes capacidades de tolerância aos riscos. Segundo Schwalbe é importante que a organização e os interessados do projeto conheçam e aceitem a sua capacidade de tolerância aos riscos. A autora sugere que existem três perfis de tolerância aos riscos [SCH07]:

- Aversão aos riscos;
- Indiferença aos riscos;
- Amante dos riscos.

Estes três perfis são definidos por uma função que relaciona a tolerância aos riscos por parte do tomador de decisão em relação aos valores envolvido na respectiva tomada de decisão [SCH07].

Prikladnicki [PEA+08] acrescenta que gerenciar riscos com sucesso requer mais do que bons processos e a capacidade de pensar intuitivamente. Exige também disciplina e método. DeMarco e Lister [DL03] definem GR como o processo de reflexão sobre as ações corretivas antes de ocorrer um problema, enquanto ainda é uma abstração. O oposto de GR é o gerenciamento de crises, que consiste em tentar resolver um problema depois que ele ocorre, elevando os impactos e custos no projeto.

Percebe-se, pelos depoimentos obtidos nos estudos pesquisados, que o Gerenciamento de Riscos constitui uma área de conhecimento que não deve ser ignorada pelas organizações, principalmente no que tange o gerenciamento de projetos de desenvolvimento de software.

3.5 TRABALHOS RELACIONADOS

Dentro dos cenários de desenvolvimento de softwares e abordagens estudados, foi realizada uma busca por estudos na área de gerenciamento de riscos que complementassem as informações pesquisadas neste trabalho, com o objetivo de expandir o conhecimento teórico a respeito do tema.

O foco principal do estudo da base teórica e dos trabalhos relacionados diz respeito às abordagens prescritivas e adaptativas no contexto de gerenciamento de risco

em projetos de desenvolvimento de software. Desta forma, os estudos que contemplam estes critérios foram identificados e são apresentados a seguir.

3.5.1 A FERRAMENTA RISKFREE

A ferramenta RiskFree [KSP+06] foi projetada com base nas boas práticas descritas pelo Guia PMBOK. A motivação para o desenvolvimento deste trabalho foi disponibilizar uma ferramenta que auxiliasse equipes de desenvolvimento de software na realização das atividades de GR em projetos de software com equipes de desenvolvimento coalocadas. A maior ligação entre o Guia PMBOK e a ferramenta está na definição do processo de gerenciamento de riscos. Cada atividades do processo possui objetivos específicos:

- **Planejar gerenciamento de riscos:** tem como principal objetivo a elaboração do plano de gerenciamento de riscos do projeto. Segundo o Guia PMBOK, este plano deve definir como e quando as atividades de identificação, análise, planejamento de resposta, monitoração e controle dos riscos irão ocorrer ao longo do projeto. Outros aspectos como orçamento, metodologias, papéis, responsabilidades e formato de relatórios também podem constar no plano;
- **Identificar riscos:** tem como principal objetivo identificar e classificar os riscos que afetam o projeto. Para cada risco, são identificados também seus sintomas e sinais de alerta. Este processo caracteriza-se por ser iterativo, à medida que o projeto avança novos riscos vão sendo identificados;
- **Analisar riscos:** para cada risco identificado deve-se realizar uma atividade de análise que tem como objetivo verificar a probabilidade de ocorrência do risco e o seu impacto em relação aos objetivos do projeto. A atividade de análise é composta pela análise qualitativa, com o objetivo de priorizar os riscos de acordo com a sua probabilidade de ocorrência e impacto caso o risco venha a ocorrer e pela análise quantitativa, com o objetivo de analisar numericamente a probabilidade e eventuais consequências de cada risco;
- **Planejar resposta aos riscos:** tem como principal objetivo determinar ações para aproveitar as oportunidade e endereçar os riscos do projeto. Esta atividade inclui a atribuição de responsabilidades para cada risco identificado, garantindo um melhor controle sobre os riscos do projeto;

- **Monitorar e controlar riscos:** tem como principal objetivo garantir que o plano de gerência de riscos seja seguido e que os riscos identificados e endereçados estejam devidamente controlados. Esta atividade caracteriza-se por ser contínua dentro do ciclo de vida do projeto.

A Figura 7 abaixo apresenta o mapeamento realizado entre o processo descrito no Guia PMBOK e o processo proposto. Percebe-se que a única adaptação realizada está na etapa de análise dos riscos, a qual é dividida em atividades de análise qualitativa e quantitativa por PMI, mas que no processo proposto foi unificada em apenas uma única etapa de análise [PMB08]. O objetivo foi facilitar a implantação do processo de gerenciamento de risco e estar de acordo com o modelo CMMI, que não descreve explicitamente esta divisão em análise qualitativa e quantitativa [KNO07].

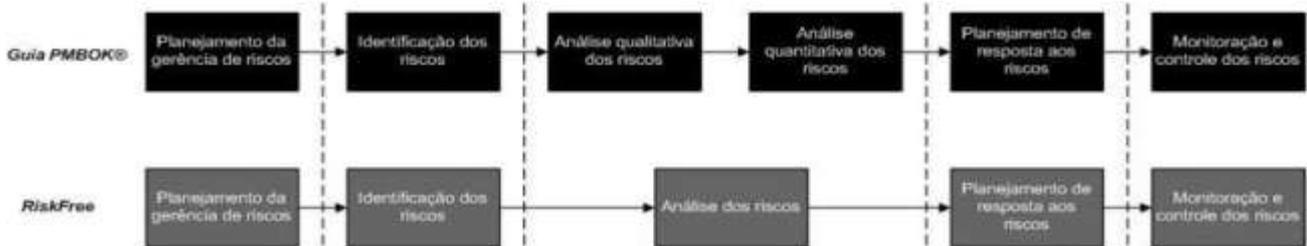


Figura 7 – Comparação entre o Guia PMBOK e a ferramenta RiskFree [KNO07]

A ferramenta RiskFree foi concebida pensando nas exigências de flexibilidade e adaptabilidade, permitindo que cada organização possa utilizar as técnicas que melhor atendem suas necessidades. Para isto, a construção da ferramenta possibilita que os componentes sejam vinculados a cada atividade do processo de GR.

Desta forma, a organização que fizesse uso da ferramenta poderia desenvolver componentes que atendessem às suas necessidades específicas, não ficando restrita a um conjunto limitado e pré-definido de técnicas.

- **Hibernate:** framework de persistência de dados utilizado para obter e persistir os dados utilizados pela ferramenta;
- **RiskFreeCore:** provê funcionalidades comuns a todos os componentes, incluindo as de acesso a dados e de autorização;
- **RiskFreeMain:** provê funcionalidades de administração e configuração da ferramenta, como o cadastro de usuários e instalação de novos componentes;
- **RiskFreeComponent:** são os componentes desenvolvidos e que implementam as técnicas e ferramentas envolvidas nas atividades que compõem o processo de gerenciamento de riscos proposto.

Os componentes desenvolvidos são instalados e vinculados a algum dos pontos adaptáveis da ferramenta, que no protótipo proposto no estudo de Knob, compreendem apenas as atividades do processo de GR, as informações gerais e resumidas sobre o projeto e os relatórios em nível de projeto e organizacional [KNO07].

3.5.2 A PROPOSTA NO-RISK

Para esta abordagem, foi realizado um estudo de vários processos de Gerenciamento de Riscos (GR) em projetos de software, tais como *Riskit*, *Odyssey*, *ARisk* e *GeRis*. Como resultado desta análise, baseado no modelo *Riskit*, desenhou-se um processo amplo cuja proposta é apresentar uma forma de gerenciar riscos de projetos de software focados em sistemas de informação [OLI06].

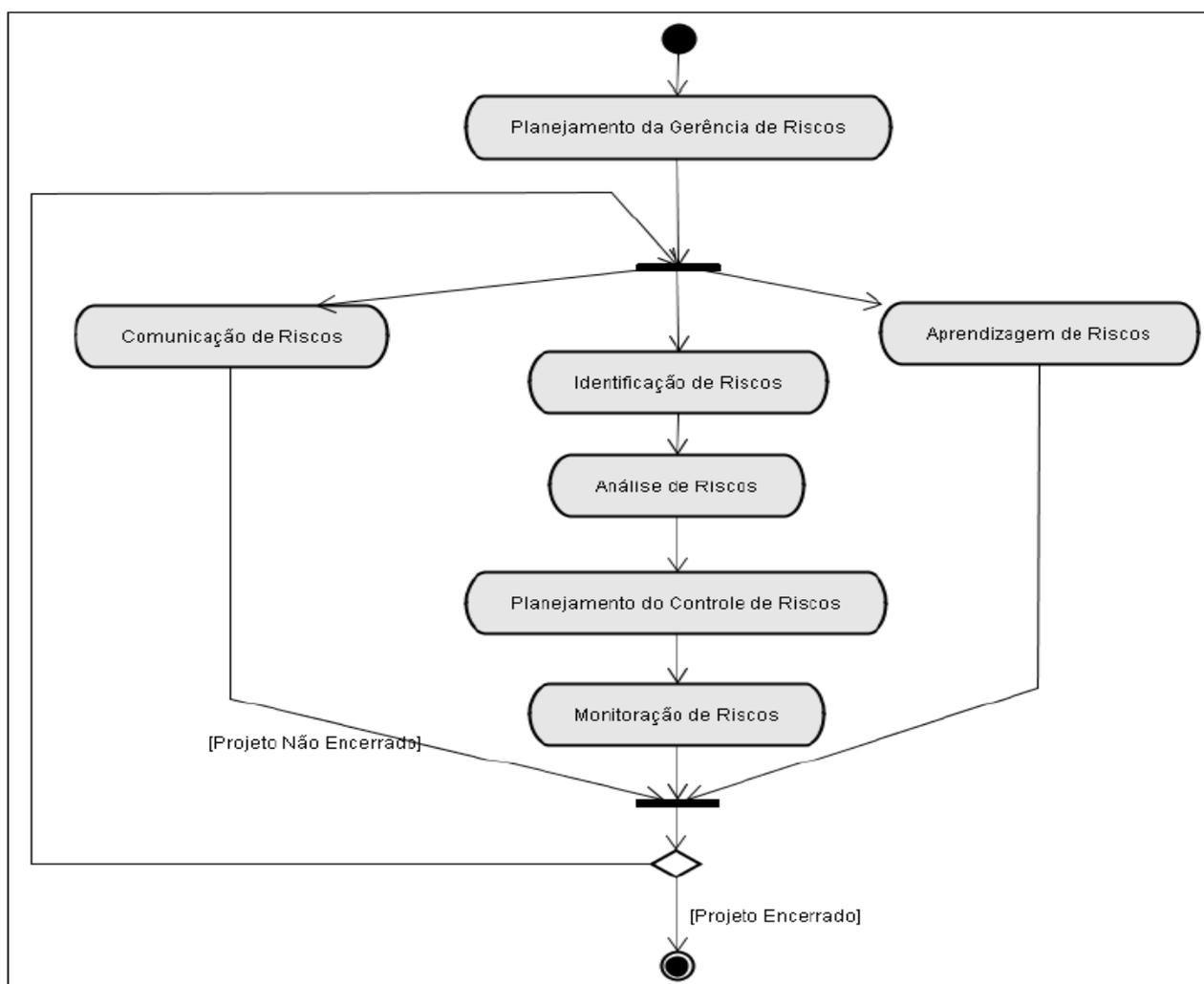


Figura 8 – Etapas da proposta No-Risk [OLI06]

Uma visão macro das etapas da proposta No-Risk é apresentada na Figura 8, através do diagrama de atividades acima. De acordo com esta proposta, as responsabilidades em cada etapa do projeto, relacionados ao GR, estão bem definidas:

- **Gerente do Projeto:** responsável pela condução do projeto;
- **Patrocinador (Sponsor):** são indivíduos ou organização envolvida nos trabalhos cujos interesses afetam o desenvolvimento do projeto;
- **Stakeholders:** pessoa ou organizações que estão ativamente envolvidos no projeto, ou cujos interesses podem ser, positiva ou negativamente, afetados pelos resultados do projeto.

A etapa inicial para este processo é a de Planejamento de GR. A partir desta etapa o processo entra em ciclo constante de monitoramento de riscos até que o projeto seja encerrado. Cabe ressaltar que as etapas de Aprendizagem e de Comunicação de Riscos são realizadas constantemente durante todo o ciclo de vida do desenvolvimento do projeto [OLI06].

3.5.3 RISK BACKLOG: UM MODELO DE GERENCIAMENTO DE RISCOS COM SCRUM

Este estudo empírico foi realizado em um trabalho de conclusão do Bacharelado em Ciência da Computação na PUCRS. Propõe a integração de processos de Gerenciamento de Riscos em projetos que utilizam o método ágil Scrum, buscando preservar os princípios do manifesto ágil. Foi elaborado com a justificativa de haver uma lacuna na utilização de GR nos métodos ágeis, considerando que a própria essência de métodos ágeis está em endereçar implicitamente os riscos de um projeto, entretanto sem que exista um processo explícito e sistemático para este fim [PZB10].

Para manter os princípios do manifesto ágil, que recomendam a utilização de comunicação visual e temporária, ao invés de repositórios eletrônicos, os autores propõem a utilização de um artefato denominado *Risk Backlog*, que seria exposto junto ao quadro de tarefas do projeto, permitindo a identificação e qualificação dos riscos, bem como a definição e acompanhamento das ações para resolução dos riscos apontados de forma visual, mantendo assim os princípios do manifesto ágil [PZB10].

A proposta *Risk Backlog* baseia o GR nos processos definidos pelo PMBOK que são relevantes para os projetos gerenciados pelo método ágil Scrum, onde o ciclo de vida é definido de acordo com a proposta de Schwaber [SB02], composto por três fases: *Pre-Game*, *Game* e *Post-Game*, conforme apresentado na Figura 9 [PZB10].

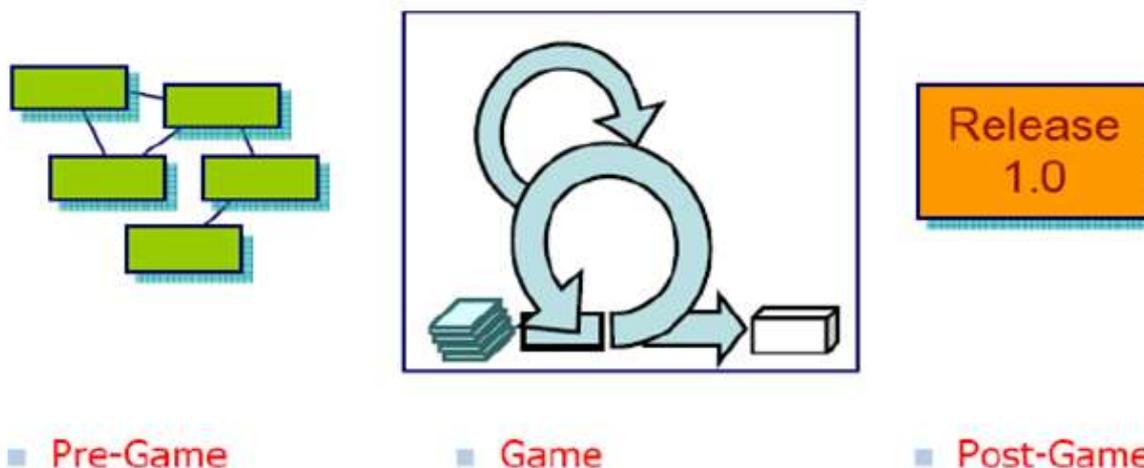


Figura 9 – Fases do ciclo de vida do Scrum [PZB10]

Neste cenário, os processos para GR propostos neste modelo são alinhados com a área de conhecimento de GR no Guia PMBOK [PMB08] e especificados nas etapas do projeto de acordo com o Scrum, conforme mapeado na Tabela 2.

Tabela 2 – Mapeamento das principais ocorrências entre GR e eventos do Scrum [PZB10]

SCRUM \ PMBOK	Pre-Game	Game			Post-Game	
		Sprint Planning Meeting	Daily Scrum Meeting	Sprint	Sprint Retrospective Meeting	
Planejar o Gerenciamento de Riscos	X		X			
Identificar Riscos	X	X		X	X	
Realizar Análise Qualitativa de Riscos	X	X		X	X	
Realizar Análise Quantitativa de Riscos	-	-	-	-	-	
Planejar Respostas aos Riscos	X			X	X	
Monitorar e Controlar Riscos	X			X	X	X

Um dos pontos importantes neste modelo está relacionado às *Daily Scrum Meeting*, onde uma nova pergunta seria acrescentada às três perguntas já realizadas nesta reunião. Cada membro da equipe responderia também a seguinte pergunta: “*Existe algum risco que pode se transformar em impedimento até a próxima Daily Scrum Meeting?*”. Dessa maneira, riscos potenciais à *Sprint* atual podem ser identificados e

abordados com tratamento adequado ao projeto [PZB10]. O controle dos processos de Gerenciamento de Riscos ficaria a cargo do *ScrumMaster*, que tem como papel principal garantir que o processo está sendo seguido e tratar todos os impedimentos do projeto [SB02]. Ao final de cada *Sprint* ocorre a *Sprint Retrospective Meeting*, onde seriam abordadas questões que auxiliariam a identificação e tratamento de novos riscos para as próximas *Sprints*, agregando conhecimento aos fatores de risco identificados até então.

3.5.4 ANÁLISE CRÍTICA SOBRE OS ESTUDOS AVALIADOS

A análise dos modelos, ferramentas e técnicas propostas para auxílio das atividades de gerenciamento de riscos em projetos de desenvolvimento de software, permite a classificação destes estudos de acordo com o modelo de gerenciamento de projeto. Considerando estes fatores, os estudos analisados foram classificados conforme pode ser identificado na Tabela 3.

Tabela 3 – Classificação dos estudos analisados

Cenário Abordagem	Gerenciamento de Risco em Projetos de Software
Prescritiva	RiskFree No-Risk
Adaptativa	Risk Backlog

A partir da análise da Tabela 3, é possível identificar na literatura alguns trabalhos que exploram o gerenciamento de risco com abordagens prescritivas e adaptativas. Entretanto, no que se refere ao modelo adaptativo, o estudo encontrado tem seu foco na adaptação do método ágil Scrum para contemplar gerenciamento de risco de forma explícita, mas nada é relatado em relação aos tipos de riscos que as abordagens adaptativas conseguem gerenciar, sendo este o foco desta pesquisa.

4 METODOLOGIA DE PESQUISA

Os estudos deste trabalho estão focados na identificação de como os riscos comuns existentes em projetos de desenvolvimento de software são tratados em projetos que utilizam abordagens adaptativas de desenvolvimento, tendo como foco o método ágil Scrum. Desta forma, o objetivo deste trabalho foi definido em verificar como os riscos comuns identificados em projetos de desenvolvimento de software são tratados por equipes de projeto que utilizam o Scrum, visando também a identificação de lacunas encontradas nas dimensões de riscos comuns que não são devidamente tratados, e a apresentação de extensões para o Scrum. Neste sentido, este capítulo apresenta detalhes acerca da metodologia de pesquisa utilizada para desenvolver este trabalho.

O termo pesquisa pode ser definido com um conjunto racional e sistemático de procedimentos que tem como objetivo responder a problemas propostos [GIL10]. Desta forma é necessária a utilização adequada de métodos e técnicas de investigação científica, ou seja, de uma metodologia, envolvendo diversas etapas que vão desde a formulação do problema até a apresentação adequada de uma proposta para solução.

4.1 ASPECTOS METODOLÓGICOS

Para garantir algum grau confiabilidade deste estudo foi planejado um rigoroso processo de pesquisa a qual contemplou a utilização de protocolos para desenvolvimento e formalização dos estudos primários e secundários executados. A revisão sistemática da literatura foi planejada seguindo o protocolo proposto por Kitchenham et al [KBB+07].

4.1.1 PROPÓSITO – OBJETIVO AMPLO

Gil afirma que as pesquisas podem ser classificadas quanto ao seu propósito – ou objetivo amplo – como exploratórias descritivas ou explicativas. As pesquisas exploratórias tem o objetivo de proporcionar maior familiarização com o problema, tornando-o mais explícito [GIL10]. As pesquisas descritivas tencionam delinear as características de uma determinada população. Já as pesquisas explicativas buscam identificar os fatores que determinam ou contribuem para a ocorrência de fenômenos.

Esta pesquisa tem como objetivo geral identificar como o gerenciamento de riscos é tratado no Scrum, buscando verificar como os riscos comuns identificados no gerenciamento de projetos de desenvolvimento de software estão sendo tratados em

projetos que utilizam o Scrum. Como não foram identificadas informações estruturadas disponíveis na literatura, é possível afirmar que esta pesquisa tem características de pesquisa exploratória, de base qualitativa.

4.1.2 DELINEAMENTO – PROCEDIMENTOS TÉCNICOS

A definição de delineamento, produzida por Gil [GIL10], descreve o termo como o planejamento da pesquisa em sua dimensão mais ampla, envolvendo os fundamentos metodológicos, concepção de objetivos, ambiente da pesquisa, além das técnicas de coleta e análise de dados. Nesta definição são considerados elementos diversos, dificultando a composição de uma classificação detalhada, pois sempre existe a possibilidade de que uma pesquisa específica não se enquadre em qualquer uma das categorias propostas ou ainda que uma mesma pesquisa possa se enquadrar em mais de uma classificação [GIL10].

Desta forma, Gil [GIL10] apresenta um sistema que considera o ambiente de pesquisa, a abordagem teórica e as técnicas de coleta e análise de dados para classificar a pesquisa nos seguintes moldes:

- **Pesquisa Bibliográfica:** presente na grande maioria das pesquisas acadêmicas sendo constituída principalmente por livros e artigos científicos;
- **Pesquisa Documental:** semelhante a pesquisa bibliográfica, diferenciando apenas na natureza das fontes, sendo esta composta por material que ainda não receberam uma análise, possibilitando uma reanálise dos dados.
- **Pesquisa Experimental:** representa o melhor exemplo de pesquisa científica, consistindo em determinar um objeto de estudo, identificar as variáveis que podem influenciá-lo, definir as possibilidades de controle e de observação dos efeitos que essas variáveis produzem no objeto observado. Possibilita o estudo da causa e efeito em determinados cenários.
- **Pesquisa Ex-Post Facto:** ou pesquisa a partir do fato passado. O objetivo é o mesmo da Pesquisa Experimental, porém com a observação de ocorrências após mudanças nas variáveis dependentes no curso natural dos acontecimentos.
- **Estudo de Coorte:** refere-se a um grupo de pessoas que tem alguma característica em comum, representando uma amostra de determinada

população a ser acompanhada por certo período de tempo. Muito utilizado em pesquisas na área da saúde.

- **Survey (Levantamento):** a característica principal deste tipo de pesquisa é a interrogação direta das pessoas cujo comportamento se deseja conhecer, mediante análise quantitativa para obtenção de conclusões referentes aos dados coletados.
- **Estudo de Campo:** muito semelhante ao *Survey*, tendo este maior alcance, porém menor profundidade. O Estudo de Campo apresenta maior flexibilidade, permitindo que seus objetivos sejam reformulados durante a pesquisa. O pesquisador realiza os trabalhos pessoalmente, possibilitando o contato próximo com a situação de estudo e conseqüentemente, a redução do viés obtido em estudos anteriores.
- **Estudo de Caso:** constitui no estudo aprofundado e exaustivo de um ou poucos objetos, permitindo seu amplo e detalhado conhecimento. Dificulta a generalização, e por isso é melhor utilizado para proporcionar uma visão global do problema, identificando fatores que influenciam ou são influenciados pelo problema em um determinado cenário.
- **Pesquisa Ação:** tipo de pesquisa com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo, no qual os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo.
- **Pesquisa Participante:** semelhante a Pesquisa Ação, caracteriza-se pela interação entre pesquisadores e membros das situações investigadas.
- **Grounded Theory:** também conhecida como *Teoria Fundamentada em Dados*, este tipo de pesquisa utiliza técnicas indutivas e incrementais baseadas na análise sistemática dos dados. É uma metodologia recente.

Para condução dos trabalhos deste estudo, o tipo de pesquisa adotado como base principal foi o Estudo de Campo, em função da possibilidade de realização das entrevistas em nível pessoal, minimizando o viés do pesquisador e a aproximação com a situação de estudo.

4.2 DESENHO DE PESQUISA

O desenho de pesquisa ilustrado na Figura 10 a seguir contempla as etapas e fases planejadas para se alcançar o objetivo deste estudo.

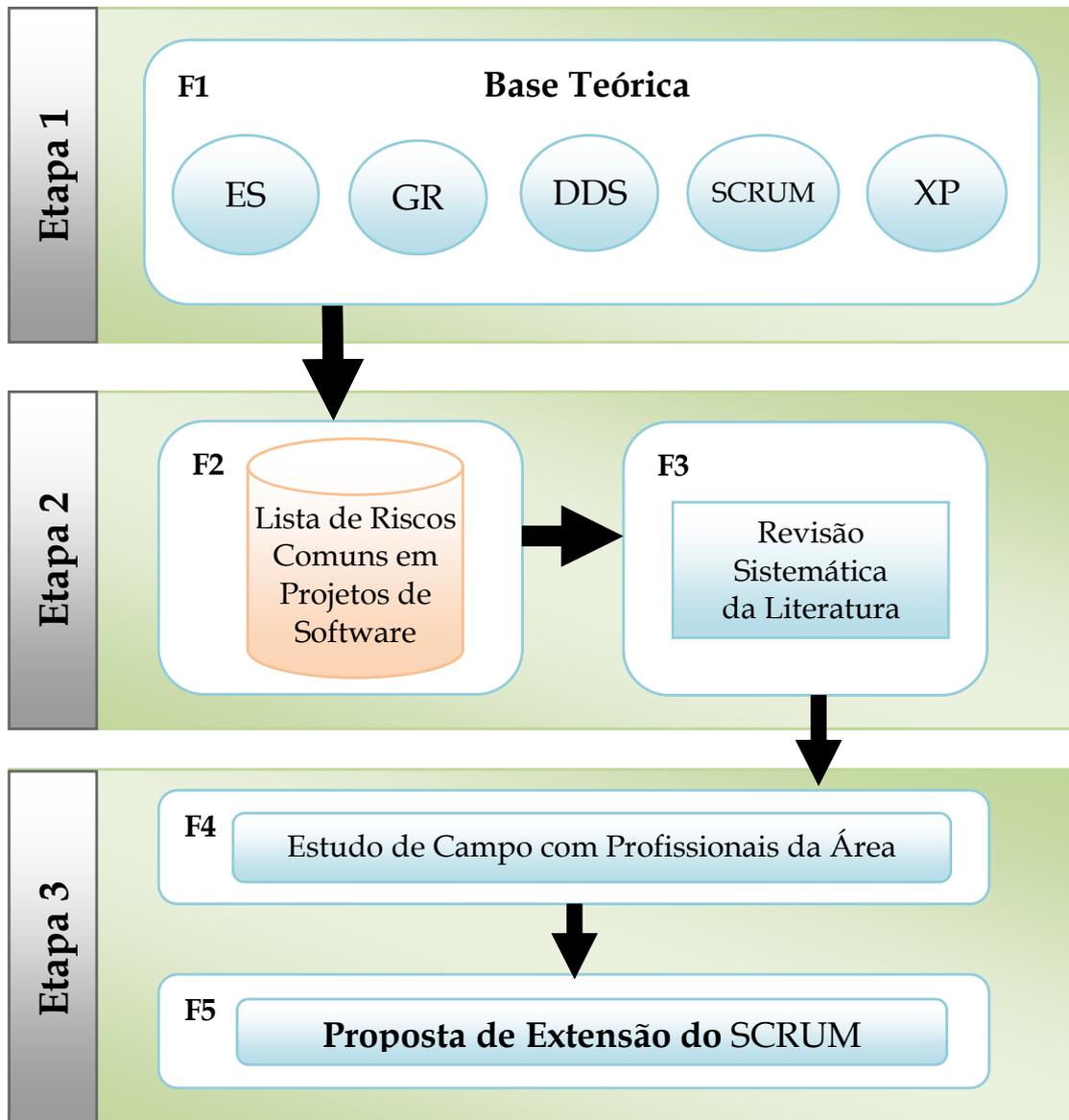


Figura 10 – Desenho de Pesquisa

- **Etapa 1:** Constituída apenas pela Fase 1 (F1), o objetivo principal desta etapa foi estudar o referencial teórico da área, envolvendo os conteúdos de engenharia de software, gerenciamento de projetos, gerenciamento de riscos e métodos ágeis para desenvolvimento de software. Um dos resultados obtidos durante o estudo da base teórica foi a elaboração de uma monografia sobre os principais temas abordados nesta pesquisa. Neste trabalho foram identificadas lacunas com relação ao tema de gerenciamento de riscos, tanto

no contexto de abordagens adaptativas quanto no cenário de desenvolvimento distribuído de software. Dadas as opções disponíveis para o avanço dos estudos nas etapas seguintes, decidiu-se focar o trabalho na área de gerenciamento de riscos no contexto de abordagens adaptativas, considerando equipes co-locadas;

- **Etapa 2:** esta etapa foi realizada em duas fases. A Fase 2 (F2) teve por objetivo a identificação de uma lista de riscos comuns em projetos de desenvolvimento de software. Esta lista foi definida a partir do estudo complementar de artigos relacionados ao tema de gerenciamento de riscos em projetos de desenvolvimento de software. Também se identificou a necessidade de verificação da literatura específica sobre o tema, buscando estudos similares na literatura. Este estudo foi conduzido na Fase 3 (F3), por meio de uma Revisão Sistemática da Literatura;
- **Etapa 3:** nesta etapa, durante a Fase 4 (F4), foi executada um estudo de campo baseado em entrevistas com gerentes de projeto e líderes de equipe que utilizam o método ágil Scrum. O objetivo deste estudo de campo foi obter conhecimento das práticas adotadas pelos entrevistados com relação ao tratamento tomado frente aos riscos comuns identificados em projetos de desenvolvimento de software. Ao final, os resultados e conclusões foram utilizados para sugerir extensões do Scrum, visando o gerenciamento efetivo dos riscos comuns encontrados em projetos de desenvolvimento de software, sendo esta a Fase 5 (F5) deste estudo.

5 LISTA DE RISCOS COMUNS EM PROJETOS DE SOFTWARE

A lista de riscos comuns em projetos de desenvolvimento de software determina uma das etapas importantes para a realização da pesquisa proposta neste estudo em razão de esta lista ter sido utilizada como base nas fases subsequentes. Esta etapa foi conduzida através da análise de estudos publicados em eventos e periódicos relevantes para as áreas de impacto da pesquisa [BOE91, SLK+01, WKA04, HH07]. Cada artigo identificado apresenta uma lista de riscos comuns em projetos de desenvolvimento de software. A seleção dos artigos foi realizada através de busca em bibliotecas digitais acadêmicas, sendo definidos os critérios descritos abaixo para a buscar os estudos na literatura:

- Possuir uma lista de riscos comuns em projetos de software;
- A classificação *Qualis*¹;
- O número de citações.

5.1 SELEÇÃO DE ESTUDOS E DIMENSÕES DE RISCOS

Como resultado da fase de seleção de estudos, foram identificados 4 artigos relevantes a partir dos critérios anteriormente definidos. Os artigos selecionados estão listados na Tabela 4.

Tabela 4 – Artigos selecionados para lista de riscos comuns

Ref.	Artigo	Qualis	Citações
1 [BOE91]	Boehm, B.W.; "Software Risk Management: Principles and Practices"; IEEE Software; vol.8; number 1; pp.32-41; 1991.	A1	1544
2 [SLK+01]	Schmidt, R.; Lyytinen, K.; Keil, M.; Cule, P.; "Identifying Software Project Risks: An International Delphi Study"; Journal of Management Information System; vol.17; number 4; pp.5-36; 2001.	A1	537
3 [WKA04]	Wallace, L.; Keil, M.; Arun, R.; "How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model"; Decision Sciences; vol.35; number 2; pp.289-321; 2004.	A1	186
4 [HH07]	Han, W.M.; Huang, S.J.; "An Empirical Analysis of Risk Components and Performance on Software Projects"; The Journal of Systems and Software; vol.80; number 1; pp.42-50; 2007.	A2	77

Os riscos identificados foram categorizados em dimensões, de acordo com o estudo proposto por Wallace et al [WKA04], conforme a Tabela 5 abaixo.

¹ Qualis é o conjunto de procedimentos utilizados pela Capes para estratificação da qualidade da produção intelectual dos programas de pós-graduação. http://www.cpgss.ucg.br/home/secao.asp?id_secao=99

Tabela 5 – Dimensões de Riscos comuns

Dimensões de Riscos	Sigla
Ambiente Organizacional	ORG
Planejamento e Controle	P&C
Complexidade	COM
Requisitos	REQ
Equipe	TEA
Usuário	USE

5.2 LISTA DE RISCOS COMUNS EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE

Após sintetizar a listagem de riscos comuns, eliminando ocorrências múltiplas de riscos encontrados simultaneamente em mais de um artigo, foi possível identificar 33 riscos comuns em projetos de desenvolvimento de software. Estes riscos comuns identificados estão listados na Tabela 6 a seguir.

Tabela 6 – Riscos Comuns em Projetos de Software

Risco	Referência	Dimensão	Descrição
1	3	ORG	Reestruturação da organização Mudança da gerência ou alteração da estrutura organizacional.
2	3	ORG	Ambiente organizacional instável Cargos vagos sem alocação de pessoas, falta de gerentes ou coordenadores, dificultando ou retardando a tomada de decisão.
3	2	ORG	Mudança de prioridade da gerência Novos objetivos definidos pela gerência podem alterar as prioridades dos projetos.
4	2	ORG	Falta de comprometimento da alta direção com o projeto Supervisão ineficiente ou inexistente dos executivos da organização.
5	4	ORG	Política corporativa com efeito negativo sobre o projeto Políticas da empresa podem ser definidas em desalinhamento com a realidade diária da condução de projetos. Metodologia Ágil, mas empresa não é.
6	3	P&C	Comunicação ineficaz Envolvimento insuficiente da equipe do projeto, falha ao indicar mudanças, falta de relatórios de status do projeto.
7	3	P&C	Tecnologia ineficaz para gestão de projetos A escolha de uma ferramenta eficaz para auxílio no gerenciamento de projetos pode influenciar a condução do mesmo.
8	2,4	P&C	Falta de metodologia eficaz no gerenciamento do projeto A equipe não emprega controle de mudanças, sem planejamento ou outras habilidades necessárias ao processo.
9	3	P&C	Etapas do projeto não definidas claramente Planejamento superficial ou projeto não está bem estruturado (falta de WBS).
10	1	P&C	Gold plating Entregar mais do que foi definido no escopo do projeto.
11	1	P&C	Deficiências na execução de tarefas e componentes fornecidos externamente Ao alocar uma atividade ou componente do projeto para um recurso externo e esta atividade ou componente apresenta falha na execução.
12	1,2,3	P&C	Estimativas irreais de tempo e custo Ocorre quando as estimativas do projeto são muito cautelosas ou ousadas demais, gerando impacto na qualidade do projeto em virtude da redução do tempo de teste ou treinamento.
13	2	P&C	Alteração de escopo ou objetivos Mudanças nos negócios ou reorganização durante o projeto.
14	2	P&C	Escopo ou objetivos não claros ou incompreendidos É impossível definir o escopo ou objetivo real do projeto devido a diferenças de opinião entre os usuários.

Risco	Referência	Dimensão	Descrição
15	2,3,4	P&C	Falta de planejamento ou planejamento inadequado A atividade de planejamento não é considerada importante ou não é realizada.
16	2	P&C	Gestão de mudanças inadequada Cada projeto necessita de um processo para gerir mudança para que o custo e escopo sejam controlados. O aumento do escopo sem controle é a evidência de uma gestão da mudança ineficaz, onde os parâmetros de sucesso não estão definidos.
17	3,4	P&C	Andamento do projeto não monitorado adequadamente Ocorre quando existe falha na definição de pontos de controle do projeto.
18	3,4	P&C	Estimativa inadequada dos recursos necessários Falta na identificação de recursos para o projeto compromete o andamento.
19	3	COM	Alto nível de complexidade técnica Ocorre quando o produto é composto por módulos de diferentes tecnologias e de difícil entendimento e integração.
20	1	COM	Falta de desempenho em tempo real Simulação, medições, modelagem, prototipagem, ajustes ou instrumentação apresentam falta de desempenho devido a ambientes limitados.
21	1,3,4	COM	Tecnologia inexistente ou imatura O produto do projeto exige a utilização de tecnologia ainda não disponível ou muito recentes e imaturas
22	1	REQ	Desenvolver as funções erradas do software Mapeamento incompleto dos componentes ou da arquitetura do software.
23	1	REQ	Desenvolver a interface de usuário errada O usuário está habituado a utilizar outro padrão de interface na organização.
24	1	REQ	Alterações tardias nos requisitos Solicitação de mudanças durante a fase de testes do produto, por exemplo.
25	2,3,4	REQ	Falta de requisitos congelados Devido às constantes necessidades de mudanças dos usuários, o software corre o risco de jamais ser concluído.
26	2,3,4	REQ	Incompreensão dos requisitos Não definir todos os requisitos do sistema antes de começar os trabalhos, gerando requisitos de forma superficial causando falha na estimativa de esforço, habilidades e tecnologia necessária para o projeto.
27	3	TEA	Membros da equipe de desenvolvimento inadequadamente treinados Falta de conhecimento na tecnologia utilizada, falta de experiência ou treinamento ineficaz.
28	1,2,3	TEA	Membros da equipe com falta de habilidade especializada exigida pelo projeto Por exemplo, tecnologia, conhecimento de negócios e experiência inexistentes entre os recursos da equipe.
29	2,3	TEA	Falta de habilidade na liderança de projetos A equipe do projeto é formada e o gerente não tem autoridade ou habilidades para obter sucesso.
30	2,3	USE	Conflitos entre usuários Diferentes objetivos dos departamentos dos usuários geram expectativas distintas.
31	2,3	USE	Falha em obter comprometimento do usuário Usuários apontam o gerente do projeto como culpado pela falta de envolvimento do cliente.
32	2,3	USE	Usuários com resistência a mudanças Obstáculo percebido quando o usuário não quer mudar de rotina ou não está motivado na empresa.
33	2	USE	Falha ao gerir expectativas do usuário final Expectativas determinam o sucesso ou o fracasso de um projeto. Expectativas incompatíveis com a entrega (muito alta ou muito baixa) causam problemas. As expectativas devem ser corretamente identificadas e constantemente ajustadas a fim de evitar o fracasso.

Esta lista de riscos comuns foi utilizada como referência para verificar a ocorrência e tratamento destes riscos em projetos de desenvolvimento de software conduzidos utilizando o método ágil Scrum.

6 REVISÃO SISTEMÁTICA DA LITERATURA

Além da busca por trabalhos relacionados ao tema desta pesquisa, foi desenvolvido um estudo de forma sistemática com o objetivo de identificar a existência de pesquisas que relacionam o gerenciamento de riscos em projetos de desenvolvimento de software e a metodologia utilizada para gerenciar estes projetos. Além disso, como objetivo secundário, buscou-se identificar estudos que apontassem como o gerenciamento de riscos implícito é tratado em projetos de desenvolvimento de software que utilizam abordagens adaptativas, especificamente com o Scrum.

Esta seção descreve o processo adotado nesta etapa do trabalho, indicando a questão de pesquisa, a qualidade e amplitude do estudo, seleção das fontes de busca, definição de critérios de inclusão e exclusão de trabalhos bem como os procedimentos para seleção dos estudos relevantes. Também são apresentados os resultados obtidos após a seleção e análise dos estudos relevantes, indicando ainda a listagem de artigos primários excluídos da pesquisa e apresentam as respostas para as questões de pesquisa propostas nesta revisão sistemática da literatura da área.

6.1 PROCESSO DE REVISÃO SISTEMÁTICA DA LITERATURA

O processo de Revisão Sistemática da Literatura é um dos mais importantes métodos de pesquisa científica em engenharia de software, conforme proposto por Kitchenham et al [KBB+07]. O processo representa um método replicável de identificar, avaliar e interpretar pesquisas relevantes para a uma determinada área de estudo ou para uma questão de pesquisa ou ainda para um fenômeno de interesse [KBB+07].

Esta revisão sistemática foi conduzida por dois pesquisadores, sendo um deles mestrando do curso de Ciências da Computação, na área de Sistemas de Informação, e o outro pesquisador doutor em Ciência da Computação.

6.2 QUESTÃO DE PESQUISA

A questão de pesquisa (QP) principal desta revisão sistemática foi:

“Como os riscos comuns em projetos de desenvolvimento de software são tratados dentro das principais metodologias de gestão de projeto de desenvolvimento de software atualmente existentes?”.

A questão principal desta pesquisa derivou duas questões secundárias:

QS1: *O que se sabe sobre gerenciamento de riscos em projetos de desenvolvimento de software no contexto de métodos de gerenciamento e desenvolvimento de projetos de software que não possuem o gerenciamento de risco de forma explícita?*

QS2: *O que se sabe sobre gerenciamento de riscos em projetos de desenvolvimento de software que utilizam o método ágil Scrum?*

6.3 QUALIDADE E AMPLITUDE DA QUESTÃO

A definição do problema a ser pesquisado nesta revisão sistemática consiste em identificar a existência de trabalhos que relacionam o gerenciamento de riscos em projetos de desenvolvimento de software e a metodologia utilizada para gerenciar estes projetos. As palavras-chave e sinônimos foram definidos de acordo com a área, abrangendo os termos mais comuns apresentados na literatura e estão listados abaixo.

- Gerência de risco, gestão de riscos, gerenciamento de riscos;
- Principais metodologias de gerenciamento de projeto de desenvolvimento de software (RUP, MSF, Scrum, XP);
- Estudos, relatórios, revisões, pesquisas, relatos.

Na elaboração do protocolo de pesquisa criado para alcançar os objetivos definidos nesta revisão sistemática, constituiu-se como intervenção os principais métodos de gerenciamento de projetos existentes, associados à área de gerenciamento de riscos. São eles RUP, MSF, Scrum e XP. Neste trabalho, não foram definidos trabalhos primários de controle devido à dificuldade de localização de pesquisas específicas nesta área.

Esta revisão sistemática se aplica a pesquisadores que estejam desenvolvendo pesquisas científicas e necessitem de informações deste tema, ou ainda a profissionais da indústria que possuem interesse em informações resultantes desta pesquisa. Não foram utilizados métodos estatísticos. Como idiomas de seleção dos estudos primários foram utilizados o inglês e o português.

6.4 SELEÇÃO DE FONTES DE PESQUISA

Para garantir uma pesquisa exaustiva por estudos primários, foram realizadas procuras sobre o tema de gerenciamento de riscos em projetos de desenvolvimento de software, utilizando para isto, as principais bibliotecas digitais relevantes e também a busca manual em páginas da Internet dos eventos nacionais mais importantes sobre o assunto. Assim, a relação de fontes de pesquisa para trabalhos primários relacionados ao contexto e objetivo desta revisão sistemática foi definida através dos critérios relacionados a seguir.

- Artigos de conferência e periódicos sobre o tema;
- Data de publicação a partir de 2001 – Manifesto Ágil;
- Base de dados atualizada;
- Estudos empíricos ou relatos de experiência contidos de forma clara;
- Pesquisa manual em sites de periódicos relevantes;
- Pesquisa em bibliotecas digitais.

Inicialmente, foram realizadas buscas nas bibliotecas digitais relevantes, para garantir uma abrangência consistente de trabalhos e estudos da área. As bibliotecas digitais pesquisadas foram:

- **IEEEXplore** (<http://ieeexplore.ieee.org>);
- **ACM Digital Library** (<http://dl.acm.org>);
- **Science Direct – Elsevier** (<http://www.sciencedirect.com>);
- **Springer Link** (<http://www.springerlink.com>);
- **Wiley** (<http://onlinelibrary.wiley.com>);
- **BDBComp** (<http://www.lbd.dcc.ufmg.br/bdbcomp>).

A composição da palavra de busca foi realizada de acordo com os critérios de definição desta revisão sistemática da literatura, escolhendo com cuidado os termos adequados em função do impacto no resultado das buscas em bibliotecas digitais. A palavra “*risk*” não foi incluída de forma isolada na palavra de busca pelo motivo de não agregar precisão aos resultados, prejudicando a busca por tratar-se de um termo comum não relacionado apenas a gerenciamento de riscos. Para evitar o surgimento de evidências falsas entre os resultados das buscas, optou-se por incluir o termo “*risk management*” como um argumento de busca devidamente relacionado à pesquisa deste trabalho. A composição das palavras de busca foi assim definida:

- **Grupo 1:** Gerenciamento de Riscos em Projetos de Desenvolvimento de Software (risk management <and> software development <and> project);
- **Grupo 2:** Métodos de gerenciamento e desenvolvimento de projetos de software que não possuem gerenciamento de risco de forma explícita (RUP <or> MSF <or> Scrum <or> XP);
- **Grupo 3:** Estudos, relatórios, pesquisas, revisões ou relatos (study <or> report <or> research <or> review <or> description);
- **Palavra de busca:** Grupo 1 <and> Grupo 2 <and> Grupo 3.

Além das pesquisas nas principais bibliotecas digitas, com o objetivo de melhorar a qualidade dos resultados da revisão sistemática da literatura, foram realizadas buscas manuais nas páginas da internet dos principais eventos brasileiros da área de estudo. Estes eventos consideram os 3 últimos anos de publicações, em virtude da dificuldade de acessar as bases de eventos brasileiros bem como da disponibilidade de artigos nestas bases. Os eventos selecionados foram:

- **SBES** – Simpósio Brasileiro de Engenharia de Software;
- **SBQS** – Simpósio Brasileiro de Qualidade de Sistemas;
- **SBSI** – Simpósio Brasileiro de Sistemas de Informação.

6.5 SELEÇÃO DE ESTUDOS

Após realização das duas etapas de busca de trabalhos primários, foi localizado um total de 466 trabalhos, considerando que a busca manual foi realizada sobre a leitura dos trabalhos aceitos nos periódicos relevantes pesquisados.

Tabela 7 – Fontes de pesquisa, palavras de busca e resultados

Biblioteca	Palavra de busca / Método de pesquisa	Resultados
ACM Digital Library	("risk management" and "software development" and project) and (RUP or MSF or Scrum or XP) and (study or report or research or review or description) and (PublishedAs:journal OR PublishedAs:transaction OR PublishedAs:magazine OR PublishedAs:newsletter)	33
IEEEExplore	("risk management" AND "software development" AND project) AND (RUP OR MSF OR Scrum OR XP) AND (study OR report OR research OR review OR description) You Refined by: Publication Year: 2001 - 2012	322
Science Direct Elsevier	pub-date > 2000 and ("risk management" and "software development" and project) and ((RUP or MSF or Scrum or XP) and (study or report or research or review or description)) [All Sources(Computer Science)]	58
Springer Link	((("risk management" and "software development" and project) and (RUP or MSF or Scrum or XP)) and ((study or report or research))' published between '1 Jan 2001' and '31 Aug 2012' with the filter: Computer Science	17
Wiley	("risk management" AND "software development" AND project) in All Fields AND (RUP OR MSF OR Scrum OR XP) in All Fields AND (study OR report OR research OR review OR description) in All Fields between years 2001 and 2012	33
SBES	Pesquisa manual nos artigos aceitos nos periódicos de 2010 a 2012	1
SBQS	Pesquisa manual nos artigos aceitos nos periódicos de 2010 a 2012	0
SBSI	Pesquisa manual nos artigos aceitos nos periódicos de 2010 a 2012	2

A busca nas bibliotecas digitais foi realizada através da aplicação da palavra de busca sobre o mecanismo de pesquisa específica de cada biblioteca digital. A listagem completa das fontes de pesquisa, suas palavras de busca específicas e a quantidade de resultados obtidos estão listados na Tabela 7.

6.6 PROCEDIMENTOS DE SELEÇÃO DE ESTUDOS

A seleção dos estudos foi feita a partir da identificação dos trabalhos resultantes das pesquisas nos mecanismos de busca, submetendo a listagem inicial aos critérios de inclusão e exclusão. Foram inicialmente identificados o título e palavras chave dos trabalhos, seguido da leitura do resumo dos mesmos e posteriormente, da leitura completa do estudo:

- **Inclusão:**
 - Artigos de natureza qualitativa e/ou quantitativa que relatem a prática do gerenciamento de riscos em projetos de desenvolvimento de software;
 - Artigos publicados a partir de 2001, ano de criação do Manifesto Ágil.
- **Exclusão:**
 - Artigos que não envolvam gerenciamento de riscos;
 - Artigos que não definam uma metodologia de gerenciamento de projetos;
 - *Short papers*.
- **Procedimentos de seleção:**
 - Identificação dos artigos obtidos nos mecanismos de busca;
 - Exclusão baseada na leitura do título, palavras chave e resumo;
 - Exclusão baseada na leitura da Introdução e Considerações finais;
 - Leitura completa dos estudos e análise crítica dos artigos;
 - Classificação dos artigos selecionados quanto ao método de gerenciamento de projetos mencionado, o modelo utilizado, indicando que se trata de um modelo adaptativo ou prescritivo; e o tipo de estudo, de acordo com a classificação de Wieringa et al [WMR06].

Após analisar os estudos obtidos na etapa de busca nas bibliotecas específica, foram identificados 43 trabalhos selecionados como estudos primários de interesse para a revisão sistemática da literatura.

6.7 ESTUDOS PRIMÁRIOS EXCLUÍDOS

Após aplicação dos critérios de exclusão definidos nesta revisão sistemática, a listagem obtida nas buscas das fontes foi reduzida de 466 resultados iniciais para 43 estudos primários, sendo excluídos 40 trabalhos após leitura dos resumos dos estudos. A listagem dos 40 trabalhos excluídos está relacionada abaixo, na Tabela 8.

Tabela 8 – Revisão Sistemática - Estudos Primários Excluídos

Título	Autores	Ano	Método	Tipo	Abordagem
Monitoring risk response actions for effective project risk management	Edouard Kujawski; Diana Angelis	2009	Other	Experience paper	-
Preventive risk management software for software projects	Murthi, S.	2002	RUP	Empirical Paper	Adaptative
Using risk to balance agile and plan-driven methods	Boehm, B.; Turner, R.	2003	XP, RUP, Scrum	Experience Paper	Adaptative
Get ready for agile methods, with care	Boehm, B.	2002	XP	Experience Paper	Adaptative
PlayScrum - A Card Game to Learn the Scrum Agile Method	Fernandes, J.M.; Sousa, S.M.	2010	Scrum	Philosophical Theoretical Paper	-
Comparing Agile Software Processes Based on the Software Development Project Requirements	Qasaimeh, M.; Mehrfard, H.; Hamou-Lhadj, A.	2008	Hybrid	Empirical Paper	Adaptative
Fuzzy model for the software projects design risk analysis	Bragina, T.; Tabunshchyk, G.	2011	Hybrid	Empirical Paper	-
Ten deadly risks in Internet and intranet software development	Reifer, D.	2002	Other	Empirical Paper	-
Quality Attribute Driven Agile Development	Sanghoon Jeon; Myungjin Han; Eunseok Lee; Keun Lee	2011	Scrum	Evaluation Research	Adaptative
Three-way cultural change: introducing agile within two non-agile companies and a non-agile methodology	Lawrence, R.; Yslas, B.	2006	XP, Scrum	Evaluation Research	Adaptative
Managing commitments and risks: challenges in distributed agile development	Kontio, J.; Hoglund, M.; Ryden, J.; Abrahamsson, P.	2004	Scrum	Philosophical Theoretical Paper	Adaptative
A Risk Management Methodology for Project Risk Dependencies	Tak Wah Kwan; Leung, H.K.N.	2011	Other	Validation Research	-
Moving from Waterfall to Iterative Development: An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP	Osorio, J.A.; Chaudron, M.R.V.; Heijstek, W.	2011	RUP	Experience paper	-
Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?	Bannerman, P.L.; Hossain, E.; Jeffery, R.	2012	Scrum	Experience paper	Adaptative
Computer-generated comprehensive risk assessment for IT project management	Wickboldt, J.A.; Bianchin, L.A.; Lunardi, R.C.; Andreis, F.G.; Santos, R.L.; Dalmazo, B.L.; Costa Cordeiro, W.L.; Sousa, A.L.R.; Granville, L.Z.; Gaspary, L.P.; Bartolini, C.	2010	Other	Evaluation Research	Adaptative
Managing Risks in Distributed Software Projects: An Integrative Framework	Persson, J.S.; Mathiassen, L.; Boeg, J.; Madsen, T.S.; Steinson, F.	2009	Other	Evaluation Research	-

Título	Autores	Ano	Método	Tipo	Abordagem
An approach to software project feasibility study using stochastic risk model during proposal preparation	Khritankov, A.	2009	Other	Solution Proposal	-
Identifying risks in XP projects through process modeling	Kirk, D.; Tempero, E.	2006	XP	Evaluation Research	Adaptative
Can Anybody Help?: Mitigating IS Development Project Risk with User Involvement	Amrit, C.; van Hillegersberg, J.; van Diest, B.	2012	RUP	Evaluation Research	Prescriptive
Development of software project risk management model review	Pu Tianyin	2011	Other	Theoretical paper	-
Insight into Risk Management in Five Software Organizations	Kajko-Mattsson, M.; Lundholm, J.; Norrby, J.	2009	Other	Experience paper	-
Risk Management and Context in a Collaborative Project Management Environment for Software Development	Lima, M.P.; David, J.M.N.; Dantas, B.T.	2010	RUP	Solution Proposal	Prescriptive
Risk Identification and Risk Mitigation Instruments for Global Software Development: Systematic Review and Survey Results	Nurdiani, I.; Jabangwe, R.; Smite, D.; Damian, D.	2011	Other	Theoretical Paper	-
The role of software process simulation modeling in software risk management: A systematic review	Dapeng Liu; Qing Wang; Junchao Xiao	2009	-	Philosophical Theoretical Paper	-
Risk Management for Web and Distributed Software Development Projects	Keshlaf, A.A.; Riddle, S.	2010	Other	Philosophical Theoretical Paper	-
Balancing Opportunities and Risks in Component-Based Software Development	Boehm, B.; Bhuta, J.	2008	RUP	Evaluation Research	Prescriptive
Towards agile security risk management in RE and beyond	Franqueira, V.N.L.; Bakalova, Z.; Thein Than Tun; Daneva, M.	2011	Other	Theoretical paper	Adaptative
Test-Driven Development for Spreadsheet Risk Management	Rust, A.; McDaid, K.	2009	Other	Experience Paper	-
Managing Risks in Global Software Engineering: Principles and Practices	Ebert, C.; Murthy, B.K.; Jha, N.N.	2008	Other	Validation Research	-
A risk taxonomy proposal for software maintenance	Webster, K.P.B.; de Oliveira, K.M.; Anquetil, N.	2005	Other	Validation Research	-
Measurement, prediction and risk analysis for Web applications	Fewster, R.; Mendes, E.	2001	Other	Experience paper	-
Communicating Risk Information in Agile and Traditional Environments	Nyfjord, J.; Kajko-Mattsson, M.	2007	RUP, Agile	Experience paper	Adaptative
Evolving beyond requirements creep: a risk-based evolutionary prototyping model	Carter, R.A.; Anton, A.L.; Dagnino, A.; Williams, L.	2001	Other	Solution Proposal	-
The agile methods fray	DeMarco, T.; Boehm, B.	2002	XP	Opinion Paper	Adaptative
Agile Methods: Crossing the Chasm	Maurer, Frank; Melnik, Grigori	2007	XP, Scrum	Philosophical Theoretical Paper	Adaptative
Value-Risk Trade-off Analysis for Iteration Planning in Extreme Programming	Xin Dong; Qiu-Song Yang; Qing Wang; Jian Zhai; Ruhe, G.	2011	XP	Evaluation Research	Adaptative
Formalizing agility, part 2: how an agile organization embraced the CMMI	Baker, S.W.	2006	XP	Opinion Paper	Adaptative

Título	Autores	Ano	Método	Tipo	Abordagem
InfoSecRM: Uma Abordagem Ontológica para a Gestão de Riscos de Segurança da Informação	Éder S. Gualberto; Rafael T. Sousa Jr; Flávio E. de Deus; Cláudio G. Duque	2012	Other	Validation Research	-
Implementação da ISO 9001 com Scrum: Um Estudo de Caso	Leonardo Carneiro; Marc A. de Queiroz; Rodolfo M. Barros; Jacques D. Brancher	2012	Scrum	Philosophical Theoretical Paper	Adaptative
An Empirical Study on the Relationship between the Use of Agile Practices and the Success of Software Projects that Use Scrum	Leila Mariz; A. César França; Fabio Q. B. da Silva	2010	Scrum	Evaluation Research	Adaptative

Apesar de terem sido excluídos, devido a falta de relevância com o objetivo desta revisão sistemática, foi possível identificar o tipo de estudo para cada trabalho primário, de acordo com a classificação de pesquisas científicas proposto por Wieringa et al [WMR06]. Também foi identificado, na grande maioria dos estudos lidos, o método de gerenciamento de projetos mencionado. Ainda foi verificada a abordagem do método, quando mencionado no resumo do estudo.

6.8 ESTUDOS SELECIONADOS

Os trabalhos selecionados foram classificados de acordo com o método de gerenciamento de projeto adotado no estudo, a abordagem utilizada, indicando se corresponde a uma abordagem prescritiva ou adaptativa, e ao tipo de estudo, de acordo com a classificação de estudos apresentado no trabalho de Wieringa et al [WMR06]. O resultado obtido pode ser verificado na Tabela 9.

Tabela 9 – Estudos resultantes da Revisão Sistemática

Referência	Título	Autores	Ano	Método	Tipo	Abordagem
[NK07]	Commonalities in Risk Management and Agile Process Models	Nyffjord, J.; Kajko-Mattsson, M.	2007	XP, Scrum	Philosophical Theoretical Paper	Adaptativa
[NK08]	Outlining a Model Integrating Risk Management and Agile Software Development	Nyffjord, J.; Kajko-Mattsson, M.	2008	Scrum	Solution Proposal	Adaptativa
[KCD+10]	Quality control and risk mitigation: A comparison of project management methodologies in practice	Khoja, S.A.; Chowdhary, B.S.; Dhirani, L.L.; Kalhor, Q.	2010	Scrum	Evaluation Research	Adaptativa

Todos os artigos selecionados indicam a avaliação ou estudo de abordagens adaptativas de gerenciamento de projetos, sendo que em todos eles, o método ágil Scrum é mencionado. Os dois primeiros artigos listados indicam uma linha de estudos recente no tema abordado sendo conduzida pelos pesquisadores da Universidade de Stockholm, na Suécia. Estes estudos referenciam o tema de gerenciamento de riscos em projetos de desenvolvimento ágil, com metodologias tais como XP e Scrum.

O primeiro artigo da listagem [NK07] realiza uma comparação entre os métodos ágeis e o gerenciamento de riscos em projetos de desenvolvimento de software. Apesar de indicar que o gerenciamento de riscos constitui um processo moroso, em contraste com os ideais defendidos pelos métodos ágeis, o estudo aponta que existem muitos pontos comuns que possibilitam uma aproximação entre métodos ágeis e gerenciamento de riscos, de forma complementar, sem que os princípios do manifesto ágil sejam afetados. Este trabalho foi classificado como “*Philosophical Theoretical Paper*”, utilizando a classificação de estudo elaborada por Wieringa et al [WMR06]. Esta classificação indica que foi realizada uma estruturação da área em forma de taxonomia ou *framework* conceitual.

O segundo artigo listado [NK08], escrito pelos mesmos pesquisadores do primeiro estudo, apresenta e avalia um modelo de integração de gerenciamento de riscos para o método ágil Scrum. O objetivo do estudo é verificar se o modelo proposto constitui uma solução válida para a lacuna existente nos métodos ágeis com relação ao gerenciamento de riscos. Este artigo foi classificado como “*Solution proposal*”, de acordo com o estudo de Wieringa et al [WMR06], indicando que o trabalho relata um problema existente e propõe uma solução para o mesmo.

O terceiro e último artigo listado [KCD+10] realiza uma comparação entre diversas metodologias de gerenciamento de projetos, classificando os projetos analisados por sua natureza, tamanho, escopo, ambiente, diversidade e tipo de projeto. A análise foi realizada no gerenciamento de projetos com foco na qualidade e gerenciamento de riscos. O tema básico do estudo indica a forma como diferentes empresas gerenciam seus projetos, através da comparação da qualidade e de risco, apontando as semelhanças entre as metodologias de gerenciamento dos projetos, definido pela área de governança de TI das empresas. De acordo com a classificação indicada por Wieringa et al [WMR06], este estudo foi classificado como “*Experience Paper*”, representando a experiência pessoal dos autores de como os fatos relatados ocorreram na prática.

6.9 RESPOSTAS ÀS QUESTÕES DE PESQUISA

A análise realizada através da leitura completa dos artigos selecionados teve o objetivo de responder às questões de pesquisa apresentadas como objetivos desta revisão sistemática da literatura da área. Entretanto, a quantidade reduzida de trabalhos

obtidos após a definição dos estudos relevantes não permite organizar os dados quantitativos de forma clara e definitiva.

Desta forma, a principal questão de pesquisa (QP) desta revisão sistemática não pode ser respondida em decorrência de não terem sido identificados estudos relevantes que utilizem metodologia de gerenciamento de projeto com gerenciamento de risco implícito. Para atender aos questionamentos secundários propostos na revisão sistemática, os artigos obtidos foram lidos e analisados. Concluiu-se que poucos estudos são realizados no contexto de gerenciamento de projetos que não definem o gerenciamento de riscos de forma explícita. Os estudos recentes originados na Universidade de Stockholm, Suécia, [NK07, NK08] podem indicar que aqueles autores identificaram esta lacuna de estudos e tem iniciado pesquisas nesta área.

Entretanto, ambos os estudos identificados [NK07, NK08] não apresentam de forma clara a real necessidade de alteração dos métodos de gerenciamento de projetos adaptativos que não possuem um procedimento para o gerenciamento de riscos de forma explícita, para que possam tratar os riscos de projetos de desenvolvimento de software de forma mais eficiente. Assim, pouco se sabe a respeito de gerenciamento de riscos em projetos de desenvolvimento de software no contexto de métodos de gerenciamento e desenvolvimento de projetos de software que não possuem o gerenciamento de riscos de forma explícita. A resposta para a questão secundária da pesquisa QS1 é atendida de forma parcial.

Quanto à segunda questão secundária de pesquisa QS2, os estudos analisados indicam que é possível adaptar o gerenciamento de riscos no método ágil Scrum, sem comprometer os ideais do manifesto ágil, de forma a incluir uma forma sistemática de gerenciamento de riscos no Scrum. Entretanto, não se pode determinar se os riscos mais comuns ocorridos em projetos de desenvolvimento de software são tratados de forma no método ágil Scrum. Os estudos apenas indicam a possibilidade de adaptação do modelo, não justificando a real necessidade de adaptação no método ágil Scrum. Desta forma, a questão QS2 “*O que se sabe sobre gerenciamento de riscos em projetos de desenvolvimento de software que utilizam o método ágil Scrum?*” é respondida, com base nos estudos identificados, sugerindo que novos estudos sejam conduzidos neste tema.

6.10 CONCLUSÃO DA REVISÃO SISTEMÁTICA DA LITERATURA DA ÁREA

A conclusão desta revisão sistemática da literatura indica que existem lacunas de estudo na área de gerenciamento de riscos em projetos de desenvolvimento de software que utilizam métodos ágeis, ou ainda, metodologias de gerenciamento de projetos, baseadas em abordagens adaptativas, que não possuam um processo de gerenciamento de riscos explicitamente definido.

Sendo esta uma característica evidente nas abordagens adaptativas, o gerenciamento de riscos em projetos de desenvolvimento de software com métodos ágeis, especificamente o Scrum, constitui uma área com potencial a ser explorado.

7 ESTUDO DE CAMPO

O estudo de campo foi elaborado a partir da constatação da necessidade de verificar como os riscos comuns identificados no gerenciamento de projetos de desenvolvimento de software estão sendo tratados em projetos que utilizam o método ágil Scrum. Também se buscou mapear possíveis lacunas que indiquem a possibilidade de adoção de algumas práticas ágeis como extensão ao Scrum para que os riscos comuns identificados sejam tratados de forma satisfatória. A seguir, serão descritos os passos adotados para condução deste estudo de campo, conforme a metodologia apresentada.

7.1 OBJETIVO

O objetivo principal deste estudo de campo foi verificar como os riscos comuns identificados no gerenciamento de projetos de desenvolvimento de software e apresentados no Capítulo 5 estão sendo tratados em projetos que utilizam o método ágil Scrum. Também se buscou identificar possíveis lacunas onde possa ser sugerida a adoção de práticas ágeis que contribuam com o método ágil Scrum para realizar o tratamento satisfatório dos riscos comuns identificados.

7.2 UNIDADE DE ANÁLISE E CONFIDENCIALIDADE

A unidade de análise deste estudo de campo é o profissional que utiliza ou utilizou o Scrum em empresas de desenvolvimento de software. A escolha dos entrevistados foi realizada com o apoio do orientador deste estudo, que inicialmente entrou em contato com as empresas, solicitando permissão para realização do estudo e pedindo o envio de uma listagem com os nomes de possíveis interessados em participar da pesquisa.

Um dos critérios respeitados na condução deste trabalho foi a questão da confidencialidade, tanto das organizações quanto dos profissionais. Os respondentes deveriam concordar em realizar a entrevista e estavam cientes do caráter de confidencialidade dos dados obtidos. Além disto, foram definidos os seguintes papéis para compor a lista dos possíveis entrevistados:

- Gerentes de Equipes de Desenvolvimento;
- Gerentes de Projetos;
- Analistas de Sistemas;
- Desenvolvedores;

- Líderes da Infraestrutura ou de Suporte.

7.3 PLANEJAMENTO DO ESTUDO DE CAMPO

O estudo de campo foi organizado em etapas, com os procedimentos identificados na Tabela 10.

Tabela 10 – Procedimentos do estudo de campo

a. Levantamento das questões e estruturação do guia para a entrevista	
Participantes	Paulo Jacó Rech
Data	12/11/2012
Local	FACIN PUCRS – Faculdade de Informática da PUCRS
b. Revisão do guia para a entrevista	
Participantes	Prof. Dr. Rafael Prikladnicki
Data	26/11/2012
Local	AGT PUCRS - Agência de Gestão Tecnológica da PUCRS
c. Validação de face e conteúdo	
Participantes	Prof. Gustavo Costa
Data	14/12/2012
Local	CAS - Centro Administrativo Sicredi
d. Pré-teste	
Participantes	Mestrando Bernardo Estácio
Data	14/12/2012 – 11:00 – 12:00
Local	FACIN PUCRS – Faculdade de Informática da PUCRS
e. Aplicação das entrevistas – Questões organizacionais	
Participantes	Gerentes de Projetos e membros de equipes Scrum
Data	17/12/2012 a 11/01/2013
Local	A combinar com cada entrevistado

7.4 RECURSOS UTILIZADOS

Os recursos tecnológicos e materiais utilizados para realização deste Estudo de Campo foram simples, permitindo a realização das entrevistas em qualquer ambiente reservado. Para as entrevistas, os recursos necessários se limitavam ao uso de uma caneta, um gravador para registro de áudio da entrevista e uma ficha de entrevista, impressa com antecedência (Anexo I), contendo a relação de riscos comuns em projetos de desenvolvimento de software e as questões relativas a esses riscos. Esta planilha foi impressa em duas vias, para o acompanhamento do respondente durante a condução da entrevista. Também foi utilizado um documento de roteiro de entrevista (Anexo II), para facilitar a condução da entrevista por parte do pesquisador.

Após a realização das entrevistas, todos os dados obtidos foram transcritos para um computador com apoio do aplicativo *Microsoft Office®*, sendo estes os recursos

tecnológicos utilizados nesta etapa do trabalho: um computador equipado com um software de planilha eletrônica e editor de textos.

7.5 DIMENSÕES DO ESTUDO

O estudo foi planejado a partir de 4 possíveis subconjuntos para classificação de cada um dos 33 riscos comuns identificados na literatura para projetos de desenvolvimento de software, de acordo com a experiência dos profissionais que utilizam o método ágil Scrum para gerenciamento de projetos. O esquema abaixo, ilustrado na Figura 11 representa graficamente estes quatro subconjuntos possíveis.

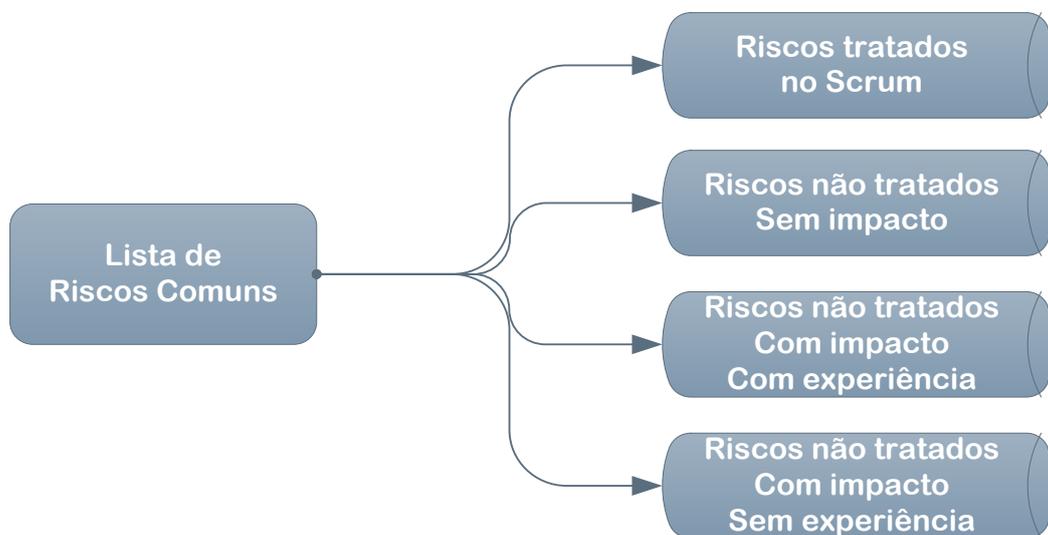


Figura 11 – Subconjuntos de Riscos Comuns com Scrum

- **Riscos tratados no Scrum:** indica o subconjunto de riscos que são tratados de forma implícita nas práticas do método ágil Scrum. Este subconjunto representa a efetividade do modelo de gerenciamento de projetos onde os riscos são tratados mesmo sem a definição de um processo específico de gerenciamento de riscos;
- **Riscos não tratados, Sem impacto:** indica o subconjunto de riscos que o método ágil Scrum não trata de forma implícita, mas que não gera impactos para o projeto, não havendo necessidade de tratamento por estar fora do escopo do projeto. Este subconjunto indica que alguns riscos comuns aos projetos de software são tratados de alguma outra forma que não pelo uso do método ágil Scrum;
- **Riscos não tratados, Com impacto, Com experiência:** indica o subconjunto de riscos que o Scrum não trata, mas que devem ser identificados,

analisados, ter respostas planejadas e monitorados. O tratamento dado aos riscos será então identificado a partir da experiência dos respondentes e indica o subconjunto de riscos que aponta uma lacuna no gerenciamento de riscos implícito no método ágil Scrum. A experiência dos entrevistados irá contribuir para uma proposta de melhoria no Scrum;

- **Riscos não tratados, Com impacto, Sem experiência:** indica o subconjunto de riscos que o Scrum não trata, mas que devem ser identificados, analisados, ter respostas planejadas, e monitorados. Não será possível identificar o tratamento dado aos riscos neste subconjunto devido ao fato de que os respondentes não possuem experiência ou casos para compartilhar no gerenciamento destes riscos no contexto do uso de Scrum.

7.6 COLETA DE DADOS

Os dados foram coletados através de um roteiro de entrevistas com questões fechadas e abertas. Este roteiro contém uma seção para informações demográficas dos entrevistados, que foram selecionados por conveniência. As informações demográficas possibilitam o mapeamento de fatores para composição e agrupamento dos dados principais da entrevista. Informações de data da realização da entrevista, hora de início e hora final também foram registradas. A Tabela 11 indica as principais informações secundárias mapeadas nesta pesquisa.

Tabela 11 – Perguntas de Informações Demográficas do Estudo de Campo

Questões	
Informações Demográficas	Nome: _____ Nascimento: _____
	Curso (nível mais alto): _____
	Instituição: _____ Concluído em: _____
	Tempo de experiência profissional na área de Informática: ____ anos.
	Tempo de experiência profissional com Métodos Ágeis: ____ anos.
	Tempo de experiência profissional com Scrum: ____ anos.
	Departamento/área: _____
	Vínculo empregatício: _____ Tempo de empresa: ____ anos.
	Papel/Função atual: _____ Quantos funcionários a empresa possui?
	Como Métodos Ágeis é utilizado na sua empresa?
	Como o Scrum é utilizado na sua empresa?
	Como o Scrum é utilizado nos seus projetos?

Quanto aos dados principais deste estudo de campo, para que a entrevista fosse conduzida de maneira a permitir a classificação dos 33 riscos comuns, de acordo com o tratamento relatado pelo profissional entrevistado e sua experiência com Scrum, foi definida uma sequência de perguntas de forma a possibilitar a análise quantitativa das

respostas obtidas. Foram planejadas duas etapas para o sequenciamento das perguntas da entrevista: a primeira etapa buscando um retorno direto, caracterizado por uma resposta simples positiva ou negativa. Esta etapa buscava mapear os dados quantitativos da análise. A segunda etapa buscando identificar a experiência do profissional da área de forma aprofundada, gerando dados qualitativos para análise. O fluxo de questionamento conduzido nas entrevistas pode ser verificado na Figura 12, onde cada risco comum seria identificado e relatado ao entrevistado, salientando as principais situações de ocorrência deste risco em projetos de desenvolvimento de software.

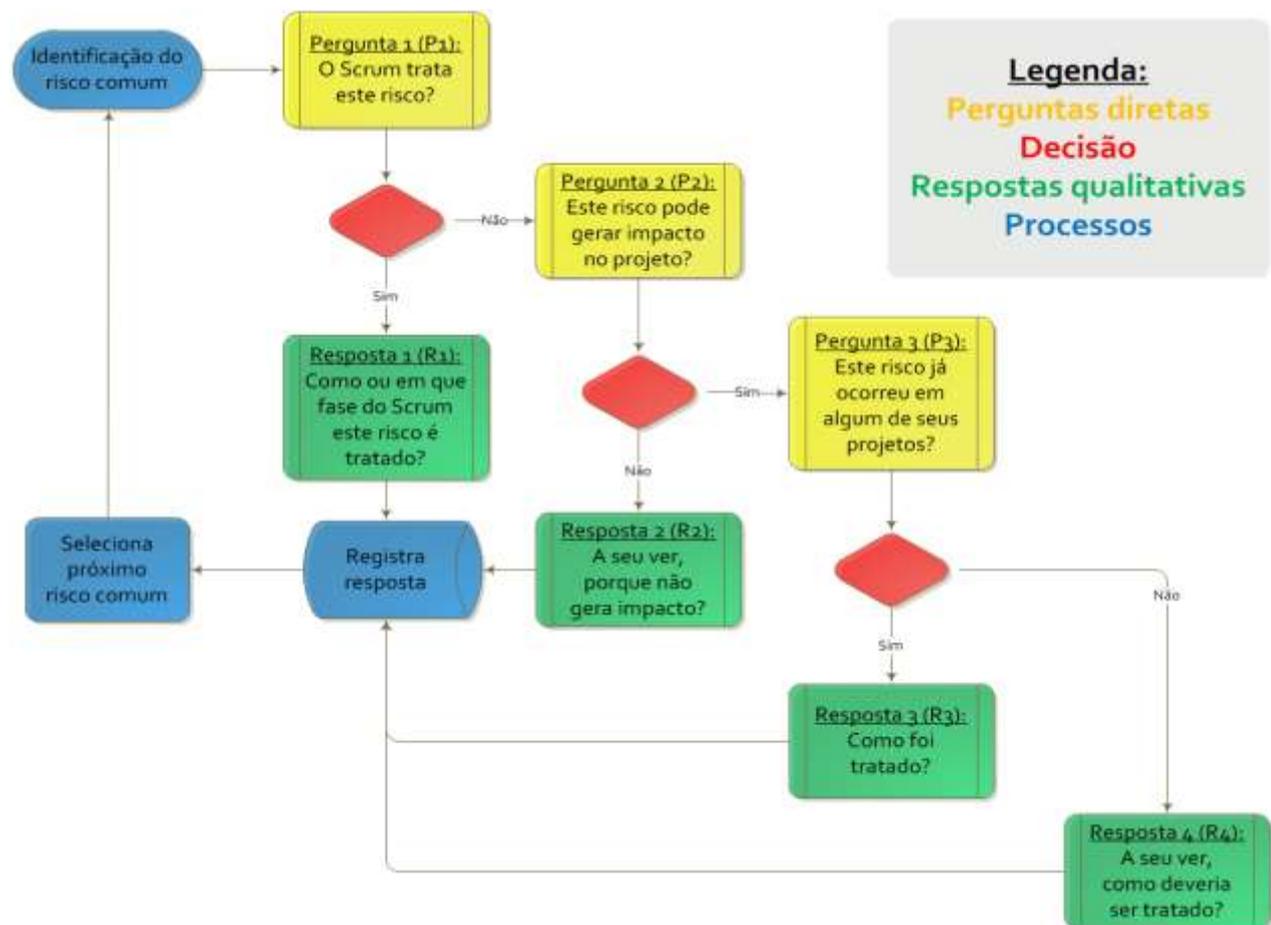


Figura 12 – Fluxo de entrevistas de Estudo de Campo

Para registro das respostas obtidas a partir da aplicação do fluxo de entrevistas identificado anteriormente, foi utilizada uma planilha contendo a listagem dos riscos comuns em projetos de desenvolvimento de software, uma descrição da ocorrência de cada risco comum, a dimensão a que pertence o risco e as lacunas para registro das respostas dos entrevistados, tanto diretas quanto dissertativas. Esta planilha de obtenção das respostas pode ser visualizada na Tabela 12 a seguir.

Tabela 12 – Planilha de Dados Principais do Estudo de Campo

#	Dim.	Identificação do Risco Comum	Perguntas			Resposta dissertativa
			P1	P2	P3	
1	ORG	Reestruturação da organização <i>Mudança da gerência ou alteração da estrutura organizacional.</i>				
2	ORG	Ambiente organizacional instável <i>Cargos vagos sem alocação de pessoas, falta de gerentes ou coordenadores, dificultando ou retardando a tomada de decisão.</i>				
3	ORG	Mudança de prioridade da gerência <i>Novos objetivos definidos pela gerência podem alterar as prioridades dos projetos.</i>				
4	ORG	Falta de comprometimento da alta direção com o projeto <i>Supervisão ineficiente ou inexistente dos executivos da organização.</i>				
5	ORG	Política corporativa com efeito negativo sobre o projeto <i>Políticas da empresa podem ser definidas em desalinhamento com a realidade diária da condução de projetos. Metodologia Ágil, mas empresa não é.</i>				
6	P&C	Comunicação ineficaz <i>Envolvimento insuficiente da equipe do projeto, falha ao indicar mudanças, falta de relatórios de status do projeto.</i>				
7	P&C	Tecnologia ineficaz para gestão de projetos <i>A escolha de uma ferramenta eficaz para auxílio no gerenciamento de projetos pode influenciar a condução do mesmo.</i>				
8	P&C	Falta de metodologia eficaz no gerenciamento do projeto <i>A equipe não emprega controle de mudanças, sem planejamento ou outras habilidades necessárias ao processo.</i>				
9	P&C	Etapas do projeto não definidas claramente <i>Planejamento superficial ou projeto não está bem estruturado (falta de WBS).</i>				
10	P&C	Gold plating <i>Entregar mais do que foi definido no escopo do projeto.</i>				
11	P&C	Deficiências na execução de tarefas e componentes fornecidos externamente <i>Ao alocar uma atividade ou componente do projeto para um recurso externo e esta atividade ou componente apresenta falha na execução.</i>				
12	P&C	Estimativas irreais de tempo e custo <i>Ocorre quando as estimativas do projeto são muito cautelosas ou ousadas demais, gerando impacto na qualidade do projeto em virtude da redução do tempo de teste ou treinamento.</i>				
13	P&C	Alteração de escopo ou objetivos <i>Mudanças nos negócios ou reorganização durante o projeto.</i>				
14	P&C	Escopo ou objetivos não claros ou incompreendidos <i>É impossível definir o escopo ou objetivo real do projeto devido a diferenças de opinião entre os usuários.</i>				
15	P&C	Falta de planejamento ou planejamento inadequado <i>A atividade de planejamento não é considerada importante ou não é realizada.</i>				
16	P&C	Gestão de mudanças inadequada <i>Cada projeto necessita de um processo para gerir mudança para que o custo e escopo sejam controlados. O aumento do escopo sem controle é a evidência de uma gestão da mudança ineficaz, onde os parâmetros de sucesso não estão definidos.</i>				
17	P&C	Andamento do projeto não monitorado adequadamente <i>Ocorre quando existe falha na definição de pontos de controle do projeto.</i>				
18	P&C	Estimativa inadequada dos recursos necessários <i>Falta na identificação de recursos para o projeto compromete o andamento.</i>				
19	COM	Alto nível de complexidade técnica <i>Ocorre quando o produto é composto por módulos de diferentes tecnologias e de difícil entendimento e integração.</i>				

#	Dim.	Identificação do Risco Comum	Perguntas			Resposta dissertativa
			P1	P2	P3	
20	COM	Falta de desempenho em tempo real <i>Simulação, medições, modelagem, prototipagem, ajustes ou instrumentação apresentam falta de desempenho devido a ambientes limitados.</i>				
21	COM	Tecnologia inexistente ou imatura <i>O produto do projeto exige a utilização de tecnologia ainda não disponível ou muito recentes e imaturas</i>				
22	REQ	Desenvolver as funções erradas do software <i>Mapeamento incompleto dos componentes ou da arquitetura do software.</i>				
23	REQ	Desenvolver a interface de usuário errada <i>O usuário está habituado a utilizar outro padrão de interface na organização.</i>				
24	REQ	Alterações tardias nos requisitos <i>Solicitação de mudanças durante a fase de testes do produto, por exemplo.</i>				
25	REQ	Falta de requisitos congelados <i>Devido às constantes necessidades de mudanças dos usuários, o software corre o risco de jamais ser concluído.</i>				
26	REQ	Incompreensão dos requisitos <i>Não definir todos os requisitos do sistema antes de começar os trabalhos, gerando requisitos de forma superficial causando falha na estimativa de esforço, habilidades e tecnologia necessária para o projeto.</i>				
27	TEA	Membros da equipe de desenvolvimento inadequadamente treinados <i>Falta de conhecimento na tecnologia utilizada, falta de experiência ou treinamento ineficaz.</i>				
28	TEA	Membros da equipe com falta de habilidade especializada exigida pelo projeto <i>Por exemplo, tecnologia, conhecimento de negócios e experiência inexistentes entre os recursos da equipe.</i>				
29	TEA	Falta de habilidade na liderança de projetos <i>A equipe do projeto é formada e o gerente não tem autoridade ou habilidades para obter sucesso.</i>				
30	USE	Conflitos entre usuários <i>Diferentes objetivos dos departamentos dos usuários geram expectativas distintas.</i>				
31	USE	Falha em obter comprometimento do usuário <i>Usuários apontam o gerente do projeto como culpado pela falta de envolvimento do cliente.</i>				
32	USE	Usuários com resistência a mudanças <i>Obstáculo percebido quando o usuário não quer mudar de rotina ou não está motivado na empresa.</i>				
33	USE	Falha ao gerir expectativas do usuário final <i>Expectativas determinam o sucesso ou o fracasso de um projeto. Expectativas incompatíveis com a entrega (muito alta ou muito baixa) causam problemas. As expectativas devem ser corretamente identificadas e constantemente ajustadas a fim de evitar o fracasso.</i>				

7.7 EXECUÇÃO E ANÁLISE DOS DADOS

Após a elaboração do planejamento do estudo de campo, o mesmo foi submetido à revisão face e conteúdo e pré-teste, conforme processo de elaboração de pesquisa científica indicado por Gil [GIL10].

O período de execução das entrevistas foi realizado entre o final do ano de 2012 e o início de 2013. Esta etapa foi conduzida com atraso relativo ao cronograma

apresentado no Seminário de Andamento da pesquisa, em decorrência do acréscimo de uma etapa de Revisão Sistemática da Literatura. Esta etapa mostrou-se muito importante neste trabalho, pois possibilitou que o pesquisador obtivesse uma melhor sustentação teórica para o tema em estudo. Entretanto, em decorrência do atraso na realização deste estudo de campo, o período de execução do mesmo mostrou-se um obstáculo por tratar-se de datas de difícil disponibilidade de profissionais qualificados a responder a pesquisa. Desta forma, foram agendadas e executadas 9 entrevistas, com profissionais de diversas empresas de desenvolvimento de software da região metropolitana de Porto Alegre – RS. A Tabela 13 apresenta as informações demográficas obtidas .

Tabela 13 – Informações Demográficas do estudo de campo

Informações Demográficas	Valor Médio
Tempo médio de experiência com Informática	15,44 anos
Tempo médio de experiência com Métodos Ágeis	4,83 anos
Tempo médio de experiência com Scrum	4,17 anos
Tempo médio de duração das entrevistas	01h02m
Vínculo empregatício	CLT: 78%, Sócio: 22%
Idade média dos entrevistados	34,56 anos

Quanto aos dados principais do estudo, obtidos após a aplicação das entrevistas e a transcrição das mesmas para a planilha de controle e análise, foi possível obter valores quantitativos para as respostas dos entrevistados. A condução das entrevistas através do fluxo identificado no protocolo de pesquisa, conforme Figura 12 acima, possibilitou que os riscos comuns fossem classificados em subconjuntos seguindo uma lógica baseada nas respostas diretas obtidas pelos profissionais da área.

- **Riscos tratados no Scrum** => P1=Sim;
- **Riscos não tratados, Sem impacto** => P1=Não E P2=Não;
- **Riscos não tratados, Com impacto, Com experiência** => P1=Não E P2=Sim E P3=Sim;
- **Riscos não tratados, Com impacto, Sem experiência** => P1=Não E P2=Sim E P3=Não.

Desta forma, conforme o fluxo de entrevista, as perguntas que exigiam respostas dissertativas (R1, R2, R3 ou R4) só poderiam ser respondidas uma única vez para cada risco comum identificado na lista, possibilitando que a experiência do profissional sendo entrevistado fosse registrada de forma qualitativa, conforme sua percepção dos fatos e cenários descritos para cada risco comum. A planilha com todas as respostas diretas, devidamente transcritas, após serem obtidas neste estudo de campo está disponível no Anexo III deste volume.

7.8 CONCLUSÃO DO ESTUDO DE CAMPO

Após a realização da etapa de coleta e análise dos dados, houve a constatação de que o método ágil Scrum, como modelo adaptativo para o gerenciamento de projetos, pode tratar os principais riscos em projetos de desenvolvimento de software de forma satisfatória. Dentre a lista dos 33 riscos comuns identificados em projetos de desenvolvimento de software, 21 são tratados adequadamente (conforme critério de linha de corte de 75%, adotado por conveniência) com a adoção do método ágil Scrum. A Figura 13 apresenta a análise dos resultados do estudo de campo, demonstrando os riscos agrupados por dimensões e qual a efetividade do uso do Scrum sobre as dimensões de risco comum.

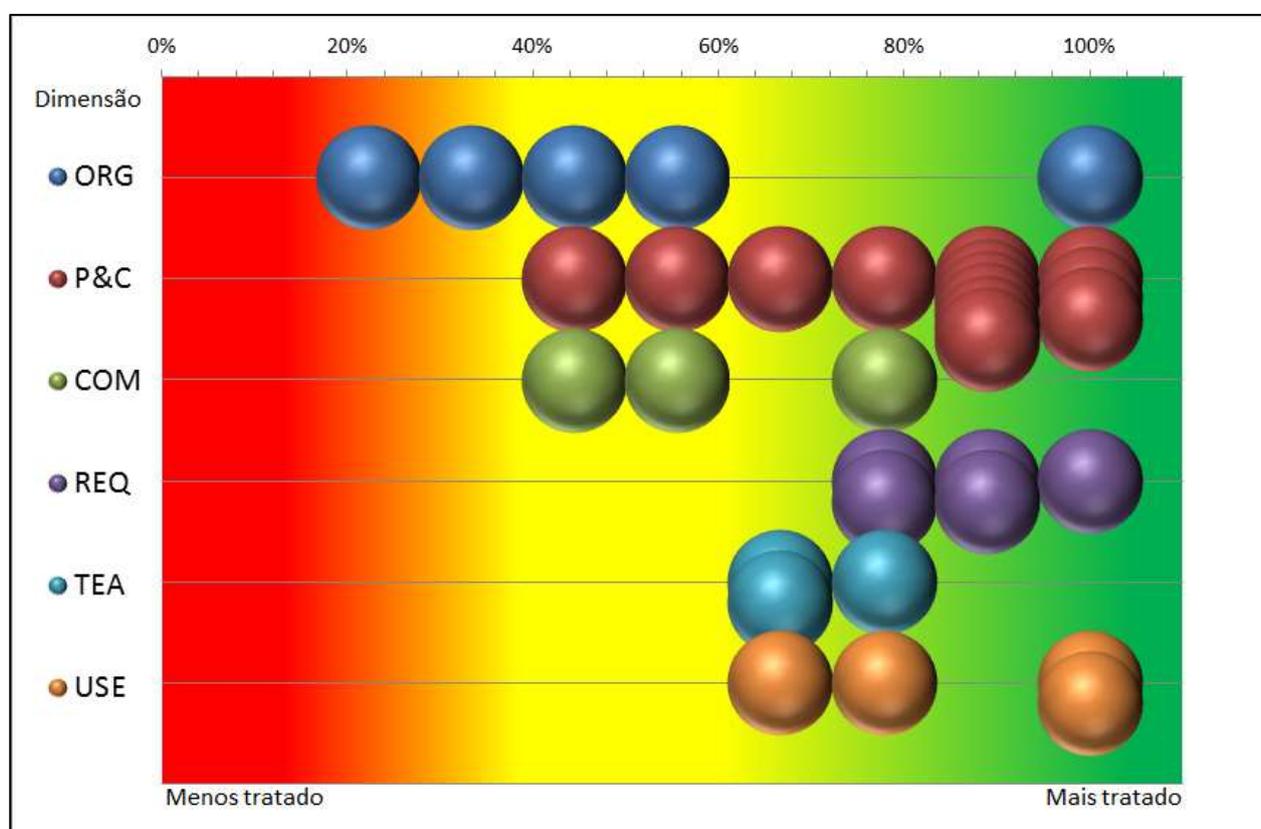


Figura 13 – Mapa de tratamento de Riscos Comuns com Scrum

No eixo vertical do gráfico, estão relacionadas as dimensões de riscos comuns, na ordem em que foram apresentadas nas entrevistas. O eixo horizontal indica o tratamento dado pelo Scrum, de acordo com a percepção dos profissionais da área, para cada um dos riscos comuns da respectiva dimensão. Quanto mais para a direita, mais o risco comum em questão é tratado adequadamente. A escala deste eixo indica o percentual de concordância entre os entrevistados quanto ao tratamento de riscos implícito no Scrum.

Percebe-se que os riscos comuns relacionados à dimensão de Ambiente Organizacional apresentam indícios de que o Scrum não trata adequadamente questões que sobrepõem o âmbito do projeto e da equipe de projeto, apresentando dificuldade em gerenciar, de forma implícita, o impacto causado pelos riscos originados na organização.

Para facilitar a análise dos dados obtidos, cada risco comum verificado neste estudo recebeu o índice de **Tratamento de Risco Comum (TRC)**, que diz respeito ao percentual de entrevistados que indicaram que o Scrum, como método ágil cujo gerenciamento de riscos está implícito, trata o risco comum em questão. Uma fórmula para descrever melhor este índice pode ser verificada na Equação 1 abaixo:

$$TRC(n) = \frac{\sum_t^1 Se (P1 = Sim), 1 senão 0}{t}$$

Equação 1 - Tratamento de Risco Comum (TRC)

Onde, n é igual ao número ordinal do risco comum em questão, t indica o total de entrevistados, $P1=Sim$ representa o número de respostas em que o entrevistado indica que o Scrum trata o risco comum em questão, para a primeira pergunta direta (P1) do fluxo de entrevista apresentado na Figura 12 acima.

Outro índice aplicado para a análise dos dados obtidos foi gerado em decorrência do conceito TRC, constituindo o conceito de **Tratamento de Dimensão de Risco Comum (TDRC)**. Este índice indica o tratamento percebido pelos entrevistados com relação a uma dimensão específica de riscos comuns. O valor representa a média dos valores de TRC dos riscos comuns de uma dimensão. A Equação 2 indica como o valor TDRC pode ser calculado para cada dimensão de risco comum.

$$TDRC(d) = \frac{\sum_{nd}^1 TRC(n)}{nd}$$

Equação 2 - Tratamento de Dimensão de Risco Comum (TDRC)

A variável d representa a dimensão em questão. O valor nd é a quantidade de riscos comuns para a dimensão d , enquanto que n representa o número do risco comum específico da dimensão d .

Para cada dimensão de risco comum, foi realizada a análise considerando os conceitos TRC e TDRC, conforme descrito anteriormente. As Tabelas 14, 15, 16, 17, 18 e 19 abaixo indicam o cálculo destes valores para cada uma das dimensões de riscos comuns. Os valores totais (TDRC) são a média dos valores obtidos em cada risco comum (TRC) das dimensões.

Tabela 14 – TDRC - Dimensão de Ambiente Organizacional

Risco	Dimensão	Descrição	TRC
1	ORG	Reestruturação da organização	56%
2	ORG	Ambiente organizacional instável	44%
3	ORG	Mudança de prioridade da gerência	100%
4	ORG	Falta de comprometimento da alta direção com o projeto	22%
5	ORG	Política corporativa com efeito negativo sobre o projeto	33%
TDRC:			51,11%

Tabela 15 – TDRC - Dimensão de Planejamento e Controle

Risco	Dimensão	Descrição	TRC
6	P&C	Comunicação ineficaz	89%
7	P&C	Tecnologia ineficaz para gestão de projetos	67%
8	P&C	Falta de metodologia eficaz no gerenciamento do projeto	100%
9	P&C	Etapas do projeto não definidas claramente	89%
10	P&C	Gold plating	89%
11	P&C	Deficiências na execução de tarefas e componentes fornecidos externamente	44%
12	P&C	Estimativas irreais de tempo e custo	100%
13	P&C	Alteração de escopo ou objetivos	100%
14	P&C	Escopo ou objetivos não claros ou incompreendidos	89%
15	P&C	Falta de planejamento ou planejamento inadequado	78%
16	P&C	Gestão de mudanças inadequada	89%
17	P&C	Andamento do projeto não monitorado adequadamente	89%
18	P&C	Estimativa inadequada dos recursos necessários	56%
TDRC:			82,91%

Tabela 16 – TDRC - Dimensão de Complexidade

Risco	Dimensão	Descrição	TRC
19	COM	Alto nível de complexidade técnica	78%
20	COM	Falta de desempenho em tempo real	44%
21	COM	Tecnologia inexistente ou imaturas	56%
TDRC:			59,26%

Tabela 17 – TDRC - Dimensão de Requisitos

Risco	Dimensão	Descrição	TRC
22	REQ	Desenvolver as funções erradas do software	78%
23	REQ	Desenvolver a interface de usuário errada	78%
24	REQ	Alterações tardias nos requisitos	89%
25	REQ	Falta de requisitos congelados	89%
26	REQ	Incompreensão dos requisitos	100%
TDRC:			86,67%

Tabela 18 – TDRC - Dimensão de Equipe

Risco	Dimensão	Descrição	TRC
27	TEA	Membros da equipe de desenvolvimento inadequadamente treinados	67%
28	TEA	Membros da equipe com falta de habilidade especializada exigida pelo projeto	67%
29	TEA	Falta de habilidade na liderança de projetos	78%
TDRC:			70,67%

Tabela 19 – TDRC - Dimensão de Usuário

Risco	Dimensão	Descrição	TRC
30	USE	Conflitos entre usuários	78%
31	USE	Falha em obter comprometimento do usuário	100%
32	USE	Usuários com resistência a mudanças	67%
33	USE	Falha ao gerir expectativas do usuário final	100%
TDRC:			86,11%

A análise dos valores de TDRC para cada dimensão de risco comum em projetos de desenvolvimento de software nos permite identificar a **Efetividade** do método ágil Scrum quanto ao tratamento implícito de gerenciamento de riscos proporcionado por este modelo adaptativo destinado ao gerenciamento de projetos, conforme apresentado na Tabela 20 abaixo.

Tabela 20 – Efetividade do Scrum por Dimensão de Risco

Dimensões de Riscos	Sigla	Riscos	TDRC
Ambiente Organizacional	ORG	5	51,11%
Planejamento e Controle	P&C	13	82,91%
Complexidade	COM	3	59,26%
Requisitos	REQ	5	86,67%
Equipe	TEA	3	70,67%
Usuário	USE	4	86,11%
Efetividade:		33	75,76%

A escala de cores da coluna TDRC identifica o nível de efetividade do tratamento de riscos implícito no Scrum, por dimensão de risco, de acordo com o relato dos profissionais da área que utilizam o Scrum para gerenciamento de seus projetos. Valores em vermelho indicam que existe pouca efetividade. Amarelo indica que há alguma efetividade, embora de forma ainda não satisfatória, conforme linha de corte, apontada por conveniência, no valor de 75% para determinar valores aceitáveis. Já os valores da coluna TDRC indicados na cor verde demonstram que a dimensão de riscos específica é tratada de forma satisfatória, sendo estes a maioria dos casos, representando uma efetividade total de 75,76%.

As análises obtidas com o gráfico apresentado na Figura 13 e na Tabela 20 são complementares. Na visualização gráfica da Figura 13, entretanto, não fica evidente a quantidade de riscos comuns enquadrados em cada escala de tratamento. Apenas percebe-se que existem riscos comuns com maior ou menor tratamento, não mostrando que pode ocorrer de que a maioria dos riscos é tratada de forma ineficaz, de acordo com a opinião dos profissionais da área.

Os dados apresentados demonstram que é possível identificar algumas lacunas no tratamento de riscos com o uso do método ágil Scrum em algumas das dimensões de riscos comuns estudadas. Para estas lacunas foram feitas sugestões de melhorias e extensão do Scrum, apresentadas a seguir.

8 GERÊNCIA DE RISCOS EM PROJETOS COM SCRUM

Conforme ficou evidenciado no estudo de campo apresentado no Capítulo 7 deste trabalho, houve a constatação de que o Scrum pode tratar os principais riscos em projetos de desenvolvimento de software de forma satisfatória. Dentre a lista dos 33 riscos comuns identificados em projetos de desenvolvimento de software, 21 são tratados adequadamente a partir dos resultados encontrados.

Desta forma, para sugerir melhorias ou extensões no Scrum que possibilitem o tratamento adequado às dimensões menos favorecidas com o tratamento de risco inerente a este método ágil, buscou-se apresentar três propostas que complementem o framework Scrum, possibilitando tratar mais efetivamente as dimensões de riscos de Ambientes Organizacionais, Complexidade e Equipe.

8.1 SUGESTÃO DE EXTENSÃO DO SCRUM

A seguir são descritas três sugestões de extensão do Scrum, visando explorar as dimensões específicas de riscos parcialmente tratados, conforme os resultados da análise apresentados anteriormente. As sugestões foram elaboradas a partir do relato dos profissionais da área, conforme a pesquisa de campo, e na literatura especializada.

8.1.1 DIMENSÃO DE AMBIENTE ORGANIZACIONAL – PORTFÓLIO - SCRUM OF SCRUMS

Esta dimensão de riscos comuns em projetos de desenvolvimento de software compreende os fatores que tratam da organização. Conforme a descrição dos riscos desta dimensão, os riscos potenciais estão relacionados à mudanças da gerência ou alteração da estrutura organizacional, cargos vagos sem alocação de pessoas, falta de gerentes ou coordenadores, dificultando ou retardando a tomada de decisão, novos objetivos definidos pela gerência podem alterar as prioridades dos projetos, supervisão ineficiente ou inexistente dos executivos da organização, políticas da empresa podem ser definidas em desalinhamento com a realidade diária da condução de projetos (a metodologia do projeto é ágil, mas a empresa não é).

Conforme a opinião de profissionais da área, segundo entrevistas realizadas na pesquisa de campo, observou-se a tendência de utilizar Gestão de Portfólio de Projetos para os projetos conduzidos com Scrum, buscando alinhar os esforços da equipe de projetos com o planejamento estratégico da organização. Desta maneira, os impactos oriundos de riscos da dimensão de ambiente organizacional seriam mitigados. No

contexto de desenvolvimento de novos produtos e de atividades de pesquisa e desenvolvimento, a gestão de portfólio consiste em gerenciar vários projetos de produtos ou pesquisa e desenvolvimento associados aos objetivos estratégicos da organização [MIG08, SIL06].

A definição de portfólio apresentado pelo PMI, de forma resumida, define o termo como *“uma coleção de projetos, programas ou outras atividades que são agrupadas para facilitar seu gerenciamento efetivo, a fim de atingir os objetivos estratégicos do negócio”* [PMB08]. Ainda seguindo o conceito do PMI, a definição apresentada para gestão de portfólio descreve a ação de *“gerenciamento coordenado dos componentes do portfólio com vistas a atingir objetivos organizacionais específicos”* [PMB08].

As empresas definem estratégias para alcançar seus objetivos e atingir a sua visão. A estratégia organizacional é determinada por um plano que descreve como as forças e competências da empresa serão utilizadas para aproveitar oportunidades, minimizar impactos de ameaças, responder aos cenários de mudanças no mercado e ampliar as atividades operacionais, entre outros objetivos [PMB08, MOO10].

A gestão de portfólio é uma disciplina dentro da governança organizacional, utilizada como um instrumento que auxilia no alinhamento entre a estratégia da organização e o uso dos recursos limitados disponíveis. A gestão de portfólio providencia os meios necessários para transformar a estratégia organizacional em um conjunto de iniciativas operacionais, possibilitando o gerenciamento das atividades com o objetivo de maximizar o uso correto dos recursos da empresa [PMB08].

Desta forma, a gestão de portfólio retrata como um dos mecanismos que tem como objetivo tratar os diversos projetos de uma empresa, em um contexto mais amplo do que a relação entre os projetos, mas também buscando o alinhamento estratégico da empresa [RMM05]. A Figura 14 a seguir demonstra o objetivo da gestão de portfólio.

Pode-se identificar que a visão, a missão, os objetivos e as estratégias organizacionais são os componentes utilizados pela empresa para definir o desempenho desejado. Complementarmente, o planejamento e gestão de portfólio, em conjunto com o planejamento e gerenciamento de alto nível das operações, estabelecem as iniciativas necessárias para atingir o desempenho desejado, sendo esta uma camada intermediária orquestrada entre o nível estratégico e o nível operacional [PMB08].



Figura 14 – Contexto organizacional da gestão de portfólio [PMB08]

No nível operacional constam as operações gerenciais e a gestão de programas e projetos autorizados, referentes à execução das iniciativas necessárias rumo à realização do desempenho desejado pela empresa. Como base nesta estrutura encontram-se os recursos organizacionais, muitas vezes escassos, devendo ser utilizados da melhor maneira possível, sendo aplicados nos projetos que resultem no maior retorno de investimento para a organização [PMB08]. Desta forma, a gestão de portfólio interliga a estratégia organizacional e a realização do conjunto de programas ou projetos autorizados.

No contexto do Scrum, uma das práticas ágeis que possibilitam a integração de equipes de projetos distintos, corroborando com a gestão de portfólio de projetos na organização, é apresentada através de uma reunião conhecida como *Scrum de Scrums*. Conforme Cohn [COH05], “*Scrum of Scrums é uma importante técnica para escalar Scrum em grandes times e projetos. Essas reuniões permitem agrupar os times para discutir seus trabalhos, focando especialmente em área de sobreposição e integração*”. A reunião *Scrum of Scrums* pode ser feita de maneira recursiva, principalmente quando há muitos times envolvidos no projeto, sendo possível dividir os diversos grupos por temas, componentes ou funcionalidades [COH05, FS12].

Conforme relata Henrik Kniberg [KNI07] em seu relato a respeito da aplicação de Scrum em sua empresa, o uso da reunião *Scrum of Scrums* auxiliou na comunicação em toda a corporação, disseminando a informação sobre a situação dos projetos corporativos e mantendo o envolvimento de toda a gestão empresarial, desde o nível de coordenação até a alta direção.

Além dos benefícios obtidos com o Portfólio de Projetos e o *Scrum of Scrum*, o pioneiro na utilização do Scrum, Ken Schwaber, relatado em seu livro “*The Enterprise and Scrum*” [SCH07b] que todo o trabalho de uma organização pode ser definido no *Product Backlog*, facilitando a organização dos projetos que dependem de esforços de outras equipes, chamando este processo de *Enterprise Product Backlog*. Para organizar o *Enterprise Product Backlog*, é necessário implementar os conceitos de Portfólio de Projetos. Schwaber salienta ainda que esta definição deve ser *top-down* na organização, para garantir o apoio de todas as áreas [SCH07b]. A Tabela 21 apresenta um resumo das sugestões de adaptações do Scrum para os riscos comuns da dimensão de ambiente organizacional.

Tabela 21 – Dimensão de Ambiente Organizacional X Portfólio – *Scrum of Scrums*

Lacunas da Dimensão de Ambiente Organizacional	Benefício de Portfólio de Projetos e Scrum of Scrums
<p>Mudança da gerência ou alteração da estrutura organizacional.</p> <p>Cargos vagos sem alocação de pessoas, falta de gerentes ou coordenadores, dificultando ou retardando a tomada de decisão.</p> <p>Supervisão ineficiente ou inexistente dos executivos da organização.</p>	<p>“...Scrum of Scrums auxiliou na comunicação em toda a corporação, disseminando o informação sobre a situação dos projetos corporativos e mantendo o envolvimento...” [KNI07]</p>
<p>Políticas da empresa podem ser definidas em desalinhamento com a realidade diária da condução de projetos. Metodologia Ágil, mas empresa não é.</p>	<p>“Criação do Enterprise Product Backlog, garante o envolvimento de todas as áreas da empresa.” [SCH07b]</p>

Desta forma, os riscos comuns em projetos de desenvolvimento de software da dimensão de ambiente organizacional, identificados como não tratados com efetividade pelo Scrum, poderiam ser tratados.

8.1.2 DIMENSÃO DE COMPLEXIDADE – SPRINT ZERO E PESQUISA E DESENVOLVIMENTO

Com relação aos riscos comuns em projetos de desenvolvimento de software que estão relacionados à dimensão de complexidade, podem-se destacar os fatores que estão ligados a ocorrência de quando o produto é composto por módulos de diferentes tecnologias e de difícil entendimento e integração, ou quando o produto do projeto exige a utilização de tecnologia ainda não disponível ou muito recente e imatura, ou ainda quando as simulações, medições, modelagem, prototipagem, ajustes ou instrumentação apresentam falta de desempenho devido à ambientes limitados.

Neste contexto, conforme o relato de profissionais da área, os riscos da dimensão de complexidade tem sido tratados com a adoção de uma técnica eficaz para a mitigação de impacto, tanto quando é identificado que o projeto envolve novas tecnologias como quando se percebe a necessidade de ambientes mais robustos para condução de testes ou simulações que envolvam o produto do projeto. Esta técnica representa a adoção de uma etapa prévia ao início das *Sprints* do projeto, onde se inclui uma *Sprint* destinada unicamente a dimensionar e identificar os requisitos de ambiente necessários para o início do projeto. Esta *Sprint* é chamada de ***Sprint Zero*** e também aborda itens de ***Pesquisa e Desenvolvimento (P&D)*** para mitigar o impacto causado pela necessidade de desenvolvimento do produto do projeto que envolve novas tecnologias, tecnologias ainda imaturas ou produtos que envolvam alto nível de complexidade devido à composição de módulos de diferentes tecnologias e de difícil entendimento e integração. Itens de P&D também podem ser incluídos no *Backlog do Produto* sempre que for percebida a necessidade de aumentar o conhecimento sobre um determinado componente ou tecnologia utilizada no projeto. Desta forma, antes de um determinado item complexo ser efetivamente desenvolvido, haverá uma etapa de P&D que ocorre na *Sprint* imediatamente anterior, possibilitando que o risco oriundo da tecnologia imatura ou alta complexidade envolvida no projeto sejam mitigados ou mesmo eliminados.

A *Sprint Zero* é considerada uma *Sprint* com o objetivo de entregar a base necessária para que a equipe possa desenvolver as funcionalidades desejadas no produto, segundo Peter Schuh [SCH05]. Conforme as experiências relatadas por *Jim Ungar* em seu artigo [UNG08], a criação de itens de P&D é essencial para o desenvolvimento de qualquer produto. O autor enfatiza que o planejamento do produto deve preceder seu desenvolvimento em uma *Sprint Zero*, envolvendo questões como:

- Identificar os objetivos de negócios e grupos de usuários;
- Conduzir entrevistas e pesquisas contextuais;
- Analisar os resultados da pesquisa e apresentar aos envolvidos
- Fazer recomendações para a condução do projeto;
- Criar protótipos para avaliação do cliente e teste de usabilidade;

Os benefícios identificados com a adoção da Sprint Zero, além da prática da criação de itens de P&D atendem as lacunas apontadas pelos profissionais da área, conforme estudo de campo, para a dimensão de complexidade de riscos. É possível mapear as lacunas apontadas e os benefícios da adoção de *Sprint Zero* e itens de P&D, mitigando ou eliminando os riscos comuns, conforme a Tabela 22 abaixo.

Tabela 22 – Dimensão de Complexidade de Riscos X Sprint Zero e P&D

Lacunas da Dimensão de Complexidade	Benefício de Sprint Zero e P&D
<p>Produto composto por módulos de diferentes tecnologias e de difícil entendimento e integração.</p> <p>O produto do projeto exige a utilização de tecnologia ainda não disponível ou muito recentes e imaturas</p>	<p>“... a criação de itens de P&D é essencial para o desenvolvimento de qualquer produto...” [UNG08]</p>
<p>Simulação, medições, modelagem, prototipagem, ajustes ou instrumentação apresentam falta de desempenho devido a ambientes limitados.</p>	<p>“A <i>Sprint Zero</i> tem o objetivo de entregar a base necessária para a equipe desenvolver o produto.” [SCH05]</p>

Desta forma, os riscos comuns em projetos de desenvolvimento de software da dimensão de complexidade, poderiam ser tratados.

8.1.3 DIMENSÃO DE EQUIPE – PROGRAMAÇÃO EM PAR

O Scrum define que a equipe do projeto deve ser auto-gerenciável e multidisciplinar, sendo este um dos pontos fortes [SCH04]. Entretanto, conforme conclusão da pesquisa de campo, realizada com profissionais da área, alguns riscos comuns de projetos de desenvolvimento de software da dimensão de equipe ocorrem em projetos conduzidos com o método ágil Scrum.

A lacuna identificada nesta dimensão de riscos comuns diz respeito a 2 dos 3 riscos mapeados nesta dimensão. Esta lacuna está relacionada a questões como falta de conhecimento na tecnologia utilizada, falta de experiência ou treinamento ineficaz, membros da equipe com falta de habilidade especializada exigida pelo projeto como conhecimento de negócios e inexperiência entre os recursos da equipe.

Uma das práticas ágeis adotada pelos profissionais da área, conforme relatado nas entrevistas da pesquisa de campo, faz menção ao uso de uma técnica abordada no método ágil *Extreme Programming* (XP) e denomina-se *Pair Programming* ou Programação em Par (PP). Como o nome sugere, trata-se de uma prática de programação que envolve dois programadores em um mesmo computador trabalhando de forma colaborativa [MCD02]. Um dos programadores se comporta como condutor do computador, escrevendo o código a ser compilado. O outro, como observador ou navegador, sendo responsável por revisar o código escrito pelo colega, prevenindo e identificando erros lógicos e sintáticos no programa sendo escrito. Ambos os programadores se revezam nos papéis e tem liberdade para decidir a frequência de troca [MCD02].

A comunicação, além da colaboração, também é um dos requisitos da programação em par. Ambos os programadores devem estar em constante contato para discutir possíveis soluções e erros de código [HAO11]. O sucesso da aplicação desta prática possui relação direta com o ambiente de trabalho, devendo ser aberto e facilitar a comunicação entre os pares [VNL07]. A complexidade da tarefa sendo desenvolvida determina a viabilidade do uso da prática de programação em par. Tarefas rotineiras, triviais ou de teste não melhoram a produtividade com o uso desta técnica [DAS+09]. Os benefícios reportados para a adoção da prática ágil de PP são vários [HAO11]:

- Aumento da produtividade;
- Aumento da qualidade de produto devido à contribuição na revisão contínua de um dos pares;
- Aumento da colaboração e comunicação do time;
- Aprimoramento da condição de trabalho (confiança e motivação), proporcionando que os programadores não se sintam isolados.

Existem muitos estudos que relatam a adoção da prática ágil de PP, tanto no contexto educacional quanto profissional. No contexto educacional, os resultados mostram que a prática cria um ambiente que proporciona um aprendizado mais efetivo aos estudantes, incentivando a pró atividade, a interação social, aumentando o desempenho e a confiança, além de incentivar o interesse na área da computação [SMG11]. Na área profissional, os resultados demonstrados nos estudos relatam o aumento da qualidade do código do produto e o desempenho no trabalho. Entretanto,

algumas críticas também foram relatadas como o aumento de recursos destinados ao projeto, custo de pessoal e conflitos entre os programadores [DAS+09].

Chong [CH07], por meio de um estudo étnico na indústria, observou que existe uma forte influência na interação dos pares dada a diferença de conhecimento dos envolvidos. Sempre que existe grande diferença entre o conhecimento dos profissionais, o programador com menor nível de conhecimento tem maior dificuldade de avaliar os argumentos técnicos apontados pelo programador mais experimentado [CH07]. Outro estudo relevante realizado por Begel [BN08], a maioria dos programadores respondentes de uma *Survey* apontou que preferem formar pares com parceiros de mesmo nível de conhecimento [BN08]. Diferentes combinações de pares quanto ao nível de conhecimento acabam proporcionando diferentes resultados na aplicação da prática PP [CX05]. A utilização de uma combinação contendo um programador experiente e outro com menor nível de conhecimento colabora para o compartilhamento mais efetivo entre os envolvidos. Quando ocorre a combinação de dois programadores experientes, surge oportunidade para a melhoria da qualidade do código gerado e para desenvolvimento de novos conhecimentos. Para Muller [MP04], que conduziu um experimento de PP na área educacional, não há efeito significativo do conhecimento dos programadores em relação à efetividade de PP. No contexto profissional, a prática de PP demonstra trazer um bom desempenho na resolução de problemas incomuns e na iniciação de novos integrantes ao projeto [LC06]. Entretanto, foram mapeados efeitos negativos na utilização de PP, com relação ao tempo e o custo [FSS+11, HA05].

A colaboração obteve um efeito positivo quando a prática de PP é utilizada entre equipes, tanto no contexto profissional [VL07] quanto educacional [BLS08]. Conforme estudo realizado por Bipp [BLS08], a rotatividade dos integrantes dos pares possibilita que todos possam obter conhecimento sobre o projeto como um todo, facilitando a interação e colaboração entre a equipe [BLS08].

A prática de PP também apresentou resultados positivos sobre a motivação da equipe [VK07]. Conforme um experimento conduzido por Muller [MP04], a prática de PP proporcionou um ambiente mais confortável e motivador durante as sessões de codificação, corroborando no desempenho das atividades do projeto [MP04]. Também foram percebidas vantagens na utilização de PP como facilitador no processo de aprendizado dos integrantes da equipe [VL07].

Os benefícios identificados com o uso da prática ágil de PP atendem as lacunas apontadas pelos profissionais da área, conforme pesquisa de campo, para a dimensão de equipe. É possível mapear as lacunas apontadas e os benefícios da prática de PP que mitigam os riscos comuns, conforme a Tabela 23 abaixo.

Tabela 23 – Dimensão de Equipe de Riscos X Benefícios de PP

Lacunas da Dimensão de Equipe	Benefício de PP
Falta de conhecimento na tecnologia utilizada	“PP demonstra trazer um bom desempenho na resolução de problemas incomuns e na iniciação de novos integrantes ao projeto.” [LC06]
Inexperiência entre os recursos da equipe	“No contexto educacional, os resultados mostram que a prática cria um ambiente que proporciona um aprendizado mais efetivo, incentivando a pró atividade, a interação social, aumentando o desempenho e a confiança, além de incentivar o interesse na área da computação.” [SMG11]
Treinamento ineficaz	“Foram percebidas vantagens na utilização de PP como facilitador no processo de aprendizado dos integrantes da equipe.” [VL07]
Membros da equipe com falta de habilidade especializada exigida pelo projeto como conhecimento de negócios	“Rotatividade dos pares possibilita que todos possam obter conhecimento sobre o projeto como um todo, facilitando a interação e colaboração entre a equipe.” [BLS08]

Conforme pode ser verificado, foi possível mapear o uso da prática ágil de programação em par para tratar alguns dos riscos comuns identificados e não tratados pelo Scrum. Neste sentido, o uso de PP, que surgiu explicitamente durante as entrevistas na pesquisa de campo, leva a uma reflexão sobre o uso de práticas de outros métodos ágeis de forma complementar ao uso das práticas do Scrum, visando assim garantir uma maior cobertura de tratamento dos riscos comuns.

9 CONCLUSÕES

O objetivo principal desta pesquisa foi planejar e desenvolver um estudo empírico visando identificar como os riscos mais comuns encontrados na literatura de gerenciamento de projetos de desenvolvimento de software são tratados no Scrum.

Para isto, ao longo da pesquisa foi possível aprofundar o tema central a partir da realização de estudos utilizando-se métodos primários e secundários de pesquisa. Assim, inicialmente se desenvolveu uma lista de riscos comuns em projetos de desenvolvimento de software (Capítulo 5). Visando determinar o melhor método científico para realização da pesquisa, desenvolveu-se uma revisão sistemática da literatura, apresentada no Capítulo 6. Além disso, através do estudo de campo (Capítulo 7) foi possível identificar quais dos riscos comuns eram tratados no Scrum, para então propor adaptações ao Scrum para os riscos não tratados de forma satisfatória (Capítulo 8). Desta forma, entende-se que os objetivos geral e específicos propostos foram alcançados.

9.1 CONTRIBUIÇÃO

Com a conclusão desse estudo houve contribuição ao conhecimento tanto para a área acadêmica quanto para a indústria, além de ter proporcionado o crescimento profissional do pesquisador. Esta pesquisa contribui cientificamente agregando conhecimento na área de gerenciamento de projetos de software, no contexto de gerenciamento de riscos em projetos que utilizam abordagens adaptativas (neste caso o método ágil Scrum).

Na indústria, empresas que utilizam Scrum como método para gerenciar projetos serão beneficiadas com o conhecimento gerado por esta pesquisa e podem aprimorar seus processos ao terem conhecimento das lacunas identificadas quanto aos riscos comuns em projetos de desenvolvimento de software que utilizam o método ágil Scrum.

O pesquisador obteve conhecimentos sobre diversas áreas da Engenharia de Software, estando atualizado com o estado da arte sobre o tema de gerenciamento de projetos, gerenciamento de riscos, abordagens adaptativas, especialmente com relação ao método ágil Scrum. Além do conhecimento científico, a contribuição pessoal obtida quanto a metodologias científicas, aplicação de estudos de campo e criação de material acadêmico proporcionou um ganho inestimável.

As principais contribuições obtidas durante a realização deste trabalho podem ser listadas conforme descrito abaixo:

- **Lista de riscos comuns em projetos de desenvolvimento de software:** a realização deste estudo possibilitou sintetizar esta lista;
- **Identificação de lacunas para continuação de estudos na área:** a realização da revisão sistemática da literatura possibilitou identificar a necessidade de ampliação de estudos na área de gerenciamento de riscos e abordagens adaptativas.

9.2 LIMITAÇÕES

As principais limitações deste trabalho referem-se principalmente ao pequeno número de entrevistas realizadas no estudo de campo. Desta forma, a generalização dos resultados obtidos ficou limitada.

Quanto aos resultados apresentados, incluindo as propostas de adaptações do Scrum, não houve a avaliação prática destas propostas. Entretanto, elas foram baseadas em um estudo científico planejado de forma rigorosa, o que garante algum grau de credibilidade nas respostas e resultados obtidos.

9.3 TRABALHOS FUTUROS

Identifica-se um potencial de crescimento nas áreas de pesquisa contempladas neste estudo. Desta forma, sugere-se explorar as seguintes pesquisas no futuro:

- Identificação de sugestões adicionais de melhorias para o Scrum, a partir de análise de cada risco não tratado adequadamente;
- Avaliação prática das sugestões de adaptações do método ágil Scrum propostas neste estudo;
- Replicação do estudo de campo realizado neste trabalho;
- Aprofundar os estudos realizados na revisão sistemática da literatura sobre o que se sabe a respeito das práticas adotadas por profissionais quanto ao gerenciamento de riscos em projetos que utilizam Scrum.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AAS+08] Arantes, E.; Anselmo, J.; Senise, L.; Sibinelli, P. "Business & Technology Review - Gerenciamento de Projetos", Rio de Janeiro, Promon, 2008.
- [AMB08] Ambler, S. W. "Agile Adoption Rate Survey Results: February 2008", Disponível em: <http://www.ambysoft.com/surveys/agileFebruary2008.html>, AmbySoft, 2008.
- [AMV06] Aspray, W.; Mayadas, F.; Vardi, M. Y. "Globalization and Offshoring of Software," A Report of the ACM Job Migration Task Force, Association for Computing Machinery, 2006.
- [AP08] Audy, J. L. N.; Prikladnicki, R. "Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas", Rio de Janeiro, Elsevier, 2008, 211p.
- [BLS08] Bipp, T; Lepper, A; Schmedding, D.; "Pair programming in software development teams - an empirical study of its benefits"; Information and Software Technology, p.231–240, 2008.
- [BN08] Begel, A; Nagappan, N.; "Pair programming: what's in it for me?"; ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '08); ACM, New York, NY; p.120-128.
- [BOE06] Boehm, B. "A View of 20th and 21st Century Software Engineering," In Proceedings of the 28th International Conference on Software Engineering, 12-29, Shanghai, 2006.
- [BOE91] Boehm, B. "Software Risk Management: Principles and Practices", IEEE Computer, v. 8, p. 32-41, Janeiro 1991.
- [CH07] Chong, J; Hurlbutt, T.; "The Social Dynamics of Pair Programming"; International Conference on Software Engineering; p.354-363; 2007
- [CLC04] Cohen, D.; Lindvall, M.; Costa, P. "An introduction to agile methods. In Advances in Computers", New York, Elsevier Science, 2004.
- [COH05] Cohen, M.; "Agile Estimating and Planning"; Prentice Hall; 2005.
- [COH12] Cohn, M.; "Managing Risk on Agile Projects with the Risk Burndown Chart"; Disponível em: <http://blog.mountangoatsoftware.com/managing-risk-on-agile-projects-with-the-risk-burndown-chart>; Acessado em: Junho/2012.
- [CP10] Cukier, D.; Prikladnicki, R. "Introduction to Agile Software Development Methods", Salvador, CBSOft, 2010.

- [CTH08] Christopher, R. N.; Taran, G.; Hinojosa, L. L.; "Explicit Risk Management in Agile Processes"; Agile Processes in Software Engineering and Extreme Programming - 9th International Conference"; Limerick, Ireland; 2008.
- [CX05] Cao, L; Xu, L.; "Activity Patterns of Pair Programming"; International Conference on System Sciences; p.88; 2005.
- [DAS+09] Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F.; "The effectiveness of pair programming: A meta-analysis"; Inf. Softw. Technol. 51; 2009
- [DBR06] De Bortoli, L. A.; Rabello, M. R. "Estrela: modelo de um processo de desenvolvimento para aplicações de comércio eletrônico", Passo Fundo, Universidade de Passo Fundo, 2006.
- [DM06] Damian, D.; Moitra, D. "Global software development: How far have we come", IEEE Computer, v. 23, Issue 5, p.17-19, 2006.
- [DL03] DeMarco, T.; Lister, T. "Waltzing with bears: managing risk on software projects", New York, Dorset House, 2003.
- [ETT12] Ettinger, D.; "A engrenagem do Scrum"; Disponível em: <http://danielettinger.com/2011/04/06/a-engrenagem-do-scrum/>; Acessado em: Julho/2012
- [FC08] Fonseca, I; Campos, A. "Por que Scrum?", Engenharia de Software Magazine, DevMedia, p. 30-35, 2008.
- [FER04] Ferreira, A.; "Mini Aurélio o Dicionário da Língua Portuguesa - Revista e Ampliada"; Brasil; Positivo; 2004.
- [FOR05] Forrester Research "Software and Services in Large Enterprises, Business Technographics", Forrester Research, 2005.
- [FS12] Flores, E.O.; Staa, A.V.; "Uma Análise de Práticas na Aplicação de Scrum em Projetos de Grande Porte"; Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro; Rio de Janeiro; 2012.
- [FSS+11] Fronza, I; Sillitti, A; Succi, G; Vlasenko, J.; "Analysing the usage of tools in pair programming sessions"; XP; 2011
- [GIL10] Gil, A.C.; "Como elaborar projetos de pesquisa"; Atlas; 5ª Edição; São Paulo, 2010.

- [GOD08] Godinho, M. C. P. “Análise de processos e ferramentas para reengenharia de software”, Passo Fundo, Universidade de Passo Fundo, 2008.
- [HA05] Hulkko,H; Abrahamsson , P.; ”A multiple case study on the impact of pair programming on product quality”; ICSE; New York, NY; p.495-504; 2005.
- [HAO11] Hao, J.; “Distributed Pair Programming in Global Software Development”; Dissertação de Mestrado; Universidade de Endiburgo; 2011.
- [HH07] Han, W.M.; Huang, S.J.; “An Empirical Analysis of Risk Components and Performance on Software Projects”; The Journal of Systems and Software; vol 80; number 1; p.42-50; 2007.
- [HSW+08] Huzita, E. H. M.; Silva, C. A.; Wiese, I. S., Tait, T. F. C.; Quinaia, M.; Schiavoni, F. L. “Um Conjunto de Soluções para Apoiar o Desenvolvimento Distribuído de Software”, II Workshop de Desenvolvimento Distribuído de Software, p. 101-110, 2008.
- [KBB+07] Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S.; “A Systematic Literature Review of Evidence-based Software Engineering”; EBSE; Technical Report; 2007.
- [KCD+10] Khoja, S.A.; Chowdhary, B.S.; Dhirani, L.L.; Kalhor, Q.; “Quality control and risk mitigation: A comparison of project management methodologies in practice”; International Conference on Education and Management Technology; p.19-23; 2010.
- [KNI07] Kniberg, H.; “Scrum e XP direto das trincheiras: Como fazemos Scrum”; C4Media; 2007.
- [KNO07] Knob, F. F. “RiskFree4PPM: uma proposta de processo para o gerenciamento de portfólios de projetos distribuídos”, Porto Alegre, PUCRS, Faculdade de Informática, 2007.
- [KRU03] Kruchten, P. “Introdução ao RUP – Rational Unified Process”. Editora Ciência Moderna, 2003.
- [KSP+06] Knob, F. F.; Silveira, F.; Prikladnicki, R.; Orth, A. “RiskFree – Uma Ferramenta de Gerenciamento de Riscos Baseada no PMBOK e Aderente ao CMMI”. V Simpósio Brasileiro Qualidade Software, Vila Velha, v. 1, p.203-217, 2006.
- [LC06] Lui, K; Chan, K.; “Pair programming productivity: Novice-novice vs. expert-expert”; International Journal of Human Computer Studies; p.915–925, 2006.

- [LDO03] Lanubile, F.; Damian, D.; Oppenheimer, H. L. “Global Software Development: technical, organizational, and social challenges”, SIGSOFT Software Engineering, v. 28, Issue 6, p.2-2, 2003.
- [MCD02] Mcdowell, C.; Werner, L. ;Bullock,H.; Fernald, J.; “The effects of pair-programming on performance in an introductory programming course”; SIGCSE '02: Proceedings of the 33rd SIGCSE technical symposium on Computer science education, p.38–42, New York, NY, USA, 2002.
- [MIG08] Miguel, P.A.C.; “Implementação da gestão de portfolio de novos produtos: um estudo de caso”; Produção, São Paulo; v.18;n.2;ABEP;2008.
- [MOO10] Moore, S.; “Strategic Project Portfolio management – Enabling a Productive Organization”; John Wiley & Sons; New Jersey; 2010.
- [MP04] Muller, M.M.; Padberg, F.; "An empirical study about the feelgood factor in pair programming"; International Symposium on Software Metrics; p.151-158; 2004.
- [NK07] Nyfjord, J.; Kajko-Mattsson, M.; “Commonalities in Risk Management and Agile Process Models”; Software Engineering Advances; ICSEA; p18; 2007.
- [NK08] Nyfjord, J.; Kajko-Mattsson, M.; “Outlining a Model Integrating Risk Management and Agile Software Development”; Software Engineering and Advanced Applications; p.476-483; 2008.
- [NSC+12] Noor, T.B.; Shakur, H.B.; Chowdhury, K.B.; Siddique, I.M.; “A Novel Approach to Implement Burndown Chart in Scrum Methodology”; Dept. of Computer Science & Engineering; Stamford University Bangladesh; IJARCSSE; v.2; Issue 10; p.421-427, 2012.
- [OLI06] Oliveira, G. C. “No-Risk: um processo para aplicação de gerência de risco de projetos de software focados em sistemas de informação”. Dissertação de Mestrado, Faculdade de Informática, PUCRS, Porto Alegre, 2006.
- [PEA+08] Prikladnicki, R.; Evaristo, R.; Audy, J.; Yamaguti, M.; “Minimizing the Challenges of Risk Management in Distributed IT Projects”; IGI Global; p.126-142; 2008.
- [PFL09] Pfleeger, S. L. “Software engineering: theory and practice”, Prentice-Hall, 2009.
- [PMB08] Project Management Institute “A guide to the project management body of knowledge: PMBOK Guide. 4rd edition”, Newton Square, Project Management Institute, 2008.

- [PMI11] Project Management Institute “PMI – Chapter Pernambuco”. Disponível em: <http://www.pmipe.org.br/web/br/pmi.php>. Acessado em: Dezembro 2011.
- [PRE10] Pressman, R. “Software Engineering: A Practioner’s Approach”, International, Macgraw-hill, 2010.
- [PZB10] Zenzen, G. L.; Baptista, J. “Uma Proposta Empírica para Utilização de Processos Explícitos de Gerenciamento de Riscos no Scrum”, TCC, , Faculdade de Informática, Porto Alegre, PUCRS, 2010.
- [REZ05] Rezende, D. “Engenharia de Software e Sistemas de Informação – 3ªed.”, Rio de Janeiro, Brasport, 2005.
- [RMM05] Rabechini, R.; Maximiano, A.C.A.; Martins, V.A.; “A adoção de um gerenciamento de portfolio como uma alternativa gerencial: o caso de uma empresa prestadora de serviço de interconexão eletrônica”; ABEP; Produção; v.15;n.13; São Paulo; 2005.
- [ROY70] Royce, W. W.; "Managing the Development of Large Software Systems"; Proceedings of IEEE WESCON, v.26, p.1-9, Agosto 1970.
- [SB02] Schwaber, K.; Beedle, M.; “Agile Software Development with Scrum”; New Jersey; Prentice Hall; 2002.
- [SCH04] Schwaber, K.; “Agile Project Management with Scrum”; Microsoft Press; 2004.
- [SCH05] Schuh, P.; “Integrating Agile Development in the Real World”; Charles River Media; 2005.
- [SCH07] Schwalbe, K. “Information Technology Project Management , Fifth Edition”. Thomson, Course Technology, 2007.
- [SCH07b] Schwaber, K. “The Enterprise and Scrum”; Microsoft Press; 2007.
- [SEI11] Software Engineering Institute; “Capability Maturity Model Integration (CMMI) Version 1.3”; Software Engineering Institute; Carnegie Mellon University; Em: <http://www.sei.cmu.edu/>; Acessado em: Outubro 2011.
- [SIL06] Silveira, A.; “Gestão de Portfolio – administrando sua carteira de projetos”; Expleo; São Paulo; 2006.
- [SIL09] Silveira, V. L. “Métodos Ágeis Aplicados em um Laboratório de Usabilidade.”, Porto Alegre, X Salão de Iniciação Científica, PUCRS, 2009.

- [SLK+01] Schmidt, R.; Lyytinen, K.; Keil, M.; Cule, P.; "Identifying Software Project Risks: An International Delphi Study"; Journal of Management Information System; vol 17; number 4; p.5-36; 2001.
- [SMG11] Salleh, N.; Mendes, E.; Grundy, J.; "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review"; IEEE Transactions on Software Engineering; p.509–525; 2011.
- [SOM07] Sommerville, I. "Engenharia de Software - 8ªed.", São Paulo, Pearson Education, 2007.
- [UNG08] Ungar, J.; "The Design Studio: Interface Design for Agile Teams"; Agile 2008 Conference; Jewelry Television; IEEE; 2008.
- [VAR05] Vargas, R. V. "Gerenciamento de projetos: estabelecendo diferenciais competitivos - 6ª Edição", Rio de Janeiro, Brasport, 2005.
- [VK07] Vanhanen, J; Korpi, H.; "Experiences of Using Pair Programming in an Agile Project"; HICSS; p.274; 2007
- [VL07] Vanhanen, J.; Lassenius, C.; "Perceived Effects of Pair Programming in an Industrial Context"; EUROMICRO Conference on Software Engineering and Advanced Applications; p.211-218; 2007
- [VNL07] Vanhanen, J.; Lassenius, C.; "Effects of pair programming at the development team level: an experiment"; Empirical Software Engineering; 2005 International Symposium; p.17-18; 2005.
- [WG10] West, D; Grant, T. "Agile Development: Mainstream Adoption Has Changed Agility", Forrester Research, 2010.
- [WKA04] Wallace, L.; Keil, M.; Arun, R.; "How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model"; Decision Sciences; vol 35; number 2; p.289-321; 2004.
- [WMR06] Wieringa, R.; Maiden, N.; Rolland, C.; "Requirements. Engineering Paper Classification and Evaluation Criteria: a Proposal and a Discussion"; Journal of Requirement Engineering; p.102-107; 2006.

ANEXO I – FICHA DE ENTREVISTA

Ficha de Entrevista - Riscos Comuns em Projetos com SCRUM

O objetivo é verificar como os 33 riscos comuns, identificados no gerenciamento de projetos de desenvolvimento de software, estão sendo tratados em projetos que utilizam o Método Ágil SCRUM. Também se busca localizar possíveis lacunas que indiquem ajustes nos processos do SCRUM, a fim de possibilitar a identificação, análise, planejamento a respostas, monitoramento e controle dos riscos comuns existentes em projetos de desenvolvimento de software. Os dados desta pesquisa são confidenciais e os resultados serão sumarizados.

Dados Gerais

Data:

Hora inicial:

Hora final:

Questões	
Dados Demográficos	Nome: _____ Nascimento: _____
	Curso (nível mais alto): _____
	Instituição: _____ Concluído em: _____
	Tempo de experiência profissional na área de Informática: ____ anos.
	Tempo de experiência profissional com Métodos Ágeis: ____ anos.
	Tempo de experiência profissional com SCRUM: ____ anos.
	Departamento/área: _____
	Vínculo empregatício: _____ Tempo de empresa: ____ anos.
	Papel/Função atual: _____ Quantos funcionários a empresa possui?
	Como Métodos Ágeis é utilizado na sua empresa?
	Como o SCRUM é utilizado na sua empresa?
	Como o SCRUM é utilizado nos seus projetos?

#	Dim.	Risco	Perguntas			Resposta
			1	2	3	
1	ORG	Reestruturação da organização Mudança da gerência ou alteração da estrutura organizacional.				
2	ORG	Ambiente organizacional instável Cargos vagos sem alocação de pessoas, falta de gerentes ou coordenadores, dificultando ou retardando a tomada de decisão.				
3	ORG	Mudança de prioridade da gerência Novos objetivos definidos pela gerência podem alterar as prioridades dos projetos.				
4	ORG	Falta de comprometimento da alta direção com o projeto Supervisão ineficiente ou inexistente dos executivos da organização.				
5	ORG	Política corporativa com efeito negativo sobre o projeto Políticas da empresa podem ser definidas em desalinhamento com a realidade diária da condução de projetos. Metodologia Ágil, mas empresa não é.				

#	Dim.	Risco	Perguntas			Resposta Dissertativa
			1	2	3	
6	P&C	Comunicação ineficaz Envolvimento insuficiente da equipe do projeto, falha ao indicar mudanças, falta de relatórios de status do projeto.				
7	P&C	Tecnologia ineficaz para gestão de projetos A escolha de uma ferramenta eficaz para auxílio no gerenciamento de projetos pode influenciar a condução do mesmo.				
8	P&C	Falta de metodologia eficaz no gerenciamento do projeto A equipe não emprega controle de mudanças, sem planejamento ou outras habilidades necessárias ao processo.				
9	P&C	Etapas do projeto não definidas claramente Planejamento superficial ou projeto não está bem estruturado (falta de WBS).				
10	P&C	Gold plating Entregar mais do que foi definido no escopo do projeto.				
11	P&C	Deficiências na execução de tarefas e componentes fornecidos externamente Ao alocar uma atividade ou componente do projeto para um recurso externo e esta atividade ou componente apresenta falha na execução.				
12	P&C	Estimativas irrealistas de tempo e custo Ocorre quando as estimativas do projeto são muito cautelosas ou ousadas demais, gerando impacto na qualidade do projeto em virtude da redução do tempo de teste ou treinamento.				
13	P&C	Alteração de escopo ou objetivos Mudanças nos negócios ou reorganização durante o projeto.				
14	P&C	Escopo ou objetivos não claros ou incompreendidos É impossível definir o escopo ou objetivo real do projeto devido a diferenças de opinião entre os usuários.				
15	P&C	Falta de planejamento ou planejamento inadequado A atividade de planejamento não é considerada importante ou não é realizada.				

#	Dim.	Risco	Perguntas			Resposta Dissertativa
			1	2	3	
16	P&C	Gestão de mudanças inadequada Cada projeto necessita de um processo para gerir mudança para que o custo e escopo sejam controlados. O aumento do escopo sem controle é a evidência de uma gestão da mudança ineficaz, onde os parâmetros de sucesso não estão definidos.				
17	P&C	Andamento do projeto não monitorado adequadamente Ocorre quando existe falha na definição de pontos de controle do projeto.				
18	P&C	Estimativa inadequada dos recursos necessários Falta na identificação de recursos para o projeto compromete o andamento.				
19	COM	Alto nível de complexidade técnica Ocorre quando o produto é composto por módulos de diferentes tecnologias e de difícil entendimento e integração.				
20	COM	Falta de desempenho em tempo real Simulação, medições, modelagem, prototipagem, ajustes ou instrumentação apresentam falta de desempenho devido a ambientes limitados.				
21	COM	Tecnologia inexistente ou imaturas O produto do projeto exige a utilização de tecnologia ainda não disponível ou muito recentes e imaturas.				
22	REQ	Desenvolver as funções erradas do software Mapeamento incompleto dos componentes ou da arquitetura do software.				
23	REQ	Desenvolver a interface de usuário errada O usuário está habituado a utilizar outro padrão de interface na organização.				
24	REQ	Alterações tardias nos requisitos Solicitação de mudanças durante a fase de testes do produto, por exemplo.				
25	REQ	Falta de requisitos congelados Devido às constantes necessidades de mudanças dos usuários, o software corre o risco de jamais ser concluído.				

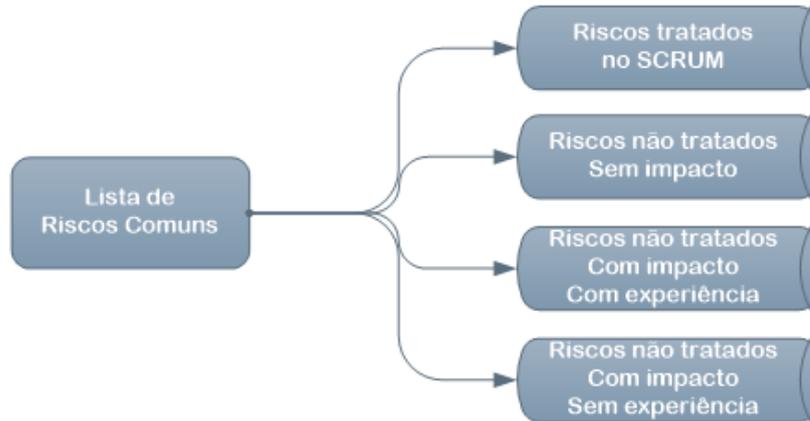
#	Dim.	Risco	Perguntas			Resposta Dissertativa
			1	2	3	
26	REQ	Incompreensão dos requisitos Não definir todos os requisitos do sistema antes de começar os trabalhos, gerando requisitos de forma superficial causando falha na estimativa de esforço, habilidades e tecnologia necessária para o projeto.				
27	TEA	Membros da equipe de desenvolvimento inadequadamente treinados Falta de conhecimento na tecnologia utilizada, falta de experiência ou treinamento ineficaz.				
28	TEA	Membros da equipe com falta de habilidade especializada exigida pelo projeto Por exemplo, tecnologia, conhecimento de negócios e experiência inexistentes entre os recursos da equipe.				
29	TEA	Falta de habilidade na liderança de projetos A equipe do projeto é formada e o gerente não tem autoridade ou habilidades para obter sucesso.				
30	USE	Conflitos entre usuários Diferentes objetivos dos departamentos dos usuários geram expectativas distintas.				
31	USE	Falha em obter comprometimento do usuário Usuários apontam o gerente do projeto como culpado pela falta de envolvimento do cliente.				
32	USE	Usuários com resistência a mudanças Obstáculo percebido quando o usuário não quer mudar de rotina ou não está motivado na empresa.				
33	USE	Falha ao gerir expectativas do usuário final Expectativas determinam o sucesso ou o fracasso de um projeto. Expectativas incompatíveis com a entrega (muito alta ou muito baixa) causam problemas. As expectativas devem ser corretamente identificadas e constantemente ajustadas a fim de evitar o fracasso.				

ANEXO II – ROTEIRO DE ENTREVISTA

Roteiro de Entrevista

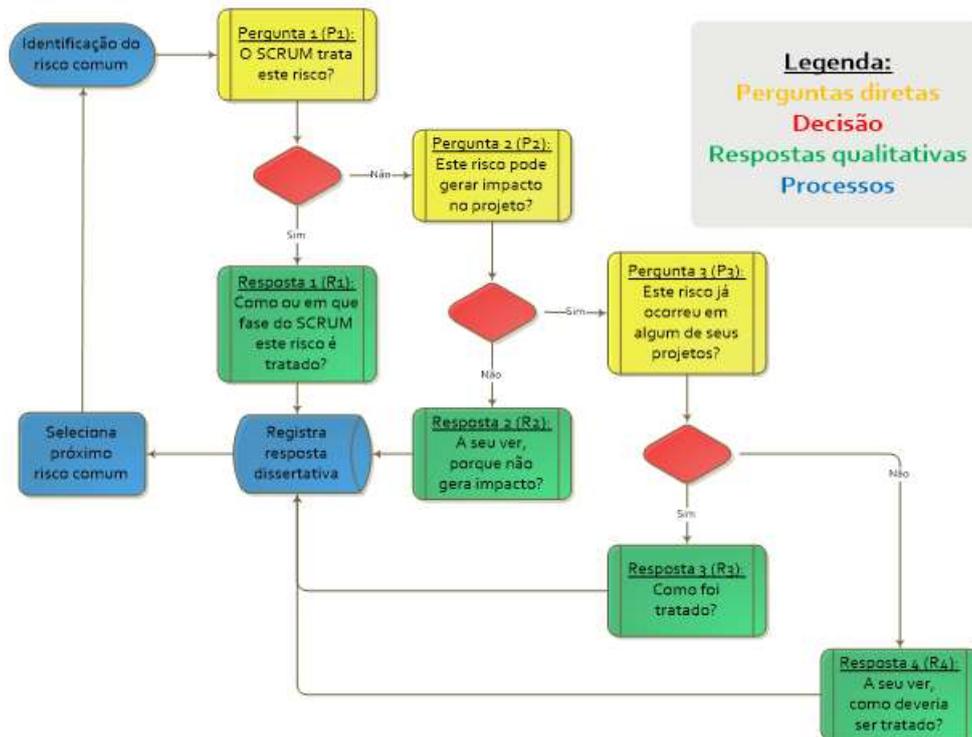
Riscos Comuns em Projetos com SCRUM

Objetivo: separar os riscos comuns em 4 subconjuntos



- **Riscos tratados no SCRUM:** Quando a Pergunta 1 for "Sim".
- **Riscos não tratados, Sem impacto:** Quando a Pergunta 2 for "Não".
- **Riscos não tratados, Com impacto, Com experiência:** Quando a Pergunta 3 for "Sim".
- **Riscos não tratados, Com impacto, Sem experiência:** Quando a Pergunta 3 for "Não".

Dimensões de Riscos	Sigla	Quantidade
Ambiente Organizacional	ORG	5
Planejamento e Controle	P&C	13
Complexidade	COM	3
Requisitos	REQ	5
Equipe	TEA	3
Usuário	USE	4



ANEXO III – PLANILHA DE RESPOSTAS DIRETAS DAS ENTREVISTAS

Risco	DIM	Descrição	Entrevista 1			Entrevista 2			Entrevista 3			Entrevista 4			Entrevista 5			Entrevista 6			Entrevista 7			Entrevista 8			Entrevista 9			Total	RT	RT	RNTSI	RNTSE	RNTTE	
			P1	P2	P3																															
1	ORG	Reestruturação da organização	N	S	S	S			S			S			N	S	S	N	S	S	S			S			S			9	56%	5	0	1	1	3
2	ORG	Ambiente organizacional instável	N	S	N	S			S			N	N		N	S	S	N	S	S	S			S			S		9	44%	4	1	2	2	0	
3	ORG	Mudança de prioridade da gestão	S		S				S			S			S			S									S		9	100%	9	0	0	0	0	
4	ORG	Falta de comprometimento da alta direção com o projeto	N	S	S	S			N	N		N	N		N	S	S	N	S	N	S	S			N	S	N	9	22%	2	2	2	3	0		
5	ORG	Política corporativa com efeito negativo sobre o projeto	N	S	S	N			N	S	S				N	S	N	N	S	N	S	S			N	S	S	9	33%	3	0	2	2	4		
6	P&C	Comunicação ineficaz	S		S				S			S			S			S								N	S	S	9	89%	8	0	0	1	1	
7	P&C	Tecnologia ineficaz para gestão de projetos	N	S	N	S			S			S			S			S								N	S	S	9	67%	6	0	1	1	2	
8	P&C	Falta de metodologia eficaz no gerenciamento do projeto	S		S				S			S			S			S								S		S	9	100%	9	0	0	0	0	
9	P&C	Tempos do projeto não definidos claramente	S		S				S			S			S			S								S		S	9	89%	8	0	0	1	1	
10	P&C	Gold plating	S		S				S			S			N	S	S	S								S		S	9	89%	8	0	0	1	1	
11	P&C	Deficiências na execução de tarefas e componentes fornecidos externamente	S		N	S			S			S			N	S	S	N	S	N	S	S			N	S	S	9	44%	4	0	1	4	4		
12	P&C	Estimativas irreais de tempo e custo	S		S				S			S			S			S								S		S	9	100%	9	0	0	0	0	
13	P&C	Aterroço de escopo ou objetivos	S		S				S			S			S			S								S		S	9	100%	9	0	0	0	0	
14	P&C	Escopo ou objetivos não claros ou incompreendidos	S		S				S			S			N	S	S	S								S		S	9	89%	8	0	0	1	1	
15	P&C	Falta de planejamento ou planejamento inadequado	S		S				S			S			N	S	S	N	S	S	S					S		S	9	78%	7	0	0	2	2	
16	P&C	Gestão de mudanças inadequada	S		S				S			S			N	S	N	S								S		S	9	89%	8	0	1	0	1	
17	P&C	Acompanhamento do projeto não monitorado adequadamente	S		S				S			S			S			S								S		S	9	89%	8	0	0	1	1	
18	P&C	Estimativa inadequada dos recursos necessários	N	S	S	S			S			N	S	S	N	S	S	S							S		S	9	56%	5	0	0	4	4		
19	COM	Alto nível de complexidade técnica	S		S				S			S			N	S	S	S								S		S	9	78%	7	0	0	2	2	
20	COM	Falta de desempenho em tempo real	S		N	S			N	S	S			N	S	S	N	S	S	N	S	S			N	S	S	9	44%	4	0	0	5	5		
21	COM	Tecnologia inexistente ou imatura	S		N	S			N	S	S			S			N	S	S	S						S		S	9	56%	5	0	0	4	4	
22	REQ	Desenvolver as funções erradas do software	S		S				N	S	S			S			N	S	S	S						S		S	9	78%	7	0	0	2	2	
23	REQ	Desenvolver a interface de usuário errado	S		N	S			S			S			N	S	N	S								S		S	9	78%	7	0	1	1	1	
24	REQ	Aterroços todos nos requisitos	S		S				S			S			N	S	S	S								S		S	9	89%	8	0	0	1	1	
25	REQ	Falta de requisitos congelados	S		S				S			S			N	S	S	S								S		S	9	89%	8	0	1	0	0	
26	REQ	Incompreensão dos requisitos	S		S				S			S			S			S								S		S	9	100%	9	0	0	0	0	
27	TEA	Membros da equipe de desenvolvimento inadequadamente treinados	S		S				S			N	S	S	N	S	S	S								N	S	N	9	67%	6	0	1	2	2	
28	TEA	Membros da equipe com falta de habilidade especializada exigida pelo projeto	S		S				S			N	S	S	N	S	S	S								N	S	N	9	67%	6	0	0	3	3	
29	TEA	Falta de habilidade na liderança de projetos	N	S	N	S			S			S			N	S	N	S								S		S	9	78%	7	0	2	0	0	
30	USE	Conflitos entre usuários	S		S				S			S			N	S	S	S								S		S	9	78%	7	0	0	2	2	
31	USE	Falta em obter comprometimento do usuário	S		S				S			S			N	S	S	S								S		S	9	100%	9	0	0	0	0	
32	USE	Usuários com resistência a mudanças	S		S				S			S			N	S	S	S								S		S	9	67%	6	0	0	3	3	
33	USE	Falta ao gerir expectativas do usuário final	S		S				S			S			S			S								S		S	9	100%	9	0	0	0	0	

Totais:	225	3	15	54
Percentuais:	75,78%	1,01%	5,05%	18,18%

ANEXO IV – RESPOSTAS DISSERTATIVAS DAS ENTREVISTAS

Risco	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5	Resposta 6	Resposta 7	Resposta 8	Resposta 9
1	Tratado como risco normal no projeto. GR Padrão.	Papéis do Scrum. Product Owner. Business Owner. Time.	User stories. Equipe não percebe mudança organizacional.	Time auto-organizado.	Cancelando a iteração. Replanejamento.	Auxílio de PMO e alta direção (escalar).	Sprint.	Processo fácil de conduzir.	É uma questão bastante delicada. A transição de liderança passa por uma influência positiva. A questão é a causa dessa mudança e como a equipe recebe isso. O Scrum não trata diretamente com isso, mas pode ajudar, principalmente pela questão de transparência.
2	Alinhamento diário. Backup do Scrum Master.	Time todo tem conhecimento da feature.	Processo Scrum mitiga o risco. Interativo e Incremental. Time auto organizado.	Não ocorre por estar sendo tratado pelo RH da empresa	Aguardar definição.	Transparência: Identificar decisão com o cliente.	Sprint.	Mitiga o risco pela facilidade de condução, sem burocracia.	O Scrum define alguns papéis e não cargos. Do ponto de vista de execução, o importante é saber quem vai fazer e sinalizar se tiver algum problema.
3	Alterar a prioridade de projeto ou parte do scope. Tratado na sprint.	Mudança bem vista. De acordo com valor para negócio.	Processo Scrum business Owner define prioridades. Entrega de valor.	Método ágil. Mudanças bem vindas.	Mudanças são naturais / normais.	Mudanças são bem vindas.	Sprint.	Adaptação rápida.	
4	Utilizar Scartation, apoio gerencial. Tratar na sprint.	Negócio envolvido com time. Desvio máximo de duas semanas.	Existe apoio estratégico (crença) da alta direção.	Não há necessidade de supervisão da alta direção.	Revisão do Backlog da Sprint. Replanejamento.	Transparência com alta direção.	Comunicação clara. Solicitação de envolvimento. Alta gestão. Manifesto ágil.	Definir decisões dentro da equipe. Autonomia.	
5	Ajuste da metodologia dos projetos.	Tratado no time. Atuar no dia-a-dia. Repassar para diretoria.	Negociação de prioridades com equipes cross escalando eventuais prioridades.	Alinhamento, envolvimento com stakeholders.	Organização é ágil.	Transparência com alta direção.	Replicar conceitos de Scrum. Em 2º momento, manifesto ágil.	Escalar e resolver problemas. Retrospectiva com experiência.	
6	Tatada diariamente. Retrospectivas.	Product Owner estão envolvidos com o time todo o tempo.	Daily Meeting. Comunicação.	Daily meetings. Comunicação aberta.	Comunicação forte no Scrum.	Burndown. Pode melhorar a documentação.	Daily meeting. Time boxes.	Implantar Burndown.	
7	Busca de ferramentas de apoio de fácil localização	Ferramenta Simples. Kambam.	Kambam físico.	Ferramentas simples. Kambam.	Kambam adaptado.	Kambam. Burndown.	Ajuste de ferramentas de gestão tradicional para SCRUM.	Ferramentas simples podem apoiar o processo.	Esta questão ela se torna relevante principalmente em projetos com equipes distribuídas. Ao meu ver, as ferramentas (e o uso de uma ferramenta eficaz) ganham força quando não podemos estar próximos. Vivenciéi isso muitas vezes, desde aspectos relacionados a Gestão (mostrar e acompanhar informações sobre projeto) quanto aspectos técnicos (fazer pair programming remotamente, por exemplo).

Risco	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5	Resposta 6	Resposta 7	Resposta 8	Resposta 9
8	Em todo ciclo do projeto.	Metodologia Ágil Processo bem definido. Eventos claros.	Sprint com Daily Comunicação.	Users stories são sempre registrados.	Cerimonias são apoio.	Flexibilidade. Mudanças são bem vindas.	Scrum é um framework.	Backlog, Processo bem definido.	
9	Conhecimento do Product Owner sobre o Backlog.	Backlog bem definido. Itens bem definidos.	Story Board, Backlog bem definido. Sprint iterativo e incremental.	Iteratividade.	Backlog, Releases, Iterações.	Fases do Scrum são bem definidas.	Mistura de métodos, Frameworks, para Scrum (SCRUM-AND).	Backlog, processo bem definido. Colaboração do time.	
10	Maturidade da equipe sobre o scrum, Product Owner Backlog.	User story - critérios de aceitação. Product Owner defini prioridades e valor para negócio.	Comunicação, Daily Meeting, Product Owner na equipe.	Permite mudanças alinhadas com o negócio.	Valor do produto. Backlog bem definido.	Podemos adicionar itens na sprint se necessário.	Planejamento detalhado com scrum.	Sprints rápidas de 15 dias.	
11	Em todo ciclo. Visão Diária, Daily Meeting.	Acompanhamento com follow-up.	Exigência de uso de scrum nos fornecedores externos.	Retrospectiva.	Clareza dependências entre iterações.	Aumentar comunicação com fornecedor.	Troca de fornecedor.	Controle de comunicação com os fornecedores. Aplicação de Scrum.	Nestes casos, é importante definir um plano para ver quando e como estas tarefas (técnicas ou não) entrarão no projeto. Tive situações de componentes de software que foram desenvolvidos pela nossa empresa tinham que ser acoplados em um produto. É fundamental entender o projeto como um todo e definir contratos (não no sentido jurídico, mas no sentido técnico) de como e quando eles precisam estar concluídos e como eles podem ser testados/verificados em sua totalidade. Um outro exemplo, relacionado a gestão, um cliente havia realizado a contratação de um servidor para hospedar uma aplicação que estávamos desenvolvendo. Este processo atrasou e, para não prejudicarmos os testes, criamos uma "solução de contorno" em usar uma das máquinas que estava disponível para realizarmos os testes e não perdermos tempo.
12	Ciclos de Sprint Retrospectiva, Planning Poker, Story Points.	Projeto é quebrado em etapas menores. Sprints de dias semanais.	Melhoria contínua. Retrospectiva, Planning, Comunicação.	Planning de estimativa. Vai sendo mais assertiva.	Priorização de itens do backlog.	Experiência melhora a precisão das estimativas.	Time boxes (planning, review).	Sprints rápidas. Time boxes.	
13	Entre ciclos e dentro das Sprints.	Mudanças bem vindas. Agragar valor ao negócio, flexibilidade.	Product Owner com equipe. Processo normal Daily meeting.	Metodologia ágil trata mudanças como bem vindas.	Mudanças bem vindas.	Mudanças são bem vindas.	Scrum baseado em alterações.	Mudanças sempre apoiadas.	

RISCO	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5	Resposta 6	Resposta 7	Resposta 8	Resposta 9
14	Conversas diárias com o Product Owner e time.	No Planning, Product Owner apresenta os itens. Pré-Planning verifica se a equipe pode desenvolver.	Scrum exige definição dos users stories.	Users stories. Time tira dúvidas em qualquer momento. Daily-integração. Comunicação.	Backlog.	Backlog definido com antecedência para a próxima sprint.	Detalhamento dos itens do backlog. DOR, DOD.	Prototipação com usuários, comunicação.	
15	Sprint Planning. Planejamento incremental. Exige maturidade da equipe.	Processo com planejamento bem definido. Itens são importantes.	Planning no primeiro dia.	Planning da sprint.	Planning.	Revisão das atividades de planejamento.	Em projetos maiores, com mescla de métodos (Scrum+PM).	Sprint Planning.	
16	Em todo ciclo do projeto Backlog. Critérios de aceitação. User stories.	Mudanças bem vindas. Escopo depende do driver da Sprint. MVP: Minimum Viable Product.	Comunicação Product owner define prioridades.	Daily. Replanejamento na daily.	Ciclos do Scrum.	Checkpoint para registro de mudanças.	Mudanças contínuas e bem vindas.	Mudanças são bem vindas. Fácil de tratar.	
17	Visão imediata. Burndown. Entregas Backlog.	Kanban acompanhado diariamente.	Comunicação Daily.	Sprints em releases. Releases são milestones.	Checkpoints, Tracker.	Burndown, Tool board. Precisa de novos controles para monitorar.	Time boxes.	Reunir com Scrum Masters. Atas de reuniões. Equipes devem ser autogerenciáveis.	
18	Deve ser definido antes do início do projeto.	Pré-Planning identifica se o time pode assumir a sprint.	Planning Users stories daily.	Replanejamento. Comunicação quando o risco for identificado.	Sprint review. Retrospectiva.	Gestão de Riscos implantada (Risk Backlog).	Planning.	Ciclos curtos sprints.	O fato é precisamos aprender com o que está acontecendo. É mais que isso, usar este aprendizado. O cliente acompanhando o projeto é fundamental, para ele estar informado do que está acontecendo.
19	Em todo ciclo. Daily Meeting. Retrospectiva. Sprints Planning.	Pre-Planning pode minimizar o risco.	Daily: Identifica rapidamente. Interativo e incremental.	Permite "quebrar" funcionalidades complexas em vários sprints.	Backlog identifica itens complexos.	Integração com outras equipes.	Time boxes. Comunicação.	Prototipação. Iteratividade. Ciclos incrementáveis.	Quanto mais complexo o produto, mais devemos gerar entregas mais rápidas. Isso vai gerar o aprendizado necessário para evolução. Isso não é uma regra, mas uma disciplina que adotei com grande sucesso. Sempre que tinha algo complexo ou até mesmo nebuloso, eu buscava entregar mais rápido com menos funcionalidade, afinal, precisamos saber onde estamos pisando.
20	Dentro das Sprints, através das entregas parciais.	Comunicação entre as equipes de sistema dando auxílio na atualização.	Comunicação Daily. Experiência. Melhoria contínua Planning.	Requisito para projetos, ambientes espelhados.	Redirecionamento do ambiente para validação de funcionalidades.	Definição com cliente para protótipo de sistema.	Kanban+Linguagem visual. Scrum. AND.	Replicar conhecimento adquirido em novos ciclos.	

RISCO	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5	Resposta 6	Resposta 7	Resposta 8	Resposta 9
21	Dentro das Sprints, através das entregas parciais.	Área de P&D (Pesquisa e Desenvolvimento). Treinamento de pessoal. Aquisição de servidores com antecedência para implantação de nova tecnologia.	P&D, Planning, P.O., validas as User Stories, Prova de Conceito.	Iteratividade.	Pré-game.	Troca de conhecimento com outras equipes.	Sprints.	Prototipação, Iteratividade, Ciclos incrementais.	Embora não trate diretamente com isso, pois o foco do Scrum é gestão e acompanhamento de projetos/produção, isso faz parte de qualquer projeto, principalmente de quem trabalha com tecnologia. Nestes casos, é importante incluir no planejamento alguns spikes de investigação de tecnologias ou até mesmo testes a serem realizados. Uma vez, precisei desenvolver um componente que "fazesse" um arquivo de áudio MP3. Foi importante incluir isso no PG e conscientizar o cliente do desenvolvimento deste componente, que na época não existia similar no mercado. Tivemos que usar a norma de especificação do arquivo para criar um analisador/parser que fizesse a divisão. Tem outros exemplos, mas acho que este dá uma ideia clara de que é algo natural para quem trabalha com tecnologia.
22	Daily Meetings, Product Owner deve participar. Retrospectivas: Ajudar no próximo Sprint.	Sprints são curtas. Product Owner disponível para esclarecer dúvidas.	Planning, Daily, Comunicação.	Etapas de teste e homologação.	Pos-game, sprint review.	BOD (desenvolvimento de testes) e refatoração constante.	Sprints.	Prototipação, Iteratividade, Ciclos incrementais.	
23	Daily Meetings, Product Owner deve participar. Retrospectivas: Ajudar no próximo Sprint.	Processo bem definido. Validação com usuário antes do desenvolvimento.	Sprint Kanban Fluxo contínuo de entregas. Product Owner envolvido diariamente.	Teste de usabilidade focus Group.	Padronização de códigos.	Planning sprint.	DDO, equipe integrada, Product Owner + Team.	Prototipação, Iteratividade, Ciclos incrementais.	
24	Sprint Planning, Product Owner deve ser flexível com o Product Backlog. Priorização do Product Backlog.	Critérios de aceitação de produto. Product Owner orienta com antecedência. Sprints curtos.	Product Owner define prioridades, Mudanças bem vindas.	Mudanças bem vindas.	Equipe autogerenciável.	Análise de impacto da mudança. Acompanhamento do cliente no projeto.	Sprints.	Sprints curtos.	
25	Sprint Planning, Product Owner deve ser flexível com o Product Backlog. Depende do Product Owner.	MVP-Mínimo seria feito. Acréscimos são feitos caso possível ou em outra sprint.	Método iterativo e incremental.	MVP-Mínimo Produto Visível.	Backlog, Planning.	Análise de impacto da mudança. Acompanhamento do cliente no projeto.	Sprints.	Fabricação dos itens, Experiências, Dependência de itens.	

RISCO	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5	Resposta 6	Resposta 7	Resposta 8	Resposta 9
26	Nas Sprints Planning Requisitos podem ser detalhados com o time.	Planning exige que os Users estarem claros. User story com critério de aceitação.	Planning, sprints de duas semanas, User stories.	Comunicação, User stories, Testes, Planning.	Backlog: (DOR), Definition of Ready, (DOD) Definition of Done.	Dentro da sprint.	Backlog, Priorização pelo Product Owner.	Visão macro, porém com objetivo curto.	
27	Através de colocation.	Métodos ágeis: time com multidisciplinaridade, comunicação Daily, pair programming.	Pair programming, comunicação Daily.	Treinamento com time interno, facilita o aprendizado.	Pré-game.	Integração com outras equipes, Treinamento.	Sprints, Daily Meetings, Time boxes.	Time multidisciplinar, Comunicação.	
28	Colocation. Todo processo.	Pair Programming coding dojo (ambiente de treinamento).	Product Owner na equipe. Conhecimento do negócio no time. Aproximar time do negócio.	Treinamento com time interno, facilita o aprendizado.	Pair-programming.	Pré-game no início do projeto.	Toda sprint precisa ter uma meta, baseada no valor agregado.	Comunicação, Techforum: reunião com o negócio e tecnologia.	
29	Nível de exigência é elevado no RH. Risco não ocorre ou pouco provável.	Product Owner prioridades, Time conduz o projeto, Scrum Master desempenho.	Equipes auto organizadas, colaborativas.	Planning, Mudanças bem vindas.	Pré game para game=>Project charter, Comunicação.	Reunião de kick-off, Apresentação do time.	Divide responsabilidade, Equipe autogerenciável.	Colaboração entre times, Entender os papéis.	
30	Product Owner, Scrum Master, Sprint Planning, Product Backlog.	Pré-Planning, Feature descuida com time, Sprint-time solicita redefinição do item, Backlog.	Product Owner administra conflitos dos usuários, Define prioridades, Valor agregado.	Time auto-organizável.	Divisão de tarefas entre os setores da empresa, Definir intenções e solicitar apoio.	Comunicação com cliente.	Transparência, Scrum Master.	Plataos, Comunicação com cliente, Incremental.	
31	Planejamento iterativo.	Retrospectiva: itens com problema são tratados com clientes.	Product Owner é o usuário e participa do time.	Daily, Time autoorganizável, Time decide.	Product Owner, Backlog=>DOR.	Envolvimento de toda equipe.	Product Owner participa do Projeto e da equipe.	Comunicação e apresentação do produto do projeto.	
32	Product Owner, Scrum Master, Entregas parciais, Discussões sobre o Product Backlog.	Scrum aproxima cliente, time, Scrum Master, Product Owner.	Comunicação Product Owner envolvido.	Daily, Time autoorganizável, Time decide.	Iterações.	Trabalho com Product Owner para demonstrar vantagens do projeto.	Manifesto Agil.	Prototipagem, piloto.	É importante apresentar como funciona o processo e a importância dos usuários em participar disso. Nós fazíamos isso sempre no início dos projetos e inclusive enfatizávamos que sem isso, ficaria muito difícil chegarmos na solução desejada.
33	Product Owner, Scrum Master, Entregas parciais, Discussões sobre o Product Backlog.	Alinhamento constante, Sprints curtos.	Interativo e incremental, Product Owner envolvido.	Iterações, Homologação.	Iterações.	Sprints de curta duração melhoram a expectativa do cliente.	Sprints curtos trabalham a expectativa do usuário final.	Ciclos curtos, Sprints.	