

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

ALEXANDRE LAZARETTI ZANATTA

**RECOMENDAÇÕES PARA TRABALHADORES NA MULTIDÃO SUPERAREM
BARREIRAS EM PROJETOS DE SOFTWARE CROWDSOURCING**

Porto Alegre

2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

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

**Recomendações para trabalhadores na
multidão superarem barreiras em
projetos de software crowdsourcing**

ALEXANDRE LAZARETTI ZANATTA

Tese apresentada como requisito
parcial à obtenção do grau de Doutor em
Ciência da Computação na Pontifícia
Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

Porto Alegre

2018

Ficha Catalográfica

Z27r Zanatta, Alexandre Lazaretti

Recomendações para trabalhadores na multidão superarem barreiras em projetos de software crowdsourcing / Alexandre Lazaretti Zanatta . – 2018.

167 p.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Prikladnicki.

Co-orientador: Prof. Dr. Igor Steinmacher.

1. Engenharia de Software. 2. Software Crowdsourcing. 3. Barreiras. 4. Recomendações. 5. Trabalhador na Multidão. I. Prikladnicki, Rafael. II. Steinmacher, Igor. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

Alexandre Lazaretti Zanatta

**RECOMENDAÇÕES PARA TRABALHADORES NA MULTIDÃO SUPERAREM BARREIRAS
EM PROJETOS DE SOFTWARE CROWDSOURCING**

Tese apresentada como requisito parcial para obtenção do grau de Doutor em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 13 de agosto de 2018.

BANCA EXAMINADORA:

Prof^a. Dr^a. Soraia Raupp Musse (PUCRS)

Prof^a. Dr^a. Tayana Uchôa Conte (UFAM)

Prof. Dr. Marcelo Soares Pimenta (UFRGS)

Prof. Dr. Igor Fábio Steinmacher (UTFPR - Co-orientador)

Prof. Dr. Rafael Prikladnicki (PUCRS - Orientador)

DEDICATÓRIA

À Deisi, Henrique e Livia.

AGRADECIMENTOS

Participar desta caminhada não é algo individual e trivial. Várias pessoas participaram desse processo e deixaram suas marcas. Marcas que jamais se apagarão. Fui apresentado à muitas pessoas, vários métodos e inúmeras técnicas que me auxiliaram a jogar uma luz em direção à um novo olhar sobre a docência e a pesquisa científica. Saio diferente de como entrei. Assim, tenho o privilégio de agradecer.

Primeiramente agradeço ao orientador, Prof. Dr. Rafael Prikladnicki, pelo direcionamento e importantes discussões ao longo desses anos. Teus apontamentos foram fundamentais na condução da tese, além de me auxiliar a tornar-me um pesquisador. Agradeço ao co-orientador prof. Dr. Igor Fabio Steinmacher o qual considero uma fonte de inspiração e imenso respeito.

Agradeço aos participantes da banca examinadora Prof^a. Dr^a. Soraia Raupp Musse, Prof^a. Dr^a. Tayana Uchôa Conte e Prof. Dr. Marcelo Soares Pimenta pelos comentários e sugestões.

Reconheço a importância da Dell Computers pela concessão da bolsa taxa. Agradeço aos professores da PUCRS, com um carinho especial à prof^a. Dr^a. Sabrina Marczak pelo seu envolvimento, dedicação e atenção dispensada para a realização de diversos estudos com os acadêmicos. A Dr^a. Letícia Machado, colega de doutoramento e grande parceira de incontáveis conversas e discussões sobre os nossos temas. Ao Luiz Vaz pelo apoio na realização de um estudo. Aos alunos da PUCRS que atuaram como voluntários nos diversos estudos. À prof^a. Dr^a. Josiane Kroll pelo suporte técnico e científico durante toda caminhada.

Esta tese não poderia ter sido realizada sem o apoio institucional da Universidade de Passo Fundo (UPF). Oportunizar a seus professores ampliarem seus conhecimentos para perceberem novos horizontes educacionais é, sem dúvida, um grande diferencial. Pertencer a uma instituição como a UPF me enche de orgulho. Agradeço aos colegas professores Dr. Juliano Tonezer da Silva e Dr. Júlio Bertolin, pelas conversas, pelos conselhos, pelas críticas e direcionamentos. Vocês foram “cajados” para minha caminhada. Aos profs. Dr. Adriano Pasqualloti, e a Dr^a. Marilene Portella pelas discussões acerca da análise de dados e pesquisa qualitativa. Ao prof. Dr. Carlos Holbig pelo apoio na análise dos dados. Aos alunos da UPF quando atuaram como voluntários em estudos, em especial aos meus orientandos de TCC, e co-orientandos no mestrado.

Agradeço aos professores Dr. Erran Carmel, Dr. Cleidson Souza, Dr^a. Helen Rozados e o Dr. Breno de França pela contribuição em diversas fases durante a realização do trabalho.

Finalmente, agradeço a minha família, eixo central nesta caminhada.

RECOMENDAÇÕES PARA TRABALHADORES NA MULTIDÃO SUPERAREM BARREIRAS EM PROJETOS DE SOFTWARE CROWDSOURCING

RESUMO

O desenvolvimento de projetos por meio de software crowdsourcing depende de um fluxo contínuo de trabalhadores da multidão para sua continuidade. Estes trabalhadores necessitam ser membros ativos, porém enfrentam dificuldades quando tentam participar em projetos de software crowdsourcing. Por esta razão, investigou-se quais foram as dificuldades que os trabalhadores da multidão enfrentam em plataformas de desenvolvimento de projetos via software crowdsourcing. Foram realizados vários estudos empíricos baseados em múltiplas fontes de dados e métodos de pesquisa, incluindo revisão da literatura, revisão por pares, estudo de campo e procedimentos da teoria fundamentada. Observou-se que os trabalhadores enfrentam muitas barreiras - relacionadas à competência, colaboração e gerenciamento do tempo - ao fazerem suas contribuições no desenvolvimento projetos por software crowdsourcing, o que pode resultar em desistências. Com base nas barreiras identificadas, revisão de literatura e sugestões dos trabalhadores foram apresentadas 13 recomendações para que estes trabalhadores possam superar tais barreiras. Estas recomendações foram avaliadas por especialistas em software crowdsourcing. As principais contribuições desta tese foram: a) identificação empírica das barreiras enfrentadas pelos desenvolvedores de software em projetos de software crowdsourcing; e b) recomendações para minimizar estas barreiras. Conclui-se que os trabalhadores da multidão precisam de competência e um esforço eficiente de gerenciamento de tempo para participar de forma colaborativa nas tarefas do desenvolvimento de projetos em software Crowdsourcing no modelo competitivo da plataforma Topcoder.

Palavras chaves: Software Crowdsourcing, barreiras, trabalhador na multidão.

RECOMENDAÇÕES PARA TRABALHADORES NA MULTIDÃO SUPERAREM

BARREIRAS EM PROJETOS DE SOFTWARE CROWDSOURCING

ABSTRACT

Software crowdsourcing development platforms require a continuous influx of crowdworkers for their continuity. Crowdworkers should be encouraged to play an important role in the online communities by being active members, but they face difficulties when attempting to participate. For this reason, we investigated the difficulties that crowdworkers face in crowdsourcing software development platforms. We conducted empirical studies relying on multiple data sources and research methods including literature review, peer review, field study, and procedures of grounded theory. We observed that crowdworkers face many barriers – related to competence, collaboration, and time management – when making their contributions in software crowdsourcing development, which can result in dropouts. Based on the identified barriers, literature review and, crowdworkers suggestions, we list 13 recommendations for participants as potential solutions to overcome such barriers. The recommendations were evaluated by surveying software crowdsourcing experts. The main contributions of this dissertation are a) empirical identification of barriers faced by crowdsourcing software development crowdworkers; and b) recommendations on how to minimize the barriers. We conclude that the crowdworkers need competency and an efficient time management effort to take part collaboratively in tasks of the Competition-Based Crowdsourcing Software Development of the Topcoder platform.

Keywords: Software crowdsourcing, crowdworker, barriers, recommendations

LISTA DE FIGURAS

Figura 1: Desenho da Pesquisa	22
Figura 2: Processo e Fases abordadas no estudo.....	23
Figura 3: Relação entre os elementos fundamentais no crowdsourcing.	35
Figura 4: Estratégias de desenvolvimento de software	47
Figura 5: Diferenças entre as estratégias.....	47
Figura 6: Modelo genérico de software crowdsourcing através de plataformas.....	52
Figura 7: Evolução do crescimento dos usuários cadastrados na Topcoder.	53
Figura 8: Etapas e fases do TopCoder.....	55
Figura 9: Processo de Desenvolvimento da AppStori.	56
Figura 10: Procedimentos da Fase 1	65
Figura 11: Consulta javascript no Topcoder.....	67
Figura 12: Primeira resposta	69
Figura 13: Segunda resposta	70
Figura 14: Barreiras Preliminares.....	77
Figura 15: Procedimentos da Fase 2	80
Figura 16: Apoio ao processo de codificação aberta	85
Figura 17: Exemplo de uma anotação.....	86
Figura 18: Categoria Central e suas relações	95
Figura 19: Categorias e suas Barreiras que emergiram da Codificação aberta	96
Figura 20: Categorias e Barreiras do trabalhador na multidão	99
Figura 21: Resultados dos graus de concordância	103
Figura 22: Etapas da Fase 3	104
Figura 23: Percentual de tempo de experiência dos especialistas.....	109
Figura 24: Nível de Concordância com a Rec#1	112
Figura 25: Nível de Concordância com a Rec#2	115

Figura 26: Nível de Concordância com a Rec#3.....	116
Figura 27: Nível de Concordância com a Rec#4.....	118
Figura 28: Nível de Concordância com a Rec#5.....	120
Figura 29: Nível de Concordância com a Rec#6.....	121
Figura 30: Nível de Concordância com a Rec#7.....	122
Figura 31: Nível de Concordância com a Rec#8.....	123
Figura 32: Nível de Concordância com a Rec#9.....	124
Figura 33: Nível de Concordância com a Rec#10.....	125
Figura 34: Nível de Concordância com a Rec#11.....	127
Figura 35: Nível de Concordância com a Rec#12.....	131
Figura 36: Nível de Concordância com a Rec#13.....	133
Figura 37: Resultado dos Graus de Concordância.....	133

LISTA DE TABELAS

Tabela 1: Desafios do DDS.....	31
Tabela 2: Definições de trabalhador na multidão	38
Tabela 3: Definições de software crowdsourcing	45
Tabela 4: Diferença entre OSS e SWCS no modelo competitivo	48
Tabela 5: Posicionamento de SWCS diante de outros modelos.	48
Tabela 6: Alvos do software crowdsourcing.	51
Tabela 7: Plataformas para projetos de software crowdsourcing.....	52
Tabela 8: Exemplo de usuários selecionados	69
Tabela 9: Origem das Barreiras da Categoria Tempo por Fonte de Dados	88
Tabela 10: Origem das Barreiras da Categoria Colaboração por Fonte de Dados .	90
Tabela 11: Origem das Barreiras da Categoria Plataforma por Fonte de Dados	92
Tabela 12: Origem das Barreiras da Categoria Pessoal por Fonte de Dados.....	94
Tabela 13: Origem das Barreiras da Categoria Tarefa por Fonte de Dados	95
Tabela 14: Transformação das barreiras em sentenças	100
Tabela 15: Frequência das barreiras.....	101
Tabela 16: Agrupamento das Barreiras.....	105
Tabela 17: Recomendações	106

LISTA DE ABREVIATURAS

AMT	<i>Amazon Mechanical Turk</i>
DDS	Desenvolvimento distribuído de software
FLOSS	<i>Free/Libre Open source software</i>
GT	<i>Grounded Theory</i>
HIT	<i>Human Intelligence Tasks</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
OSS	<i>Open source software</i>
SBQS	<i>Simpósio Brasileiro de Qualidade de Software</i>
SWCS	Software crowdsourcing
Swebok	<i>Software Engineering body of knowledge</i>
TFD	Teoria Fundamentada nos Dados
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1.	INTRODUÇÃO.....	14
1.1.	Questão de Pesquisa	18
1.2.	Objetivos da Pesquisa	19
1.3.	Método de Pesquisa.....	19
1.4.	Escopo.....	23
1.5.	Principais contribuições	25
1.6.	Outros resultados	25
1.6.1.	Artigos Publicados	25
1.6.2.	Participação em co-orientações.....	27
1.6.3.	Participação em orientação	27
1.6.4.	Outras contribuições.....	28
1.7.	Organização da tese.....	28
2.	BASE TEÓRICA	29
2.1.	Desenvolvimento distribuído de software	29
2.1.1.	Desafios do DDS	30
2.1.2.	Desafios do DDS relacionados às pessoas.....	31
2.2.	Crowdsourcing.....	32
2.2.1.	Elementos do Crowdsourcing.....	34
2.2.2.	Tipos de Crowdsourcing	41
2.2.3.	Software crowdsourcing	43
2.2.4.	Software crowdsourcing e suas propriedades	46
2.2.5.	Motivação para Software Crowdsourcing	49
2.2.6.	Projetos de software crowdsourcing mediados por plataformas.....	51
2.2.7.	Desafios do Software Crowdsourcing.....	57
2.3.	Reflexões sobre o capítulo	59
3.	TRABALHOS RELACIONADOS.....	60
3.1.	Trabalhadores em projetos de software crowdsourcing	60
3.2.	Trabalhadores em projetos de OSS	62
4.	BARREIRAS PARA CONTRIBUIR EM PLATAFORMAS DE SOFTWARE CROWDSOURCING	64
4.1.	Identificação preliminar das Barreiras e das Recomendações - Fase 1	64
4.1.1.	Revisão da literatura.....	65
4.1.2.	Primeiro estudo de campo.....	65
4.1.3.	Segundo estudo de campo.....	72
4.2.	Barreiras e Recomendações - Fase 2	80

4.2.1. Estudo de caso	81
4.2.2. Execução da coleta de dados.....	82
4.2.3. Barreiras Identificadas	84
4.2.4. Barreiras identificadas em todos os elementos do crowdsourcing	86
4.2.5. Propostas de Recomendações.....	97
4.3. Barreiras identificadas apenas para os indivíduos da multidão	98
4.4. Avaliação das Barreiras específicas do indivíduo da multidão	99
4.4.1. Coleta de dados para a avaliação das Barreiras	99
4.4.2. Resultados da avaliação das Barreiras	100
4.5. Reflexões sobre o capítulo	103
5. RECOMENDAÇÕES PARA CONTRIBUIR EM PLATAFORMAS DE SOFTWARE CROWDSOURCING	104
5.1. Recomendações aos trabalhadores da multidão.....	105
5.2. Coleta dos Dados para a avaliação das recomendações.....	107
5.3. Resultados e análise das recomendações	108
5.4. Reflexões sobre o capítulo	134
6. CONCLUSÕES.....	135
6.1. Limitações do trabalho.....	136
6.2. Trabalhos futuros.....	137
Referências	139
APÊNDICE A – E-mail enviado aos participantes Fase 1	152
APÊNDICE B – Questões abertas – Fase 1	153
APÊNDICE C – Protocolo do estudo de caso	154
APÊNDICE D – Termo de consentimento.....	155
APÊNDICE E – Descrição da Tarefa – Fase 2	156
APÊNDICE F – Questões abertas – Fase 2.....	157
APÊNDICE G – Questões fechadas – Fase 2	158
APÊNDICE H – E-mail enviado – Fase 3.....	161
APÊNDICE I – Questões – Fase 3.....	162
APÊNDICE J – Gráficos de caixa e de divergência	167

1. INTRODUÇÃO

O conhecimento ou inteligência coletiva, é uma prática comum de troca de conhecimento por novas formas de organização e de coordenação flexível e em tempo real [1]. Para Levy [2] a inteligência coletiva é “uma inteligência distribuída por toda parte, incessantemente valorizada, coordenada em tempo real, que resulta em uma mobilização efetiva das competências”. A partir desse conhecimento coletivo “as estruturas de hierarquização nas empresas estão sendo remodeladas”, favorecendo a participação de atores externos em processos de criação em massa, alterando, inclusive, as relações de trabalho entre empregador e empregados [3], [4].

A globalização e os avanços tecnológicos que guiam essas transformações e relações sociais dependem, essencialmente, da interação entre os atores envolvidos neste sistema. Os sistemas de software são um exemplo onde estas transformações tem sido percebida com uma velocidade significativa. Complexos por natureza, devido a diversos domínios de aplicação e setores da economia, o desenvolvimento de software dentro deste contexto de mudança tem sido dependente dos indivíduos envolvidos neste processo.

Crowdsourcing surge nesse contexto. Conforme Howe [5], crowdsourcing “*é o ato de pegar um trabalho tradicionalmente designado à um empregado e externá-lo para um grupo indefinido, e geralmente grande, de pessoas através da internet*”. Por meio de indivíduos independentes, distribuídos geograficamente, com conhecimentos e habilidades específicas, a criatividade advinda de uma grande quantidade de pessoas interagindo para executar diversas atividades e resolver problemas, cria oportunidades, mas também potencializa as dificuldades. O crowdsourcing e o mercado laboral on-line surgiram como novas formas de trabalho humano os quais permitem que as organizações dimensionem, de maneira flexível, sua força de trabalho e contratem especialistas, normalmente por um preço menor [6].

O crowdsourcing tem chamado a atenção como um novo ambiente de desenvolvimento de acordo com Deloitte [4] e Thuan *et al.* [7], pois utiliza o conhecimento e a sabedoria da multidão, para, a partir da colaboração, contribuir na resolução de problemas da sociedade nos mais diferentes segmentos da economia. Exemplos de aplicação do crowdsourcing são abordados por Cooper *et al.* [8], Brabham [9] e Xiao e Paik [10], incluindo a área de Engenharia de Software, e, por conseguinte, o desenvolvimento

de software [11], [12], [13], [14], [15]. Pode-se dizer que crowdsourcing é um caso extremo de desenvolvimento distribuído de software e conforme Howe [16], como a aplicação de vários princípios de software livre em empresas de software.

Quando uma organização investe em um projeto, normalmente a mesma torna-se proprietária da solução com os respectivos direitos de propriedade intelectual, pois oferece uma recompensa financeira à multidão pela solução proposta como destacam Marjanovic *et al.* [17]. Isto é diferente do que ocorre nos princípios do software livre. Ou seja, a multidão é paga para oferecer uma solução. Essa forma de trabalho é vista como inovadora e sem precedentes, como afirmam Doan *et al.* [11], Kaganer *et al.* [18] e Johns *et al.* [19]. Entretanto, em software livre, a simples identificação de um problema e uma provável solução para esse problema, pode ser considerada como a própria motivação e satisfação e, não envolve retribuição financeira direta.

Para o Prof. Brian Fitzgerald¹, crowdsourcing tem chamado a atenção da comunidade de engenharia de software: *“Crowdsourcing is emerging as an alternative outsourcing strategy which is gaining increasing attention in the software engineering community”*. Para Wu *et al.* [15], crowdsourcing para desenvolvimento de software, ou Software Crowdsourcing (SWCS), tem se tornado uma área emergente e ainda pouco explorada na engenharia de software.

Alguns estudos centram-se em entender as motivações dos trabalhadores em contribuírem em projetos de software crowdsourcing como apontados por Mao *et al.* [14], Stol e Fitzgerald [20] e LaToza *et al.* [21]. Autores como LaToza e van der Hoek [22] e Schmid-Druner [23] indicam que a primeira motivação para trabalhadores é receber uma compensação financeira para atividades em crowdsourcing. No modelo do crowdsourcing competitivo, Felstiner [24] afirma que:

Contest-based crowdsourcing (or competitive crowdsourcing) functions as an open competition, with firms broadcasting a problem or complex task to the crowd in the understanding that many crowd members may accept the challenge and perform the work but only one (or a small group) will receive the reward.

Escassos estudos mostram a aplicação do SWCS na análise do comportamento individual do trabalhador na multidão em projetos de software crowdsourcing no modelo

¹ Um dos principais pesquisadores em temas relacionados ao desenvolvimento de software com trabalho alternativo, durante palestra de abertura intitulada - *Crowdsourcing Software Development: Silver Bullet or Lead Balloon* – no *13th IEEE/ACM International Conference on Global Software Engineering / 2018*.

competitivo [25], [26]. A multidão de indivíduos e o cliente devem estar intrinsecamente ligados e relacionados sendo que essa multidão deve ser reconhecida e tratada como parceira em atividades de crowdsourcing. As necessidades, aspirações e motivações da multidão merecem alta consideração, como destaca Branquinho [27]. Ainda conforme Branquinho [27] uma questão crítica para o sucesso no processo de crowdsourcing pode estar relacionada a, dentre outras, *“à orientação e incentivo à multidão participante para completar a tarefa designada sem perder o foco”* além da *“preservação, após a conclusão do processo de crowdsourcing, dos contatos com os melhores provedores de soluções, quando externos à organização”*. Segundo Wu *et al.* [15], apenas 20% dos usuários da Topcoder, que oferece a possibilidade de desenvolvimento de projetos de software crowdsourcing são participantes ativos. Dos usuários registrados na Topcoder², 17,3% tem participado, ao menos, de uma competição de algoritmo, enquanto que apenas 0,3% em competições de Design e 0,7% em competições de Desenvolvimento conforme aponta Archak [28]. Será que esses percentuais são suficientes para manter a plataforma operando? Além disso, Hosseini *et al.* [29] realizaram um mapeamento sistemático na literatura para classificação das características da multidão e, não encontraram evidências de quantos indivíduos, efetivamente, terminam suas tarefas em ambientes crowdsourcing.

Nesse contexto, Qureshi e Fang [30] discutem que é razoável perceber que em projetos de software livre dependem da constante entrada e contribuição de novatos da multidão para a sua sobrevivência. Novos trabalhadores na multidão são a fonte da inovação para novos trabalhos como relatam Kraut *et al.* [31]. Entretanto, de acordo com Steinmacher [32] eles enfrentam diversas barreiras e dificuldades, as quais podem acarretar em desistências na execução das tarefas, muitas vezes antes de completá-las. Isto foi comprovado em um recente estudo com novatos em projetos de software livre. Por exemplo, se um trabalhador ao participar de uma competição, não ganhar por algumas vezes, será que ele continuará a participar em novos projetos?

Por exemplo, um dos principais desafios do modelo crowdsourcing, de acordo com Saengkhattiya *et al.* [33] é garantir que os participantes submetam suas tarefas com um mínimo de qualidade, especialmente quando esses participantes estão ansiosos em receber um retorno financeiro da atividade enviada.

² Topcoder é uma plataforma on-line que organiza competições no modelo crowdsourcing para desenvolvimento de software.

Diferentes fusos horários, distribuição geográfica das pessoas, e a falta de conhecimento da língua nativa da tarefa são apontados por Saremi e Yang [34] como aspectos motivacionais que estimulam o trabalhador da multidão a terminar sua tarefa. Neste sentido, Yang *et al.* [35] apontam um índice de 82,9% de taxa de desistência de submissão de tarefas de software crowdsourcing, considerando que a maioria dos participantes em comunidades on-line não contribuem, como apontam Hill *et al.* [36].

Além disto, os participantes que têm muita experiência com crowdsourcing podem ter uma vantagem significativa sobre aqueles que são inexperientes, na medida em que podem ser mais proficientes com uma plataforma e, portanto, podem ter mais chances de ganhar uma competição em software crowdsourcing como afirmam Stol e Fitzgerald [37], resultando, com isso, que os outros participantes desistam de submeter a tarefa. Assim, e na mesma direção, Mehta [38] considera um grande desafio em como manter ativos os trabalhadores da multidão.

Ademais, além de ter uma massa crítica de usuários, o crowdsourcing também depende da motivação e participação das pessoas conforme Zhao e Zhu [39]. Uma questão enfrentada pela maioria das comunidades virtuais é que os participantes *“têm uma alta tendência a interromper sua participação depois de algum tempo”* como destacam Lu *et al.* [40]. Straub *et al.* [41] discutem aspectos de formas de recompensação financeira devidas aos trabalhadores como fatores que influenciam o trabalhador a completar a tarefa em plataformas para micro-tarefas como a MTurk³. Também encontraram que a performance está diretamente relacionada a capacidade do trabalhador em entregar a tarefa.

Brabham [42] comenta que *“qualquer projeto baseado no modelo crowdsourcing é tão vibrante quanto sua comunidade online. Uma multidão insatisfeita com uma organização é livre para sair, e um grande êxodo pode fazer com que um projeto colapse completamente”* e, complementa ainda que entender exatamente quem participa em projetos de crowdsourcing é importante porque diferentes tipos de pessoas (indivíduos) possuem diferentes capacidades, talentos e motivações.

Conhecer a multidão ainda é um tema pouco explorado,

Little is known about who these workers and there are many open questions that would be of great interest. Some initial work has been done on workers' motivation, but we need a better understanding of why participants engage in this 'insecure' form of employment. Furthermore, while 'normal' development jobs typically require certain levels of formal education, little is known about

³ <https://www.mturk.com/>

the background of crowdsourcing workers. Other considerations include the reliability of workers, both in terms of customers being able to deliver products in a timely fashion, as well as careful consideration of product specific knowledge and IP. [37]

O conhecimento adquirido na execução da presente tese, auxiliará não apenas o próprio indivíduo que faz parte da multidão, mas também os outros três elementos do crowdsourcing: o cliente, a plataforma e a tarefa. O cliente no que se refere a criação de mecanismos que o auxiliem na elaboração de tarefas, enquanto que na plataforma o foco será nos subsídios que fundamentam a inserção de recursos tecnológicos capazes de antever possíveis problemas de desistências dos trabalhadores na multidão. E, na tarefa no que tange ao seu detalhamento e decomposição.

Assim, justifica-se entender quais são os fatores que impedem os trabalhadores a completarem suas tarefas em ambientes crowdsourcing e que mecanismos poderiam ser propostos para auxiliar estes trabalhadores a finalizarem estas tarefas. Entende-se, portanto, que existe oportunidade para uma investigação sobre o comportamento do indivíduo integrante da multidão em projetos de software crowdsourcing, uma vez que a desistência ou desinteresse dessas pessoas pode ser fatal para o sucesso da plataforma, do negócio do cliente e, por consequência, do modelo crowdsourcing.

Registra-se, por fim, que a presente tese é inspirada no trabalho desenvolvido por Steinmacher [32]. Em seu trabalho, o autor identificou 58 barreiras que os novatos enfrentaram após realizarem a primeira contribuição em projetos de desenvolvimento de software livre, agrupadas em seis categorias, entre elas: diferenças culturais, características dos novatos, problemas de recepção, orientação dos novatos, obstáculos técnicos, e problemas de documentação. Apresentou também um portal, denominado de FLOSScoach⁴, para apoiar os novatos em sua primeira contribuição.

1.1. Questão de Pesquisa

A questão de pesquisa da presente tese é:

“ Como apoiar os trabalhadores na multidão a participarem de tarefas em projetos de Software Crowdsourcing no modelo competitivo? ”

⁴ <http://www.flosscoach.com/>

Para responder à questão de pesquisa central serão investigadas as seguintes questões secundárias:

QP1: Quais são as barreiras enfrentadas pelos trabalhadores na multidão ao participarem em projetos de Software Crowdsourcing no modelo competitivo?

QP2: Quais são as recomendações que auxiliam os trabalhadores da multidão a superarem barreiras e minimizar desistências?

1.2. Objetivos da Pesquisa

O objetivo geral desta pesquisa é oferecer mecanismos para que os trabalhadores na multidão, registrem e submetam tarefas em projetos de Software Crowdsourcing no modelo competitivo.

Para alcançar o objetivo proposto, identificam-se os seguintes objetivos específicos:

- a) Entender, identificar e descrever as barreiras e dificuldades dos trabalhadores na multidão ao submeterem tarefas e, como os afetam em projetos de software crowdsourcing no modelo competitivo;
- b) Identificar e avaliar as recomendações que possam auxiliar os trabalhadores na multidão a superarem barreiras identificadas em projetos de software crowdsourcing no modelo competitivo.

1.3. Método de Pesquisa

O presente trabalho pode ser considerado como uma pesquisa científica aplicada. Ele foi planejado para ser executado como pesquisa qualitativa, pois teve enfoque no aprofundamento da compreensão do comportamento de indivíduos em ambientes sociais como mostram Dittrich *et al.* [43]. Em relação aos objetivos, esta pesquisa é classificada como descritiva, onde o pesquisador não interfere no ambiente alvo da pesquisa. Quanto aos procedimentos, esta pesquisa classifica-se como *ex-post-facto*, caracterizando que o estudo foi realizado após a ocorrência no ambiente pesquisado.

Foram utilizados vários métodos empíricos para alcançar o problema de pesquisa e responder as questões de pesquisa. Entre os quais destacam-se a revisão de literatura *ad-hoc*, o estudo de campo, o estudo de caso, as discussões em grupo, a revisão por pares, a análise quantitativa e, sobretudo, a análise qualitativa.

Vale ressaltar que a pesquisa foi realizada em diversas fases, considerando um período de tempo específico com relação ao fenômeno considerando, particularmente, eventos ocorridos e documentos produzidos entre os anos de 2016 e 2018. Para tanto, o processo de pesquisa também foi baseado em técnicas e práticas da Teoria Fundamentada nos Dados, apresentada por Strauss e Corbin [44], que é um método para extrair os resultados mais significativos, ou seja, descobrir, por meio das condições em que os fenômenos ocorrem, as barreiras ou dificuldades enfrentadas em atividades de projetos de software crowdsourcing no modelo competitivo na plataforma Topcoder.

A seguir apresentar-se-á a descrição de cada fase.

Fase 1 - Identificação Preliminar das Barreiras e Recomendações: nesta fase foram conduzidos alguns estudos para delimitação e fechamento do escopo, atender o primeiro objetivo específico e auxiliar no processo para responder a primeira questão de pesquisa da presente tese. Inicialmente, executou-se uma revisão de literatura *ad-hoc*, para auxiliar na contextualização para o problema bem como na análise das possibilidades presentes na literatura para a concepção do referencial teórico da pesquisa. Como método para a coleta dos dados optou-se pelo Estudo de Campo. O estudo de campo caracteriza-se pelo aprofundamento de uma realidade específica. Inicialmente fez-se uma observação direta das atividades do grupo estudado, além de entrevistas com os participantes para auxiliar no entendimento e interpretações que ocorreram no caso estudado [45]. Salienta-se, portanto, que o Estudo de Campo foi, por característica, “mais” aberto e não linear.

Enfim, como resultado desta fase, compreendeu-se o modelo crowdsourcing e seus elementos, analisou-se e discutiu-se o software crowdsourcing e suas relações e aplicações, identificou-se um dos problemas a ser abordado pela presente tese, bem como, elaborou-se uma lista preliminar das barreiras enfrentadas por usuários quando participam em projetos de software crowdsourcing no modelo competitivo.

Fase 2 – Definição e Avaliação das Barreiras, e elaboração de uma lista preliminar das recomendações: o objetivo da execução dessa fase foi atender à primeira questão de pesquisa bem como o primeiro objetivo específico da presente tese. Nesta fase foram realizados novos estudos de campo e um estudo de caso para a definição das barreiras além da elaboração preliminar de uma lista de recomendações para que os trabalhadores da multidão possam superar as barreiras identificadas para registro e submissão de tarefas em projetos de software crowdsourcing no modelo competitivo. Considerando o objetivo da presente tese, nessa fase, foram selecionadas com auxílio de

revisores, apenas as barreiras que os trabalhadores na multidão possam, por meio de alguma intervenção pessoal e individual, supera-las sem a participação efetiva e direta da plataforma ou do cliente.

No que se refere ao estudo de caso, o mesmo foi eleito devido a impossibilidade de controlar o ambiente e os eventos comportamentais processuais e, que segundo Yin [46] o Estudo de caso é *"uma inquirição empírica que investiga um fenômeno contemporâneo dentro de um contexto da vida real, quando a fronteira entre o fenômeno e o contexto não é claramente evidente e onde múltiplas fontes de evidência são utilizadas"*. O estudo de caso foi composto como um único caso com duas unidades de análise. Como o foco está na descrição de determinadas intervenções e, no contexto na qual ocorreu, o estudo de caso foi do tipo descritivo. Em relação às unidades de análise, o estudo é do tipo integrado, pois existe apenas um contexto único, composto por duas unidades integradas de análise.

Para a avaliação das barreiras, realizou-se um novo estudo de caso onde os participantes realizaram tarefas em uma plataforma de crowdsourcing, participaram de uma atividade de discussão em grupo bem como responderam um questionário para que avaliassem suas percepções sobre as dificuldades enfrentadas quando da realização das tarefas propostas. A partir da execução dessa fase fez-se uma avaliação individual das barreiras.

Fase 3 – Identificação e Avaliação das Recomendações: o objetivo da execução dessa fase foi atender à segunda questão de pesquisa e o segundo objetivo específico da presente tese. Com as barreiras avaliadas e uma lista preliminar de recomendações realizou-se uma busca na literatura por soluções semelhantes em áreas correlatas para a confecção das recomendações finais. Especialistas em software crowdsourcing avaliaram as recomendações por meio de um questionário. Como resultado obteve-se uma avaliação das recomendações propostas aos trabalhadores na multidão em projetos de software crowdsourcing no modelo competitivo na plataforma Topcoder.

Salienta-se, portanto, que em todas as fases para o processo de codificação, utilizou-se a lógica indutiva para a construção analítica dos códigos. Foram inferidas categorias teóricas dos dados e suas propriedades e redigidos memorandos (anotações e diagramas) para auxiliar na escrita das categorias à medida que as mesmas emergiam. A comparação constante entre dados, memorandos, categorias e códigos foi realizada. Para o registro dos dados, utilizou-se o software Nvivo®. Desta maneira, ao utilizar práticas da Teoria Fundamentada nos Dados, acredita-se que se desenvolveu um modelo coeso para

representar o fenômeno ora estudado. A constante participação de especialistas para os processos de codificação aberta (códigos preliminares), na categorização e na revisão por pares das barreiras específicas do trabalhador na multidão, e, por fim, na avaliação das recomendações, foram fundamentais para a consolidação desses processos sistematizados na pesquisa qualitativa. O desenho da pesquisa empregado durante todo o processo de desenvolvimento do trabalho pode ser observado na Figura 1.

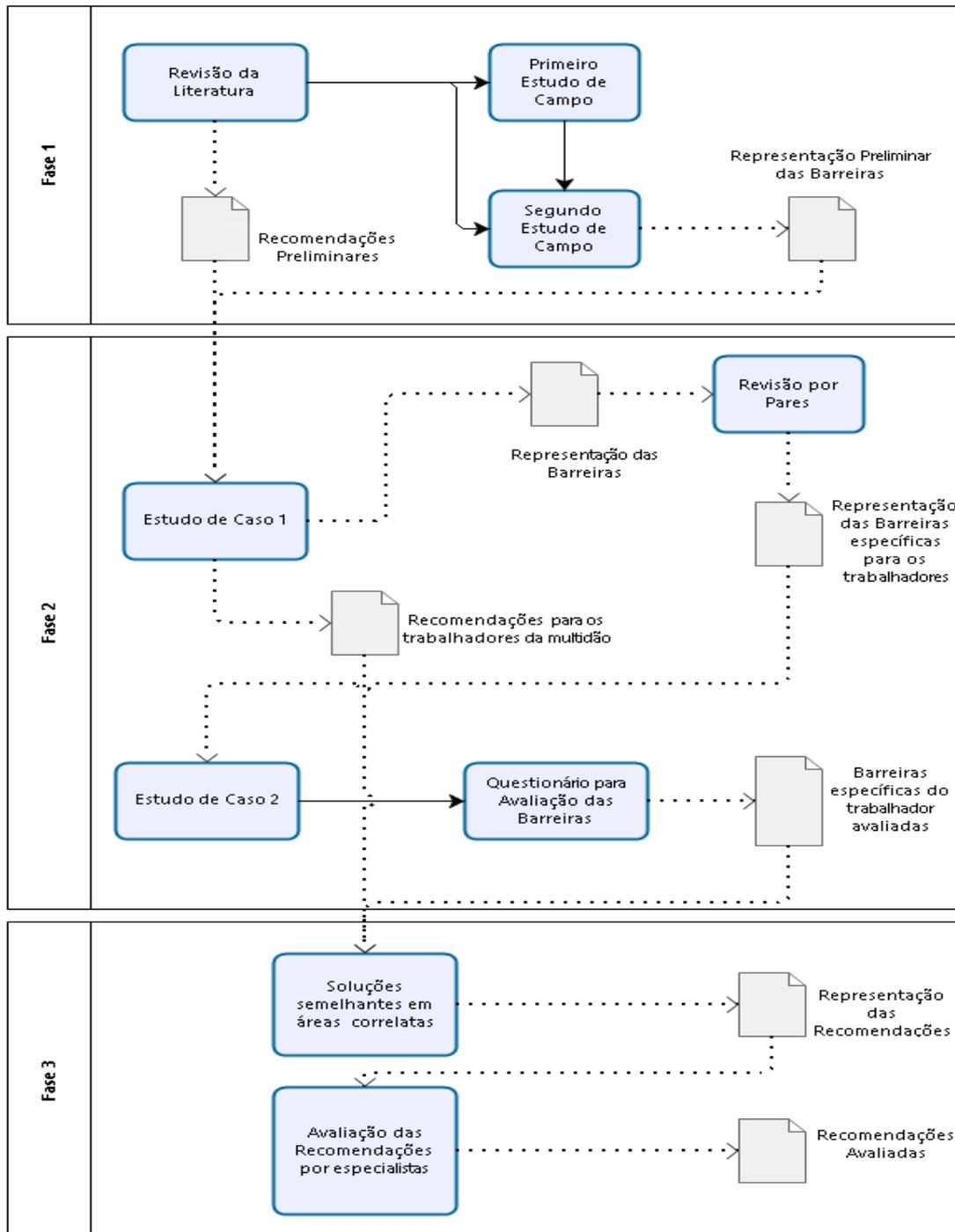


Figura 1: Desenho da Pesquisa

1.4. Escopo

Nesta seção é descrito o escopo bem como a definição de alguns termos utilizados neste trabalho. No que se refere ao foco, foi restritivo a software crowdsourcing devido a sua característica peculiar quando comparado, por exemplo, com Desenvolvimento Distribuído de software e Software Livre. Assim, os resultados não podem ser generalizados para esses dois modelos.

O termo crowdsourcing pode ser referenciado como *peer production, user-powered systems, user-generated content collaborative systems, community systems, social systems, social research, social media, collective intelligence, wkinomics, crowd wisdom, smart mobs, mass collaboration* e *human collaboration*, conforme Doan *et al.* [11]. A presente tese utilizará o termo crowdsourcing. Para software crowdsourcing, utilizar-se-á a definição proposta por Mao *et al.* [47], que é “o ato de transferir, externamente, qualquer tarefa no processo de desenvolvimento de software a um grande grupo indefinido de trabalhadores online em formato de chamada pública”.

O “trabalhador na multidão” é o indivíduo que faz parte da multidão e que realiza uma tarefa em um projeto de software crowdsourcing no modelo competitivo. Esta base conceitual está em consonância com a definição de Brabham [42]. No que se refere ao termo “participação do trabalhador na multidão”, é importante observar que essa participação ocorreu na execução de tarefas na plataforma Topcoder especificamente na etapa do processo “*Desenvolvimento de Componente*” e nas fases do “Registro” feito durante o prazo de inscrição, e na consequente fase da “Submissão” da mesma. Entende-se pelo termo “participar” o registro e a submissão de uma tarefa em uma competição. O processo e as fases utilizadas podem ser observados na Figura 2:



Figura 2: Processo e Fases abordadas no estudo.

É importante observar a diferença entre tarefa e atividade. Tarefa é a menor peça identificável e essencial de um trabalho e como meio de diferenciação entre os vários componentes de um projeto ou simplesmente como “um trabalho a ser feito ou realizado⁵”. *Tarefa é o trabalho prescrito, e refere-se àquilo que a pessoa deve realizar, segundo sua chefia, seus colegas ou segundo ela mesmo, enquanto atividade é trabalho como efetivamente realizado e refere-se ao modo como a pessoa realmente realiza sua tarefa⁶.*

Existem três categorias de trabalhos executados no modelo crowdsourcing, como trabalho voluntário, financiamento coletivo e trabalho pago como definem Green et al [48]. A presente tese utilizará a categoria crowdsourcing para trabalho pago, no modelo competitivo utilizado pela plataforma Topcoder como destacam LaToza e van der Hoek [22]. Registre-se, portanto, que o presente trabalho tem como pressuposto que o trabalhador na multidão está interessado em uma compensação financeira ao realizar uma tarefa em projeto de software crowdsourcing no modelo competitivo.

A partir de um conjunto de barreiras específicas para o trabalhador na multidão elaboraram-se recomendações para que esse trabalhador possa supera-las. Assim, tem-se como premissa que uma recomendação é aquilo que tenta guiar, ou um conselho, ou uma advertência para o trabalhador na multidão no registro e submissão de tarefas de projeto de software crowdsourcing, ficando, a livre escolha do trabalhador, optar, ou não, pela execução da recomendação apresentada. Importante observar que as recomendações propostas podem ser realizadas apenas pelo indivíduo, sem a intervenção direta ou indireta da plataforma ou do cliente.

Salienta-se, portanto, e dado o posicionamento do trabalho e todas as definições, que não é objeto desse trabalho tratar questões de atração e motivação do trabalhador na multidão em participar de competições no software crowdsourcing, e sim, o foco está em auxiliar por meio de recomendações específicas ao trabalhador, que possa se registrar e submeter uma tarefa. Outro aspecto que merece destaque é que as recomendações não foram avaliadas para os demais elementos do crowdsourcing: o Cliente, a Plataforma e a Tarefa.

⁵ <https://en.oxforddictionaries.com/definition/task>

⁶ http://www.labiutil.inf.ufsc.br/hiperdocumento/unidade2_1_2.html

1.5. Principais contribuições

Essa tese possui duas contribuições inéditas:

- a) Identificação empírica e representação das barreiras encontradas por trabalhadores na multidão em projetos de software crowdsourcing no modelo competitivo.
- b) Recomendações para que os trabalhadores na multidão possam superar as barreiras identificadas.

1.6. Outros resultados

Como parte integrante do processo de aprendizagem do modelo crowdsourcing, software crowdsourcing, e as dificuldades que os trabalhadores na multidão enfrentam em plataformas de software crowdsourcing no modelo competitivo, o autor da presente tese obteve outros resultados advindos de várias atividades, dentre as quais destacam-se:

1.6.1. Artigos Publicados

Durante o processo de doutoramento, vários artigos foram publicados na temática da presente tese. Foram dois artigos na principal conferência sobre Software Crowdsourcing (CSI-SE) nos anos de 2017 e 2018, além de um artigo em um *Special Issue*, da IEEE Software, sobre software crowdsourcing, no ano de 2017. Em 2016, na IEEE, publicou-se um trabalho sobre as plataformas de software crowdsourcing. Publicou-se em 2015, durante a Escola Latino Americana de Engenharia de Software. A seguir, uma lista dos artigos aceitos especificamente na temática da tese:

- a) ZANATTA, A. L.; MACHADO, L. S.; STEINMACHER, I.F. Competence, Collaboration and Time Management: Barriers and Recommendations for crowdworkers. In: The 5th International Workshop on Crowd Sourcing in Software Engineering, 2018, Gotemburgo. The 5th International Workshop on Crowd Sourcing in Software Engineering, 2018.

A realização desse trabalho contribui para a avaliação, por meio de um estudo de caso, das barreiras enfrentadas por trabalhadores na multidão em projetos de software crowdsourcing no modelo competitivo e um conjunto de recomendações para que esses trabalhadores possam superar essas barreiras.

b) ZANATTA, A. L.; Steinmacher, I.; MACHADO, L. S.; Souza, C.; PRIKLADNICKI, R. Barriers Faced by Newcomers to Software-Crowdsourcing Projects. *IEEE SOFTWARE*, v. 34, p. 37-43, 2017.

A realização desse trabalho contribui para a elaboração de uma lista preliminar das barreiras enfrentadas por novatos em suas primeiras contribuições em projetos de software crowdsourcing no modelo competitivo. Os resultados desse estudo foram essenciais para a identificação das recomendações. Este trabalho foi selecionado para apresentação oral no programa “*Journal First*” durante o Simpósio Brasileiro de Engenharia de Software-2018.

c) MACHADO, L. S.; ZANATTA, A. L.; MARCZAK, S.; PRIKLADNICKI, R. The Good, the Bad and the Ugly: An Onboard Journey in Software Crowdsourcing Competitive Model. In: *The 4th International Workshop on Crowd Sourcing in Software Engineering, 2017, Buenos Aires. The 4th International Workshop on Crowd Sourcing in Software Engineering, 2017.*

Este trabalho apresentou um conjunto das barreiras enfrentadas e prováveis sugestões para trabalhadores em suas primeiras contribuições em projetos de software crowdsourcing no modelo competitivo. O resultado foi obtido por meio da realização de um estudo de caso e entrevistas com os participantes.

d) ZANATTA, A. L.; MACHADO, L. S.; PEREIRA, G. B.; PRIKLADNICKI, R. Software Crowdsourcing Platforms. *IEEE Software*, v. 33, p. 112-116, 2016.

O objetivo principal desse trabalho foi conhecer e analisar as principais plataformas para o desenvolvimento de software crowdsourcing. A análise comparativa foi fundamental para a eleição da plataforma utilizada no desenvolvimento desta tese.

e) PEREIRA, G. B.; ZANATTA, A. L.; PRIKLADNICKI, R. What challenges project managers face in software crowdsourcing? *2nd Latin-American School on Software Engineering, 2015, Porto Alegre. ELA-ES 2015: [anais da] II Escola Latino Americana de Engenharia de Software. Porto Alegre: UFRGS, 2015. v. 1. p. 142-145.*

Primeiro artigo publicado durante a realização do doutorado. O objetivo desse trabalho foi, além de conhecer temas relacionados ao crowdsourcing, entender o funcionamento do desenvolvimento de software baseado no crowdsourcing, seus desafios e oportunidades no que tange ao gerenciamento de projeto de software. Também, possibilitou-se conhecer os métodos de pesquisas associados à busca de barreiras enfrentadas pelos gerentes de projetos associadas à projetos de SWCS.

1.6.2. Participação em co-orientações

Como professor Colaborador do Mestrado do Programa de Pós-Graduação em Computação Aplicada da Universidade de Passo Fundo, foi co-orientado um aluno e estão sendo co-orientados outros dois. A seguir, uma lista dos alunos com temas relacionados a presente tese:

- a) Mateus Henrique Dal Forno, sob título: “CPFTL: Uma proposta de processo adaptável para testes funcionais utilizando o crowdsourcing”, dissertação defendida em 2016.
- b) Fernando Costela, na área de software crowdsourcing, com título indefinido. (*em andamento*).
- c) Tiago Moraes Ferreira, na área de sistemas de recomendação com aplicação no software crowdsourcing, com título indefinido. (*em andamento*).

1.6.3. Participação em orientação

Como professor Titular I⁷ no Bacharelado em Ciência da Computação da Universidade de Passo Fundo, tive a oportunidade em orientar, entre 2015 e 2018, sete alunos de graduação no Trabalho de Conclusão de Curso com temas relacionados à presente tese. A seguir, uma lista dos orientandos e seus respectivos trabalhos:

- a) Petronio Bedin, com trabalho sob título “*Software Testing* Crowdsourcing: Uma análise comparativa com o Swebok”, concluído em 2015.
- b) Filipe Tiago Menegatti, com trabalho sob título “Uma Análise Comparativa de Plataformas para Crowdsourcing”, concluído em 2016.
- c) Jessica Bossio, com trabalho sob título “Uma proposta de soluções para usuários novatos superarem barreiras em atividades de crowd testing”, concluído no ano de 2017.
- d) Rafael Koegler, com trabalho sob título “Sistemas de Reputação: Uma Análise em Plataformas de Crowd Testing”, concluído em 2017.
- e) Angélica Meneghini Boff, com trabalho sob título “Barreiras e recomendações em testing crowdsourcing”, concluído em junho de 2018.

⁷ Titulação denominada conforme plano de carreira próprio da instituição.

- f) Lucas Darigo Altmann, com trabalho sob título “Uma proposta de ferramenta para Testing Crowdsourcing”, concluído em junho de 2018.
- g) Regis Sganzerla, com trabalho em andamento e previsão de defesa de TCC em novembro de 2018.

1.6.4. Outras contribuições

- a) Elaboração de uma revisão sistemática de literatura sobre testes em software crowdsourcing e produção de um artigo.
- b) Mantenedor de um site brasileiro que trata das questões relacionadas ao projeto de software crowdsourcing, disponível em: <http://www.softwarecrowdsourcing.com.br/>
- c) Difusão do conceito, suas aplicações e características do modelo software crowdsourcing, por meio do auxílio na preparação de:
 - o Um minicurso intitulado: “Software Crowdsourcing: Desafios e soluções”, apresentado pela professora Dr^a. Sabrina Marczak no 15º Simpósio Brasileiro de Qualidade de Software em 2016.
 - o Um tutorial intitulado: “Software crowdsourcing”, a ser apresentado durante o CBSOFT 2018. Foram também autores deste tutorial – Leticia dos Santos Machado, Ricardo Rodrigo Marinho Melo, Cleidson R. B. de Souza, Rafael Prikladnicki e Sabrina Marczak.

1.7. Organização da tese

Este volume está organizado em 6 capítulos. Na sequência, o capítulo 2 apresenta a base teórica desta pesquisa, no que se refere aos conceitos de Engenharia de Software, Desenvolvimento Distribuído de Software, e, principalmente o crowdsourcing e o Software Crowdsourcing.

No capítulo 3, apresentam-se os trabalhos relacionados acerca do tema central dessa tese. No capítulo 4, apresentam-se as barreiras identificadas para os trabalhadores na multidão. No capítulo 5, são apresentadas as recomendações que auxiliam o trabalhador na multidão a superarem as barreiras identificadas, bem como, a avaliação dessas recomendações. Finalmente, no Capítulo 6 são apresentadas as considerações finais, com ênfase nas contribuições. Também são discutidas algumas limitações. Conclui-se destacando os trabalhos futuros, bem como algumas oportunidades de pesquisas.

2. BASE TEÓRICA

O referencial teórico é uma importante etapa da pesquisa pois contém os principais elementos, conceitos e teorias fundamentais para o desenvolvimento do projeto de pesquisa. Na seção 2.1 apresentam-se os conceitos e as principais características do Desenvolvimento Distribuído de Software, caracterizando, em especial, os desafios existentes ao utilizar esse modelo. O modelo crowdsourcing e seus elementos, suas características e aplicações, além do software crowdsourcing são apresentados na seção 2.2.

2.1. Desenvolvimento distribuído de software

O software é considerado como um importante ativo nos negócios empresariais, tornando-se um elemento fundamental como ferramenta de competição, pois conforme Herbsleb e Moitra [49] está sendo cada vez mais oneroso e menos competitivo desenvolvê-lo no mesmo espaço físico, na mesma organização ou até no mesmo país. Para minimizar esses aspectos algumas organizações buscam pessoas ou equipes em diferentes localizações geográficas para desenvolver softwares.

Desenvolvimento Distribuído de Software (DDS) é uma abordagem caracterizada pelo desenvolvimento de software realizado por equipes dispersas geograficamente. Essas equipes podem estar no mesmo país, ou fora dele, pertencerem a mesma organização ou terceirizadas. Diversas razões conduzem para o desenvolvimento distribuído de software, entre as principais destacam-se a “Demanda de Custos”, a “Sinergia Cultural”, o “Mercado Global”, a “Escala”, o “*Time-to-market*”, e, por fim, o “Rigor e Experiência” como demonstram Audy e Prikladnicki [50]. Ainda conforme esses autores, a disponibilidade de pessoas qualificadas não acompanha o crescente aumento de produção de software, gerando, com isso, uma busca por indivíduos com recursos equivalentes em regiões diferentes com menor custo de produção. Pressões para entregar o produto ou serviço em um menor espaço de tempo, também é apontado pelos autores como uma razão para o uso do DDS. Sugerem também, o desenvolvimento conhecido como *follow-the-sun*, onde as equipes distribuídas trabalham 24 horas por dia, isto é, quando uma equipe terminar seu turno, a outra continua o seu trabalho independentemente do local físico onde estão localizadas. Contudo, se faz necessário, que essas equipes utilizem mecanismos formais de controles e documentação utilizando novas formas de resolver os problemas. Assim,

quanto maior uma organização produtora de software, mais necessidade de controles, tornando-a, por conseguinte, difícil de gerenciar.

Ao transferir alguma parte de sua operação para outro local, alguns aspectos organizacionais do DDS necessitam ser discutidos, como apresenta Karolak [51]. Primeiro, se deve definir se a produção será *offshore* (em outro país fora do mercado onde o produto será vendido), ou *onshore* (no mesmo país e mercado onde o produto será vendido). Depois, se faz necessário definir se as atividades serão desenvolvidas em operação ou se as atividades devem ser terceirizadas (*outsourcing* tradicional), ou ainda, se serão estabelecidas outras formas entre as possibilidades oferecidas pelo DDS.

De acordo com Audy e Prikladnicki [50] o *outsourcing* tradicional “é caracterizado pelo cenário onde uma empresa delega o controle sobre uma ou mais atividades para uma empresa externa cujo serviço contratou”. O *insourcing* consiste “na relação onde a empresa cria seus próprios departamentos e centros de desenvolvimento de software”. Existem as seguintes formas de relacionamento entre fornecedores e clientes (*outsourcing* e *insourcing*) e a distribuição geográfica (*offshore* e *onshore*) como apontam Audy e Prikladnicki [50]:

- *Onshore insourcing*: demanda doméstica;
- *Onshore outsourcing* ou *outsourcing*: terceirização fora do país;
- *Offshore outsourcing* ou *offshoring*: terceirização no país;
- *Offshore insourcing* ou *captivel/internal offshoring*: filial da empresa em outro país.

A decisão em optar pela forma de relacionamento e a distribuição geográfica no modelo DDS é dependente do modelo de negócio adotado pela empresa para alcançar seus objetivos organizacionais.

2.1.1. Desafios do DDS

O DDS oferece grandes mudanças na maneira como os produtos ou serviços são pensados, desenvolvidos, e testados, por isso, se faz necessário um suporte ou uma estrutura necessária para o desenvolvimento desse tipo de modelo, logo, são inerentes ao novo modelo alguns desafios.

Na Tabela 1, e de acordo com Audy e Prikladnicki [50], podem ser observados todos os desafios do DDS separados pelas categorias, Pessoas, Processo, Tecnologia, Gestão e, por fim, a Comunicação.

Tabela 1: Desafios do DDS

Categorias	Desafios
Pessoas	Confiança; Conflitos; Diferenças culturais; Ensino de DDS; Espírito de equipe; Formação de equipes e grupos; Liderança; Tamanho da equipe.
Processo	Arquitetura de software; Engenharia de requisitos; Gerência de configuração; Processo de desenvolvimento.
Tecnologia	Tecnologia de colaboração; Telecomunicações.
Gestão	Coordenação, controle e interdependência; Gestão de portfólio de projetos; Gerência de risco; Legislação (incentivos fiscais e tributários); Legislação (propriedade intelectual); Modelos de negócio; Seleção e alocação de projetos.
Comunicação	<i>Awareness</i> ; Contexto; Dispersão geográfica; Estilo de comunicação; Formas de comunicação; Fusos horários.

Fonte: Audy e Prikladnicki [50]

Devido um provável relacionamento das relações dos desafios das pessoas no DDS, com as pessoas trabalhadoras na multidão no software crowdsourcing, a seguir, apresentar-se-á apenas o detalhamento dos desafios da categoria Pessoa.

2.1.2. Desafios do DDS relacionados às pessoas

As pessoas são componentes fundamentais em qualquer processo de desenvolvimento de software, por isso, acrescentar o elemento da distribuição no desenvolvimento potencializa e faz surgir novos desafios relacionados às pessoas nesse modelo. No DDS, são apresentados por Audy e Prikladnicki [50] os desafios como a Confiança, Conflitos, Diferenças culturais, Ensino de DDS, Espírito de equipe, Formação de equipes e grupos, Liderança, e Tamanho da equipe.

Ainda conforme os autores, a construção de relações de confiança é fator essencial para o funcionamento da equipe. Mesmo em equipes distribuídas manter uma integração presencial com todos os integrantes é fundamental para estabelecer a confiança da equipe no início do projeto. Conflitos em ambientes distribuídos de software são comuns como em qualquer outra atividade de desenvolvimento. O gerenciamento da diversidade cultural é fundamental para efetividade de uma equipe distribuída, principalmente em âmbito global.

Como as equipes são “unidades sociais frágeis que podem facilmente ser quebrada” é importante manter o chamado “espírito da equipe”.

As equipes em DDS precisam de mecanismos de relacionamentos eficientes para criar uma visão compartilhada por meio de interações. E, o líder desempenha o papel de facilitador nesse processo além de, democraticamente, discutir e receber opiniões da equipe. Quanto mais pessoas e papéis envolvidos, mais desafiadores serão os problemas relacionados à coordenação e ao controle.

Existem muitos desafios e oportunidades associados ao DDS, como brevemente relatados. Porém, ao acrescentar pessoas “desconhecidas” no desenvolvimento de software, outros desafios aparecem aos gestores. Nesse sentido, abordagens como o crowdsourcing, caracterizadas como um caso extremo de DDS, de acordo com Howe [16], potencializam esses desafios.

2.2. Crowdsourcing

As relações entre empregados e empresas estão mudando. Onde os trabalhadores eram empregados em tempo integral com salários e benefícios bem definidos, agora passam para a execução de tarefas remuneradas sob demanda com contratos específicos, deixando-os livres para executarem tarefas a sua escolha.

Nesse sentido e conforme Deloitte:

Transforming the workforce: The growth of alternative work arrangements
Technology is transforming more than the way individual jobs are done—
it’s changing the way companies source labor. Many global companies
already actively use crowdsourcing efforts to generate new ideas, solve
problems, and design complex systems. [4]

Assim, o termo crowdsourcing, (neologismo de “*crowd*” – multidão e “*outsourcing*” – terceirização), sem tradução para o português significa, literalmente, “fonte de informações de uma multidão”, onde a multidão tem o objetivo de, ao colaborar, contribuir com um propósito em específico tendo como vantagem a acessibilidade econômica e de infraestrutura. Jeff Howe e Mark Robinson publicaram, em junho de 2006, o artigo denominado *The Rise of Crowdsourcing* na revista Wired Magazine, definindo assim crowdsourcing:

é o ato de pegar um trabalho tradicionalmente designado à um empregado e externá-lo para um grupo indefinido, e geralmente grande, de pessoas através da internet. (tradução nossa) [5].

Outra definição de crowdsourcing:

“Simply defined, crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call”. This can take the form of peer production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers [5].

Por ser um termo recente, Estellés-Arolas *et al.* [52] encontraram aproximadamente 40 definições originais do termo “crowdsourcing” sendo as mais frequentes utilizadas pelo criador do termo, Jeff Howe e o professor Daren C. Brabham [53].

O uso do crowdsourcing está crescendo⁸ e desenvolveu-se rapidamente na última década impulsionado pelo advento da Web 2.0, como apontam Peng *et al.* [54] e Stol e Fitzgerald [20]. A Web 2.0, como destaca O’Reilly [55], é um termo utilizado para definir um padrão de computação o qual utiliza a internet como um meio facilitador para a colaboração e compartilhamento de informações entre os indivíduos, tornando-os, assim, mais participativos.

O modelo de produção que o crowdsourcing utiliza é totalmente dependente da inteligência coletiva, ou seja, do conhecimento e da experiência dos indivíduos conectados à internet para solucionar diversos problemas. É uma maneira eficaz de realizar tarefas que podem ser resolvidas por seres humanos, todavia, são difíceis de serem solucionadas por computadores, como comentam Allahbakhsh [53] e Keimel [56]. O crowdsourcing proporciona as organizações o aumento da produtividade, a redução de custos e a melhoria dos processos de apoio, como discute Allahbakhsh [53]. Trata-se de um modelo de negócio que envolve terceirização colaborativa, podendo ser ou não remunerada de acordo com Peng *et al.* [54].

Destaca-se que o crowdsourcing diferencia-se basicamente de *outsourcing*, pois no crowdsourcing não se “conhecem”⁹ os executores das tarefas solicitadas. O *outsourcing* consiste, basicamente, conforme Audy e Prikladnicki [50] na transferência das atividades conhecidas como atividades meio, e nunca as atividades fins para uma organização terceirizada, sendo mais focada como parceria. A contratação de serviços periféricos visa reduzir custos internos aproveitando a especialização de empresas externas que, em determinadas áreas específicas, se revelam como opção vantajosa, permitindo maiores

⁸ Andrew Karpie, em seu blog <http://spendmatters.com/> comenta que “Gartner has predicted that, by 2018, over 75% of high-performing enterprises will be using some type of crowdsourcing for business process services”

⁹ Apenas a plataforma possui acesso a todos os dados dos usuários cadastrados para a execução da tarefa.

rentabilidades, uma vez que os empregados subcontratados normalmente percebem vencimentos reduzidos. Ramakrishnan e Srinivasaraghavan [57] comentam que o crowdsourcing é nova geração do *outsourcing* em tecnologia da informação. A aplicação do crowdsourcing é diversificada e está presente em diversos setores econômicos, desde a venda de diversos produtos como aponta Howe [16], até em pesquisas de fármacos conforme destaca Kittur *et al.* [58]. Um exemplo que a multidão poder trazer bons resultados é a Wikipédia, que destaca a inteligência coletiva valorizando as capacidades e habilidades do indivíduo, independentemente de sua formação de acordo com Giles [59] e Howe [16].

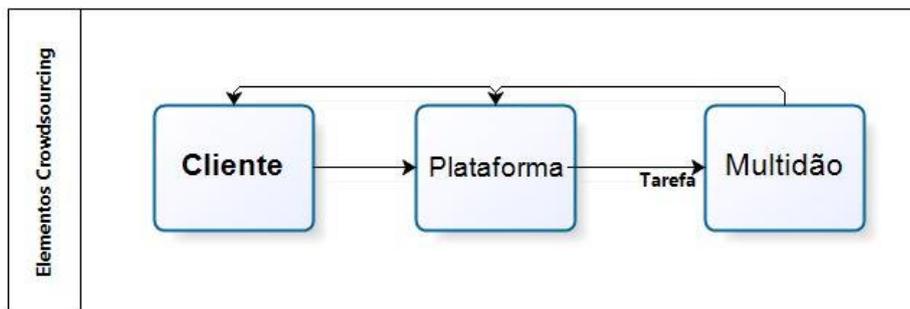
2.2.1. Elementos do Crowdsourcing

Autores como Stewart *et al.* [60] e Whitley [61] tratam o crowdsourcing como um processo que envolve vários atores e operações, enquanto outros como Vukovic [62], Schenk e Guittard [63] e Kittur *et al.* [64] considera-o como uma plataforma com características e funções específicas que auxiliam na realização de processos on-line. Um tratamento tão diversificado mostra o nível de especificidade sobre o qual o crowdsourcing é estudado como destacam Zhao e Zhu [39].

Crowdsourcing pode ser analisado como um conjunto de processos que transforma entradas em saídas utilizando seus quatro elementos: o cliente, a plataforma, a multidão e a tarefa. Esse processo é criado por pessoas com distintas visões de comportamento, de planejamento, requisitos, projeção e previsão, implementação, desdobramento e operacionalização de acordo com Branquinho [27].

Para compreender o modelo do crowdsourcing é importante definir e delimitar esses elementos fundamentais, onde, o **Cliente** solicita um serviço para a **Plataforma** a qual o encaminha por meio de uma **Tarefa** para que o trabalhador na **Multidão** a realize. Quem define o que será feito é o Cliente, porém, quem gerencia esse processo é a plataforma, mas quem dita o ritmo são os trabalhadores da multidão.

A Figura 3, ilustra e resume a relação existente entre esses quatro elementos fundamentais no crowdsourcing.



Fonte: adaptada de [65]

Figura 3: Relação entre os elementos fundamentais no crowdsourcing.

A seguir apresenta-se uma descrição dos quatro elementos fundamentais do crowdsourcing.

a) **Cliente:** No contexto do crowdsourcing, o cliente é aquele que solicita um serviço para a multidão, por meio de uma plataforma. Essas tarefas podem envolver, por exemplo, a elaboração de projetos de design gráfico, de desenvolvimento de software, teste de software, de criação/compra de marketing/conteúdo criativo, de vendas e suporte, dentre outros, como apontado por Vukovic [62]. Em relação às vantagens para os clientes que utilizam esta abordagem, cita-se a facilidade da realização de tarefas “sob demanda”, ou seja, o trabalhador na multidão que realiza as tarefas para a organização não possui vínculo empregatício¹⁰ com a mesma, e a organização se beneficia de uma força de trabalho que é mobilizada de acordo com sua demanda. A organização realiza o pagamento pela execução da tarefa somente após sua conclusão. Desta forma, e como aborda Li [66], ocorre uma redução de custos para a organização, que não necessita pagar pelo tempo demandado para que a tarefa fosse cumprida, como ocorre nas organizações em que os funcionários são pagos pela sua jornada de trabalho.

b) **Plataforma:** A plataforma é considerada o segundo elemento do crowdsourcing, apoia e facilita a mobilização da multidão, gerenciando e controlando a realização das tarefas. Vukovic [62] e Prikladnicki *et al.* [65] comentam que as plataformas são mantidas e gerenciadas por organizações, sendo as responsáveis por aspectos técnicos da realização das tarefas, que incluem, basicamente, a criação das tarefas a serem desenvolvidas, a

¹⁰ Conforme o artigo 3º da Consolidação das Leis Trabalhistas Brasileiras define o empregado como: "toda pessoa física que prestar serviços de natureza não eventual a empregador, sob a dependência deste e mediante salário".

seleção dos indivíduos para sua realização, a coleta e envio dos resultados ao cliente e o pagamento pela realização das tarefas aos indivíduos.

Algumas características comuns nas plataformas são descritas por Tsai *et al.* [67] e a seguir destacadas:

1. Uso de painéis administrativos, utilizados pelos administradores do sistema e pelos organizadores das atividades para o gerenciamento das tarefas;
2. Ferramentas de colaboração e comunicação, que possibilitam a realização de discussões, com a participação dos indivíduos e os responsáveis pelas plataformas;
3. Classificação de participantes e ferramentas de recomendação;
4. Ferramentas que possibilitam o desenvolvimento da tarefa solicitada;
5. Gestão de pagamento;
6. Repositório de arquivos, disponibilizando informações que envolvem a especificação da tarefa, bem como critérios de qualidade e de aceitação da solução a ser desenvolvida.

São exemplos de plataformas para crowdsourcing a *Amazon Mechanical Turk* (AMT), *Turkit*, *Mob4hire*, *uTest*, *Freelancer*, *eLance*, *Guru*, *Trada*, *99design*, a *InnoCentive*, *CloudCrowd* e *CloudFlower*, *Topcoder*, entre outras como destacam Doan *et al.* [11], Mao *et al.* [47] e, Zanatta *et al.* [68].

Como argumentam Stol e Fitzgerald [20], é difícil desenvolver softwares apenas por meio de HIT, acrônimo do inglês *Human Intelligence Tasks*, pois as tarefas são, em sua grande maioria, complexas e precisam de indivíduos com habilidades e conhecimentos específicos, além, de necessitarem tempo para a sua execução, muito diferente do que ocorre com as HITs. Como desenvolver software necessita de indivíduos especializados, e Schenk e Guittard [69] comentam que a multidão é basicamente composta por amadores, isso acrescenta um desafio para a atividade de projetos de software crowdsourcing. Portanto, existem plataformas que utilizam o modelo do crowdsourcing que podem ser utilizadas para o desenvolvimento de software, como a *Topcoder* – considerada por Li [66] a mais popular, a *CrowdPlat* e a *Upwork*.

c) **Multidão:** O terceiro elemento, e considerado por Saxton *et al.* [70] o elemento “*mais importante*” do crowdsourcing é a multidão. A multidão é composta por um grande¹¹ número de pessoas conectadas, que executam as tarefas solicitadas pelo cliente, por meio de uma plataforma [71]. Esses trabalhadores estão dispersos geograficamente, possuem diferentes conhecimentos, falam diversos idiomas, vivenciaram muitas experiências, são de várias culturas, possuem diversas idades e escolaridades, entre muitas outras características típicas de trabalhadores em uma multidão, como comenta Brabham [42]. Essas múltiplas características dos trabalhadores trazem um componente diferenciador e desafiador ao modelo crowdsourcing como ocorre no DDS conforme Audy e Prikladnicki [50]. Olson e Rosacker [72] acrescentam ao perfil do trabalhador na multidão no crowdsourcing como competitivo, transitório, dinâmico, atuando em curto prazo.

O crowdsourcing utiliza a capacidade criativa, a inteligência coletiva e a heterogeneidade desses indivíduos para a realização das tarefas solicitadas pelos clientes, que são disponibilizadas por meio de diversas plataformas existentes. Cada trabalhador na multidão pode realizar a tarefa de forma individual ou coletiva, tendo a iniciativa de selecionar as tarefas que deseja realizar, de acordo com suas necessidades e interesses pessoais.

A diversidade da multidão traz soluções e ideias inovadoras. O retorno obtido por meio da realização de determinada tarefa pode ocorrer pelo pagamento de um valor monetário definido, e também em forma de reputação, demonstrando, com isso, o comprometimento do trabalhador na multidão na realização das tarefas e evidenciando o nível de qualidade de sua produção, [54], [65], [73], [62]. Uma comunidade online sobrevive apenas se manter uma massa crítica de participantes ativos [74], [75], [70].

Dada a diversidade da participação da multidão em atividades de crowdsourcing, Brabham [42] apresenta um estudo com várias proposições de taxonomias dessa multidão. Essa taxonomia é útil para entender os diferentes níveis de engajamento que os trabalhadores na multidão podem ter, e as habilidades necessárias para obter soluções de qualidade de multidões diversas, além de suas motivações para executarem as tarefas. Na maioria das comunidades on-line seguem a regra 90-9-1, onde 90% dos usuários apenas observa e nunca contribuem, denominados de “*lurkers*”, 9% contribui ocasionalmente, chamados de “*Contribute casually*” e, por fim, apenas 1% são responsáveis pelas efetivas

¹¹ No crowdsourcing definitions place a minimum or maximum limit as to the size of a crowd. However, the crowd is defined in some instances as generally a large network of people [73].

contribuições na comunidade denominados de “*Heavy contributors*”, conforme Nielsen [79]. Martineu [80] também classifica como “*lurkers*” aqueles trabalhadores que apenas observam e não contribuem.

A Tabela 2 apresenta um quadro comparativo da definição do termo “crowd worker”.

Tabela 2: Definições de trabalhador na multidão

Definição de trabalhador na multidão	Referências
<i>“Workers are the individuals who perform the work—they develop the chunks of software that are ‘outsourced’ by a customer”.</i>	[20]
<i>“In this sense, crowd work is a sociotechnical work system constituted through a set of relationships that connect organizations, individuals, technologies and work activities”.</i>	[58]
<i>“An online community of individuals engaged in a crowdsourcing activity.”</i>	[42]
<i>“The people working through on-line platforms have been called different things: ‘micro-entrepreneurs’, ‘gigs’, ‘contractors’, ‘on-demand workers’, ‘freelancers’”</i>	[76]
<i>“A crowd worker is an individual who apply for jobs / tasks posted on crowdsourcing platforms. We consider these workers a part of external workforce.”</i>	[77]
<i>“Crowdsourcing is essentially driven by an open call, which implies that the requester knows little to nothing about potential contributors. However, practically the crowd that can be reached via the call and should respond to it is determined by the crowdsourcing platforms and advertising channels the requester uses or by knowledge and skills prerequisites.”</i>	[78]

Um resumo de diferentes tipologias e papéis da participação dos usuários em comunidades on-line no que se refere na gestão, contratação, ativação e retenção desses usuários foi discutido por Rohrmeier [81]. Ainda para o autor, mesmo com a diversidade de participação em uma comunidade, é crucial que os participantes forneçam conteúdo suficiente e mantenham uma interação com os demais participantes para preservação da comunidade. Nas comunidades sociais on-line, o valor é inerente à conexão de nós, a comunicação entre eles e a troca e combinação de conhecimento, informação e bens digitais [81]. A participação do usuário em ferramentas de comunicação como fóruns e chats, deve ocorrer sempre de forma a discutir assuntos que agreguem valor às experiências. Assim, percebe-se certa similitude ao comportamento dos trabalhadores em atividades do crowdsourcing.

Conforme Stol e Fitzgerald [20], Ramakrishnan e Srinivasaraghavan [57] e Kittur *et al.* [58], motivação é um dos grandes desafios no apoio a trabalhos em crowdsourcing. Ainda e nesse sentido, Hosseini *et al.* [29] entendem que mecanismos de feedback aos trabalhadores na multidão apoiam a motivação e, por conseguinte, mantem os trabalhadores nas atuais e futuras atividades no crowdsourcing. Brabham [42] e LaToza *et al.* [82] classificam a motivação do trabalhador na multidão em extrínseca e intrínseca. Motivação extrínseca ocorre sempre que uma atividade é feita para alcançar algum resultado tangível, como por exemplo, uma recompensa financeira. Por outro lado, a

motivação intrínseca é definida como a realização de uma atividade por suas satisfações inerentes e intangíveis como por exemplo, executar a tarefa apenas por diversão ou como um desafio [83].

Por fim, e como comentam Fuller [84], Morgan e Wang [85], Leimeister *et al.* [6] e Martinez *et al.* [86] o ponto principal em competições no crowdsourcing é encontrar e atrair a multidão “certa”, e, fundamentalmente, gerenciar a efetiva participação do trabalhador na multidão para a resolução de um problema em particular.

d) **Tarefa:** Além dos três elementos no crowdsourcing, autores como Hosseini *et al.* [29], Rouse [87], Geiger e Schader [88], reconhecem um quarto elemento: a Tarefa.

Conforme Rouse [87] as tarefas podem ser classificadas como de *baixa complexidade*, ou simples, que são aquelas que podem ser realizadas por um trabalhador com baixa demanda cognitiva¹² ou moderado treinamento para executar determinadas tarefas e de “fácil” avaliação. Outras tarefas são consideradas como complexas que exigem alta ou média demanda cognitiva para sua realização além de serem difíceis de avaliar. Algumas tarefas não são nem simples, nem complexas, pois envolvem um moderado nível de complexidade e moderado nível para avaliação, para esses tipos de tarefas, a autora as denomina de *moderada*.

Macro tarefas são consideradas aquelas tarefas que *“levam um pouco mais de tempo, porque são menos automatizadas e exigem mais conhecimento por parte do trabalhador”*. Suzuki e Igarashi [89] contribuem nesse sentido e afirmam que macro tarefas são normalmente executadas por trabalhadores profissionais. Desenvolver um website, criar uma base de dados, ou escrever um código (programação) são consideradas macro tarefas [24].

Entretanto, pequenas tarefas ou micro tarefas são aquelas que não exigem trabalhadores com alta demanda cognitiva, ou aqueles ditos, “profissionais” e podem ser

¹² OECD PISA 2015 - Programa Internacional de Avaliação de Estudantes, a demanda cognitiva é o processo mental necessário à resolução da tarefa de um item, e essa demanda pode ser baixa, média e alta. Os níveis definidos para essa avaliação são: Baixa Realizar um procedimento de uma única etapa, por exemplo, recordação de um fato, termo, princípio ou conceito ou localizar uma única informação em um gráfico ou tabela. Média: Usar e aplicar conhecimento conceitual para descrever ou explicar fenômenos; selecionar procedimentos apropriados envolvendo dois ou mais passos; organizar/apresentar dados, relatórios ou utilizar conjuntos de dados simples ou gráficos. Alta: Analisar informação ou dados complexos; resumir ou avaliar evidências; justificar e argumentar a partir de várias fontes de informação; desenvolver um plano ou sequência de passos para abordar um problema.

submetidas a um número ainda maior de pessoas quando comparadas a macro tarefas. A plataforma para crowdsourcing AMT, é considerada por Ipeirotis [90], Kittur *et al.* [64], e Berinsky *et al.* [91] como uma das mais populares que utiliza o conceito de pequenas tarefas ou HIT. As micro-tarefas se caracterizam por serem atividades que podem ser realizadas em um “pequeno espaço de tempo”, não necessitem de um conhecimento específico (baixa demanda cognitiva) para sua realização e que sejam repetíveis e atômicas conforme Agerfalk *et al.* [92]. Para que isto ocorra, o solicitante normalmente divide os trabalhos em pequenas (micro) tarefas, e as repassa para os trabalhadores na multidão para a sua execução de acordo com LaToza *et al.* [13]. Um exemplo de uma HIT, na AMT, é o reconhecimento e marcação de imagens e a avaliação de qualidade de vídeo como discute Figuerola Salas *et al.* [93]. Após a realização da tarefa – ou um conjunto de tarefas – o trabalhador recebe uma “pequena¹³” compensação financeira [90]. A maior parte das contribuições na AMT é oriunda dos Estados Unidos e Índia [94].

Em fevereiro de 2017, o professor Panos Ipeirotis – autor de diversos artigos sobre crowdsourcing, entre eles, [90], [95], [96], publicou, em seu blog¹⁴, o resultado de um aplicativo denominado *MTurk Tracker* em uma crônica intitulada “*The Decline of Amazon Mechanical Turk*”, onde afirmava que o número de clientes que utilizam o serviço do AMT estava diminuindo consideravelmente ano após ano. Após a publicação desse estudo, curiosamente, a AMT retirou o acesso aos dados do público em geral. Em novo post¹⁵ o professor Ipeirotis fez o seguinte comentário: “*Amazon blocked MTurk tracker after these articles were posted. MTurk Tracker will not come back live*”.

E, comenta ainda

It seems that after years of neglect, Mechanical Turk starts losing its appeal. In our latest measurement, we see Mechanical Turk losing 50% of its requesters in a YoY measurement.

Por fim, ao transformar entradas em saídas utilizando os quatro elementos em um sistema integrado, sinérgico e equilibrado, o modelo crowdsourcing potencializa e incrementa mecanismos de criação de inteligência coletiva.

¹³ Por exemplo, a plataforma AMT disponibilizou uma micro tarefa com recompensa financeira de US\$0,05. Disponível em <https://www.mturk.com/mturk/findhits?match=false> (Acesso em set/17)

¹⁴ <http://www.behind-the-enemy-lines.com/>

¹⁵ <http://www.behind-the-enemy-lines.com/2016/02/the-decline-of-amazon-mechanical-turk.html>

2.2.2. Tipos de Crowdsourcing

Tipificar um tema emergente como o crowdsourcing requer precaução. Existem diferentes tipos que utilizam o conceito de colaboração oriundo do crowdsourcing, desde o financiamento coletivo até a criação de conteúdo pela multidão.

Conforme Branquinho [27], foram encontrados 25 tipos e uma variada nomenclatura e modalidades de crowdsourcing de acordo com a aplicação. Os tipos encontrados pela autora foram: *“broadcast search”, “crowd creation”, “crowd creativity”, “crowdcasting”, “crowd collaboration”, “crowd competition”, “crowd innovation”, “crowdfunding”, “crowd labor”, “crowd manufacturing”, “crowd marketing”, “crowd project”, “crowdstartup”, “crowdstorm”, “crowd tuning”, “crowd voting”, “crowd wisdom”, “distributed knowledge”, “idea jam”, “external crowdsourcing”, “internal crowdsourcing”, “intra-corporate crowdsourcing”, “prediction market”, “problemsourcing”, “reversed crowdsourcing”*. Percebe-se uma preocupação em designar na formação do nome, em geral, a palavra *“crowd”* adicionada da *“modalidade”* ou *“aplicação”* da tarefa pela multidão, por exemplo: *“crowd creation”* é, em tradução livre igual a *“criação da multidão”*.

Um dos principais sites¹⁶ de difusão do modelo do crowdsourcing divulgou, na segunda versão de um relatório técnico, as sete categorias que representam as diferentes aplicações funcionais de crowdsourcing a seguir destacadas: Inovação aberta, Construindo a comunidade, Criatividade coletiva, Engajamento cívico, Conhecimento coletivo, *Crowdfunding*, e, por fim, a Nuvem de Trabalho. A Inovação aberta caracteriza-se pelo uso de fontes externas para gerar, desenvolver e implementar ideias. Construindo a comunidade é o desenvolvimento das comunidades por meio do engajamento ativo de indivíduos que compartilham interesses em comum. A Criatividade coletiva objetiva reunir talentos criativos para projetar e desenvolver conteúdo ou mídia original. O Engajamento Cívico trata de questões de interesse público. O Conhecimento coletivo é o desenvolvimento de ativos de conhecimento ou recursos de informação de um grupo distribuído de contribuintes. O *Crowdfunding*, é conhecido como financiamento coletivo. Por fim, a Nuvem de Trabalho, que por meio de uma força de trabalho distribuída e disponível *on-demand*, é capaz de cumprir determinadas tarefas.

¹⁶ <http://www.crowdsourcing.org/>

A conferência - HCOMP¹⁷ elaborou 14 tópicos de interesse em crowdsourcing, destacando: “*citizen science*”, “*collective action*”, “*collective knowledge*”, “*crowdsourcing contests*”, “*crowd creativity*”, “*crowd funding*”, “*crowd ideation*”, “*crowd sensing*”, “*distributed work*”, “*freelancer economy*”, “*open innovation*”, “*microtasks*”, “*prediction markets*”, “*wisdom of crowds*”, entre outros. Apesar da diversidade de nomes, essas classificações permitem distinguir para que tipo de plataforma delegar os projetos, pois, nem todos os mercados de ideias fornecem os mesmos serviços, cada plataforma está focada em uma área particular, e, portanto, pode oferecer um serviço adequado ao tipo de público específico. Trabalho realizado por Green *et al.* [48] pela Comissão Europeia categorizou o trabalho da multidão como: a) Crowdsourcing para financiamento; b) Crowdsourcing para trabalho pago e, c) Crowdsourcing para trabalho voluntário.

Por exemplo, o *Crowdfunding*, também conhecido como financiamento coletivo, é um modelo de colaboração cujo objetivo é arrecadar fundos de uma multidão. “*O crowdfunding mobiliza o bolso coletivo, permitindo que as pessoas financiem os projetos em que acreditam, com pequenas doações*” [16]. O modelo de *crowdfunding* baseia-se no conceito em que várias pessoas contribuem financeiramente para viabilizar uma ideia, uma causa, um negócio ou um projeto no qual a pessoa se identifique. Aos apoiadores do projeto, é oferecida uma recompensa que geralmente varia de acordo com o valor doado. Resumindo, o autor apresenta a ideia do negócio ou do projeto, reporta sua proposta e especifica o quanto necessita, ou espera, obter. Assim, por meio da plataforma, indivíduos que se interessarem pela causa fazem doações.

Outro exemplo é o modelo *crowdcasting*, cujo objetivo é construir uma rede de usuários, para então, entregar desafios ou tarefas para serem resolvidas com o propósito de se extrair conhecimento ou ideias inovadoras. Esses desafios, geralmente, ocorrem em formato de uma competição. “*Divulgar um problema ao maior número de pessoas possível, na esperança de que, em algum lugar, alguém, apareça com uma solução*” [16]. O *crowdcasting* senta-se na perspectiva de que, ao tornar público um problema, as chances de encontrar uma solução aumentam na mesma proporção do tamanho do grupo para o qual foi divulgado.

Existem aspectos positivos e negativos no crowdsourcing como apresentado em Olson e Rosacker [72]. Críticas a baixa remuneração é apontada por diversos autores,

¹⁷ <http://www.humancomputation.com/2016/>

como Silberman *et al.* [97], Postigo [98] e Wexler [99] que o consideram uma forma de exploração de trabalho via internet. Por outro lado, Busarovs [100] mostra que os usuários trabalhadores no crowdsourcing não se sentem explorados desde que a plataforma ofereça mecanismos de transparência e algum tipo de compensação, financeira ou não. Sublinha-se que no crowdsourcing competitivo a remuneração ao final do trabalho realizado é caracterizada como importante incentivo extrínseco ao trabalhador na multidão como indica Schmid-Druner [23].

Algumas organizações têm receio em delegar tarefas a pessoas externas ao seu ambiente, onde as negociações se realizam basicamente, e, apenas, por meios virtuais como relatam Kaganer *et al.* [18]. Devido a isso, as organizações procuram utilizar o modelo crowdsourcing apenas para a realização de projetos que possuam baixo orçamento, sem cronograma para sua realização e fraco acoplamento com o negócio. Portanto, o crowdsourcing é uma maneira diferenciada de como a multidão, normalmente online, oferta serviços, ideias ou conteúdo para um determinado fim de um solicitante. Assim, desenvolver software pode ser considerado como uma possibilidade no uso do modelo crowdsourcing.

2.2.3. Software crowdsourcing

Uma das possibilidades de aplicação do crowdsourcing é no desenvolvimento de software. Software crowdsourcing conforme Mao *et al.* [47], é o

ato de transferir, externamente, qualquer tarefa no processo de desenvolvimento de software a um grande grupo indefinido de trabalhadores online em formato de chamada pública (tradução do autor)

Importante observar que um processo de desenvolvimento de software é um conjunto de atividades ordenadas e relacionadas, que envolvem restrições e recursos necessários para que, ao final da execução das atividades seja obtido um software [101]. Ainda conforme o autor, as atividades de “Especificação de software”, “Desenvolvimento de software”, “Validação de Software” e “Evolução do Software”, são comuns a todos os modelos de processo de software, sejam dirigidos a planos ou ágeis.

As primeiras iniciativas da utilização de trabalhadores na multidão para a realização de atividades presentes nos processos de desenvolvimento de software surgiram a partir das comunidades de software livre. São exemplos dessas iniciativas o desenvolvimento do

Sistema Operacional Linux, o Apache¹⁸, o Rails¹⁹ e o Firefox²⁰ que foram feitos por diversas pessoas dispersas geograficamente de forma colaborativa e sem recompensa financeira. Esse exemplo evidencia que é possível a criação de software por meio da multidão.

O uso de terceirização de etapas do processo de desenvolvimento de software é realizado há algum tempo como destacam Audy e Prikladnicki [50], todavia o uso de estratégias baseadas no uso de crowdsourcing para o desenvolvimento de software é emergente como apontam Stol e Fitzgerald [20] e Wu *et al.* [67]. Por meio da realização de uma chamada pública em plataformas disponíveis na internet, pode-se obter a participação de trabalhadores na multidão neste processo.

Usui e Morisaki [73] propuseram uma abordagem para dividir software em muitas partes e terceirizar cada parte para a multidão por meio da Internet. Essa abordagem permite a terceirização e sugere que cada parte desenvolvida do software seja testada por meio de casos de teste. Seguindo essa abordagem Li *et al.* [102] comentam que

essentially, software crowdsourcing is a distributed and open software development paradigm with the basis of online labor market and developer community.

De acordo com Stol e Fitzgerald [20] projetos de software crowdsourcing seguem essencialmente o ciclo de vida Cascata²¹. Por outro lado, outros autores como We *et al.* [67] comentam que o crowdsourcing pode ser incorporado a outros processos de desenvolvimento de software sejam eles dirigidos a planos (o modelo Cascata) ou adaptativos (os Métodos Ágeis).

Os ciclos de vida ditos tradicionais são inadequados para a produção em massa e pela própria natureza do crowdsourcing, por isso, propuseram o modelo denominado de Metropolis para o desenvolvimento de software por meio do crowdsourcing argumentam Kazman e Chen [103]. O modelo Metropolis apresenta três tipos de papéis. A plataforma chamada de *kernal*, as aplicações construídas na *kernal*, (referindo-se como *perifery*) e os usuários finais, conhecidos como *masses*. Sete princípios do modelo são introduzidos para gerenciar o processo de projetos de software crowdsourcing. Os autores alertam que sejam

¹⁸ <https://www.apache.org/>

¹⁹ <http://rubyonrails.org/>

²⁰ <https://www.mozilla.org/pt-BR/firefox/new/>

²¹ Cascata é o nome do modelo que desenvolve software de forma sequencial, seguindo as fases da análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de software. [48]

definidas diretrizes que apontem em quais etapas do processo de desenvolvimento o crowdsourcing pode ser utilizado, bem como, e, em que parte do software pode ser desenvolvido utilizando-se dessa abordagem. Com o uso de crowdsourcing, as tarefas deixam de ser realizadas somente pelos colaboradores da organização, ou por empresas contratadas, e passam a ser executadas por qualquer trabalhador na multidão [104], [105]. Com o uso de recompensas financeiras e de mecanismos de reputação, há o incentivo para que a multidão participe de todo o processo de desenvolvimento do software. O número de participantes em um projeto realizado por meio do crowdsourcing varia de acordo com sua natureza e com as recompensas envolvidas [67]. Desta forma, surge um novo modelo de desenvolvimento de software denominado de software crowdsourcing (SWCS) [67], [15].

Apesar de recente, foram identificadas algumas definições do termo “software crowdsourcing”. A Tabela 3 apresenta as 8 definições, em inglês, dos trabalhos selecionados por Mao *et al.* [47].

Tabela 3: Definições de software crowdsourcing

Definição	Referência
<i>“The way of organizing software development in all these practices of software crowdsourcing is changing from traditional software factory or open source teams to decentralized, peer-production based ecosystems of software developers.”</i>	[15]
<i>“Crowd Development is a development process designed for transient workers of varying skill. ...Is a development process that organizes work into microtasks...”</i>	[13]
<i>“Crowdsourced software development utilizes an open call format to attract geographically distributed developers to accomplish various types of software development tasks”</i>	[25]
<i>“The accomplishment of specified software development tasks on behalf of an organization by a large and typically undefined group of external people with the requisite specialist knowledge through an open call.”</i>	[20]
<i>“To denote the applications of crowdsourcing techniques to support software development (in its broadest sense).”</i>	[47]
<i>“Software crowdsourcing practices blur the distinction between users and developers, and follow the cocreation principle — that is, a regular user becomes a codesigner, codeveloper, and comaintainer. This is a paradigm shift from conventional industrial software development, with developers distinct from users, to a crowdsourcing-based peer-production software development, in which many users can participate.”</i>	[67]
<i>“... software development via crowdsourcing, which has the IT artifact in its core... The model of crowdsourced software development that we have chosen and is more common in practice is of the latter form in which participants compete against each other by submitting pieces of work (design, architecture, code, bug report, etc.).”</i>	[105]
<i>“SW CS has become a viable development paradigm for Software-as-a-Service (SaaS) ecosystems.”</i>	[106]

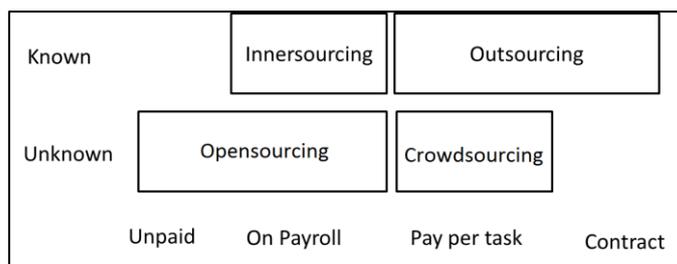
Dentre as citações apresentadas na Tabela 3, percebe-se a preocupação dos autores na manutenção dos princípios do crowdsourcing como, o trabalho executado por meio de uma chamada pública à um grupo indefinido de pessoas conectadas a internet além, é claro, de evidenciar as principais atividades no processo de desenvolvimento de software.

2.2.4. Software crowdsourcing e suas propriedades

A origem do software crowdsourcing remete ao desenvolvimento de software livre e ao desenvolvimento distribuído de software, herdando algumas características desses modelos e incorporando novas propriedades. Ou seja, software crowdsourcing é um caso extremo de desenvolvimento distribuído de software com a aplicação de vários princípios de software livre em empresas de desenvolvimento de software de acordo com Howe [16]. Olson e Rosacker [72] demonstram também que existe uma forte relação entre software livre e crowdsourcing visto que, ainda segundo os autores “*Simply stated, Open Source Software (OSS) is crowdsourcing applied to software development*”. Dessa maneira, se faz necessário reforçar e apresentar os conceitos advindos desse novo modelo, e compará-los com os anteriores, com o intuito de mitigar qualquer dúvida e minimizar certa confusão de conceitos e entendimentos que, naturalmente surgem.

Nesse sentido Agerfalk *et al.* [92] apresentam algumas propriedades do modelo de projetos de software crowdsourcing no que se refere a estratégia ou abordagem do negócio (“*innersourcing*” e “*outsourcing*”), na transferência de valor ao negócio, na forma de pagamento (“*Unpaid*”, “*On Payroll*”, “*Pay per tasks*” e “*contract*”), e “o quanto se conhece” da multidão (“*knownness*”).

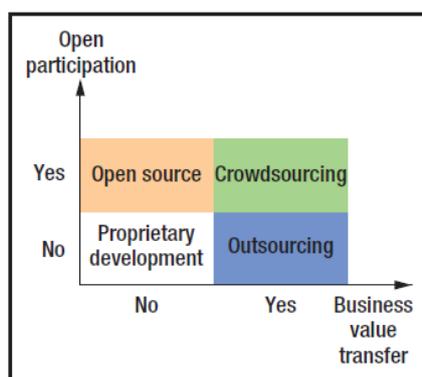
Como pode ser observado na Figura 4, sobre a dimensão “o quanto se conhece a multidão”, na abordagem *innersourcing* e *outsourcing* os executores das tarefas são “conhecidos”, diferentemente do que ocorre no *Opensourcing* e *crowdsourcing*, onde os executores são “desconhecidos”. Ou seja, no modelo crowdsourcing não se conhece a multidão por parte do cliente, apenas os gestores da plataforma de crowdsourcing, contratado para tal, pode ter acesso aos dados privados dos trabalhadores. Na dimensão sobre a “forma de pagamento”, destaca-se que no crowdsourcing o pagamento é feito por tarefa e não por contrato como ocorre no modelo *outsourcing*.



Fonte: Agerfalk *et al.* [92]

Figura 4: Estratégias de desenvolvimento de software

As diferenças entre as estratégias de desenvolvimento de software nas dimensões de “participação aberta” e “transferência de valor ao negócio” nas perspectivas de software livre, Crowdsourcing, *Outsourcing* e *Desenvolvimento proprietário (in-house)* são apresentadas por Peng *et al.* [54]. Essas diferenças podem ser observadas na Figura 5.



Fonte: [54]

Figura 5: Diferenças entre as estratégias

Percebe-se, ao analisar a Figura 5, que apenas nos modelos OSS e Crowdsourcing ocorre a possibilidade de chamada pública para a participação de eventos, onde exclusivamente o crowdsourcing oportuniza a transferência de valor de negócio entre os envolvidos. No desenvolvimento proprietário não há a possibilidade de chamada, tampouco, transfere valor ao negócio. Ou seja, na perspectiva dos autores o crowdsourcing é indicado quando da utilização das estratégias de desenvolvimento de software nas dimensões de “participação aberta” e “transferência de valor ao negócio”.

Ademais e pela semelhança entre OSS e SWCS como comentam Olson e Rosacker [72] e [105], existem algumas diferenças que podem ser observadas na Tabela 4.

Tabela 4: Diferença entre OSS e SWCS no modelo competitivo

Característica	Software livre	Software crowdsourcing
Participação	Voluntária	Coletiva
Pagamento	Normalmente não	Sim
Motivação	Identidade social	Renda
Recompensa	Status	Social
Estabilidade	Estável	Transitória
Prazo	Longo	Curto
Propriedade Intelectual	Usuário	Cliente
Solicitação dos projetos	Independente	Cliente

No OSS os aspectos como participação voluntária e de longo prazo diferem em muito das características competitivas, de curto prazo e transitórias do modelo crowdsourcing. Não obstante, a essência dos dois modelos em ter a multidão como fonte de inteligência coletiva é mantida.

Assim, Agerfalk *et al.* [92] discutem e apresentam o posicionamento de software crowdsourcing quando comparada a sua definição com *In-house*, *Global Development*, *OSS* e *Outsourcing*. A Tabela 5 apresenta algumas definições.

Tabela 5: Posicionamento de SWCS diante de outros modelos.

Modelo	Definição	Referência
In-house	<i>Software that is developed by its only corporate user, or one of its major corporate users.</i>	[107]
Global Development	<i>is carried out by teams of knowledge workers located in various parts of the globe developing commercially viable software for a company</i>	[49]
OSS	<i>Is understood as a kind of production that involves allowing access to the essential elements of a product to anyone for the purpose of collaborative improvement to the existing product</i>	[108]
Outsourcing	<i>a mean of procuring from external suppliers services or products that are normally part of organization outsourcing task usually performed by people closely connected with an institution to a 'crowd' or people outside the institution.</i>	[109]
Crowdsourcing ²²	<i>the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call.</i>	[5], [12]

²² http://dec.bournemouth.ac.uk/staff/rali/Supplementary/RCIS_Crowdsourcing_Supplementary_Material.pdf

2.2.5. Motivação para Software Crowdsourcing

Ao optar por projetos de software no modelo crowdsourcing é fundamental que o cliente ou solicitante, o qual demanda o serviço, defina quais são seus objetivos e que benefícios deseja atingir ao utilizar esse modelo. Dessa forma, Wu *et al.* [15] e [67] enumeram alguns dos objetivos e motivações apontados por organizações que utilizam o software crowdsourcing que são: qualidade do software, aquisição rápida, identificação de talentos, redução de custos, soluções diversificadas, criação de ideias, ampliar a participação, educação dos participantes e marketing. A seguir, uma breve descrição de cada um dos objetivos e motivações.

- **Qualidade do software:** obter um software de qualidade é um objetivo comum para qualquer projeto, inclusive em projetos de software crowdsourcing. Em virtude disso, os organizadores devem definir metas e objetivos bem específicos a serem atingidos em relação a confiabilidade, o desempenho, a segurança, a manutenção e a usabilidade do software, e, principalmente, quais serão os seus critérios de avaliação.

- **Aquisição rápida:** ao invés de desenvolver internamente o software, as plataformas de crowdsourcing podem postar uma competição e aguardar algo semelhante que já tenha sido desenvolvido. Isso acarreta em uma redução do tempo de entrega do software.

- **Identificação de talentos:** um cliente pode estar interessado em identificar talentos. Para isso, lança um desafio cujo objetivo é, estritamente, encontrar desenvolvedores de acordo com o perfil desejado. Esse é um fator determinante que faz com que, por exemplo, programadores de computadores participem ativamente de competições crowdsourcing. Porém, as plataformas dificultam o contato direto dos clientes com os desenvolvedores.

- **Redução de custos:** a aquisição de um software por um custo menor é um objetivo comum para todos os clientes, independente do processo utilizado para desenvolvê-lo. Contudo, além de pagar aos vencedores é necessário pagar a plataforma. O objetivo é minimizar os custos do software crowdsourcing, incluindo a organização, a estimativa de custo e o preço a ser pago. Este preço pode consistir tanto de recompensas monetárias ou apenas no reconhecimento do trabalhador.

- **Soluções diversificadas:** é possível obter diversas soluções de um mesmo software por meio de um software crowdsourcing no modelo competitivo. A diversidade de soluções geradas pelos concorrentes, pode ser utilizada para melhorar a confiabilidade no sistema. Schenk e Guittard [69] reforçam que é importante ter competidores de diferentes regiões e com formação e habilidades profissionais diversificadas para trabalhar em conjunto e produzir software de qualidade, favorecendo, com isso, a diversidade de soluções.

- **Criação de Ideias:** em algumas competições, o objetivo é apenas obter novas ideias, na esperança de que no futuro, essas ideias venham a se tornar soluções reais. Nesse caso, o problema abordado pode ser muito desafiador, e assim, os organizadores não estão muito interessados nas soluções apresentadas, e sim nas prováveis brilhantes ideias dessas soluções que podem gerar novas oportunidades de negócios.

- **Ampliar a participação:** algumas competições de crowdsourcing tem por objetivo recrutar o maior número possível de participantes. Um dos motivos para isso, é na obtenção de soluções de qualidade, pois o organizador pode selecionar a melhor solução entre todas as enviadas.

- **Educação de participantes:** em competições crowdsourcing, os organizadores, algumas vezes, não estão interessados nas soluções apresentadas, ou na identificação de talentos, mas sim, ensinar a esses novos competidores conhecimentos ou qualificações através dessas competições.

- **Marketing:** algumas organizações podem realizar competições utilizando o modelo crowdsourcing apenas como uma ação de marketing, isto é, gerar, por exemplo, um aumento da publicidade da sua organização, fazendo, com isso, que muitos participantes reconheçam o nome da organização e sua natureza de seus negócios.

Autores como Wu *et al.* [15], [67] apresentam também alguns “alvos” de projetos de software no modelo crowdsourcing categorizados conforme o objetivo que as organizações desejam atingir ao propuserem tarefas por meio desse modelo. Por exemplo, ao solicitar uma tarefa a organização pode eleger como “alvo” o participante, pois deseja identificar talentos ou criar uma comunidade sobre algum determinado tema. A Tabela 6 resume alguns dos alvos de projetos de software crowdsourcing.

Tabela 6: Alvos do software crowdsourcing.

Alvo	Fundamentação
Soluções submetidas	<ol style="list-style-type: none"> 1. Obter soluções de qualidade a um custo baixo. 2. Adquirir soluções rapidamente. 3. Obter ideias nas soluções.
Participantes	<ol style="list-style-type: none"> 1. Identificar talentos. 2. Realizar os projetos ou organizações reconhecidas. 3. Criar uma comunidade de usuários com certo conhecimento. 4. Treinar os participantes com determinados conhecimentos.
Observadores	<ol style="list-style-type: none"> 1. Torná-los cientes das organizações crowdsourcing. 2. Motivá-los a financiar projetos semelhantes.

Pelo exposto, percebe-se que as motivações para projetos de software crowdsourcing reforçam e estão alinhados com os principais objetivos do crowdsourcing como um todo, no intuito de reduzir os custos de produção, acelerar o *“time to market”* e ampliar as opções de soluções alternativas para o desenvolvimento de software. Nota-se, também, uma preocupação na criação de mecanismos capazes de treinar e manter uma comunidade on-line ativa e apta com talentos capazes de realizar determinadas tarefas, visto que desenvolver software exige habilidade, demanda cognitiva e criatividade dos indivíduos.

2.2.6. Projetos de software crowdsourcing mediados por plataformas

Projetos de software crowdsourcing somente podem ser executados a partir da presença dos quatro elementos fundamentais do crowdsourcing, isto é, o cliente, a plataforma, a multidão e a tarefa. Para que a plataforma execute seu papel no processo é fundamental que possua processos adaptados ou específicos para o desenvolvimento de software. Esses processos devem abordar, em algum momento, as fases no processo de desenvolvimento de software, entre as quais, destacam-se a “Especificação de software”, “Desenvolvimento de software”, “Validação de Software” e, por fim, a “Evolução do Software”.

A Figura 6 mostra um modelo genérico de um típico projeto de software crowdsourcing onde todo o processamento, desde a entrada até a saída, é gerenciado por alguma plataforma. Este modelo, de maneira ampla, pode ser descrito dessa forma: o cliente solicita à plataforma o desenvolvimento de um software, a plataforma organiza as tarefas e as atividades para o desenvolvimento do software e as delega para o trabalhador

na multidão executá-las, a plataforma gerencia esse processo e entrega o software pedido ao cliente.

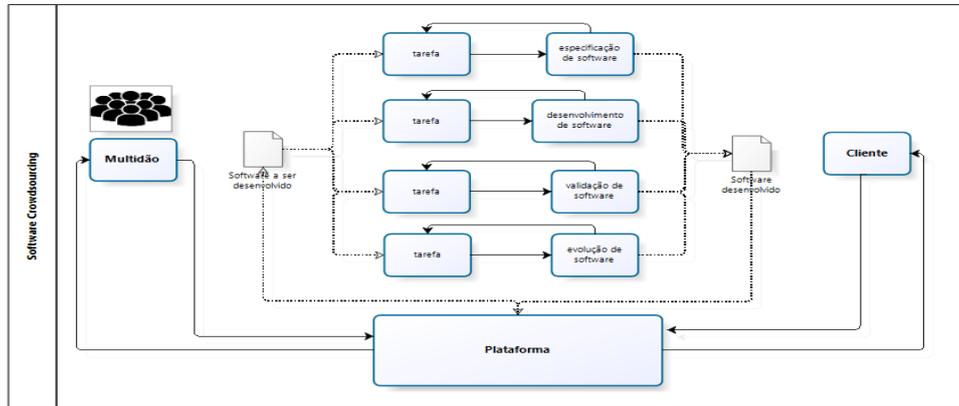


Figura 6: Modelo genérico de software crowdsourcing através de plataformas.

Existem plataformas que cobrem todas fases, ou atividades, do ciclo de vida de projetos de software crowdsourcing, apesar disso, há aquelas que possuem apenas uma dessas fases/atividades.

A Tabela 7 apresenta uma relação de algumas plataformas indicando o nome da plataforma, bem como, em qual fase/atividade a plataforma se enquadra no que tange ao desenvolvimento de projetos de software crowdsourcing. Pode ser observado na

Tabela 7, a fase do ciclo de vida “Validação de software” que contém a atividade de testes de software é aquela que possui o maior número de plataformas quando comparada a outras fases/atividades. Isso demonstra, de certa forma, que a área de testes de software crowdsourcing tem recebido atenção tanto pela comunidade científica, quanto pelos proprietários de plataformas de testes de software [110].

Tabela 7: Plataformas para projetos de software crowdsourcing

Nome da Plataforma	Fase ou atividade do Ciclo de Vida do software
Utest	Testes
Passbrains	Testes
99tests	Testes
TestBirds	Testes
BugFinders	Testes
WeDoLogos	Projeto
99Desing	Projeto
DesignCrowd	Projeto
CrowdREquire	Requisitos
CrowdSpring	Projeto
GetACoder	Codificação

Entretanto, existem plataformas que podem executar projetos de software crowdsourcing cobrindo todas as fases/atividades do ciclo de vida de um software. São alguns exemplos dessas plataformas a Topcoder, a UpWork e a AppStori. Estas plataformas são descritas a seguir.

a) Topcoder

Inaugurada em abril de 2001, a plataforma TopCoder²³ é, de acordo com Schenk e Guittard [63], uma das principais plataformas para projetos de software crowdsourcing no mundo, reunindo, profissionais e amadores que realizam tarefas comerciais e que competem entre si visando, também, uma compensação financeira. A Figura 7 mostra a evolução dos usuários cadastrados na Topcoder, com destaque ao forte crescimento a partir do ano de 2012, fato esse que pode ter acontecido após a mudança de gestão e venda da Topcoder, em setembro de 2013, para a Appirio²⁴. Em 2018, Topcoder já consta com aproximadamente 1,2 milhões de membros²⁵. O Facebook, como comenta Li [66], entre tantas outras grandes organizações, por exemplo, é cliente dessa plataforma.

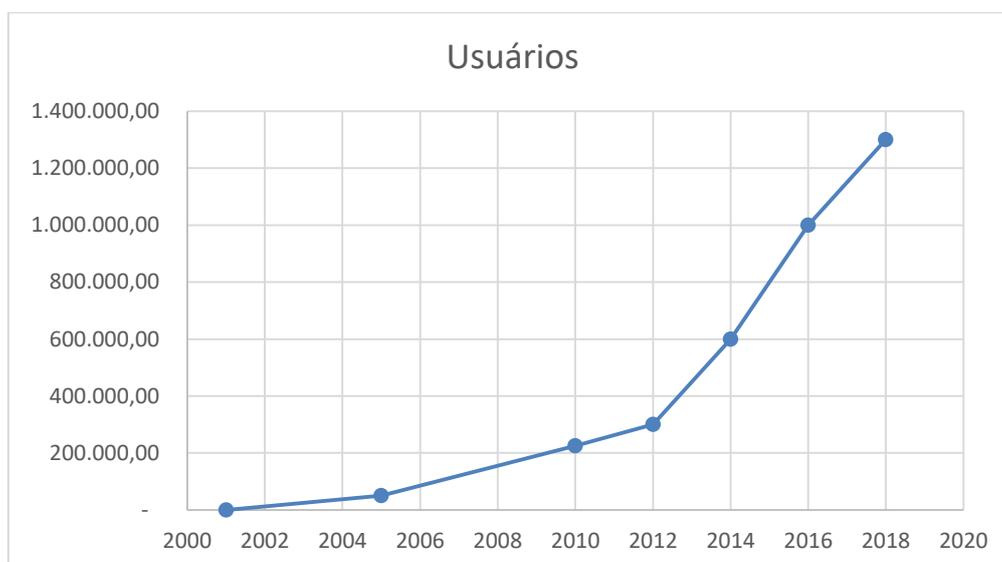


Figura 7: Evolução do crescimento dos usuários cadastrados na Topcoder.

A plataforma Topcoder oferece os serviços organizados em categorias de desafios divididos em: Desafios, Desafios de Desenvolvimento, Desafios de *Data Science* e Programação Competitiva. Quem gerencia a plataforma é a própria proprietária da

²³ <http://www.topcoder.com>

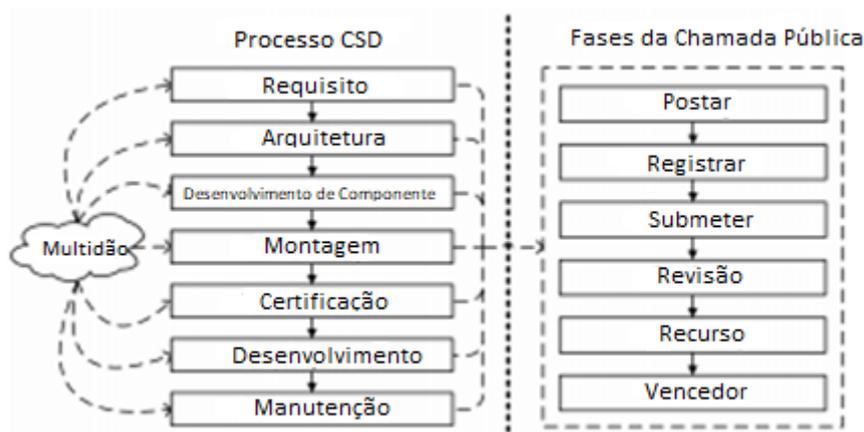
²⁴ <https://appirio.com/>

²⁵ <https://www.deccanherald.com/business/we-are-banking-digital-drive-our-growth-668438.html>

empresa. Informações relevantes como requisitos do sistema, cronograma, orçamento e outras, são negociados pela organização diretamente com o cliente. Cada tarefa possui um escopo e é composta por requisitos técnicos que definem o comportamento esperado daquele componente e as interfaces necessárias para integrar esse componente com as demais partes do sistema. Como no desenvolvimento tradicional de software, o processo no Topcoder de acordo com Lakhani *et al.* [26] abrange todas as fases do ciclo de vida sendo que para cada uma dessas fases do ciclo de vida, uma chamada pública é realizada. A chamada pública é composta por seis tarefas: publicação, registro, submissão, revisão, recurso e vencedor [102].

A Figura 8, apresenta a relação entre o processo e a chamada pública. A seguir, uma breve descrição das seis fases da chamada pública.

1. Publicação do projeto na plataforma, constando como algumas informações como pagamento pela realização do projeto, prazos, entre outros.
2. Registros por parte dos interessados, feito durante o prazo de inscrição definido para este projeto. Após esse período, o candidato deve enviar uma proposta para a implementação da tarefa.
3. Os trabalhadores apresentam a implementação da proposta anterior, conforme o prazo definido.
4. As submissões são avaliadas por revisores. Avançam para etapa seguinte apenas as aprovadas.
5. Se a submissão for aprovada pelos revisores, ela é avaliada para identificar o índice de qualidade das soluções desenvolvidas, seguindo aspectos previamente definidos. Os trabalhadores podem solicitar revisão da avaliação realizada.
6. Após a passagem por estas etapas, o trabalho melhor avaliado recebe o pagamento. O segundo colocado pode, e depende da competição, receber um percentual do valor definido para o projeto. O cliente recebe apenas a melhor solução.



Fonte: [25]

Figura 8: Etapas e fases do TopCoder

Merece destaque o desenvolvimento de software pela plataforma Topcoder, pois, conforme Howe [16] p. 11 “Os programas Topcoder têm uma média de 0,98 bug a cada mil linhas de código. O padrão do setor é 6”. O autor comenta ainda sobre a manifestação de um cliente da plataforma: “Uma empresa tradicional teria designado seis ou sete desenvolvedores para o projeto e levaria mais de um ano para concluí-lo. Nós terminamos em pouco mais de cinco meses”. Dadas as características apresentadas, a plataforma Topcoder coloca-se, atualmente, como principal referência em projetos de software crowdsourcing, como destacam Yang *et al.* [35].

b) UpWork

A plataforma de crowdsourcing UpWork, anteriormente conhecida como “oDesk” pode ser utilizada para o desenvolvimento de software além de outras tarefas de designers, de tradução, de profissionais de marketing, de vendas e finanças. O funcionamento básico da plataforma inicia com a publicação de um projeto com a descrição do trabalho, objetivos, resultados esperados, habilidades desejadas e o prazo de entrega. O responsável do projeto recebe inscrições de candidatos qualificados para a realização da tarefa. Os candidatos, previamente cadastrados, possuem um perfil com informações pessoais e profissionais, como histórico de trabalho, habilidades, feedbacks de clientes, competências linguísticas e de comunicação, valor da hora trabalhada, total de projetos trabalhados, e, total de horas trabalhadas.

Na plataforma, avaliações, pontuações e comentários de outros solicitantes, auxiliam no processo de seleção dos candidatos para a realização da tarefa. Além disso, uma

ferramenta da plataforma monitora os trabalhadores a cada dez minutos. A plataforma paga os clientes apenas por trabalhos aprovados. No momento a plataforma recebe cerca 10% de cada pagamento feito ao trabalhador. Não foi possível identificar, em detalhes, o processo de desenvolvimento de software utilizado pela plataforma, todavia, conforme indicado no *site*²⁶ da plataforma, todas as atividades de desenvolvimento de software são oferecidas com destaque.

c) AppStori

A plataforma de crowdsourcing AppStori pode ser utilizada para desenvolvimento de software bem como para *Crowdfunding*. Essa plataforma tem como objetivo possibilitar o financiamento de uma proposta e estimular a discussão e o surgimento de ideias para novos aplicativos para smartphones e futura disponibilização na loja de aplicativos *Apple Store*.

A Figura 9, adaptada de Wu *et al.* [15], descreve o funcionamento da plataforma.

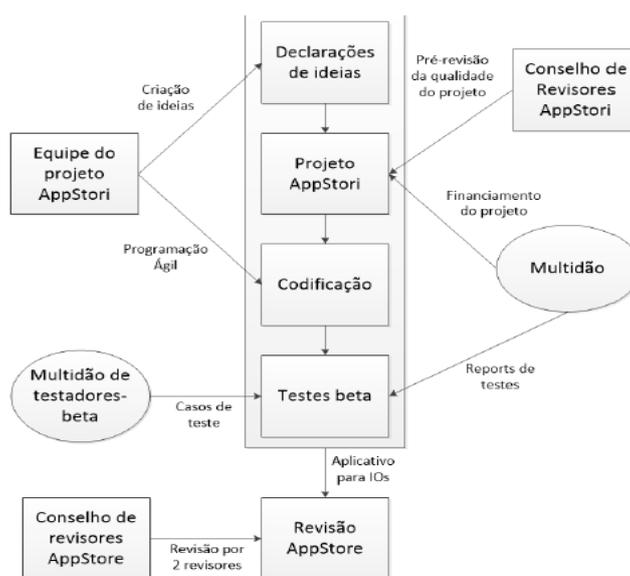


Figura 9: Processo de Desenvolvimento da AppStori.

Basicamente, o processo de desenvolvimento na AppStori inicia com a criação de uma proposta de projeto. Essa proposta é composta pela especificação dos recursos do aplicativo, bem como a receita que se espera após a conclusão do projeto. O orçamento do projeto. Esta proposta será revisada e avaliada por um conselho de revisores, cujo objetivo

²⁶ <http://www.upwork.com>

é garantir a qualidade dos projetos disponibilizados. Após esse conselho, definirá se a proposta tornar-se-á um projeto, ou não.

Caso seja aprovado, o projeto é disponibilizado na plataforma para ser financiado. Após a equipe faz a divulgação do projeto para atrair os recursos financeiros da multidão. A plataforma oferece ferramentas de gestão do projeto. Se o projeto atingir o valor solicitado, 7% do total da arrecadação é destinado para a plataforma. Segundo Wu *et al.* [15] a AppStori diferencia-se das demais plataformas crowdsourcing, pelo fato de que não há competição por parte dos participantes. Nessa plataforma, a equipe de desenvolvimento é responsável pela mobilização da multidão e arrecadar os fundos necessários para o financiamento do projeto.

2.2.7. Desafios do Software Crowdsourcing

Os desafios do desenvolvimento distribuído de software foram apresentados por Audy e Prikladnicki [50], enquanto Steinmacher [32] mostrou os desafios em projetos de software livre. Devido a semelhança entre essas duas abordagens e crowdsourcing como comentam Olson e Rosacker [72], em projetos de software crowdsourcing também possuem desafios, alguns deles herdados do DDS e OSS. Existe consenso na literatura sobre os principais desafios encontrados em projetos em software crowdsourcing. Os autores como Mao *et al.* [14], LaToza *et al.* [13], LaToza *et al.* [21], Peng *et al.* [54], Mao *et al.* [47], [102], [111] e Lykourantzou *et al.* [112] destacam os desafios estruturados em seis áreas a seguir descritas:

- 1) **Decomposição de tarefas.** O trabalho decomposto em um conjunto de tarefas significativamente menores (micro tarefas) é uma questão chave, ao definir um cenário de terceirização. Tarefas de desenvolvimento de software usando o crowdsourcing geralmente dependem umas das outras, isso se torna um problema levando em conta que diferentes desenvolvedores executam as suas tarefas sem saber como elas serão integradas ao produto final, que é resultante da soma de todas as submissões. Em se tratando de desenvolvimento de software, e que as tarefas sejam mais complexas e dependentes umas das outras, o desafio é encontrar uma decomposição adequada do produto de software em tarefas que possam ser enviadas à multidão. Decompor o projeto em pequenos módulos, possuir requisitos “claros” e limitar a dependência das tarefas pode definir o sucesso do projeto.

2) **Coordenação e comunicação.** Ao tratar de tarefas complexas usando crowdsourcing, como é o caso do desenvolvimento de software, a comunicação e coordenação das tarefas tornam-se um fator determinante para o sucesso do projeto.

3) **Planejamento e programação.** Ao delegar uma tarefa para uma força de trabalho desconhecida, a organização pode perder o controle do andamento da tarefa. Se por um lado isso é interessante pois a tarefa pode ser concluída em um período de tempo menor, por outro, pode causar uma incerteza em relação da tarefa ser entregue a tempo. Uma das características do crowdsourcing é a agilidade na resolução de uma tarefa, entretanto isso não garante que essa tarefa será concluída dentro do prazo determinado, uma vez que os trabalhadores com conhecimento e domínio específico, nem sempre estão disponíveis, quando se necessita. É importante garantir que foi atribuído tempo suficiente aos executores da tarefa, e esse planejamento é feito com base no tamanho e extensão da tarefa. Projetos extensos e de longa duração podem resultar em um desinteresse da multidão em realizar a tarefa acarretando, com isso, em um número menor de submissões.

4) **Garantia da qualidade.** Como o cliente não tem o conhecimento dos desenvolvedores que fornecem o software e nem do processo que eles utilizam, garantir a qualidade da solução desenvolvida surge como uma grande preocupação para o projeto de software crowdsourcing. Pelo fato da multidão ser formada também por amadores, existe a possibilidade do surgimento de solução com baixa qualidade. Para amenizar essa baixa qualidade da solução, se faz necessário mobilizar o maior número possível de concorrentes para participar. *“Ainda há dúvidas sobre a qualidade das respostas dos participantes, especialmente quando os desafios são relacionados a questões científicas e tecnológicas para inovação”* [27]. Por exemplo, 30% ou mais de submissões na AMT podem ser de baixa qualidade de acordo com Kittur *et al.* [64], [113].

5) **Conhecimento e Propriedade intelectual.** O vazamento de dados confidenciais ou passíveis de propriedade intelectual é uma das preocupações de um cliente ao usar o crowdsourcing. Esse vazamento pode causar uma perda em relação a vantagem competitiva da organização sobre a sua concorrência. Algumas organizações podem hesitar em fornecer detalhes sobre uma determinada tarefa, para proteger a sua propriedade intelectual. Porém os detalhes tornam-se fundamentais na especificação para os desenvolvedores entenderem o que a organização está solicitando. Outro risco inerente ao processo crowdsourcing, se deve ao fato de um desenvolvedor apresentar uma solução

que não é sua, por exemplo um código fonte aberto, no entanto, com uma licença de uso restritiva.

6) **Motivação e Remuneração.** Motivação é um tema que tem recebido atenção e é considerada como um fator crítico de sucesso em projetos de software de acordo com Sharp *et al.* [114] e Beecham *et al.* [115]. Os fatores de motivação podem ser intrínsecos, onde o incentivo ocorre por meio de satisfação pessoal, por exemplo, ou extrínsecas que ocorre por meio, por exemplo, de uma recompensa monetária como comentam Stol e Fitzgerald [20], LaToza *et al.* [21], e Ramakrishnan e Srinivasaraghavan [57]. Para atrair e motivar os indivíduos para a realização das tarefas, a remuneração deve ser oferecida de forma significativa pois a execução das tarefas no desenvolvimento de software, muitas vezes, são complexas e demoradas como afirmam Zhu *et al.* [116]. Kinnaird *et al.* [117] comentam que *“Work quality is generally low on such platforms, in part because workers have a fleeting relationship with the employer and low pay”*

2.3. Reflexões sobre o capítulo

Desenvolvimento distribuído de software, software livre e software crowdsourcing possuem muitas semelhanças e desafios. Howe [16] comenta que o crowdsourcing utiliza conceitos e princípios de software livre, sendo um caso extremo de desenvolvimento distribuído de software.

Assim, estas semelhanças favorecem o crescimento do modelo software crowdsourcing, por outro lado, aumentam os desafios no processo de utilização, tanto pelas empresas, quanto pelos trabalhadores da multidão, deste novo modelo. Um novo desafio pode ser acrescentado em projetos de software crowdsourcing. Mesmo motivados e com a perspectiva de recebimento monetário futuro, os trabalhadores na multidão em projetos de software crowdsourcing **enfrentam muitas barreiras para registrarem e submeterem sua tarefa.** Nesta perspectiva tem-se um novo desafio que é encontrar mecanismos que os apoiem a superarem essas barreiras.

Autores como Stol e Fitzgerald [20] reforçam a necessidade de estudos para que sejam identificados os limites do desenvolvimento de software utilizando o modelo crowdsourcing.

3. TRABALHOS RELACIONADOS

A literatura de software crowdsourcing é recente e oferece uma perspectiva sobre os seus quatro elementos fundamentais: o cliente, a plataforma, a multidão e a tarefa, porém, no que tange aos desafios que os trabalhadores na multidão enfrentam ao desenvolverem software no modelo crowdsourcing é escassa. Portanto, a seguir, são apresentados alguns trabalhos relacionados na perspectiva da multidão em software crowdsourcing, bem como, em OSS.

3.1. Trabalhadores em projetos de software crowdsourcing

Sob a luz do elemento multidão Mao *et al.* [25], Vukovic [62], Jayakanthan e Sundararajan [118], Jayakanthan e Sundararajan [119], Varshney [120], dedicam certa atenção à motivação, a atração, a retenção e, por fim, para a qualidade da tarefa produzida [90]. Entretanto, existe uma lacuna de estudos que abordem desafios e problemas que os trabalhadores enfrentam, e como os enfrentam, quando colocados em situação de registro e submissão de tarefas em projetos de software crowdsourcing competitivo.

Desenvolver uma modelagem e um *framework* para a avaliação de feedback dos usuários em tempo real, de modo que seja mais acessível aos usuários se expressarem e aos desenvolvedores entenderem esse feedback foi o objetivo no trabalho realizado por Sherief [121]. Este feedback de usuários são informações significativas dados pelos usuários sobre o software, baseados em suas experiências de uso, reutilizando o conhecimento da multidão, os quais contribuem em identificar problemas no software, modificar requisitos existentes ou solicitar novos requisitos conduzindo, com isso, a melhor aceitação dos usuários de software. Observam que a literatura ainda é limitada no fornecimento de abordagens de engenharia no desenvolvimento de frameworks de crowdsourcing de avaliação de software.

Apresentar o conceito de HITs e seu uso dentro de desenvolvimento no crowdsourcing foram discutidos em LaToza *et al.* [13]. Apresentou uma barreira sobre a participação de atividades de relatório de erros, entre outras atividades que bloqueiam o aumento de desenvolvedores a se tornarem comprometidos com este tipo de trabalho. Os autores ainda apontam algumas técnicas para garantia de qualidade, porém sem um conceito apresentado e demonstra com conceitos sobre alguns benefícios que a multidão pode trazer para o desenvolvimento de software. Apresentam também a garantia de qualidade e a redundância na decomposição de tarefas de modo que múltiplas alternativas

sejam geradas e comparadas. Destaca ainda que testes do tipo A/B poderiam diminuir as barreiras da experimentação de software na engenharia, aumentando a disponibilidade de novas ferramentas e práticas de utilidade.

LaToza *et al.* [82] apresentam uma plataforma, denominada de CrowCode para compartilhamento de conhecimento por meio de um sistema de perguntas e respostas. Os autores afirmam que essa plataforma auxilia no processo de decompor macro tarefas em micro tarefas contribuindo no entendimento da tarefa por parte do trabalhador na multidão, aumentando a participação desses trabalhadores no desenvolvimento de software baseado em software livre. Para avaliar a viabilidade da plataforma, realizou-se um estudo que mostrou que uma pequena multidão de 12 trabalhadores foi capaz de codificar, com sucesso, 480 linhas de código e 61 testes unitários em 14,25 horas / hora.

Difallah *et al.* [122] propuseram um sistema que seleciona quais trabalhadores devem executar uma determinada tarefa tendo como base seus perfis de usuários extraídos de redes sociais. Para isso foi desenvolvido o framework *Pick-a-Crowd* que auxiliam no processo de selecionar, automaticamente, a multidão para determinadas tarefas. Para avaliar a proposta, os autores desenvolveram uma extensão (componente) denominado de "OpenTurk" no navegador Chrome que permite a utilização mais eficiente do gerenciamento de atividades, do tipo HIT, no Amazon MTurk. Resultados iniciais mostram um aumento de produtividade de 29% maior do que quando utilizado apenas com o modelo Amazon MTurk.

O sucesso do desenvolvimento de software crowdsourcing depende de uma confiável multidão de desenvolvedores de software on line que registrarão e enviarão suas tarefas em troca de uma compensação financeira. No trabalho de Yang *et al.* [35], uma análise preliminar, por meio de estudos empíricos, sobre o comportamento de trabalhadores na multidão no desenvolvimento de software revelou uma considerável taxa de desistência de 82,9% desses trabalhadores. Assim, foi proposto uma metodologia de suporte à decisão com base analítica para orientar esses trabalhadores a registrarem e submeterem suas tarefas. A metodologia, denominada de DCS-DS, foi avaliada em dados reais obtidos na plataforma TopCoder. Como resultados, e aplicando conceitos de aprendizado de máquina, foi possível prever um provável desistente com uma precisão média de 99%.

Demonstrar conceitos e preocupações no uso de crowdsourcing como uma estratégia de terceirização para o contexto de desenvolvimento de software foi o trabalho desenvolvido por Stol e Fitzgerald [37] e trouxeram perspectivas e preocupações no uso de

crowdsourcing como alternativa de desenvolvimento de software. Os autores ainda reforçam que é fundamental que os três elementos do modelo crowdsourcing estejam unidos e sincronizados no entendimento real do projeto que o cliente deseja. Como contribuição apresentam alguns desafios e oportunidades em projetos de software crowdsourcing.

Um modelo colaborativo, denominado de Turkomatic, foi apresentado por Kulkarni *et al.* [123] para auxílio no desenvolvimento de tarefas complexas. Os autores relatam que quando a multidão não recebe acompanhamento ou supervisão, as tarefas são executadas parcialmente com qualidade questionável. Entretanto, quando existe acompanhamento a qualidade da tarefa melhora significativamente.

Incorporar trabalhadores na multidão em projetos de software crowdsourcing pode ser sinônimo de aumento de recursos, pois esses trabalhadores precisam familiarizar-se com o domínio da aplicação, com os requisitos e visão do projeto, sua arquitetura e tecnologias associadas. Além disso, se faz necessário conhecer a ferramenta e o processo de desenvolvimento do software, antes do efetivo início do trabalho para que possa contribuir de maneira eficiente no projeto. Esses recursos constituem, então, barreiras no processo de participação que os trabalhadores na multidão devem superar muito antes de contribuírem em novos projetos.

3.2. Trabalhadores em projetos de OSS

Se em projetos de desenvolvimento de software proprietário são escassos os estudos que avaliam as dificuldades que novatos enfrentam em suas contribuições Begel *et al.* [124], Berlin *et al.* [125] e Sim *et al.* [126], em projetos de software livre isso também ocorre. Projetos de software livre utilizam o conceito de software livre o qual significa que um código aberto de software possa ser adaptado para diferentes fins, respeitando quatro liberdades. Segundo a *Free Software Foundation*²⁷ e German [127] a primeira Liberdade refere-se ao ato de executar um software como desejar, para qualquer propósito. Ser capaz de estudar como um software funciona (com acesso ao código-fonte), e adaptá-lo às suas necessidades é a segunda liberdade. A terceira liberdade é poder redistribuir cópias. Por fim, a quarta e última liberdade é poder distribuir cópias das versões modificadas a outros.

²⁷ <http://www.fsf.org/>

Projetos de software livre alcançaram notoriedade e auxiliaram o crescimento de diversos segmentos da sociedade [108], [128], [129]. A maioria desses projetos dependem fortemente de uma competente comunidade de voluntários ativa e constante entrada de novatos para a sua sobrevivência. Contudo, ao realizar sua primeira contribuição, muitos novatos enfrentam barreiras, como por exemplo, problemas de recepção, quebras de expectativa e curva de aprendizagem, entre outros, e, isso, podem levá-los a desistir do projeto.

Para compreender melhor quais são os fatores que constituem as barreiras na primeira contribuição, e, como ela é composta, para, se possível, mitigar seus efeitos, realizou-se um estudo com desenvolvedores de software que deviam modificar o código-fonte de um aplicativo *Free/Libre Open source Software* (FLOSS). Foi solicitado, para identificarem quais foram os fatores que os motivaram, bem como, quais foram as questões que complicaram o processo de contribuição, ou mesmo, que impediram de completá-la. Von Krogh *et al.* [130] descreveram o conceito de uma "barreira na contribuição" em sua análise do projeto Freenet²⁸. Foram encontrados quatro obstáculos específicos que impedem os desenvolvedores novatos no processo de modificar esse código.

Foi no trabalho de Steinmacher *et al.* [129] que os autores identificaram e compreenderam as barreiras que os novatos enfrentaram após realizarem a primeira contribuição em projetos de desenvolvimento de software livre. Apresentaram, também, algumas estratégias para amenizá-las. Foram identificadas 58 barreiras agrupadas em seis categorias, entre elas: diferenças culturais, características dos novatos, problemas de recepção, orientação dos novatos, obstáculos técnicos, e problemas de documentação. A partir dessa constatação desenvolveu-se um portal denominado de FLOSScoach²⁹, para apoiar os novatos em sua primeira contribuição.

Portanto, pode-se concluir que, não foram identificados estudos na literatura que abordam especificamente o tema de barreiras encontradas por trabalhadores em software crowdsourcing no modelo competitivo.

²⁸ <https://freenetproject.org/#services>

²⁹ <http://www.flosscoach.com/>

4. BARREIRAS PARA CONTRIBUIR EM PLATAFORMAS DE SOFTWARE CROWDSOURCING

Para identificar, organizar e compreender as barreiras encontradas por trabalhadores em projetos de software crowdsourcing no modelo competitivo, foram executados vários métodos empíricos, entre eles: o estudo de caso, o estudo de campo, entrevista, questionário, revisão por par, discussão em grupo, anotação e memorando, distribuídos em algumas fases.

Os dados foram analisados a partir da perspectiva da Teoria Fundamentada nos Dados, conforme Strauss e Corbin [44], pois, pretendeu-se compreender a realidade e a atitude de indivíduos, os significados atribuídos às determinadas situações, interações e certas experiências de suas vidas em situações subjetivas do seu dia-a-dia.

4.1. Identificação preliminar das Barreiras e das Recomendações - Fase 1

O objetivo da Fase 1 foi atender o primeiro objetivo específico e auxiliar no processo para responder a primeira questão de pesquisa. Inicialmente, fez uma revisão da literatura, depois, foi realizado estudo qualitativo com três fontes de dados. A realização dos estudos de campo teve como objetivo, além de conhecer o software crowdsourcing e suas plataformas de desenvolvimento, auxiliar no processo de elaboração de um conjunto preliminar de barreiras [68].

Um dos resultados desse estudo pode ser observado em Zanatta *et al.* [131]. O segundo estudo de campo teve como objetivo identificar, entender e descrever as barreiras enfrentadas por trabalhadores na multidão, bem como, elaborar uma lista preliminar de sugestões para que os trabalhadores possam sobrepor as barreiras encontradas. As etapas desenvolvidas na Fase 1 podem ser observadas na

Figura 10.

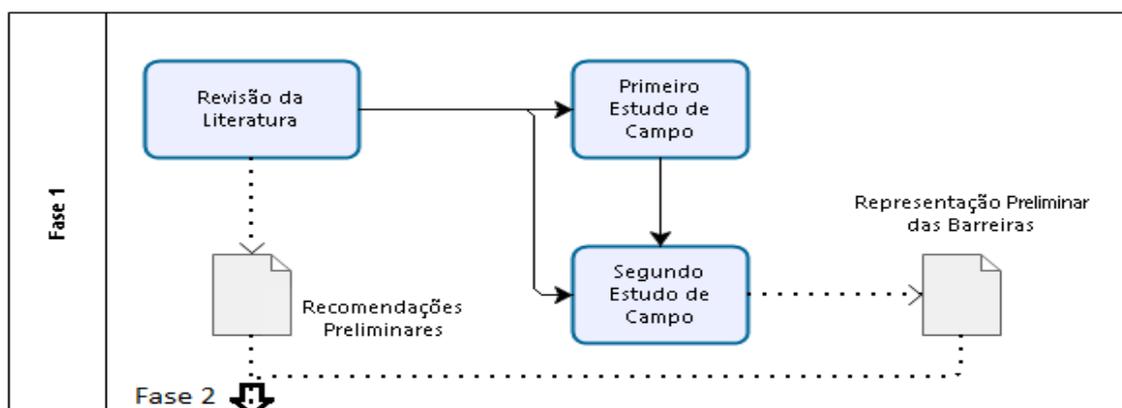


Figura 10: Procedimentos da Fase 1

4.1.1. Revisão da literatura

Essa etapa de natureza exploratória, teve como principal objetivo o aprofundamento no referencial teórico sobre crowdsourcing, software crowdsourcing e seus desafios na perspectiva do trabalhador na multidão, para isso, realizou-se uma revisão da literatura do tipo narrativa em uma base de dados. A revisão da literatura do tipo narrativa caracteriza-se por não exigir um protocolo rígido para sua confecção, sendo frequentemente menos abrangente, e, principalmente, a seleção dos artigos arbitrária. Entretanto esse tipo de revisão de literatura possibilita a interferência da percepção subjetiva [132]. Assim, foi realizada a análise de 213 trabalhos (artigos científicos, teses e dissertações, relatórios técnicos) disponibilizados pelos autores em <https://github.com/Rhapsod/software-crowdsourcing-papers> publicados entre os anos de 2008 e 2016, proposto por Mao *et al.* [47].

Foram esses trabalhos que serviram de base teórica da área de software crowdsourcing, a busca dos desafios em software crowdsourcing, bem como, das recomendações. Na análise realizada no trabalho de Mao *et al.* [47] foi possível identificar alguns, e poucos, desafios presentes nos elementos no modelo software crowdsourcing. Consequentemente, percebeu-se uma carência na literatura analisada de desafios exclusivos da multidão quando participam - em específico no registo e na submissão de tarefas - em projetos de software crowdsourcing.

Vale mencionar que Machado *et al.* [133], cujo trabalho não está presente neste repositório - identificaram os desafios presentes nos quatro elementos do crowdsourcing. Nos desafios encontrados específicos para o elemento Multidão destacam-se: a dificuldade em conhecer um conjunto significativo de tecnologias exigidas para a execução de determinadas tarefas; a falta de domínio de outro idioma, em especial o inglês, para o entendimento, e execução da tarefa, além da colaboração com outros membros da plataforma; o baixo retorno (*feedback*) por parte da plataforma, tanto das tarefas enviadas, como na elucidação de dúvidas e, por fim; a baixa colaboração entre os participantes de atividades de projetos em software crowdsourcing.

4.1.2. Primeiro estudo de campo

O objetivo do primeiro estudo de campo foi elaborar uma lista preliminar de barreiras encontradas por desenvolvedores de software novatos (recém-chegados), quando da

primeira contribuição, em uma plataforma de projetos de software crowdsourcing no modelo competitivo.

A plataforma utilizada para a realização do primeiro estudo de campo foi a Topcoder, por possibilitar o desenvolvimento de software em todas as suas fases, desde a concepção até a evolução, disponibilizar *Application Programming Interface* (API³⁰) com acesso a alguns dados, possuir mais de 1 milhão de usuários cadastrados e ser objeto de vários estudos por parte da comunidade científica de crowdsourcing [26], [68]. A plataforma AppStori foi descartada para o estudo, por não ser caracterizada como crowdsourcing no modelo competitivo. As plataformas CrowdPlat e UpWork não foram selecionadas para o estudo, por não divulgarem o processo de desenvolvimento de software. As demais plataformas não foram selecionadas por não possuírem todas as fases do ciclo de desenvolvimento de software.

A Topcoder possui um sistema de classificação para os usuários calculado por algoritmos específicos, onde, após cada competição são mantidos os dados do usuário como a volatilidade, a avaliação e o número de vezes classificado antes de competir. Conforme a classificação do usuário é designada a ele uma cor para facilitar o entendimento quanto do acesso à plataforma pela comunidade ou os próprios gestores da plataforma. Ou seja, o *handle* do usuário recebe uma cor. Caso o usuário seja classificado entre 0 e 899 atribui-se, a ele, a cor “Cinza” que representa que o usuário está em estágio inicial (novato) e cadastrado a menos de um ano na plataforma. Se o usuário for classificado entre 900-1199 é atribuído a ele a cor “Verde”, a qual indica também que o usuário está em um estágio inicial e submeteu uma solução, entretanto, não obteve vitória em alguma competição. A cor Lilás indica que o usuário é considerado como intermediário, pois obteve vitória em alguma competição. Se o usuário for classificado nas cores “Amarelo” ou “Vermelho” ele é considerado experiente pois já obteve alguma vitória em mais de um tipo de competição na Topcoder. Para o estudo piloto, foram selecionados apenas usuários categorizados como Cinza ou Verde, ou seja, com classificação (rating) até 1.199. O sistema de atribuição de cores aos usuários demonstrou ser eficiente e intuitivo. Recentemente, a Topcoder disponibilizou, em seu site, uma funcionalidade para busca e seleção de usuários com vários filtros para critérios de escolha.

³⁰ <http://api.topcoder.com/v2/users/validateEmail?email=?>

Após a análise do funcionamento e a estrutura da Topcoder, iniciou-se o processo de busca dos usuários cadastrados na plataforma para uma possível entrevista. Para a realização dessa busca, definiu-se como critério de escolha usuários que participaram de uma competição de resolução de algoritmos nos últimos 180 dias (considerando a época), e com sistema de classificação máximo de 1.199 (*Algorithm rating Max*). Após a aplicação destes critérios na funcionalidade de busca da Topcoder foram encontrados 3.403 usuários e a amostra selecionada, com grau de confiança de 95% e margem de erro de 6%, foram selecionadas para a primeira rodada até 271 usuários. Como a plataforma não disponibiliza o e-mail dos usuários, de maneira intuitiva, atribui-se para cada nome de usuário (conhecido como *handle*) do Topcoder selecionado, um endereço de e-mail do gmail. Por exemplo, o usuário cujo *handle* é XXX, foi atribuído a ele o e-mail XXX@gmail.com.

De posse destes e-mails fez-se uma consulta à plataforma, por meio de uma API pública, para verificar se o e-mail atribuído estava disponível para uso na plataforma Topcoder. Caso a consulta retorne "false", isso significa que o e-mail existe, e não pode ser utilizado por outro, logo, é muito provável que seja do usuário Topcoder. Assim, dos 271 usuários foram selecionados apenas 42 usuários com e-mails válidos.

O código escrito em "Javascript" utilizado para consulta do e-mail válido na base Topcoder pode ser observado na Figura 11.

```
var lista=["XXX"];
for(i in lista)
{ var j = i;
$.ajax
("http://api.topcoder.com/v2/users/validateEmail?email="
+ lista[j] + "@gmail.com").success(function(data)
{console.log(
data.requesterInformation.receivedParams.email + "\t" +
data.available);});});
```

Legenda: "XXX" é a lista de todos os *handles* obtidos na pesquisa no Topcoder

Figura 11: Consulta javascript no Topcoder

Com os 42 e-mails válidos poder-se-ia iniciar o processo de contato com os usuários, porém, sabendo que a Topcoder não disponibiliza o nome do usuário, e, para evitar a impessoalidade do envio do e-mail e, aumentar a probabilidade de retorno destes e-mails buscou-se uma forma de identificar o usuário. O objetivo deste procedimento foi encontrar o nome do usuário para colocar no início do e-mail o "nome da pessoa".

Habitualmente a comunidade participante de plataformas crowdsourcing, também atua em outras comunidades como Bitbucket³¹ e Github³² [67]. O Bitbucket e Github são serviços de *web hosting* compartilhado para projetos que utilizam mecanismos de controle de versão e colaboração. Na busca por mais dados de usuários, além de certificar que os e-mails correspondiam aos usuários da Topcoder, fez-se uma consulta ao Github por meio de uma API³³ pública do Github. Após nova consulta, foram encontrados 8 usuários com o mesmo nome de usuário no Topcoder e no Github. Essa operação foi realizada apenas para confirmar e constatar que, muitos usuários no Topcoder também utilizam o Github com o mesmo nome de usuário (*nickname ou handle*). Isso pode auxiliar no processo de contato com o usuário, pois, em muitos casos, o usuário disponibiliza seu nome no Github, fato esse que não ocorre no Topcoder. Além disso, foi desenvolvido um código, em Python, para selecionar alguns dados do usuário como como país de origem, mês de entrada, e o ano de entrada na plataforma Topcoder.

Após esse aprendizado realizou-se nova consulta na base do Topcoder obtendo-se 490 usuários, sendo 164 usuários novatos no Topcoder com ano de entrada entre os anos de 2015 e 2016. Destes 164 usuários, foram obtidos 35 e-mails válidos.

Um exemplo do resultado da consulta na plataforma Topcoder com os usuários de e-mails válidos, poder ser observado na Tabela 8. Os dados são: o *nick* (usuário), *country* (país de origem do usuário), *wins* (número de vitórias do usuário naquela competição), *mês de entrada* (mês de entrada do usuário na plataforma Topcoder), e *ano entrada* (ano de entrada do usuário na plataforma Topcoder). Os *nicknames* foram substituídos por letras por questões de privacidade dos dados dos participantes.

³¹ <https://bitbucket.org/>

³² <http://www.github.com>.

³³ API Github <https://api.github.com/users/>

Tabela 8: Exemplo de usuários selecionados

NICK	COUNTRY	WINS	Month Entry	Year Entry
A	Japan	0 Wins	February	2016
B	United States	0 Wins	February	2016
C	China	0 Wins	February	2016
D	France	0 Wins	April	2016
E	Taiwan	0 Wins	January	2016
F	Republic of Korea	0 Wins	January	2016
G	United States	0 Wins	February	2015
H	India	0 Wins	August	2015
I	Bangladesh	0 Wins	April	2015
J	China	0 Wins	January	2015
K	Switzerland	0 Wins	October	2015
L	Russian Federation	0 Wins	February	2015
M	Russian Federation	0 Wins	October	2015

Com a posse dos dados dos usuários novatos na plataforma Topcoder, foram enviados e-mails convidando-os para uma entrevista do tipo semiestruturada. O modelo de e-mail enviado para os prováveis entrevistados pode ser observado no Apêndice A.

Para evitar um provável acúmulo de respostas dos usuários solicitando uma agenda para a entrevista, os e-mails foram enviados em lotes de aproximadamente 10 e-mails separados por 3 dias. O primeiro lote de e-mails foi enviado no dia 28 de maio de 2016. Importante registrar que de todos e-mails enviados, nenhum retornou como e-mail inválido. Nessa primeira rodada apenas dois usuários responderam o e-mail, curiosamente, ambos brasileiros. A seguir, conforme a Figura 12, pode ser observado um trecho da primeira resposta por e-mail. A identificação do remetente foi suprimida por questões de confidencialidade e privacidade.

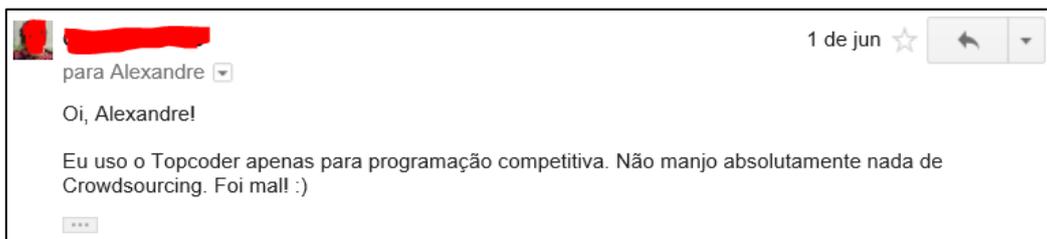


Figura 12: Primeira resposta

Pela primeira resposta, percebe-se que o usuário, mesmo participando de uma plataforma de crowdsourcing, não está familiarizado com termo crowdsourcing. Apesar do retorno, não houve evolução para uma entrevista. No segundo retorno, é possível observar que o usuário mesmo fazendo parte de uma plataforma de crowdsourcing, para ele, participar de competição de algoritmos não é parte integrante de projetos crowdsourcing. Apesar da troca de três e-mails entre o autor da presente proposta com o usuário em questão, também, não houve evolução para uma entrevista. Um trecho da resposta pode ser observado na Figura 13.

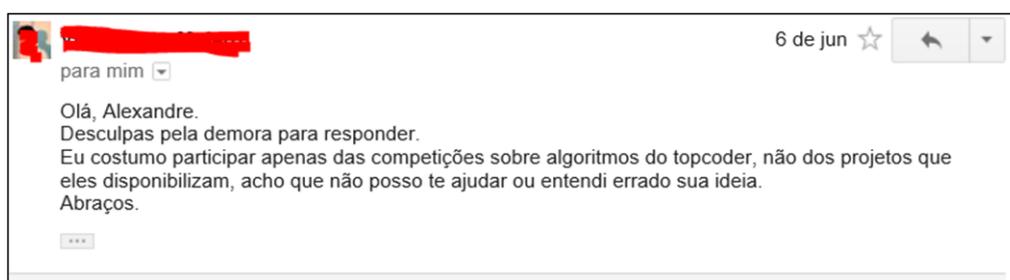


Figura 13: Segunda resposta

Mesmo com as duas respostas, infelizmente, nenhuma entrevista foi realizada. Vale lembrar que de um total de 3.403 foram selecionados apenas 271 dos usuários, restando ainda 3.133 para futura análise. Na tentativa de obter novos usuários fez-se uma alteração nos critérios de busca na Topcoder. Alterou-se o critério de busca para os usuários que participaram de uma “competição de algoritmos” para “competição de desenvolvimento”, permanecendo o sistema de classificação de no máximo 1.199. A alteração do tipo da competição foi motivada pelo insucesso na busca e pelo comentário realizado na segunda resposta (Figura 13). Salienta-se, portanto, que no site ao escolher a opção “*Development rating*”, não existe a possibilidade de selecionar as contribuições dos últimos 180 dias, como ocorreu na primeira seleção.

Após nova consulta à base Topcoder foram encontrados 1.059 usuários, entretanto, ao ordená-los pela última contribuição do usuário na coluna “*Development Rated Events*”, percebeu-se que, com rating máximo de 1199, a última contribuição foi no ano de 2014. Ou seja, tendo como base que os novatos são aqueles que entraram no Topcoder, entre os anos de 2015 e 2016, e, que os usuários selecionados fizeram a última contribuição a dois anos (2014) o resultado dessa busca foi descartado.

Com o descarte dos dados advindos da consulta por “*Development Rated Events*”, fez-se uma nova consulta pelo termo “*Test Suites rating*”, onde, agora, foram encontrados

apenas 41 usuários. Destes 41 usuários, apenas 4 contribuíram no ano de 2016. Foram enviados e-mails para esses quatro usuários, onde, infelizmente, e, novamente, não se obteve êxito no retorno destes usuários. Importante destacar que nenhum e-mail retornou como inválido ao autor remetente (autor da presente tese). Com nova negativa de retorno, foram selecionados os 37 e-mails restantes dessa última busca para novo envio, mesmo não adequados aos critérios. Mesmo assim, não se obteve o retorno. Após o envio de aproximadamente 76 e-mails válidos e nenhuma entrevista realizada, o primeiro estudo piloto foi encerrado no dia 12 de junho de 2016.

Mesmo não alcançado o objetivo inicialmente proposto para o primeiro estudo de campo piloto, a principal lição aprendida foi a imensa dificuldade em encontrar o contato, além de contatar os indivíduos na multidão em projetos de software crowdsourcing no modelo competitivo.

A política das plataformas em não expor o contato (o e-mail por exemplo) dos usuários indica o caráter competitivo do crowdsourcing na Topcoder. É provável que essa restrição de acesso auxilie as plataformas no processo de evitar um provável assédio por outras organizações que desejam contar em seus quadros com os grandes ganhadores de competições. Um exemplo de restrições de comunicações entre os usuários pode ser observado no “Termo de uso para membros da comunidade do Utest”, disponível no site³⁴ da Utest, pioneira em teste para software crowdsourcing, que determina

You are prohibited from making direct contact or engaging in commercial transactions, with other Members or uTest and Applause Customers, whether through the Site or otherwise. Members agree not to post their email address, phone number, or any other method of contact outside of the Site or give any of the foregoing information to another User. Members also must not contact uTest or Applause Customers outside of the uTest Site and Platform.

Como as plataformas Topcoder, Crowdplat e Upwork não disponibilizam um contato com seus usuários por questões de ética e confidencialidade, isso tornou o processo de encontrar e contatar os usuários difícil. Dado o exposto e devido as dificuldades apresentadas, além de não alcançar o objetivo proposto no primeiro estudo piloto que era elaborar uma lista preliminar de barreiras enfrentadas por novatos em projetos de software crowdsourcing, fez-se um segundo estudo de campo piloto, a seguir descrito.

³⁴ <https://www.utest.com/terms-and-conditions>

4.1.3. Segundo estudo de campo

Devido à dificuldade em contatar e, por conseguinte, entrevistar os usuários na plataforma Topcoder realizou-se um novo estudo. Dessa vez, fez-se um convite para um conjunto pré-definido de participantes (amostra de conveniência) para que participassem, voluntariamente, do estudo.

O objetivo do segundo estudo de campo foi identificar, de forma preliminar, as barreiras que podem afetar novatos em projeto de software *crowdsourcing* no modelo competitivo. Para isso, foi enviado um questionário por e-mail, no dia 13 de junho de 2016, para 20 desenvolvedores de software convidando-os à participarem em uma competição na Topcoder. Estes desenvolvedores foram divididos em dois grupos. O primeiro composto por 10 alunos finalistas do curso de bacharelado em Ciência da Computação de uma Instituição de Ensino Superior e o segundo grupo com 10 profissionais com mais de dois anos de experiência em desenvolvimento de software. Os indivíduos dos dois grupos foram selecionados obedecendo como critério de escolha nunca terem participado ou utilizado alguma plataforma de software *crowdsourcing*, além de classificados por Steinmacher [128] como “*onboarding newcomers*” isto é, “*volunteers that were trying to lace their first contributions*”.

Solicitou-se para cada indivíduo que realizasse o cadastramento, compreendesse a estrutura e o funcionamento da plataforma para que, passada essa etapa, participasse ao seu livre critério de escolha de uma tarefa em alguma competição de desenvolvimento de software (codificação) que pudesse ser realizada em um prazo máximo de 12 dias. Foram acrescidos 5 dias ao prazo para que o participante pudesse familiarizar-se com a Topcoder [21].

O questionário possuía a seguinte questão: “*Em sua opinião quais foram as principais dificuldades enfrentadas para contribuir, pela primeira vez, na plataforma Topcoder?*” A questão foi adaptada de Steinmacher [32]. Também se solicitou que o participante anotasse suas percepções sobre:

- facilidade em encontrar uma tarefa para participar.
- percepções durante a participação como:
 - forma de interação/comunicação com outros participantes e gestores da tarefa,
 - forma de apresentação
 - descrição da tarefa

- coordenação da tarefa
- critérios de escolha do vencedor.
- documentação de apoio para a realização das tarefas.
- percepções do uso em geral da plataforma.

Foram recebidos por e-mail, entre os dias 25 de junho de 2016 e 10 de julho de 2016, oito formulários com as respostas dos participantes. Entre os respondentes, três alunos e cinco de profissionais da área de desenvolvimento de software. Os participantes possuem idades entre 20 e 25 anos. Destes, 66% (5) possuem formação superior na área da tecnologia da informação com mais de 7 anos de experiência em atividades relacionadas ao desenvolvimento de software (incluindo experiência acadêmica). Todos jamais participaram em projetos de desenvolvimento de software crowdsourcing no modelo competitivo.

Os formulários recebidos dos alunos foram identificados pela letra “A” seguido de um número indicando a ordem de recebimento, por exemplo, o primeiro formulário recebido de um aluno foi identificado como “A1” e assim consecutivamente. Os formulários recebidos dos profissionais foram identificados pela letra “P” seguido de um número indicando a ordem de recebimento, por exemplo, o primeiro formulário recebido de um profissional foi identificado como “P1” e assim consecutivamente.

Para exemplificar, um trecho transcrito do comentário do aluno (A3), pode ser observado a seguir:

Em linhas gerais os desafios estavam escritos de maneira clara, tornando fácil o entendimento daquilo que é proposto e com uma documentação também detalhada. Após o registro em uma tarefa há uma grande comunicação não apenas na ferramenta mas também via e-mails que são enviados a cada novo review. Ao longo do uso encontrei algumas dificuldades na tela de login que as vezes ficava em uma espécie de “loop” que mesmo após a realização do login ela retornava a tela anterior e mesmo sem nenhuma ação, carregava e continuava a voltar a tela de login. Não consegui averiguar se esse problema ocorria em decorrência da plataforma ou de oscilações em minha internet. Outra questão que gostaria de constar é que talvez seria interessante a inserção de atividades que demandam um menor tempo de execução, afim de incentivar os novos usuários. Além disso, apesar da língua inglesa ser difundida em todo mundo, a tradução da plataforma para outros idiomas ao meu ver ajudaria a alavancar ainda meu seu uso. – A3

Um trecho transcrito do comentário do profissional (P2), pode ser observado a seguir:

“Uma dificuldade que notei foi que a plataforma não tem uma funcionalidade de anexar documentos necessários para suporte do projeto e para programadores. Em alguns projetos é utilizado links para Google Drive, GitLab(que é obrigatório um cadastro do usuário para acessar os documentos) ou o fórum. Acredito que com a implementação de uma funcionalidade de anexar os documentos como uma nova aba seria uma forma mais eficaz, rápida para o usuário. Obs: Links de documentos pelo fórum pode oferecer riscos de usuários mal intencionados.” - P2

De posse destes dados, realizou-se uma análise qualitativa dos dados por meio da *Grounded Theory*, traduzida para o português para Teoria Fundamentada nos Dados – TFD. Como o objetivo foi identificar as barreiras e não criar uma teoria substantiva como sugere a Teoria Fundamentada nos Dados, utilizou-se apenas a codificação aberta e a axial e não foi realizada a etapa da categorização. Uma versão de avaliação da ferramenta ATLAS.ti³⁵ foi utilizada no processo armazenamento dos dados e de apoio à codificação.

A seguir, apresentar-se-ão as seis barreiras preliminares identificadas.

1. **Dificuldade no conhecimento do idioma inglês para compreender / entender o enunciado da tarefa:** O Inglês é o idioma universal para projetos baseados na tecnologia da informação. Trabalhadores na multidão devem ser aptos a comunicar-se, na mesma linguagem, em comunidades on-line. Após o uso na plataforma Topcoder, alguns participantes comentaram: *“Eu gostaria de ver a plataforma Topcoder com um sistema multi-lingual (tradução simultânea) pois eu não consigo nem ler, e nem escrever bem o idioma Inglês”*. Outros participantes reforçam esse aspecto, *“Ter habilidades no idioma Inglês é muito importante”* e, outro participante comenta: *“Eu acho que o mais importante é que a língua inglesa é a mais falada no mundo”* A plataforma Topcoder utiliza o inglês como língua oficial e, não possui mecanismos automáticos de tradução simultânea a outros idiomas.

2. **Falta de conhecimento da linguagem de programação para a realização da tarefa:** Entender um código-fonte não é uma tarefa fácil. É muito importante que este código tenha uma boa estrutura lógica, seja bem escrito e auto-organizado, para que os indivíduos que o manterão possam identificar a lógica do funcionamento do código, suas rotinas (ou módulos) e funções. Um dos entrevistadores mencionou que *“é necessário estruturar regras de programação para mantê-lo compreensível, como cada função (ou módulo) deve ter comentários, nomes de variáveis devem ser significativos e assim por diante”*. Problemas para entender a estrutura do código foram mencionados. Normalmente, não é uma atividade trivial entender um código-fonte, devido, muitas vezes, pelo seu tamanho e complexidade.

3. **A documentação da tarefa não foi entendível, insuficiente ou ausente para realização da tarefa:** A maioria dos participantes relatou problemas ou procedimentos inadequados relacionados à documentação de tarefas durante a atividade realizada. Seja

³⁵ <http://www.atlasti.com>

pela falta ou insuficiência de documentação. Em ambos os casos, os participantes comentam que essa falta ou omissão dificultam o entendimento da tarefa. Sobre isso, comentam que muitas tarefas, simplesmente não tinham documento associado, ou, quando a tinham, era insuficiente para o seu entendimento. Alguns consideram, ainda, a documentação “pouco clara” ou incompleta. Um participante da plataforma Topcoder disse: *"O link para o GitHub também não ajuda e não está claro para mim como isso funciona"*. Outro entrevistado mencionou: *"Não consigo encontrar documentação adequada para realizar essas tarefas"*.

4. Dificuldade no entendimento da funcionalidade da plataforma: A ISO 9241-11³⁶ define a usabilidade como *“a medida em que um produto pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado”*. Em uma percepção de eficácia, os participantes comentam que é difícil completar a tarefa, exigindo, muito esforço cognitivo para executar a tarefa. Os participantes mostraram-se motivados para discutir com outros membros das plataformas sobre problemas, tanto para execução da tarefa, quanto para o seu entendimento. O canal eleito para essa comunicação, foi, preponderantemente, o fórum. Do ponto de vista da eficiência, os participantes comentam que *“é muito difícil encontrar as informações sobre as tarefas”*. Sobre a percepção de satisfação alguns participantes comentam que *“a plataforma é legal”* e outros dizem que *“a plataforma é intuitiva e a velocidade de navegação é muito boa”*. A maioria dos participantes acredita, que o Topcoder *“é fácil de usar”*. Segundo um dos participantes do estudo: *"A estratégia de layout utilizada para menus e filtros nos ajuda a procurar por competições na plataforma"*. No geral, a percepção dos participantes em relação à usabilidade foi positiva, com exceção da eficácia, que é um aspecto fundamental da usabilidade da plataforma.

5. Dificuldade no entendimento da tarefa e dificuldade no entendimento do fluxo das atividades para a realização da tarefa: Participantes relatam sobrecarga de informações durante a realização de tarefas. Isso causa dificuldade no entendimento da tarefa bem como no fluxo de submissão da tarefa. Um participante mencionou: *"Estou me sentindo um pouco estressado com tantas informações do que é necessário para tomar uma decisão"*. Outro participante reforçou essa questão: *"Eu tenho muitas informações [disponíveis] na Plataforma TopCoder e não sei o que fazer"*.

³⁶ ISO/IEC. 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E), 1998.

6. **Pouco tempo disponível para realizar a tarefa:** Encontrar uma tarefa com o tempo destinado pela plataforma para sua conclusão, com o tempo disponível pelo trabalhador para realizar a tarefa foi considerado um importante problema. Participantes comentaram que foi difícil desenvolver uma compreensão de qual é o esforço (cognitivo) e tempo necessário para desenvolver a tarefa. Nesse caso, há muitas dificuldades associadas ao mal gerenciamento do tempo e, não são de razoável resolubilidade, porque em um projeto de software crowdsourcing no modelo competitivo, tempo é dinheiro. De acordo com a análise, “tempo para realizar a tarefa” refere-se ao gerenciamento pessoal do tempo. Um dos participantes comentou: *“estou procurando as tarefas de acordo com minhas habilidades e tempo disponível para o desafio”*. Outro participante reforçou isso: *“Acredito que quem tem tempo integral para as tarefas, deve ter a vantagem de quem tem meio período, obviamente”*

Observou-se uma sobreposição entre algumas barreiras com problemas “tradicionais” no desenvolvimento de software distribuído ou local. Por exemplo, a falta de documentação nos requisitos ou no código-fonte é um problema constante no desenvolvimento de software [128]. Entender o código é outro problema comum em qualquer modelo de desenvolvimento de software [32]. No entanto, esses problemas provavelmente são amplificados quando ocorrem em um contexto de software crowdsourcing: a falta ou documentação inadequada não pode ser minimizada por meio de comunicação direta formal ou informal com os clientes, porque esse tipo de comunicação geralmente não é possível em um contexto de crowdsourcing. Entender o código-fonte é problemático nesse contexto, já que os desenvolvedores não têm acesso a todo o projeto e não estão cientes da arquitetura ou das decisões que podem influenciar suas tarefas.

Na Figura 14 podem ser visualizadas as seis barreiras preliminares que emergiram como resultado da análise qualitativa realizada.

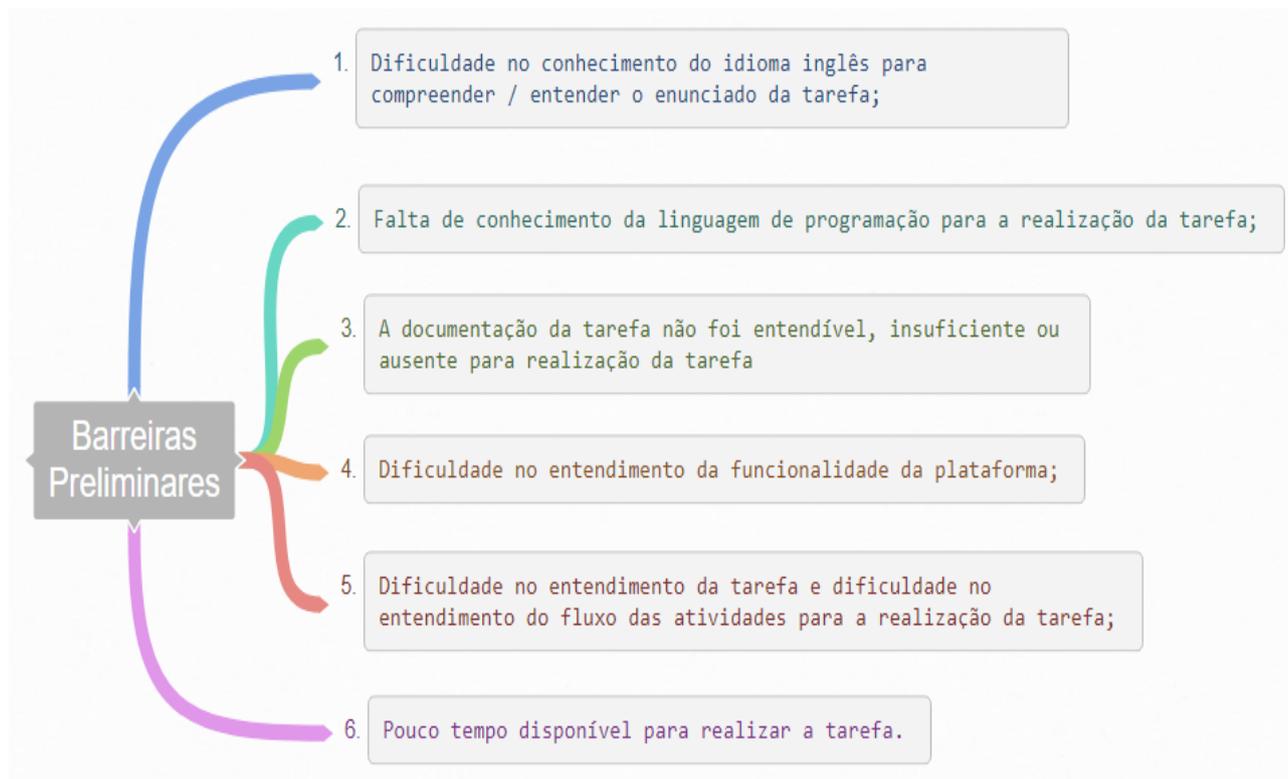


Figura 14: Barreiras Preliminares

Interessante observar que, das seis barreiras preliminares encontradas (observadas na Figura 14), apenas duas (“*Dificuldade no entendimento da tarefa e dificuldade no entendimento do fluxo das atividades para a realização da tarefa*”, e “*pouco tempo disponível para realizar a tarefa*”) não foram encontradas por Machado *et al.* [133]. Isso pode ter ocorrido pela natureza da tarefa.

Baseado nessas barreiras preliminares, elaborou-se, um conjunto de seis sugestões para que os novatos participantes na plataforma Topcoder pudessem sobrepor essas barreiras. As sugestões estão descritas a seguir.

1. **A plataforma deve fornecer documentação clara, completa e consistente sobre a tarefa e o projeto:** Com base nas entrevistas, é importante atender às necessidades dos novatos e aos requisitos de tarefas em projetos de software crowdsourcing. Quando a documentação não estiver disponível ou o trabalhador não a compreender, é fundamental oferecer mecanismos de comunicação para que os mesmos possam o utilizar para questionamentos ou comentários, quando necessário. Abordagens síncronas como bate-papo, teleconferências, entre outros, e assíncronas como por exemplo, e-mail e fóruns são relevantes nesses cenários. Os dois elementos do modelo crowdsourcing, clientes e a plataformas, precisam prestar atenção ao tempo gasto em

práticas de documentação adequadas e artefatos adicionais durante o envio de tarefas, pois isso, provavelmente reduzirá o tempo gasto em respostas a consultas dos trabalhadores na multidão, sejam eles novatos, ou não.

2. **A plataforma deve estabelecer mecanismos que auxiliem o trabalhador na multidão a escolher uma tarefa de acordo com seu perfil e habilidades.** Esses mecanismos podem auxiliar a motivar os desenvolvedores e, conseqüentemente, permitir que eles calculem melhor o tempo necessário para concluir as tarefas escolhidas. Portanto, uma sugestão aos gestores das plataformas é que incluam mecanismos de pesquisa que permitam aos desenvolvedores, especialmente os novatos, identificar as tarefas de acordo com seu perfil e o grau de complexidade da tarefa proposta pela plataforma. De acordo com LaToza e van der Hoek [22] um princípio fundamental do crowdsourcing é que cada participante deve ser informado com precisão sobre a tarefa a ser executada”.

3. **O trabalhador da multidão, novato em projetos de software crowdsourcing, deve escolher uma tarefa de acordo com suas expectativas de maior benefício:** LaToza e van der Hoek [22] mencionaram que “*antes de aprender em sistemas baseados no crowdsourcing, o desenvolvedor de software deve conhecer a cultura e modelo de negócio de projetos em software crowdsourcing*”. O software crowdsourcing é uma nova forma de organização do trabalho que é mediada por plataformas e tarefas. Assim, esse modelo pode sobrecarregar de informações e atividades, as quais, podem extrapolar a real capacidade do trabalhador em gerenciar o seu tempo para submeter a tarefa. Ao perder o controle, o trabalhador pode desistir de registrar e, por conseqüência, submeter a tarefa.

4. **O trabalhador da multidão, novato em projetos de software crowdsourcing, deve escolher tarefas nas quais possuam mecanismos de guia ou diretrizes para a execução da tarefa.** Semelhante ao apresentado por Steinmacher *et al.* [129] no contexto de projetos de OSS, sugere-se o desenvolvimento de páginas web específicas para novatos, fornecendo a eles recursos de que precisam. Importante observar, que, não é indicado sobrecarregar os trabalhadores na multidão com todos os recursos possíveis, pois, muita informação pode confundi-los no processo de registro e submissão da tarefa.

5. **A plataforma deve oferecer ao novato em projetos de software crowdsourcing, mecanismos que permitam entender a estrutura de um código-fonte.** De acordo com o tamanho do projeto, sugere-se associar a documentação da tarefa,

diagramas da UML gerados automaticamente a partir do código-fonte. Porém, cabe observar, que poderão ocorrer preocupações em relação à propriedade intelectual do código e documentação produzida.

6. **A plataforma deve oferecer ao novato em projetos de software crowdsourcing, mecanismos de tradução automática.** Na fase exploratória dessa seção, foi possível identificar que, a maioria dos participantes em atividades são originados de países do continente asiático, sendo muitos deles não nativos em inglês e, por conseguinte, com prováveis dificuldades no entendimento das tarefas, bem como, na comunicação com os demais membros da plataforma. Portanto, plataformas de crowdsourcing, como o Topcoder, poderiam implementar recursos como a tradução automática assistida por computador. No trabalho de Machado *et al.* [133] a falta de conhecimento na língua inglesa foi apontada como um desafio para o trabalhador da multidão em atividades de software crowdsourcing.

Salienta-se, portanto, que o objetivo da Fase 1 foi encontrar barreiras enfrentadas por novatos em primeiras contribuições em projetos de software crowdsourcing. Contudo, para não limitar apenas a novatos, na Fase 2 do presente trabalho ampliou-se para qualquer tipo de trabalhador na multidão, sendo ele novato, ou não.

Existiram algumas limitações no segundo estudo. Uma limitação para avaliar o presente estudo pode ser a possibilidade de os participantes serem tendenciosos (viés) a encontrarem algumas barreiras, visto que, o pedido foi justamente para isso. Importante destacar que os indivíduos que participaram deste estudo pertencem, ou estão alocados, na região norte do estado do Rio Grande do Sul no Brasil e, isso pode ser também uma limitação do estudo.

O resultado demonstrou a existência de barreiras encontradas pelos usuários ao participarem pela primeira vez em projetos de software crowdsourcing, observado em Zanatta *et al.* [131] e como ocorreu em projetos de software livre de acordo com Steinmacher *et al.* [128].

Importante ressaltar que a realização destes estudos foi fundamental no processo de constante revisitação aos dados coletados para a análise comparativa. Também contribuiu significativamente no refinamento dos procedimentos de coleta, registro e análise dos dados. Além de alcançar o objetivo proposto com a realização deste segundo estudo piloto, outro ganho foi a familiarização com a Teoria Fundamentada nos Dados e a ferramenta de apoio à análise de dados qualitativa. Merece destaque também, que segundo

Yin [46] esses estudos realizados podem ser considerados como um projeto piloto para a realização de um novo estudo (demonstrado na seção 4.2). Importante registrar que as recomendações propostas, não foram avaliadas, porém, servirão como base para a confecção das recomendações finais da presente tese.

4.2. Barreiras e Recomendações - Fase 2

O principal objetivo da Fase 2 foi responder e atender o primeiro objetivo da presente tese que é “Entender, identificar e descrever as barreiras e dificuldades dos usuários quando participam de competições e como os afetam em projetos de software crowdsourcing” além de auxiliar na resposta da primeira questão de pesquisa da presente tese.

A partir das reflexões e análises realizadas dos dados obtidos na Fase 1, conduziu-se um estudo de caso para identificar as barreiras enfrentadas por trabalhadores na multidão na realização de tarefas em projetos de software crowdsourcing no modelo competitivo. Após identificadas as barreiras, por meio do processo de Revisão por Pares, foram selecionadas apenas as barreiras relativos ao trabalhador. Um conjunto de recomendações emergiu do estudo de caso.

As etapas desenvolvidas nesta fase podem ser observadas na Figura 15.

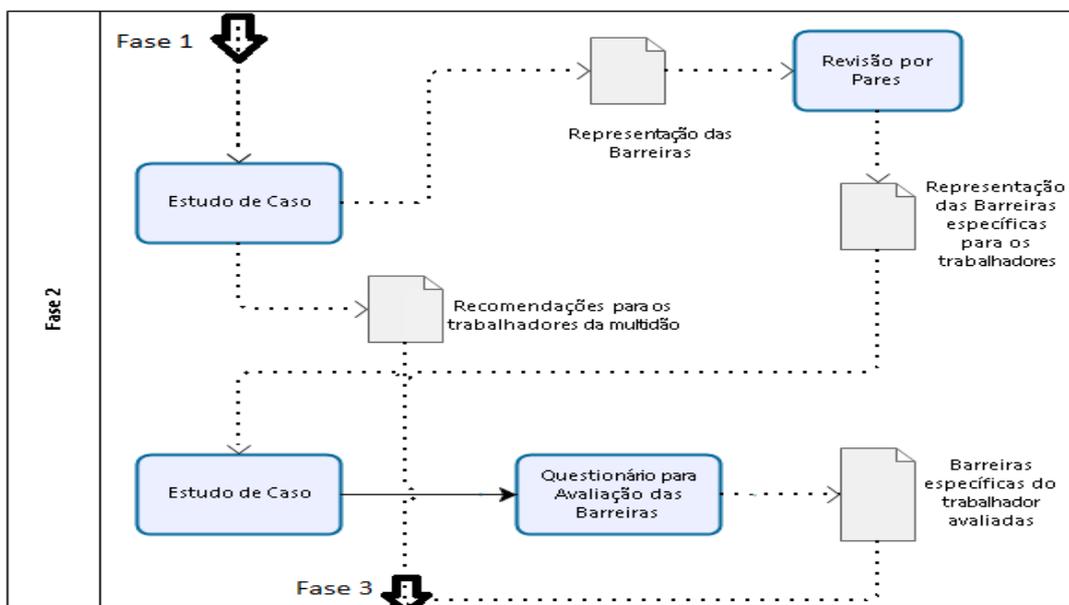


Figura 15: Procedimentos da Fase 2

4.2.1. Estudo de caso

De posse dos dados coletados nos estudos de campo realizados na Fase 1, planejou-se e executou-se um estudo de caso para aprofundar o conhecimento destes dados, em especial a lista preliminar de barreiras, bem como, as recomendações. O protocolo do estudo de caso pode ser observado no Apêndice C. A seguir é descrito o estudo de caso.

A unidade de análise I, denominada de “IE1”, ocorreu como um projeto na disciplina “Desenvolvimento de Software Colaborativo” do Programa de Pós-Graduação em Ciência da Computação da Instituição de Ensino 1 (IE1). A disciplina discute a história de sistemas colaborativos bem como princípios e tipos de desenvolvimento de software colaborativo como Desenvolvimento de Software Global e Crowdsourcing. A disciplina é oferecida uma vez ao ano. Durante as 16 sessões (aulas) os acadêmicos participam de aulas expositivas, grupos de discussão, apresentação de artigos relacionados e, por fim, da realização de um projeto final da disciplina. Nesta unidade de análise, participaram 18 homens e 3 mulheres, totalizando 21 participantes, com idades entre 20 e 35 anos. Destes, 61,9% (13) possuem formação superior na área da tecnologia da informação com mais de 6 anos de experiência em atividades relacionadas ao desenvolvimento de software (incluindo experiência acadêmica). Mesmo que 76,2% (16) nunca tenha participado em projetos de desenvolvimento de software crowdsourcing, e, que 90,5% (19) não conseguiram vencer a competição proposta, 85,7% (18) acreditam, ainda, que é possível desenvolver softwares utilizando o modelo crowdsourcing.

A unidade de análise II, denominada de “IE2”, ocorreu com a execução do mesmo projeto da IE1 e, em duas disciplinas. Um projeto foi desenvolvido na disciplina “Qualidade e Teste de Software” do Bacharelado (graduação) em Ciência da Computação da Instituição de Ensino 2 (IE2). A disciplina discute a história da qualidade do software bem como novas abordagens para testes de software como *software testing* crowdsourcing. O mesmo projeto foi desenvolvido na disciplina de “Engenharia de Software” do Curso Superior de Tecnologia em Análise e desenvolvimento de Sistemas da Instituição de Ensino 2. As duas disciplinas são oferecidas uma vez ao ano. Durante as 18 aulas, os acadêmicos participaram de aulas expositivas, e da realização de um projeto final da disciplina. Nessa unidade de análise, no projeto da disciplina “Qualidade e Teste de Software” participaram 12 homens e uma mulher, com idades entre 20 e 25 anos. Todos são acadêmicos do bacharelado em Ciência da computação, sendo destes, 84,6% (11) entre 1 e 5 anos de

experiência em atividades relacionadas ao desenvolvimento de software (incluindo experiência acadêmica). Mesmo que 76,9% (10) nunca tenha participado em projetos de desenvolvimento de software via crowdsourcing, e, que 69,2% (9) não venceram a competição proposta, 92,3% (12) acreditam, ainda, que é possível desenvolver software utilizando o modelo crowdsourcing.

Nesta mesma unidade de análise, porém, agora no projeto da disciplina “Engenharia de Software” participaram 3 homens e 2 mulheres totalizando 5 participantes, com idades entre 20 e 25 anos, com 1 a 5 anos de experiência em atividades relacionadas ao desenvolvimento de software (incluindo experiência acadêmica). Mesmo que 60% (3) nunca tenham participado em projetos de desenvolvimento de software crowdsourcing, e, que nenhum participante tenha vencido a competição proposta, todos acreditam que é possível desenvolver software utilizando o modelo crowdsourcing. Justificam-se duas unidades de análise e os estudos na Fase 1, pois, conforme Strauss e Corbin [44], na Teoria Fundamentada nos Dados, para enriquecer e ampliar os significados sobre o fenômeno é fundamental que a coleta de dados seja realizada com sujeitos e práticas diferenciadas com a finalidade de possibilitar a interpretação sistemática comparativa dos dados.

No total, 39 desenvolvedores participaram deste estudo de caso, sendo 33 homens e 6 mulheres, com idades entre 20 e 35 anos, e com experiência em desenvolvimento de software. Interessante observar, mesmo que a maioria nunca tendo participado de atividades de software crowdsourcing acreditam que é viável desenvolver software utilizando esse modelo.

4.2.2. Execução da coleta de dados

Para a coleta de dados do estudo de caso, os dados dos 39 participantes foram arquivados com apoio da QSR Nvivo³⁷ versão 10. Na primeira fase deste trabalho, utilizou-se a ferramenta Atlas.ti, entretanto, como tratava-se de uma versão de avaliação, foi descartada. Observa-se que o propósito da utilização desses softwares não foi para automatização do processo analítico, mas sim, apenas na organização, classificação e disponibilização dos dados.

Prevendo a execução da Teoria Fundamentada nos Dados, e por ser um processo indutivo do conhecimento, foram elaboradas questões abertas e fechadas que permitem

³⁷ <http://www.qsrinternational.com/>

uma análise mais flexível das respostas, permitindo, com isso, um profundo estudo envolvido com os limites do problema em questão. Uma proposta de roteiro da entrevista semiestruturada, adaptada de Steinmacher [32], pode ser observada no Apêndice B. Vale lembrar que, conforme a Teoria Fundamentada nos Dados, as questões podem mudar e não se esgotam no início do processo investigativo, fato que ocorreu no processo de passagem da Fase 1 para a Fase 2 do presente trabalho. Novamente, para a execução da coleta dos dados foi eleita a plataforma TopCoder. A coleta foi realizada nas duas unidades de análises. Após o início das atividades no semestre acadêmico, apenas na quarta semana de aula, os estudantes foram apresentados ao projeto que deveria ser realizado. Esse projeto pode observado no Apêndice E. Os estudantes tiveram duas semanas (semana 6) para selecionar a tarefa dentro da categoria “*Development Challenge (DC)*”, com prazo para realização da tarefa em até seis semanas (semana 12), incluindo mais duas semanas para responderem o questionário. Uma discussão em classe foi realizada nas duas unidades de análise (em momentos separados), e o retorno dos relatórios ocorreu na semana 16. Os três relatórios produzidos foram:

a) Relatório 1 – Descrição da Tarefa (semana 6): os estudantes reportaram a tarefa selecionada nos oferecendo uma pequena descrição do objetivo da tarefa, a sub-categoria eleita na categoria de “*Development Challenge*” e o tempo esperado para a realização da tarefa. Uma breve explanação das razões da escolha da tarefa também foi entregue.

b) Relatório 2 – Relato da experiência (semana 10): os estudantes responderam 9 questões abertas que podem ser observadas no Apêndice F, sobre a sua experiência em termos de colaboração e as dificuldades encontradas para a realização das tarefas, a realização da plataforma e o uso da plataforma Topcoder.

c) Relatório 3 – Questionário com questões fechadas (semana 14): esse questionário, que pode ser encontrado no Apêndice G, contém dados demográficos dos estudantes relacionados, principalmente a formação, habilidades, competências, entre outros.

Para tanto, e justifica-se sublinhar que o processo de pesquisa executado nessa fase, foi novamente baseado em técnicas e práticas da Teoria Fundamentada nos Dados discutidos por Strauss e Corbin [44], que é um método para extrair os resultados mais significativos, ou seja, descobrir, por meio das condições em que os fenômenos ocorrem as barreiras e as dificuldades enfrentadas em atividades de projetos de software crowdsourcing no modelo competitivo. Cabe registrar que na visão da Teoria

Fundamentada nos Dados, são amostrados os fatos e os incidentes e não pessoas ou instituições e que a amostragem não é estatística como referenciada na pesquisa quantitativa.

Os dados coletados no Estudo de Caso por meio dos questionários, análise dos relatórios, discussão em grupo, anotações e memorandos foram utilizados para a definição das categorias até que elas se estabilizassem em que pese, novos eventos não agregassem nada mais ao estudo em questão. Assim, embora utilizando-se de análise documental, as fontes de dados permanecem sendo os questionários, e, em segundo lugar os documentos, analisados à luz das observações do autor da presente tese além de outras técnicas complementares.

Executou-se, em sua plenitude apenas a codificação aberta como indicam Strauss e Corbin [44]. Inicialmente, na codificação aberta os dados foram decompostos em unidades de análise para serem examinados, comparados, conceituados e categorizados. Na codificação axial, fez-se uma relação preliminar entre as categorias. Por fim, em uma tentativa de aproximação à codificação seletiva, elaborou-se um conjunto de categorias essências em torno do elemento relacionadas especificamente ao trabalhador da multidão, objeto de estudo da tese.

Importante destacar que o processo de codificação aberta teve a participação, de forma colaborativa e paralela, de outros dois pesquisadores. Após a codificação fez-se uma discussão com outro pesquisador, em busca do consenso sobre as codificações, suas características e nomenclaturas. Esta discussão serviu como base para evitar prováveis vieses. O resultado dessas discussões foi a representação das barreiras descritas na próxima seção.

4.2.3. Barreiras Identificadas

Para a construção de códigos e categorias analíticas a partir dos dados, as respostas com as questões abertas (Relatório 2), e os códigos advindos da Fase 1 foram, além de armazenadas na base de dados do software Nvivo, impressas para auxiliar no processo de comparação constante durante a etapa da análise. Importante destacar que a Teoria Fundamentada nos Dados é “*um método geral da análise comparativa [...] e um conjunto de procedimentos capazes de formar uma Teoria Fundamentada nos Dados*” [44]. Este procedimento auxiliou a construção dos códigos e categorias.

Um momento do processo de análise da codificação aberta (códigos preliminares) com os pesquisadores que atuaram como especialistas (juizes) pode ser observado na Figura 16. Apenas para ilustrar, foram afixadas em uma parede as respostas dos 21 respondentes das questões abertas (Apêndice F) da unidade de análise da IE1. O especialista que atuou como juiz, pode realizar a codificação aberta, identificando ou marcando os códigos preliminares com o auxílio de pequenos blocos de notas composto por folhas de papel coloridas com adesivos. Visualizar as questões auxiliou o procedimento de identificação e categorização. Este processo de fixar em uma parede as respostas das questões também foi realizado para as respostas da unidade de análise IE2.

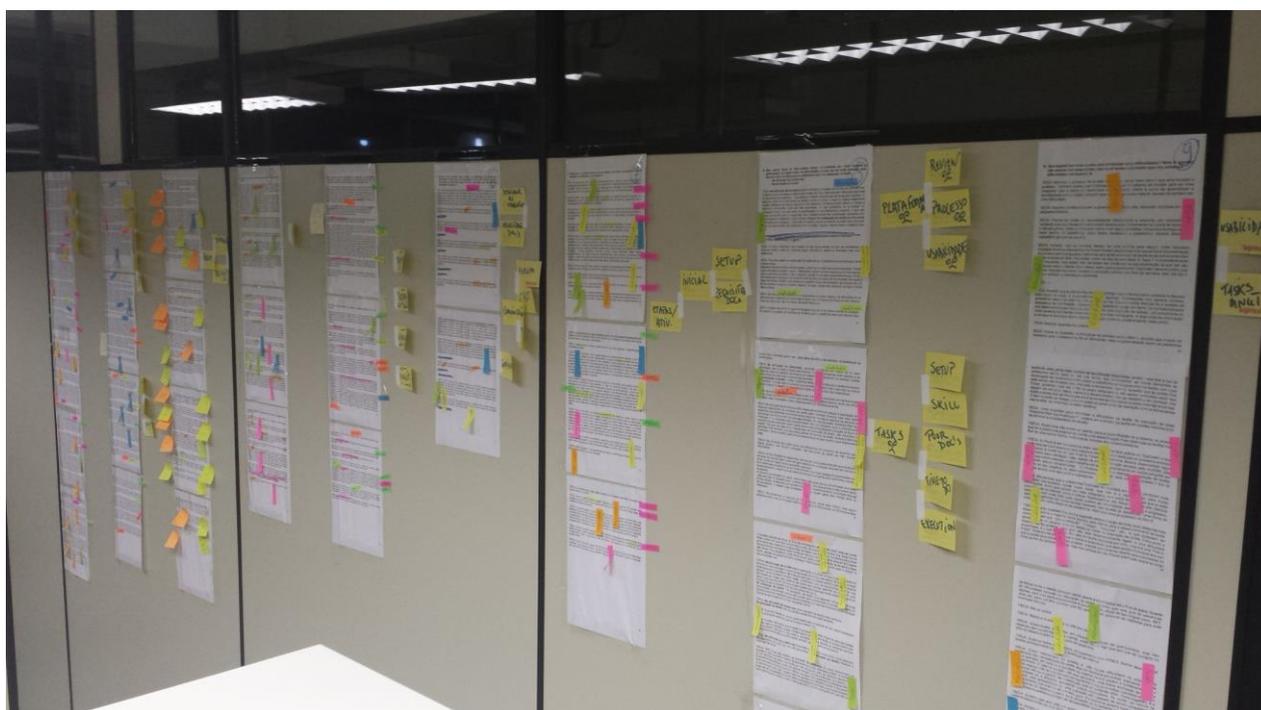


Figura 16: Apoio ao processo de codificação aberta

Como auxílio no processo de coleta de dados, fez-se na semana 15, uma sessão de discussão em classe com os participantes das duas unidades de análise. Para isso, foram respeitadas algumas características, como a homogeneidade dos participantes, a natureza qualitativa, e a discussão focada em um tópico. Algumas práticas como o planejamento e condução das discussões também foram utilizadas. Para a discussão no grupo dos participantes da unidade de análise I, atuaram como moderadores a professora responsável pela disciplina, o autor da presente proposta bem como, outra especialista. Na discussão com os participantes, na unidade de análise II, o moderador foi apenas o autor da presente tese. Os dados levantados, durante as duas sessões, foram transcritos e as anotações dos moderadores foram armazenados também na base de dados do software Nvivo.

A redação de memorandos e as anotações auxiliou na elaboração das categorias, especificação das propriedades, determinação das relações, e a identificação das lacunas. A seguir, um exemplo de uma anotação realizada durante a sessão realizada pela responsável da disciplina na Instituição de Ensino 1 pode ser observada na Figura 17.

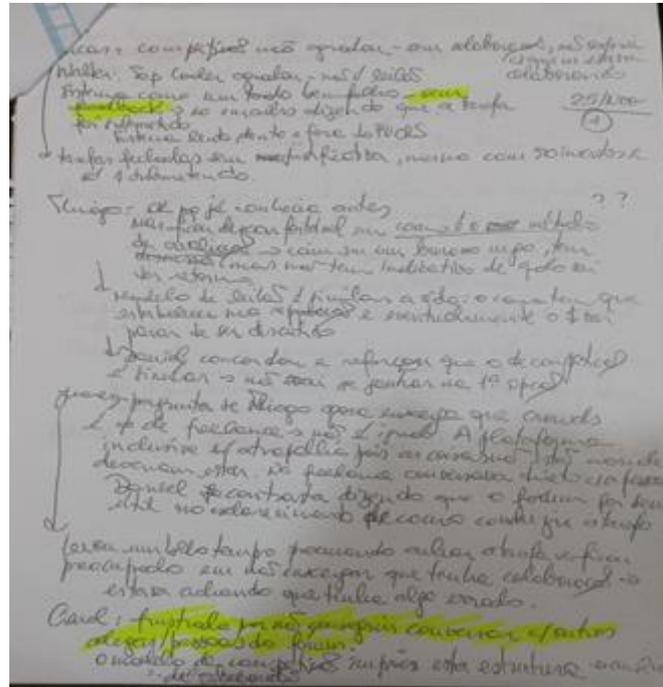


Figura 17: Exemplo de uma anotação

As respostas das questões abertas (Relatório 2), e os códigos advindos da Fase 1, foram assim organizadas. Cada resposta foi identificada por um número, a qual indica a ordem de recebimento, seguido pela letra “P” que significa, “participante”, por exemplo, a primeira resposta recebida foi identificada como “1P” e assim consecutivamente. Como foram utilizadas duas unidades de análise para a realização do Estudo de Caso, os mesmos também foram identificados como “IE1” para a unidade de análise da Instituição de Ensino 1 e “IE2” para a unidade de análise da Instituição de Ensino 2. A seguir dois exemplos:

1P_IE1 = Primeiro Participante da Unidade de Análise da IE1.

2P_IE2 = Segundo Participante da Unidade de Análise da IE2.

4.2.4. Barreiras identificadas em todos os elementos do crowdsourcing

Essa seção apresenta a partir dos dados analisados³⁸, os resultados da codificação aberta, das categorias, e dos conceitos construídos a partir das barreiras enfrentadas pelos

³⁸ Os dados estão disponíveis em <https://github.com/ALZana/doutorado>

trabalhadores quando participam em tarefas em projetos de software crowdsourcing no modelo competitivo. Importante observar, que as barreiras emergiram de todos os elementos do modelo crowdsourcing: o cliente, a plataforma, a tarefa e a multidão. Fragmentos de alguns textos capturados dos entrevistados que auxiliaram para a construção e imersão das barreiras serão expostos.

A seguir, far-se-á uma descrição das categorias, Tempo, Plataforma, Pessoal, Tarefa e Colaboração, com suas respectivas barreiras.

a) **Categoria Tempo**

A primeira grande categoria é o *Tempo*. Tempo que pode ser dividido em duas dimensões. A primeira no que se refere ao tempo **pessoal** disponível para entender, aprender e executar a tarefa, e, a segunda, no que tange ao tempo que a **plataforma**, em conjunto com o cliente, determina para que a tarefa seja executada e entregue ao cliente.

Mesmo tendo o perfil necessário para realizar a tarefa e interesse em realizá-la, o trabalhador leva em consideração seu tempo disponível para fazer a tarefa. Ou seja, o tempo pessoal disponível foi considerado um elemento importante, mesmo que negativo, por grande parte dos participantes e, nesse sentido, é o primeiro valor a ser elencado que merece atenção.

“11P_IE1: levou vários dias para que eu encontrasse uma tarefa cuja execução eu pudesse fazer no tempo proposto, visto que a maioria das tarefas dentro do tipo exigido na descrição do trabalho é voltada ao desenvolvimento ou correção de bugs de aplicações, e essas duas atividades não são fazem parte das minhas skills ou interesses.”

“2P_IE2: Em uma segunda fase precisei decidir qual dos desafios realizar, programação ou análise. Como não sou um exime programador pensei não vou me ater a estes desafios, então parti diretamente para a parte das análises e casos de testes. Ao analisar os desafios, acabei optando por uma mais simples devido a meu curto período de tempo em decorrência de estar abrindo uma empresa e estar sempre em prospecção de novos clientes”.

“3P_IE2: Entretanto é necessário uma grande disponibilidade, pois a maioria das tarefas são complexas e dispendiosas em questão de tempo, além de demandarem conhecimento técnico de tecnologias específicas de cada desafio, muitas vezes alheias ao repertório do usuário participante. Muito tempo para serem realizadas”.

Os gestores de projetos da plataforma em conjunto com o cliente determinam, em função de necessidades do cliente e como em qualquer outro negócio, o prazo que cada tarefa ficará disponível para ser entregue. No modelo crowdsourcing a redução de custos é fundamental, como discute Allahbakhsh [53]. O trabalhador tem o livre arbítrio em definir se fará ou não a tarefa, pois ele deverá avaliar, onde cabível, e determinar se possui as

competências necessárias, o tempo disponível e os recursos suficientemente mínimos para entender e executar a tarefa.

“5P_IE1: Eu vinha monitorando a relação de tarefas na categoria Development Challenge desde que me inscrevi no TopCoder, em busca de uma que fosse compatível com as minhas habilidades e, também, pudesse ser realizada em um tempo relativamente curto (estava planejando fazê-la em um final de semana)”.

“6P_IE1: A Seleção da tarefa ocorreu através da busca de challenge F2F. Nas buscas eu tive algumas dificuldades com filtros pois buscava uma atividade que atendesse o prazo referente a entrega do trabalho. Também fiz uma leitura geral das atividades encontradas, com diferentes filtros, a fim de conhecer melhor o tipo de premiação das mesmas. Escolhi uma atividade que não fosse tão complexa e principalmente que atendia os prazos”.

Enfim, nota-se, e muito presente, a preocupação do trabalhador com aspectos relacionados ao seu tempo pessoal que deverá dispor no que tange, para entender e preparar todo o ambiente quando comparado ao tempo disponibilizado pela plataforma, para poder executar a tarefa.

Neste contexto, as barreiras relativas a tempo são apresentadas a seguir.

1. Pouco tempo pessoal disponível para executar a tarefa
2. A tarefa foi considerada muito complexa e com tempo limitado para a sua execução
3. Pouco tempo que a plataforma destinou para a execução da tarefa
4. Dificuldade no tempo alocado para preparar o ambiente para executar a tarefa.

A Tabela 9 mostra as barreiras da categoria Tempo de acordo com a fonte (origem) dos dados em respectivas Fases (1 e 2). Entende-se por origem quando da ocorrência de, ao menos, uma barreira.

Tabela 9: Origem das Barreiras da Categoria Tempo por Fonte de Dados

Barreiras da Categoria Tempo	Origem		
	Fase 1	Fase 2	
		IE1	IE2
Pouco tempo pessoal disponível para executar a tarefa	•	•	•
A tarefa foi considerada muito complexa e com tempo limitado para a sua execução	•	•	•
Pouco tempo que a plataforma destinou para a execução da tarefa		•	
Dificuldade no tempo alocado para preparar o ambiente para executar a tarefa	•	•	•

Legenda: IE1: Instituição de Ensino 1 / IE2: Instituição de Ensino 2

b) Categoria Colaboração

A categoria Colaboração refere-se aos problemas ou dificuldades no processo de colaboração que os trabalhadores encontraram durante a utilização da plataforma e a execução da tarefa. Percebeu-se, basicamente, dois tipos de colaboração: A colaboração entre os membros da multidão e, a colaboração entre o trabalhador e a plataforma.

No que tange a colaboração entre os membros da multidão, nota-se que a percepção dos participantes foi que não houve colaboração.

“4P_IE1: Na tarefa que desenvolvi havia bastante interação entre o cliente e os membros inscritos na tarefa dentro do fórum de discussões. Foram criados vários tópicos pelos membros com dúvidas que foram solucionadas rapidamente pelo cliente. Então acredito que houve colaboração entre as essas duas pontas. Porém, fiquei com a impressão de não existir nenhuma colaboração direta entre desenvolvedores. Alias, a própria natureza da plataforma, competição, faz com que esse tipo de colaboração seja desencorajada, o que na minha opinião é algo negativo”

Reforçando a teoria que não houve colaboração entre os membros da plataforma, muitos participantes foram taxativos ao afirmar, como é o caso do participante 7P_IE1.

“7P_IE1: Entre os membros da plataforma: nenhuma dentro da plataforma.”

Entretanto, é interessante observar que as percepções dos trabalhadores foi de que não houve colaboração entre eles, porém, constatou-se que a maioria dos participantes buscou informações sobre a tarefa nos fóruns da plataforma. Assim, pode-se entender que quando um trabalhador busca informações na plataforma postadas por alguém, é possível considerar esta ação de busca como um processo de compartilhamento de dados. Então, quando alguém compartilha algo útil para outrem, deduz-se que houve colaboração.

“8P_IE1: Na atividade que desenvolvi, eu recorri ao fórum para encontrar mais detalhes, visto que a descrição da atividade era muito sucinta. No fórum, o dono da atividade postou algumas regras de negócios que não estavam na descrição da tarefa. Também li algumas postagens com dúvidas de outros desenvolvedores, e as respostas enviadas me ajudaram a obter um melhor entendimento. Se considerarmos o fórum como parte da plataforma, podemos dizer que existe colaboração entre a plataforma e os membros dela. Já a interação entre os membros, vi pouca colaboração, pois na atividade que trabalhei, eu apenas consumi as respostas do dono da atividade”

“4P_IE2: [...] fiquei com a impressão de não existir nenhuma colaboração direta entre desenvolvedores.”

Fato positivo e relevante, mesmo não sendo colocado como uma barreira, é que os participantes perceberam a colaboração entre eles e a plataforma por meio do canal Fórum. Percebe-se, com isso, uma preocupação da plataforma em estreitar uma relação e interação com os participantes.

“4P_IE1: Na tarefa que desenvolvi havia bastante interação entre o cliente e os membros inscritos na tarefa dentro do fórum de discussões. Foram criados vários tópicos pelos membros com dúvidas que foram solucionadas rapidamente pelo cliente. Então acredito que houve colaboração entre as essas duas pontas. Porém, fiquei com a impressão de não existir nenhuma colaboração direta entre desenvolvedores. Alias, a própria natureza da plataforma, competição, faz com que esse tipo de colaboração seja desencorajada, o que na minha opinião é algo negativo.”

“5P_IE1: “Achei interessante que, mesmo em uma tarefa F2F, os membros da plataforma se ajudaram compartilhando informações no fórum, dando dicas, etc.”

“6P_IE1: “Sem dúvida a colaboração foi essencial e funcionou de forma objetiva. Especificamente na atividade que eu escolhi não encontrei os arquivos para download e busquei estas informações no Fórum da atividade.”

Desta forma, o fórum foi considerado um importante canal de comunicação entre os trabalhadores e a plataforma, principalmente com o objetivo para elucidar dúvidas do trabalhador e auxílio no processo de documentação na interpretação da tarefa.

“5P_IE1: No caso da tarefa que realizei, o cliente pareceu bem disponível, interagindo bastante no fórum e respondendo as dúvidas dos participantes”

“11P_IE1: “Em minha experiência a única colaboração entre membros e o cliente ocorreu em um post no fórum, onde um dos membros havia identificado que faltava ‘um arquivo no pacote disponibilizado para download”

3P_IE2: “Em relação a forma de comunicação e interação, as atividades se dão por meio de fóruns”.

Na Categoria Colaboração foram encontradas evidências das seguintes barreiras:

1. Dificuldade na comunicação com outros membros
2. Identificar outros membros é difícil

A Tabela 10 mostra as barreiras da categoria Colaboração de acordo com a fonte (origem) dos dados em respectivas Fases (1 e 2).

Tabela 10: Origem das Barreiras da Categoria Colaboração por Fonte de Dados

Barreiras da Categoria Colaboração	Origem		
	Fase 1	Fase 2	
		IE1	IE2
Dificuldade na comunicação com outros membros		•	
Identificar outros membros é difícil		•	•

Legenda: IE1: Instituição de Ensino 1 / IE2: Instituição de Ensino 2

c) Categoria Plataforma

A categoria Plataforma reúne as dificuldades em geral apresentadas pelos trabalhadores no que tange ao uso da plataforma, nos aspectos gerais sobre a tarefa – desde seu entendimento até a sua execução, além de aspectos pessoais para realizar a tarefa.

Mesmo considerando a plataforma Topcoder como uma das mais importantes do modelo crowdsourcing, vários participantes comentaram que a mesma não facilita a comunicação entre os membros participantes de uma tarefa. É importante recordar que

tanto no desenvolvimento de software quanto no modelo software crowdsourcing, desenvolver software é uma atividade, em sua essência, comunicativa.

“1P_IE1: “topcoder não facilita o conhecimento/interação entre os usuários”

“2P_IE1: “Acredito que a plataforma deveria possibilitar uma forma de garantir a interação entre o demandante da tarefa e os participantes além de fórum, como um chat com horário marcado para questionamentos, assim quem tem dúvidas poderia usar esses “pontos de encontros”.

12P_IE1: “identificar outros membros na plataforma é bem difícil.”

6P_IE2: “Ao fazer o uso da plataforma TopCoder, tive as seguintes percepções da mesma. Em relação a facilidade de encontrar atividades para realizar, tive um pouco de dificuldades, devido a plataforma utilizar o idioma inglês, no qual não possuo fluência, além de envolver diversas linguagens, frameworks que não possuo conhecimento necessário para realiza-las.”

7P_IE2: “Observou-se que muitas tarefas são complexas, e principalmente dispõem de pouco tempo para finalização, o que se aproxima muito de modelos reais de desenvolvimento de software. Outra observação, é que a plataforma não é para usuários que buscam conhecimento em alguma linguagem de programação, pois, e sua maioria os desafios são complexos e com pouco tempo para trabalhar.”

A falta de retorno das tarefas enviadas pelo trabalhador, com comentários ou observações foi citada como um aspecto muito negativo. Percebeu-se uma fraqueza da plataforma no processo de retorno (*feedback*). Mesmo sendo o fórum o principal canal de comunicação entre os participantes, alguns sugerem a existência de outros canais como a utilização da comunicação síncrona, como por exemplo, o *chat*.

“6P_IE2: Falta de interações e comunicações mais ágeis, como por exemplo chat com quem pôs a atividade, ficando apenas refém dos fóruns e documentações disponíveis.”

“1P_IE1: A documentação e onboarding da tarefa que eu executei eram muito pobres, tive que buscar informações em todos os cantos (descrição da tarefa, fóruns, mockups, Google Drive disponibilizado, código) para ter o entendimento total do que estava sendo proposto como tarefa e também para configuração do ambiente. Nesse sentido tive dificuldades na busca por informação e entendimento da tarefa. Por ter sido uma tarefa com prazo curto, o processo de revisão e appeal foi muito curto, e não pude interagir com as pessoas que fizeram a revisão do meu código.”

“1P_IE1: “achei confuso o processo de revisão”

“10P_IE1: Na utilização da plataforma achei confuso o processo de revisão por dois motivos, o primeiro é que as telas do sistema de revisão da TopCoder são bem diferentes das telas do Dashboard por exemplo... nem parece o mesmo sistema se não fosse pelo logotipo. O segundo ponto foi o processo de revisão como um todo, o qual eu estava esperando receber feedback do código escrito e num segundo momento poder alterar o código e submeter novamente, mas isso não aconteceu.”

Um destaque negativo da plataforma muito comentado foi a sua usabilidade. Usabilidade pode ser entendida pela definição da ISO (*International Organization for*

Standardization), “a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico”. Como citado, vários (24) participantes relataram esse aspecto. Um trecho coletado de um participante pode ser observado a seguir:

“19P_IE1: Não sei como funciona a plataforma para os clientes, porém, acredito que se a mesma fosse mais simples e indicasse de forma mais clara onde o código fonte do projeto deveria estar este problema poderia ser evitado.”

Na categoria Plataforma foram encontradas evidências das seguintes barreiras:

1. A plataforma é lenta
2. Difícil utilizar a plataforma (usabilidade)
3. É confuso utilizar a plataforma
4. É confuso processo de revisão
5. Novatos sentem dificuldades em usar a plataforma
6. Pouco feedback dado pela plataforma.
7. A plataforma Topcoder não facilita a comunicação.
8. Um problema é a não existência de um chat para comunicação

A Tabela 11 mostra as barreiras da categoria Plataforma de acordo com a fonte (origem) dos dados em respectivas Fases (1 e 2).

Tabela 11: Origem das Barreiras da Categoria Plataforma por Fonte de Dados

Barreiras da Categoria Plataforma	Origem		
	Fase 1	Fase 2	
		IE1	IE2
A plataforma é lenta			•
Difícil utilizar a plataforma (usabilidade)	•	•	•
É confuso utilizar a plataforma		•	•
É confuso processo de revisão	•	•	•
Novatos sentem dificuldades em usar a plataforma		•	
Pouco feedback dado pela plataforma.	•	•	•
A plataforma Topcoder não facilita a comunicação.		•	•
Um problema é a não existência de um chat para comunicação		•	•

Legenda: IE1: Instituição de Ensino 1 / IE2: Instituição de Ensino 2

d) Categoria Pessoal

A categoria Pessoal reúne as dificuldades pessoais enfrentadas pelo trabalhador no uso da plataforma, aspectos gerais sobre a tarefa além de, principalmente, aspectos pessoais para realizar a tarefa.

O segundo aspecto negativo mencionado – com um número significativo de ocorrências (15) – foi preparar o ambiente para entender/aprender ou fazer a tarefa.

“12P_IE1: [...] O segundo foram ambientes complexos que se gastava muito tempo para serem montados, com muitas dependências, ao meu ver, sem uma ligação direta com a tarefa referida na descrição.”

Novamente, compreender o idioma inglês foi apontado pelos participantes como um problema. Percebe-se aqui que, os participantes da unidade de análise I (IE1) tiveram menos problemas com o idioma inglês quando comparados com os participantes da unidade de análise II (IE2).

“18P_IE1: “Outro fator que pode ser complicador é o idioma, as vezes com palavras e expressões técnicas, ou na conversa com outro usuário que não escreva de maneira correta.”

“2P_IE2: Ao fazer o uso da plataforma TopCoder, tive as seguintes percepções da mesma. Em relação a facilidade de encontrar atividades para realizar, tive um pouco de dificuldades, devido a plataforma utilizar o idioma inglês, no qual não possuo fluência, além de envolver diversas linguagens, frameworks que não possuo conhecimento necessário para realiza-las.”

3P_IE2: “devido a plataforma utilizar o idioma inglês”

7P_IE2: “é que faz uso apenas da língua inglesa, o que para algumas pessoas pode dificultar ou até mesmo impossibilitar o entendimento da tarefa (dependendo da força de vontade de cada um)”

8P_IE2: “Um ponto negativo é a questão da língua inglesa”

O tipo da tarefa selecionada pelo participante também foi apontada como um problema como reporta:

“11P_IE1: [...] A tarefa selecionada era do tipo First to Finish, mas foram aceitas 63 submissões antes do encerramento, mesmo que apenas o primeiro a enviar tenha sido realmente o vencedor do desafio.”

As barreiras identificadas na categoria Pessoal são descritas a seguir.

1. Dificuldade em encontrar uma tarefa em aberto de acordo com minha habilidade.
2. Dificuldade em entender a tarefa devido a minha limitação do conhecimento.
3. Dificuldade em entender a tarefa devido a minha limitação do conhecimento da língua inglesa
4. Dificuldade para preparar o ambiente para fazer a tarefa
5. Dificuldade na gestão da tarefa (aprender, fazer)

A Tabela 12 mostra as barreiras da categoria Pessoal de acordo com a fonte (origem) dos dados em respectivas Fases (1 e 2).

Tabela 12: Origem das Barreiras da Categoria Pessoal por Fonte de Dados

Barreiras da Categoria Pessoal	Origem		
	Fase 1	Fase 2	
		IE1	IE2
Dificuldade em encontrar uma tarefa em aberto de acordo com minha habilidade.	•	•	•
Dificuldade em entender a tarefa devido a minha limitação do conhecimento.	•	•	•
Dificuldade em entender a tarefa devido a minha limitação do conhecimento da língua inglesa	•	•	•
Dificuldade para preparar o ambiente para fazer a tarefa		•	•
Dificuldade na gestão da tarefa (aprender, fazer)		•	•

Legenda: IE1: Instituição de Ensino 1 / IE2: Instituição de Ensino 2

e) Categoria Tarefa

A categoria Tarefa apresenta aspectos relacionados à tarefa apresentados pelos trabalhadores durante a utilização da plataforma e a execução da tarefa, desde o entendimento (demanda cognitiva) até a montagem por parte do trabalhador de uma configuração computacional para executá-la, e a execução da tarefa propriamente dita. O principal problema da tarefa relaciona-se a má definição da mesma. Vários participantes comentaram que não conseguiram entender o objetivo da tarefa. Interessante observar que os participantes escolheram a tarefa conforme suas habilidades técnicas e, mesmo assim, utilizaram o fórum para entender mais sobre a tarefa.

Pela falta ou inexistência de documentação sobre as tarefas, os participantes relatam a importância do Fórum para a busca de informações complementares.

“1P_IE1: Nos fóruns do desafio não identifiquei nenhum tipo de colaboração com relação a construção do código, somente dúvidas relacionadas ao desenvolvimento da tarefa, que não estavam bem descritas nos detalhes da tarefa.”

“5P_IE1: [...] Em resumo, achei as especificações muito pobres e considero algo negativo o cliente ter tido que complementá-las dias depois do início da tarefa no fórum. Também achei que falou um feedback melhor após a submissão da solução.”

“8P_IE1: Na atividade que desenvolvi, eu recorri ao fórum para encontrar mais detalhes, visto que a descrição da atividade era muito sucinta.”

Preparar o ambiente para fazer a tarefa, aliada ao grau de complexidade comparada ao baixo retorno financeiro foram outros problemas comentados pelos participantes. Assim, é fundamental que os participantes consigam encontrar, entender e fazer a tarefa.

“9P_IE1: [...] achei a tarefa bastante complexa e exigente para o prêmio que estava sendo pago.”

“13P_IE1: Na realização da tarefa, tive algumas dificuldades técnicas relacionadas à tecnologia que escolhi utilizar para desenvolver a solução, entretanto foi possível encontrar respostas para tais dificuldades por meio de consultas à documentação da própria tecnologia e fóruns de discussão online.”

As barreiras da categoria Tempo são:

- Tarefa mal definida ou mal elaborada
- A documentação associada a tarefa é insuficiente ou inexistente
- Alto grau de complexidade da tarefa comparada ao baixo retorno financeiro

A Tabela 13 mostra as barreiras da categoria Tarefa de acordo com a fonte (origem) dos dados em respectivas Fases (1 e 2).

Tabela 13: Origem das Barreiras da Categoria Tarefa por Fonte de Dados

Barreiras da Categoria Tarefa	Origem		
	Fase 1	Fase 2	
		IE1	IE2
Tarefa mal definida ou mal elaborada	•	•	•
A documentação associada a tarefa é insuficiente ou inexistente	•	•	•
Alto grau de complexidade da tarefa comparada ao baixo retorno financeiro		•	

Legenda: IE1: Instituição de Ensino 1 / IE2: Instituição de Ensino 2

Por fim, ao elaborar as categorias e especificar suas propriedades e lacunas percebeu-se o surgimento das primeiras relações entre as categorias. Importante registrar que a relação entre as categorias emergiu após a seleção da categoria central “Pessoal”. A categoria que apareceu frequentemente nos dados foi denominada de central. É importante registrar que a categoria central “Pessoal”, foi eleita devido ao escopo da presente tese que é investigar as barreiras individuais dos trabalhadores na multidão quanto executam tarefas em projetos de software crowdsourcing no modelo competitivo.

O dendograma, que pode ser observado na Figura 18, apresenta a categoria central “Pessoal”, ou seja, todas as outras categorias importantes estão relacionadas com ela. Percebe-se que o primeiro nível de similaridade (ou distância) da categoria central foi com a categoria “Plataforma”, e isto ocorreu porque as principais dificuldades apontadas pelos participantes foi justamente envolvendo questões de usabilidade da Plataforma, depois com problemas de entendimento, documentação e execução da categoria “Tarefa”. As categorias “Colaboração” e “Tempo” possuem o mesmo nível de similaridade com a categoria “Tarefa”.

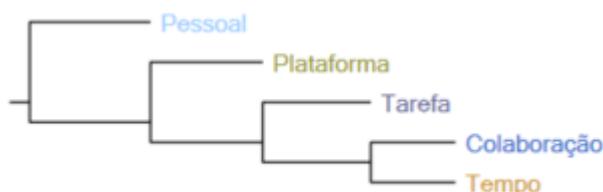


Figura 18: Categoria Central e suas relações

A Figura 19 mostra uma representação das barreiras que emergiram em cada uma das categorias.



Legenda: F1: Barreira também presente na Fase 1

Figura 19: Categorias e suas Barreiras que emergiram da Codificação aberta

Enfim, cada elemento do crowdsourcing (cliente, plataforma, tarefa, multidão) foi alocado como uma dimensão das categorias. A utilização do método comparativo constante auxiliou para a realização da codificação aberta. Percebe-se que ao final dessa etapa, o não surgimento de novos códigos, novas descrições ou novas categorias. Isto auxiliou que, e em torno dessas, as outras categorias foram integradas, para, posteriormente realizar-se a construção inicial da narrativa descritiva do fenômeno de estudo.

4.2.5. Propostas de Recomendações

Além das barreiras identificadas, solicitou-se aos participantes que apontassem ou sugerissem algumas recomendações que pudessem minimizar as barreiras, ou dificuldades enfrentadas durante a execução da tarefa no projeto realizado. Os resultados dessa fase podem ser observados em Machado *et al.* [134]

A questão que os participantes responderam foi “*Quais sugestões você propõe para minimizar estas dificuldades?*” Com base nos dados (relatório 2) foram feitas algumas recomendações separadas em três categorias: Usabilidade da Plataforma, Gestão da tarefa e Feedback (Retorno da atividade) [131].

Para a categoria “Usabilidade da Plataforma” as sugestões foram:

- Tornar o layout do sistema de revisão mais integrado com o painel da plataforma;
- Padronizar as necessidades relacionadas à localização e download de arquivos de tarefas;
- Melhorar os padrões de interação com outros usuários;
- Maior transparência e interação no fluxo de trabalho da tarefa (por exemplo, status de cada etapa);

Para a categoria “Gestão da tarefa” as sugestões foram:

- Indicar um ambiente de desenvolvimento para uma tarefa;
- Integrar tarefas com outras ferramentas de desenvolvimento e repositório colaborativo (Github, Stackoverflow, etc);
- Atualizar a documentação dos requisitos da tarefa (quando alterada) no fórum;
- Melhorar o processo de divisão e descrição das tarefas
- Recomendar novas tarefas com base no perfil dos participantes

Para a categoria “Feedback” as sugestões foram:

- Melhorar o processo de revisão de código;
- Fornecer um canal de comunicação síncrono.

Por fim, é importante observar que essas recomendações se referem a todos os elementos do crowdsourcing e não foram avaliadas ou validadas, porém, servirão de base para a confecção das recomendações finais, um dos objetivos da presente tese.

4.3. Barreiras identificadas apenas para os indivíduos da multidão

Essa seção apresenta os resultados das barreiras específicas do indivíduo da multidão. É importante destacar que o processo de seleção das barreiras relativas ao trabalhador na multidão, ocorreu utilizando o constante método comparativo, além, da efetiva participação de três revisores e de uma revisita às codificações já realizadas.

Foram separadas ou selecionadas apenas as categorias e barreiras relacionadas ao indivíduo da multidão. O processo de seleção das barreiras específicas do trabalhador ocorreu da seguinte forma: as barreiras identificadas foram apresentadas a dois revisores que, individualmente, e a seu critério de livre escolha, pudessem categorizar as barreiras em algum (um ou mais) elemento do crowdsourcing (cliente, plataforma, multidão e tarefa). As barreiras que não foram categorizadas por, ao menos dois especialistas no indivíduo do elemento Multidão foram excluídas. Após realizou-se uma discussão em grupo para a categorização final.

Salienta-se, portanto, que o foco no indivíduo do elemento Multidão ocorre porque, justamente, o objetivo da presente tese é identificar as barreiras que os trabalhadores da multidão enfrentam quando no registro e submissão de tarefas em projetos de software crowdsourcing e propor um conjunto de recomendações que o próprio indivíduo possa realizar sem a intervenção direta ou indireta da plataforma ou do cliente.

A Figura 20 apresenta as 10 barreiras (específicas do trabalhador da multidão) resultantes da análise realizada pelos especialistas. As barreiras foram novamente categorizadas e divididas assim: a) Dificuldade na Gestão do Tempo, b) Falta de Competência e c) Dificuldade na Colaboração. A letra “B” seguida por um número, em sequência, localizada no início de cada barreira, indica um identificador de cada barreira.

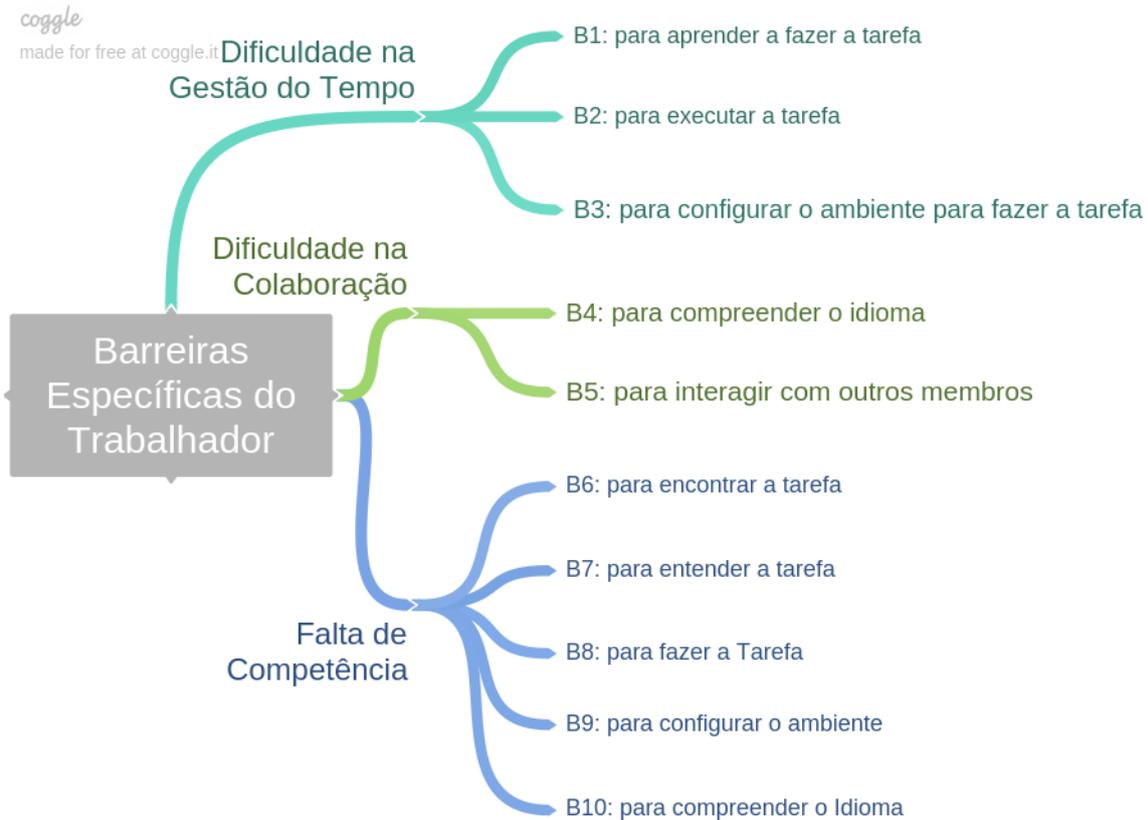


Figura 20: Categorias e Barreiras do trabalhador na multidão

Enfim, desenvolver projetos de software crowdsourcing são atividades colaborativas nas quais o trabalhador na multidão necessita de conhecimento prévio sobre a tarefa que irá executar, bem como, uma eficiente gestão do seu tempo pessoal e disponível para executar uma tarefa. Ressalta-se que muitos problemas em projetos de software crowdsourcing são potencializados quando comparados ao desenvolvimento de software tradicional [99], [135], [136], [137].

4.4. Avaliação das Barreiras específicas do indivíduo da multidão

De posse de um conjunto de barreiras específicas para o trabalhador na multidão, fez-se necessário a avaliação dessas barreiras. Portanto, realizou-se um novo estudo de caso, seguido da aplicação de um questionário aos participantes deste estudo. Foi utilizada a apenas a análise quantitativa para avaliação dos resultados.

4.4.1. Coleta de dados para a avaliação das Barreiras

Para avaliar as barreiras exclusivas do trabalhador da multidão fez um novo estudo de campo. Novamente esse procedimento ocorreu em um projeto na disciplina

“Desenvolvimento de Software Colaborativo” do Programa de Pós-Graduação em Ciência da Computação de uma Instituição de Ensino Superior. Durante as aulas os acadêmicos participam de aulas expositivas, grupos de discussão, apresentação de artigos relacionados e, por fim, da realização de um projeto final da disciplina. Na realização deste estudo de campo, utilizou-se novamente a plataforma Topcoder, bem como, foram repetidas as instruções apresentadas no estudo de caso executado na Fase 2. O período de execução, juntamente com as análises, foi entre os meses de agosto a dezembro de 2017.

As descrições das barreiras foram reescritas para evitar o viés dos respondentes. Por exemplo, as barreiras B3 e B9 foram agrupadas na sentença 4, e as barreiras B1 e B8 foram agrupadas na sentença 5. Percebe-se aqui, uma diminuição da quantidade de barreiras de 10 para 8, sem prejuízo em sua interpretação. Esse processo passou pela avaliação de um revisor especialista. A seguir, na Tabela 14 apresenta-se a transformação da descrição das barreiras em sentenças que foram apresentadas aos participantes.

Tabela 14: Transformação das barreiras em sentenças

Barreiras	Sentenças
B1: Dificuldade na gestão do tempo para aprender a fazer a tarefa E B8: Falta de competência para fazer a tarefa	BA5 - Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para executá-la.
B2: Dificuldade na gestão do tempo para executar a tarefa	BA8 - Foi difícil gerenciar o meu tempo disponível para executar a tarefa.
B3: Dificuldade na gestão do tempo para configurar o ambiente para fazer a tarefa E B9: Falta de competência para configurar o ambiente	BA4 - Tive problemas para configurar o ambiente necessário para realizar a tarefa.
B4: Dificuldade na colaboração para compreender o idioma	BA6 - Ao realizar a tarefa, evitei a colaboração com os demais membros da plataforma, pois eles se comunicam apenas em inglês.
B5: Dificuldade na colaboração para interagir com outros membros	BA7 - Tive dificuldades na colaboração com os demais membros da plataforma ao realizar a tarefa.
B6: Falta de competência para encontrar a tarefa	BA2 - Foi difícil encontrar uma tarefa para executá-la de acordo com minhas habilidades ou perfil.
B7: Falta de competência para entender a tarefa	BA3 - Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para compreender a tarefa.
B10: Falta de competência para compreender o Idioma	BA1 - Como a plataforma está em inglês, tive dificuldades para compreender a tarefa.

Legenda: BA-Barreira Avaliada.

4.4.2. Resultados da avaliação das Barreiras

Para o processo de avaliação das Barreiras, participaram 21 pessoas, 85% (18) com ao menos 6 anos de experiência no mercado de desenvolvimento de software, e 90% (19) participam pela primeira vez no desenvolvimento de software no modelo crowdsourcing

competitivo. Todos os envolvidos no estudo de caso concordaram com as cláusulas presentes no “Termo de Consentimento” que pode ser observado no Apêndice D.

Importante observar que o planejamento e execução dessa fase ocorreu conjuntamente a realização de atividades do mestrando (na época) do PPGCC Luiz Fernando Vaz. Assim, os participantes responderam a seguinte questão: “*Como você percebe as seguintes barreiras quando da realização da tarefa na plataforma Topcoder?*”

Após a execução das tarefas, os participantes avaliaram as barreiras encontradas por meio de um questionário com a utilização de uma escala Likert de 5 pontos com neutro, com as seguintes opções: (1) Discordo fortemente, (2) Discordo, (3) Nem concordo nem discordo, (4) Concordo e, por fim, (5) Concordo fortemente. Uma discussão em grupo também foi realizada para auxiliar no processo de entendimento das dificuldades encontradas pelos participantes.

A Tabela 15 mostra a frequência (ordenada pela Moda) das respostas por barreiras, na perspectiva do trabalhador na multidão.

Tabela 15: Frequência das barreiras.

Barreiras	Moda	1	2	3	4	5
BA8 - Foi difícil gerenciar o meu tempo disponível para executar a tarefa.	5	3	1	0	5	12
BA2 - Foi difícil encontrar uma tarefa para executá-la de acordo com minhas habilidades ou perfil.	5	1	3	1	6	10
BA4 - Tive problemas para configurar o ambiente necessário para realizar a tarefa.	5	4	2	6	2	7
BA5 - Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para executá-la.	4	2	5	2	9	3
BA7 - Tive dificuldades na colaboração com os demais membros da plataforma ao realizar a tarefa.	3	7	0	14	0	0
BA3 - Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para compreender a tarefa.	2	6	8	1	3	3
BA6 - Ao realizar a tarefa, evitei a colaboração com os demais membros da plataforma, pois eles se comunicam apenas em inglês.	1	11	0	9	0	1
BA1 - Como a plataforma está em inglês, tive dificuldades para compreender a tarefa.	1	15	3	1	2	0

Legenda: BA: Barreira Avaliada / (1) Discordo fortemente, (2) Discordo, (3) Nem concordo nem discordo, (4) Concordo (5) Concordo fortemente.

Os resultados mostram que a barreira que mais dificultou os participantes durante a execução da tarefa foi a dificuldade em gerenciar o seu tempo pessoal (Barreira 8). Isso ocorre porque o grupo de estudantes que participaram do estudo, como acontece na multidão do Topcoder de acordo com Yang *et al.* [35], não possui dedicação exclusiva na plataforma, trazendo, consigo, um provável viés.

Observa-se que, para maioria dos participantes, cerca de 60% (13) (Concordam fortemente) foi difícil encontrar uma tarefa de acordo com suas habilidades (Barreira 2). Isso pode ter ocorrido porque se faz necessário que o trabalhador na multidão seja

proficiente em determinadas tecnologias, com isso, pode excluir ou diminuir sua possibilidade de participar em determinadas competições. Para cerca de 56% (12) dos participantes, a dificuldade para preparar o ambiente computacional para executar a tarefa (Barreira 4), foi apontada como uma importante barreira. Entretanto, e nesse caso, vale destacar que esta barreira está diretamente relacionada ao tipo de tarefa, visto que, determinadas tarefas não necessitam de um ambiente específico (por exemplo, instalar alguma ferramenta) para a sua execução.

Os participantes tiveram dificuldades em entender e executar a tarefa (Barreira 5 e Barreira 3) mesmo escolhendo uma tarefa de acordo com suas habilidades e preferências pessoais. Isso ocorre devido a natureza intrínseca do software crowdsourcing, isto é, o trabalhador na multidão encontra, muitas vezes, além do fracionamento da tarefa, um contexto na qual a tarefa encontra-se um tanto escasso. Isso ocorre, pois, as tarefas são independentes e intencionalmente descritas superficialmente devido as regras de negócios específicas dos clientes que as delegam.

Mesmo considerando que esse estudo de campo foi executado por participantes brasileiros, a barreira sobre a falta ou carência no conhecimento da língua inglesa (Barreira 6), foi observada apenas por 5% (1) como um problema no que tange a dificuldade em interagir com outros participantes.

Com relação aos participantes que relataram forte discordância para evitar colaborar com outros membros por meio do idioma inglês, mesmo assim, é possível considerar que os participantes colaboraram no idioma inglês via fórum de discussões (postam mensagens ou simplesmente leem e coletam informações das postagens de outros participantes da tarefa). Para os participantes que responderam como “Nem concordo e nem discordo” pode-se interpretar que eles não se sentem confortáveis em comunicar-se via mensagens de texto utilizando uma linguagem não nativa. Isto pode ser observado Figura 21. (Barreira 6).

Na Figura 21, pode-se observar que 70% (15) dos participantes não acharam difícil entender as especificações e requisitos da tarefa que estavam escritos em inglês (Barreira 1). Isso é interessante porque contrasta com resultados de fases anteriores, nos quais os participantes relataram o idioma não nativo como uma barreira de colaboração. Portanto, seria pertinente explorar este aspecto para entender melhor os fatores que contribuem para a compreensão das especificações ou requisitos das tarefas.

Os resultados do grau de concordância podem ser observados na Figura 21.

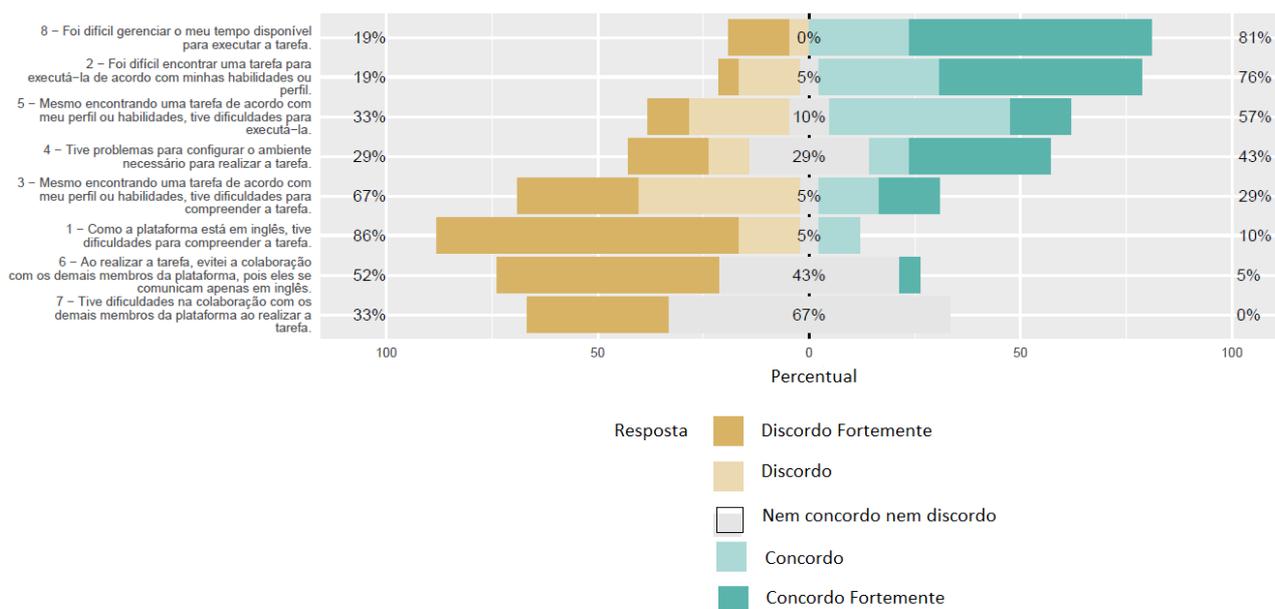


Figura 21: Resultados dos graus de concordância

4.5. Reflexões sobre o capítulo

Por meio da efetiva utilização de diversos métodos empíricos foi possível identificar as barreiras enfrentadas pelos trabalhadores na multidão quando da realização de tarefas em projetos de software crowdsourcing no modelo competitivo.

Barreiras relacionadas ao Tempo, à Colaboração, a Plataforma, a Tarefa, e, por fim, dificuldades de ordem pessoal ou individual. Enfim, barreiras presentes em todos os quatro elementos do crowdsourcing. Também foram separadas as barreiras específicas do trabalhador.

No que se refere a dificuldades específicas do trabalhador da multidão, percebeu-se em linhas gerais, aspectos relacionados a gestão do pessoal, a colaboração e a falta de competência. Ao avaliar as barreiras, percebeu-se uma grande dificuldade, por parte dos trabalhadores na multidão na gestão do tempo pessoal para realizar as atividades propostas pela plataforma, bem como, a dificuldade em encontrar uma tarefa para executá-la de acordo com as habilidades do trabalhador. Merece atenção problemas para configurar o ambiente necessário para realizar a tarefa. Por outro lado, não tiveram empecilhos com o idioma inglês, tanto para entender a tarefa, quanto para comunicar-se com demais membros da plataforma. Como limitações destes estudos, destacam-se que os indivíduos que participaram desse estudo pertencem ou estão alocados em uma determinada região do estado do Rio Grande do Sul no Brasil, são brasileiros, e, por fim, as atividades propostas tenham sido realizadas apenas na plataforma Topcoder.

5. RECOMENDAÇÕES PARA CONTRIBUIR EM PLATAFORMAS DE SOFTWARE CROWDSOURCING

O principal objetivo deste capítulo foi responder a segunda questão de pesquisa da presente tese, “QP2: *Quais são as recomendações que auxiliam os trabalhadores da multidão a superarem barreiras e minimizar desistências?*”, bem como o segundo objetivo específico: “*Identificar e avaliar as recomendações que possam auxiliar os trabalhadores na multidão a superarem barreiras identificadas em projetos de software crowdsourcing no modelo competitivo*”.

Avaliadas as categorias e suas respectivas barreiras encontradas pelos trabalhadores na multidão em projetos de software crowdsourcing no modelo competitivo, fez-se a prospecção de recomendações para que esses trabalhadores possam as utilizar para superar as barreiras encontradas. Salienta-se, portanto, que o foco está no indivíduo, no trabalhador da multidão para que ele próprio possa realizar as recomendações sem a necessidade da intervenção da plataforma ou do cliente.

Após, as recomendações foram avaliadas por especialistas em software crowdsourcing por meio de um questionário. A avaliação é um critério imprescindível na Teoria Fundamentada nos Dados para consolidar a pesquisa e mostrar seu rigor científico. Conforme Yin [46] o objetivo da avaliação não é testar o modelo como acontece na Pesquisa Quantitativa, mas sim, comparar os conceitos e suas relações com os dados, e determinar, se estão, em consonância ou apropriados à investigação executada. A avaliação das recomendações está consonância com as relações entre as categorias de barreiras e delas com as recomendações. As etapas desenvolvidas na Fase 3 podem ser observadas na Figura 22.

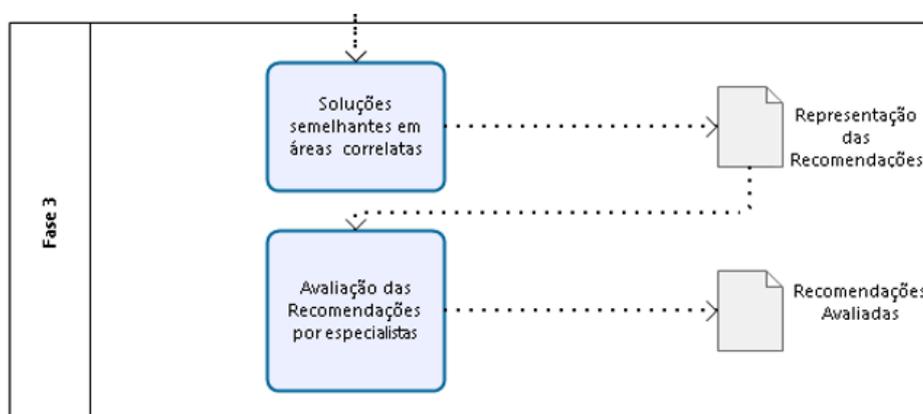


Figura 22: Etapas da Fase 3

5.1. Recomendações aos trabalhadores da multidão

A partir de um conjunto final de barreiras e de uma lista preliminar de barreiras identificadas na Fase 2, foi dada uma atenção especial ao processo de confecção das recomendações. Assim, foram elaboradas 13 recomendações cujo objetivo é auxiliar o trabalhador na multidão a superar as barreiras em projetos de software crowdsourcing no modelo competitivo, ou seja, para cada barreira identificada e avaliada, fez-se uma recomendação. Como as recomendações emergiram de acordo com as barreiras encontradas durante o processo de elaboração das recomendações percebeu-se que algumas dessas recomendações poderiam ser indicadas para uma ou mais barreiras. Isto ocorreu simplesmente porque as recomendações eram similares.

Assim, as barreiras “BA3” e “BA5” foram agrupadas para a Barreira Final (BF) “BF1”, e as barreiras “BA1” e “BA6” foram agrupadas na barreira “BF6”. Percebe-se aqui, novamente uma “redução” da quantidade de barreiras de 8 para 6, sem prejuízo em sua interpretação. Esse processo passou pela avaliação de um revisor especialista. A Tabela 16 apresenta o agrupamento das barreiras para a elaboração das recomendações.

Tabela 16: Agrupamento das Barreiras.

Barreiras Avaliadas	Barreiras Finais
BA2: Foi difícil encontrar uma tarefa para executá-la de acordo com minhas habilidades ou perfil. E BA5: Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para executá-la.	BF1: É difícil encontrar uma tarefa, entendê-la e executá-la de acordo com as habilidades do trabalhador na multidão.
BA3: Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para compreender a tarefa.	BF2: Mesmo encontrando uma tarefa de acordo com as habilidades do trabalhador na multidão é difícil terminar e submeter a tarefa.
BA8: Foi difícil gerenciar o meu tempo disponível para executar a tarefa.	BF3: É difícil gerenciar o tempo disponível para executar uma tarefa.
BA4: Tive problemas para configurar o ambiente necessário para realizar a tarefa.	BF4: É difícil configurar o ambiente necessário para executar a tarefa
BA7: Tive dificuldades na colaboração com os demais membros da plataforma ao realizar a tarefa.	BF5: É difícil colaborar com outros membros da plataforma para auxiliar na execução da tarefa.
BA1: Como a plataforma está em inglês, tive dificuldades para compreender a tarefa. E BA6: Ao realizar a tarefa, evitei a colaboração com os demais membros da plataforma, pois eles se comunicam apenas em inglês.	BF6: Para trabalhadores não fluentes em inglês, e, como a plataforma está nesse idioma, é difícil entender a tarefas. Durante a execução da tarefa, é difícil colaborar com outros membros da plataforma porque eles apenas comunicam-se em inglês.

Legenda: BA-Barreira Avaliada; / BF-Barreira Final

A Tabela 17 mostra as recomendações e as origens que fundamentaram ou serviram de base para a sua confecção. As recomendações descritas a seguir, estão ordenadas de acordo com uma provável execução das tarefas por parte do trabalhador na multidão.

Tabela 17: Recomendações

Descrição da Recomendação	Recomendação incluída baseada:			Barreiras
	Soluções correlatas da literatura	Fases		
		1	2	
Rec#1: O trabalhador na multidão deve utilizar algum mecanismo que o auxilie a encontrar uma tarefa na plataforma de acordo com suas habilidades ou preferências e expectativas de recompensas financeiras pessoais.	[26], [82], [105], [138], [139], [140], [141], [142], [143]	X	X	BF1: É difícil encontrar uma tarefa, entendê-la e executá-la de acordo com as habilidades do trabalhador na multidão.
Rec#2: O trabalhador na multidão deve utilizar algum mecanismo que o auxilie a decompor a tarefa para melhorar o entendimento da mesma	[29], [88], [144], [40], [145], [41], [146], [147], [137], [148]	-	-	BF2: Mesmo encontrando uma tarefa de acordo com as habilidades do trabalhador na multidão é difícil terminar e submeter a tarefa.
Rec#3: O trabalhador na multidão deve avaliar e gerenciar o tempo disponível para aprender como executar a tarefa.	[23], [83], [41], [149], [150]	-	X	BF3: É difícil gerenciar o tempo disponível para executar uma tarefa.
Rec#4: O trabalhador na multidão deve criar seu próprio plano e gerenciar sua agenda para dispor de tempo suficiente para terminar a tarefa.				
Rec#5: O trabalhador na multidão deve definir pequenas atividades para finalizar a tarefa.				
Rec#6: O trabalhador na multidão deve determinar a duração das atividades				
Rec#7: O trabalhador na multidão deve preparar o ambiente para a execução de cada tarefa.				
Rec#8: O trabalhador na multidão deve definir sua agenda para finalizar a tarefa	[32], [131]	-	X	BF4: É difícil configurar o ambiente necessário para executar a tarefa
Rec#9: O trabalhador na multidão deve utilizar máquinas virtuais com Ambientes de Desenvolvimento Integrados (IDEs) para fazer a tarefa				
Rec#10: O trabalhador na multidão deve utilizar um ambiente de desenvolvimento integrados (IDEs) que possua uma Wiki, como apoio para fazer a tarefa				
Rec#11: O trabalhador na multidão deve utilizar ambientes de desenvolvimento integrados (IDEs) sob o conceito de computação em nuvem para fazer a tarefa	[5], [29], [58], [70], [99], [135], [136], [137], [151], [152], [153], [40], [154], [155], [156], [157],	X	X	BF5: É difícil colaborar com outros membros da plataforma para auxiliar na execução da tarefa.
Rec#12: O trabalhador na multidão deve utilizar diferentes canais para comunicar-se sincronamente ou assincronamente (software crowdsourcing é baseado na interação).				
Rec#13: O trabalhador na multidão deve utilizar algum mecanismo de tradução da tarefa para facilitar o entendimento da tarefa e a colaboração com outros membros da plataforma	[58], [76], [158], [159], [160], [161], [162], [163]	X	-	BF6: Para trabalhadores não fluentes em inglês, e, como a plataforma está nesse idioma, é difícil entender a tarefas. Durante a execução da tarefa, é difícil colaborar com outros membros da plataforma porque eles apenas comunicam-se em inglês.

Para a representação das recomendações, buscou-se por soluções semelhantes na literatura de áreas correlatas a partir da Revisão da Literatura (seção 4.1.1) e das sugestões advindas dos participantes durante a execução das Fases 1 e 2. As recomendações foram elaboradas baseadas na validação conceitual das relações entre as categorias de barreiras e delas com o tema central das barreiras do trabalhador na multidão.

5.2. Coleta dos Dados para a avaliação das recomendações

Para o processo de coleta de dados para a avaliação das recomendações, utilizou-se o questionário como instrumento de pesquisa para conhecer a opinião de especialistas se as recomendações sugeridas são suficientes para que os trabalhadores na multidão possam superar as barreiras identificadas.

Para a seleção da população-alvo dos potenciais respondentes (especialistas) adotou-se o seguinte critério: O especialista deve ter participado do comitê científico (incluindo *chair* ou *co-chair*) ou ter publicado artigo científico em ao menos uma das cinco edições (2018, 2017, 2016, 2015 e 2014) do *Workshop on Crowd Sourcing in Software Engineering*. Justifica-se a escolha desse evento por tratar-se do único com a temática relacionada a software crowdsourcing. A busca dos nomes dos potenciais respondentes ocorreu por meio da visita em cada uma das páginas web do evento. Foram excluídos da lista dois revisores do piloto do questionário, além daqueles que participaram em coautoria com o autor da presente tese em trabalhos científicos. Ao total foram selecionados 91 participantes (população-alvo). Para após, de posse destes nomes, fez-se uma coleta do contato dos selecionados como o e-mail, Twitter, página no Facebook, páginas web pessoais e o número de telefone. Foi fundamental obter esses contatos para o envio do questionário.

O questionário, que pode observado no Apêndice I, foi validado por 3 especialistas, e é composto por duas seções com um conjunto de questões abertas e fechadas. A primeira seção contém as barreiras e as sugestões das recomendações para cada barreira. Na segunda seção constam as questões relacionadas aos dados demográficos dos participantes, bem como, uma solicitação se o respondente poderia participar de uma entrevista. Nas questões fechadas as alternativas foram organizadas em escala Likert de 5 pontos com neutro, de maneira que o respondente indique o seu posicionamento diante da afirmação, dispostas assim: (1) Discordo fortemente, (2) Discordo, (3) Nem concordo nem discordo, (4) Concordo e (5) Concordo fortemente. Ou seja, foi utilizada essa escala para auxiliar no entendimento do nível de concordância dos especialistas com relação as

recomendações. Importante observar que ao final de cada afirmação do tipo escala, fez uma pergunta aberta para que o respondente pudesse explicar ou fundamentar a sua escolha.

5.3. Resultados e análise das recomendações

Esta seção tem como objetivo apresentar os resultados dos questionários enviados aos especialistas para a avaliação das recomendações aos trabalhadores da multidão em projetos de software crowdsourcing no modelo competitivo. Inicialmente serão apresentados os dados demográficos dos participantes e, após, as avaliações e análises.

Cada recomendação foi analisada sob duas perspectivas. Na primeira, fez-se uma análise qualitativa das respostas textuais dos especialistas. A segunda foi executada a partir da análise do gráfico de caixa (*Boxplot*) utilizados para avaliar a distribuição empírica dos dados. O gráfico de caixa é formado pelo primeiro e terceiro quartil e pela mediana. Assim, para a categoria “**Concordam**” foram selecionadas apenas as recomendações que receberam o grau de concordância como “*concordam*” ou “*concordam fortemente*”. Para a categoria “**Discordam**” foram selecionadas apenas as recomendações que receberam o grau de concordância “*discorda* ou *discorda fortemente*” e, por fim, para “**Neutro**” foram apenas selecionadas as recomendações que receberam o grau de concordância “*nem concordo nem discordo*”.

Observa-se, também, que por tratar-se de uma análise qualitativa os comentários e as fundamentações feitas pelos respondentes foram considerados para a categorização de cada recomendação. Importante registrar que, ao utilizar a estratégia sequencial exploratória³⁹, os dados quantitativos auxiliaram nas interpretações dos dados qualitativos.

No Apêndice J podem ser observados os gráficos de caixa e de divergência que auxiliaram na interpretação dos graus de concordância dos especialistas com as recomendações propostas.

Os questionários foram enviados, por e-mail, a 91 participantes na segunda quinzena de abril de 2018, com prazo de devolução até 7 de maio de 2018. Retornaram 9 e-mails inválidos. Os modelos de e-mail enviados podem ser observados no Apêndice H.

³⁹ Com exceção das Rec#8, Rec#9 e Rec#10, pois não foram encontrados comentários dos entrevistados.

Dos 82 e-mails válidos com os questionários, retornaram 14⁴⁰ especialistas, desses, cerca de 84% (12) possuem entre dois e cinco anos envolvidos em projetos de software crowdsourcing (Figura 23). Foi excluído da análise um único participante (E3) que, ao responder o questionário, se enquadrava como “não tenho experiência em SWCS”, além de que esse especialista marcou para todas as perguntas fechadas, apenas o item 3 (nem concordo nem discordo) e, também, não respondeu a nenhuma das questões abertas, logo, foi retirado do processo de análise.

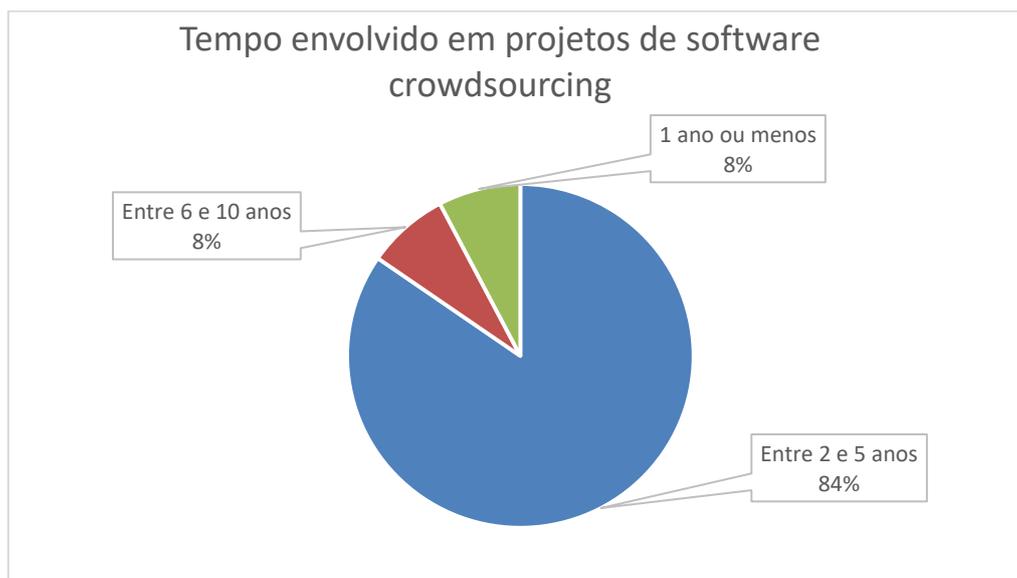


Figura 23: Percentual de tempo de experiência dos especialistas

Para preservar a identidade dos especialistas, cada um foi identificado pela letra “E” que significa “especialista” seguido por um número que indica a ordem de recebimento do questionário. Por exemplo, E1 significa o primeiro entrevistado e assim por diante.

Antes de preencher o questionário, cada especialista foi informado sobre o termo de consentimento das condições da pesquisa. Cada recomendação possui um identificador único com o seguinte formato. “Rec” que significa Recomendação, seguida pelo símbolo # e por um número sequencial. Por exemplo: Rec#1 é o acrônimo de primeira recomendação.

A seguir apresentar-se-á cada recomendação com a atribuição da sua categoria, que representa o grau de concordância dos especialistas.

⁴⁰ Não existe consenso na literatura sobre o tamanho da amostra em análise qualitativa, pois depende do projeto de pesquisa e da saturação teórica e conceitual observada pelo pesquisador. *Flick, Uwe in Baker, S. E. e Edwards, R How many qualitative interviews is enough. (2012) <http://eprints.ncrm.ac.uk/2273/4>*, coletaram a opinião de vários especialistas e apontam: “Thus, to end this Introduction as we began, the answer to ‘How many qualitative interviews is enough’ is ‘it depends’”.

Rec#1 - O trabalhador na multidão deve utilizar algum mecanismo que o auxilie a encontrar uma tarefa na plataforma de acordo com suas habilidades ou preferências e expectativas de recompensas financeiras pessoais.

Esta recomendação tem como objetivo auxiliar o trabalhador na multidão a superar a barreira **BF1: É difícil encontrar uma tarefa, entendê-la e executá-la de acordo com as habilidades do trabalhador na multidão.**

Crowdsourcing caracteriza-se também pela capacidade em atrair um grande número de pessoas com competência para a realização de determinadas tarefas. Ou seja, competência, é uma característica da multidão que auxilia a solução de problemas, a inovação, a co-criação entre outras. Possuir recursos humanos com a competência⁴¹ esperada para a resolução de problemas é algo intrínseco ao desenvolvimento de um projeto, porém, vale lembrar, que é relacionado diretamente às características específicas do trabalhador, sobretudo, o motivo é que basicamente um participante competente resolverá as tarefas complexas e será mais eficiente nas atividades baseadas no crowdsourcing como comentam Hosseini *et al.* [29].

Straub *et al.* [41] reforçam que, em atividades de micro-tarefa, a capacidade do trabalhador é associada diretamente ao processo de completar a tarefa como comentam:

Individual capability is correlated to task completion in a rank-order tournament. Capable workers finish the task more often. (p.10)

Mediocre and strong competitors are correlated to task completion when feedback is given in a rank-order tournament; it is not related to task completion when strength of competitors is weak. Workers quit the task more often when facing stronger competitors. (p.10)

A dificuldade em encontrar uma tarefa pode ser causada por problemas de usabilidade da plataforma pois um sistema de comunidades on-line com “boa” usabilidade deve ter uma estrutura de navegação que permita aos membros navegar facilmente e encontrar o que precisam [144]. Yang *et al.* [35] propuseram um mecanismo dinâmico de apoio à decisão do trabalhador da multidão para orientar os trabalhadores na aceitação das tarefas de desenvolvimento oferecidas pela plataforma Topcoder. Além disso, a busca por

⁴¹ Segundo Philippe Perrenoud – sociólogo suíço - em A Teoria das Competências, Artmed 1999, “as competências são traduzidas em domínios práticos das situações cotidianas que necessariamente passam compreensão da ação empreendida e do uso a que essa ação se destina. Já, habilidades são representadas pelas ações em si, ou seja, pelas ações determinadas pelas competências de forma concreta”.

tarefas que apenas maximizam os ganhos esperados é uma motivação importante no processo de encontrar uma tarefa [145].

Alguns comentários extraídos dos participantes são apresentados a seguir como elemento no processo avaliativo da Rec#1. Para os participantes E1, E6, E10 é válida essa recomendação. O entrevistado E8 alerta que pode ocorrer uma perda de trabalhadores para determinadas tarefas. A evasão de participantes em ambientes de software crowdsourcing é um dos principais problemas enfrentados pelas plataformas [35].

E1: Tools or systems are welcomed to reduce the overhead of selecting tasks.

E6: Finding appropriate tasks certainly seems important

E8: suggesting task based on workers skill set and learning curve will lead to less mismatching and hopefully higher chance of receiving qualified submissions. however this method may lead to losing available workers due to less diversity in terms of available task in case worker wants to explore their skill set in a new area. So it is very important that who is the end user of this question"

E10: Matching task with expertise is an important part of a crowdsourcing platform.

Para o entrevistado "E14", em OSS a procura por projetos realizados por estudantes é frequentemente associada a aprendizagem. Mesmo em ambientes competitivos, a busca por aperfeiçoamento profissional é um dos objetivos da multidão quando participa de tarefas em projetos de software crowdsourcing [84], [85], [6] e [86].

E14: In Open Source projects, newcomers who search for tasks (instead of approaching the project with a task at hand already) are often students. Students also rate learning as an important motivation for joining. Thus, the recommendation sounds very reasonable.

O entrevistado E13 concorda com a recomendação, porém alerta que é responsabilidade da plataforma oferecer esses mecanismos de seleção ou recomendação de tarefas.

E13: Some support for crowdworkers in finding suitable tasks is very useful. But I only selected "3" at this question because I think such task selection/recommendation should be part of crowdsourcing systems by default rather. That is, it should not be the responsibility of the crowdworker to use these task selection systems, but these should be an integral part of crowdsourcing systems (which they are obviously not at this stage).

Utilizar mecanismos que auxiliem o trabalhador a encontrar uma tarefa na plataforma de acordo com suas habilidades é considerado um importante aliado caso o trabalhador busca apenas compensação financeira e não o aprendizado.

E2: For many crowdworkers, financial incentives are equally important.

E5: If you mean "personal learning" by the workers skills, yes, I think a system that helps them filter tasks by their skills/interests would help increasing the overall quality/quantity of workers by making tasks discoverable.

E7: this could be very helpful to separate people interested in the topic or interested in earning money.

Um dos entrevistados comenta que talvez não tenha entendido a questão, mesmo assim, ao assumir que o trabalhador busque compensações financeiras pela realização da tarefa concorda em utilizar algum mecanismo. O entrevistado “E4” não acredita que seja necessário utilizar esses mecanismos que apoiem trabalhador a encontrar uma tarefa na plataforma de acordo com suas habilidades, infelizmente ele não apresenta alguma sugestão.

E4: dont see this as necessary

E9: I'm not sure I fully comprehend the question. Are you talking about a system that balances learning and earning objectives to make some kind of tradeoff between the two? Or are these alternatives (the system optimizes learning, thereby possibly leading to suboptimal earnings)? I'm assuming the latter.

Além da análise dos comentários textuais feitos pelos especialistas, realizou-se uma análise estatística descritiva das respostas (escala Likert), que pode ser observada na Figura 24, cujo objetivo foi conhecer o nível ou grau de concordância desses especialistas com a Recomendação 1 (Rec #1).

Percebe-se ao analisar a Figura 24, que 67% (9) dos especialistas concordam (*Concordo ou Concordo fortemente*), 33% (4) nem concordam nem discordam e, por fim, nenhuma ocorrência de discordância (*Discordo fortemente, Discordo*) com essa recomendação, assim, a maioria dos especialistas confirmam a Rec#1.

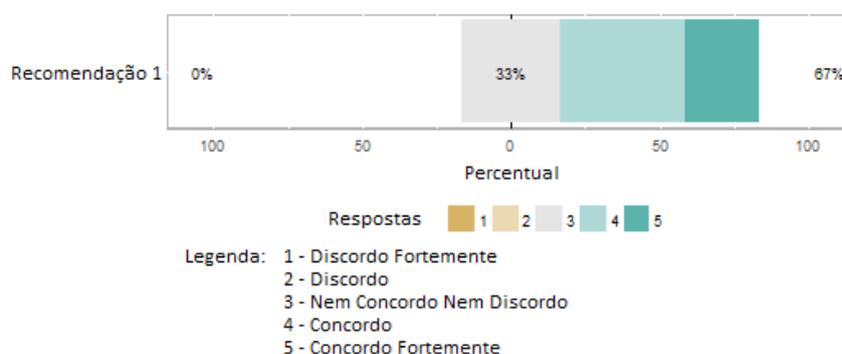


Figura 24: Nível de Concordância com a Rec#1

Diante do exposto, conclui-se que os especialistas “**Concordam**” com a Rec#1, isto é, os trabalhadores da multidão devem ter acesso a algum sistema de recomendação para dar suporte a encontrar tarefas de acordo com seu perfil.

Rec#2 - O trabalhador na multidão deve utilizar algum mecanismo que o auxilie a decompor a tarefa para melhorar o entendimento da mesma

O objetivo dessa recomendação é auxiliar o trabalhador na multidão a superar a barreira **BF2: Mesmo encontrando uma tarefa de acordo com as habilidades do trabalhador na multidão é difícil terminar e submeter a tarefa.**

Para a dificuldade em entender uma tarefa para executá-la, Tajedin e Nevo [105] sugerem que projetos que podem ser decompostos em pequenos módulos com requisitos claros e interdependências limitadas são mais propensos a ter sucesso. No entanto, ao decompor uma tarefa de desenvolvimento de software, é necessário estabelecer um equilíbrio entre o fornecimento de uma especificação suficientemente detalhada para que a tarefa seja executada, em detrimento ao tempo para realizar essa especificação e a própria tarefa. LaToza *et al.* [82] apresentam uma plataforma, denominada de CrowCode para compartilhamento de conhecimento por meio de um sistema de perguntas e respostas. Os autores afirmam que essa plataforma baseada em micro tarefas auxilia no entendimento da tarefa por parte do trabalhador na multidão. Dustdar e Gaedke [138] mostram uma ferramenta que auxilia o gerente de projetos a delegar as tarefas para a multidão conforme o seu perfil e, isso pode contribuir para o entendimento da tarefa pelo trabalhador. Preist *et al.* [140] sugerem que o pagamento baseado na competição “funciona bem” para trabalhadores qualificados. Por fim, e apenas para complementar, Luther *et al.* [139] propõe o uso do modelo crowdsourcing para aumentar o processo de aprendizagem de um indivíduo.

Para fazer a tarefa, a capacidade é substancialmente associada com a finalização da tarefa, além de que apenas trabalhadores aptos a fazer a tarefa, frequentemente a terminam, autores como Straub *et al.* [41] e Satzger *et al.* [137] complementam que em ambientes complexos, é fundamental considerar a experiência do trabalhador na delegação das tarefas. Afirmam ainda que a falta de conhecimento é uma importante fonte de mal-entendidos e erros no desenvolvimento de software tradicional, e, potencializados no software crowdsourcing. Outros estudos como Fershtman e Gneezy [146] e Eriksson *et al.* [147], relatam e reforçam a relação entre a capacidade do trabalhador, incentivado por alguns mecanismos de recompensa, são associados com a finalização de uma tarefa.

A seguir, alguns comentários extraídos dos participantes que auxiliaram no processo avaliativo da Rec#2. Para os participantes E1 e E6, é válido utilizar algum mecanismo que auxilie na decomposição da tarefa para melhorar o seu entendimento, tanto é que, o entrevistado E1 comenta que utiliza essa recomendação com seus alunos. Porém, para o

entrevistado E6 que, mesmo concordando com a recomendação pensa que a implementação dessa recomendação é muito difícil de ser implementado.

E1: That's what we do in a software engineering capstone course where I participe. We encourage students to take fine-grained tasks.

E6: I generally would agree, but it can be really difficult to break down tasks. Depends on the context I guess

E10: Decomposing too much would not be good for quality control. A task should take half a day to one week to complete ideally with a relatively independent chunk of work.

E11: I agree that it is up to the requesters to ensure clarity of the task requirements.

Para o entrevistado E7, deve haver um equilíbrio de informações da tarefa, ou seja, entre o contexto e o escopo da tarefa para não sobrecarregar tanto a tarefa quanto o provável executor da mesma.

Para os entrevistados E5, E8 e E9 é dever da plataforma ou do contratante realizar a decomposição da tarefa, não cabendo ao trabalhador realizar a decomposição.

E5: The question is a little bit confusing. You mention "the crowdworker should decompose..." and then "It's the requester's responsibility to decompose a task...". So you mean the REQUESTER or WORKER? If you're talking about the worker, I believe more research is needed on that area. If the requester breaks down the tasks into microtasks, I believe it would help reducing ambiguity. However, it would be interesting to assign the decomposition itself to a crowd. Meaning there would be 2 stages: 1) the crowd define the microtasks 2) the crowd works on each microtask.

E8: this recommendation may have some overlap with hiding the intelligence of the project

E9: "There's different statements on this page. I'm not sure which one I should rate. As a requestor: Why should I require the crowdworkers to fulfill the tasks in any particular way? I may not want to decompose tasks, because decomposition may be part of the task.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 2 (Rec #2) podem ser observados na Figura 25. Percebe-se ao analisar a Figura 25 **Erro! Fonte de referência não encontrada.**, que apenas 25% (3) dos especialistas concordam (*Concordo ou Concordo fortemente*), 25% nem concordam nem discordam e, por fim, 50% (6) de discordância (*Discordo fortemente, Discordo*) com essa recomendação.

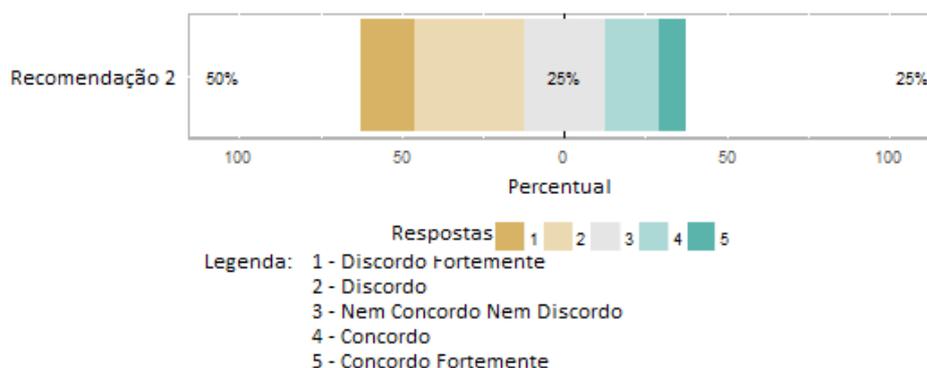


Figura 25: Nível de Concordância com a Rec#2

Assim, e diante do exposto os especialistas **“Discordam”** da Rec#2. Isto é, os especialistas refutam a recomendação de que o trabalhador na multidão deve utilizar algum mecanismo que o auxilie a decompor a tarefa para melhorar o entendimento da mesma. Os especialistas recomendam que é dever ou obrigação da plataforma ou do cliente decompor a tarefa. Não foram encontradas nas respostas dos especialistas, evidências quando o trabalhador decompõe a tarefa, melhora o seu entendimento.

Rec#3: O trabalhador na multidão deve avaliar e gerenciar o tempo disponível para aprender como executar a tarefa.

O objetivo dessa recomendação é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa.**

A gestão do tempo refere-se a capacidade do trabalhador incluir alguns processos necessários para gerenciar o término pontual da tarefa ofertada pela plataforma. Isto é, a dificuldade em gerenciar o tempo pelo trabalhador, inclui processos necessários para entender, preparar o ambiente, executar e, principalmente, terminar a tarefa.

O trabalhador deve avaliar e gerenciar o tempo para aprender a fazer a tarefa também chamada de curva de aprendizagem. Curva de aprendizagem é uma representação do nível médio cognitivo de aprendizagem para uma determinada atividade. Park e Jensen [149] comentam que usuários gastam um significativo período de tempo de aprendizado sobre o projeto antes de participarem efetivamente de algumas atividades. Straub et al [41] perceberam a importância do conhecimento prévio que o trabalho deve possuir para executar a tarefa em projetos de crowdsourcing. Outra discussão apontada por Schmid-Druner [23] diz respeito ao trabalho das multidões como "atividade de tempo livre", e comenta que o trabalhador de plataformas europeias não tem a intenção de fazer esse tipo de trabalho, como sua principal fonte de renda, mas sim, considerá-lo como uma atividade de tempo livre em paralelo a de outro trabalho. Um estudo da *International Labour*

*Office*⁴² indica que, para “38 por cento dos trabalhadores americanos e 49 por cento dos trabalhadores indianos da *Amazon Mechanical Turk*, o trabalho da plataforma constitui sua principal fonte de renda”. Inclusive, a plataforma Topcoder, tem ofertado sistematicamente, programas educacionais (*Topcoder Education Week Marathon Match*) aos seus usuários como apoio a participação em competições.

Yang *et al.* [164] entendem que, em tarefas de crowdsourcing, o tempo é uma dimensão do esforço associado para fazer a tarefa.

Time is the dimension along which the human cost (e.g., fatigue, stress) associated with the execution of crowdsourcing tasks has been typically quantified.

A seguir, alguns comentários extraídos dos participantes que auxiliaram no processo avaliativo da Rec#3. Para o participante E1, é válido avaliar e gerenciar o tempo disponível para aprender como executar a tarefa, porém pensa ser muito difícil para determinados tipos de tarefas especificar tempo e cronograma de execução.

E1: There are tasks in which it is very difficult to be very specific when defining hours or timelines.

E8: Using Crowdworkers in all the task preparation is helpful in terms of time and budget however it may not reflect client production policy!

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 3 (Rec #3) podem ser observados na Figura 26. Percebe-se ao analisar a Figura 26, que 75% (9) dos especialistas concordam (*Concordo ou Concordo fortemente*), 25% (3) nem concordam nem discordam e, nenhuma ocorrência de discordância (*Discordo fortemente, Discordo*) com essa recomendação.

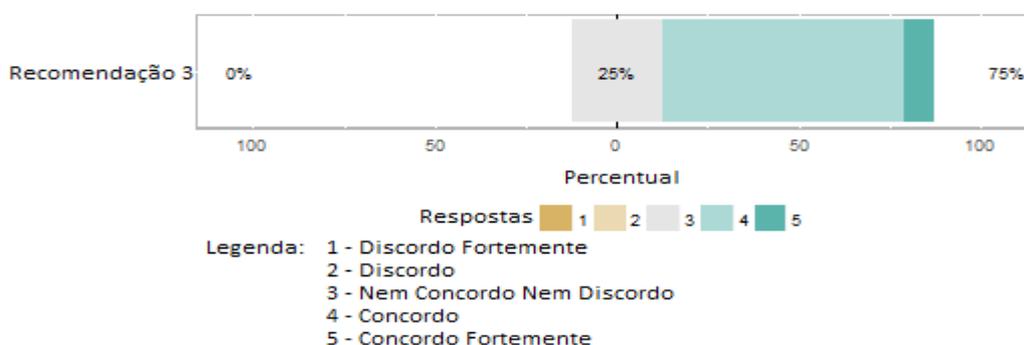


Figura 26: Nível de Concordância com a Rec#3

⁴² Berg, *Income security in the on-demand economy: Findings and policy lessons from a survey of crowdworkers*, ILO, Geneva 2016 (Berg 2016)

Diante do exposto conclui-se que os especialistas “**Concordam**” com a Rec #3, isto é, o trabalhador na multidão deve avaliar e gerenciar o tempo disponível para aprender como executar a tarefa

Rec#4: O trabalhador na multidão deve criar seu próprio plano e gerenciar sua agenda para dispor de tempo suficiente para terminar a tarefa

O objetivo dessa recomendação, como aconteceu na Rec#3, é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa.**

Normalmente, as tarefas em software crowdsourcing possuem informações como um título, uma descrição textual, data de término, tempo alocado, recompensa monetária – se pertinente - e algumas qualificações exigidas para executar a tarefa. Assim, de posse da data do término e tempo alocado para a execução da tarefa, se faz necessário dimensionar o tempo pessoal que será destinado para, planejar e gerenciar o cronograma, definir algumas atividades, estimar os recursos e as durações das atividades, e, por fim, controlar o cronograma estipulado como aponta [165].

Hoßfeld *et al.* [83] comentam que o trabalhador leva em consideração para a sua tomada de decisão se irá, ou não executar a tarefa, o tempo estimado que foi atribuído pelo gestor da tarefa na plataforma antes de executar uma tarefa, por isso, para a plataforma é fundamental ser assertiva ao estimar o tempo da execução da tarefa e, para o trabalhador ter mecanismos que o auxiliem a definir o tempo para a execução da tarefa.

Organizar a agenda para a realização de uma tarefa é considerado fundamental para o trabalhador, visto que Schmid-Drune [23] mostra que, em muitos casos, os trabalhadores em plataformas são classificados como autônomos, e, nessa condição, caso não sejam capazes de gerenciar sua própria agenda podem não concluir as tarefas agendadas. Neste sentido, E13 entende que essa recomendação não é específica para ambientes de crowdsourcing, isto é, pode ser generalizada para várias comunidades on-line.

E13 : These planning are valid for all workers. nothing special in crowdsourcing.

A dificuldade em gerenciar o tempo pessoal disponível para executar a tarefa é decorrente muitas vezes das exigências da plataforma na entrega rápida das atividades,

Platform work encourages a rapid pace of work without breaks: “Online crowd workers may be working to tight deadlines (more skilled freelance workers) or on low piece-rates for micro-tasks (lowerskilled clerical workers) while offline workers are under pressure to complete fixed-fee jobs and move on to the next” [23]

Os trabalhadores na multidão devem organizar um plano organizacional para adquirir conhecimentos e habilidades necessárias para fazerem completarem suas tarefas [149]. Nesse sentido Straub *et al.* [41] encontraram evidências que relacionam a entrega da tarefa com o tempo associada à ela para sua execução. O participante E5 comenta que fornecer um plano detalhado das tarefas isso depende do tamanho dessa tarefa.

E5: I believe that for highly specialized tasks such as software development, the crowd is required to have a certain degree of expertise, which means they're in a better position than the requester (mostly) to define how much effort / time is required, pretty much like software engineers give estimates to their tasks to management in a work environment. As to providing a detailed plan with tasks, it depends on the size of the microtask. Assigning a straightforward method implementation to a worker wouldn't require planning or further task breakdown. Assigning a higher-level and ambiguous design task would

Nesta linha, o E7 entende que pelo menos alguma estimativa de esforço deve ser fornecida. Se houver um conjunto de tarefas menores, um cronograma geral poderá ser fornecido.

E7: at least some effort estimation should be provided. If there is a set of smaller tasks an overall timeline as well as timelines for smaller pieces could be provided.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 4 (Rec #4) podem ser observados na Figura 27. Percebe-se ao analisar a Figura 27, que 75% (9) dos especialistas concordam (Concordo ou Concordo fortemente), e 17% (2) nem concordam nem discordam e, 8% (1) de discordância (*Discordo fortemente, Discordo*) com essa recomendação.

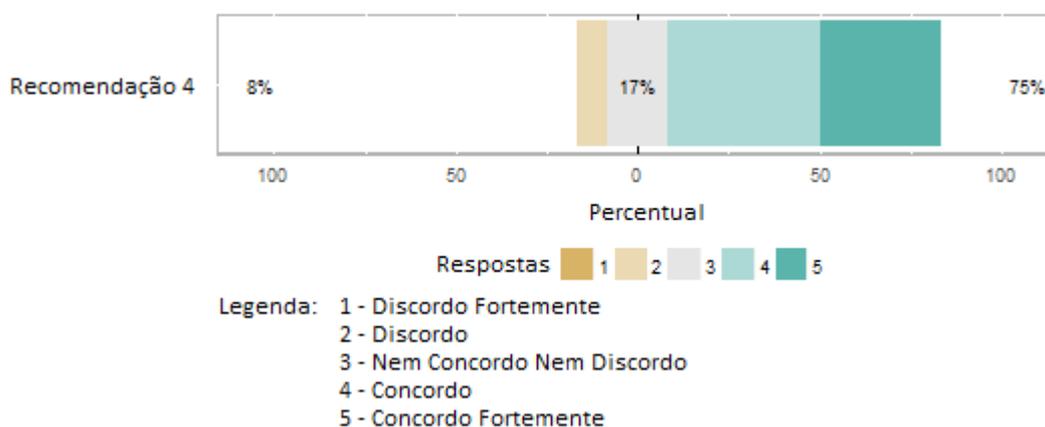


Figura 27: Nível de Concordância com a Rec#4

Diante do exposto, conclui-se que os especialistas “**Concordam**” com a Rec. #4, isto é, o trabalhador na multidão deve criar seu próprio plano e gerenciar sua agenda para dispor de tempo suficiente para terminar a tarefa.

Rec#5: O trabalhador na multidão deve definir pequenas atividades para finalizar a tarefa.

O objetivo dessa recomendação, como aconteceu nas Rec3# e Rec#4, é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa.**

Decompor as tarefas pode significar o sucesso do projeto em atividades de software crowdsourcing sendo considerado um importante desafio para esse novo modelo de desenvolvimento de software. [14], [13], [21], [47], [54], [102], [111], [112].

O trabalhador na multidão deve definir pequenas atividades no auxílio da gestão do tempo executar a tarefa. Se para a plataforma ou o cliente já é um desafio decompor a tarefa, isso é potencializado quando fica sob a responsabilidade do trabalhador da multidão, como comenta o entrevistado E6

E6: Depends on the particular tasks... estimation and decomposition can be really difficult

Neste sentido, Kittur *et al.* [58] ao analisarem plataformas, como a Mturk, baseadas essencialmente em micro-tarefas para tarefas consideradas como “simples” como rotular uma imagem, propõe um framework para decompor tarefas complexas também em micro-tarefas. Destacam a dificuldade nesse processo devido a escassez de informações por parte da plataforma nas tarefas, para a possibilidade de decomposição. Corroboram nessa análise Zhang *et al.* [148]

Para o participante E4, parece desnecessário definir pequenas atividades para finalizar a tarefa, porém, infelizmente, não sugere uma solução alternativa para essa recomendação.

E4: seems unnecessary

Na perspectiva do entrevistado E11, a divisão da tarefa criará muito “overhead”, e segue na mesma direção do E12.

E11: The division creates too much overhead. Also see my previous answer.

E12: More onus should be on requesters vs. the crowdworker. We can't expect too much for them for small tasks.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 5 (Rec #5) podem ser observados na Figura 28. Percebe-se ao analisar a Figura 28, que 67% (8) dos especialistas nem concordam nem discordam e, 17% (3) concordam (*Concordo ou Concordo fortemente*), e 16% (2) de discordância (*Discordo fortemente, Discordo*) com essa recomendação. Nota-se, maior percentual neutro dentre todas as recomendações. Assim, os especialistas mantiveram-se **neutros** para a recomendação de que o trabalhador na multidão deve definir pequenas atividades para finalizar a tarefa.

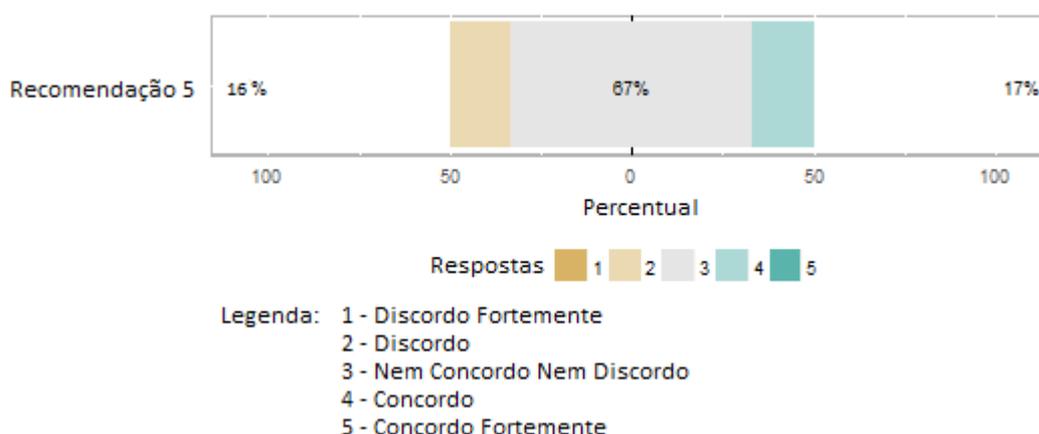


Figura 28: Nível de Concordância com a Rec#5

Rec#6: O trabalhador na multidão deve determinar a duração das atividades

O objetivo dessa recomendação, como aconteceu nas Rec3#, Rec#4 e Rec#5, é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa**.

Na perspectiva do entrevistado E11, com a divisão da tarefa isso acarretará mais ônus ao trabalhador na multidão e, isso deve ser evitado.

E12: More onus should be on requesters vs. the crowdworker. We can't expect too much for them for small tasks.

Não foram identificados comentários dos participantes com sugestões que pudessem auxiliar o processo avaliativo de que o trabalhador na multidão deve determinar a duração das atividades, talvez isso possa ter contribuído para que a maioria dos entrevistados não tenham se posicionado favoravelmente ou não a essa recomendação.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 6 (Rec #6) podem ser observados na Figura 29. Percebe-se ao analisar a Figura 29, que 58% (7) dos especialistas nem concordam nem

discordam e, 17% (3) concordam (*Concordo ou Concordo fortemente*), e 25% (3) de discordância (*Discordo fortemente, Discordo*) com essa recomendação. Nota-se, o segundo maior percentual neutro dentre todas as recomendações. Observa-se, a “alta” **neutralidade** apontada pelos especialistas, isto é, os especialistas mantiveram-se neutros para a recomendação de que o trabalhador na multidão deve determinar a duração das atividades.

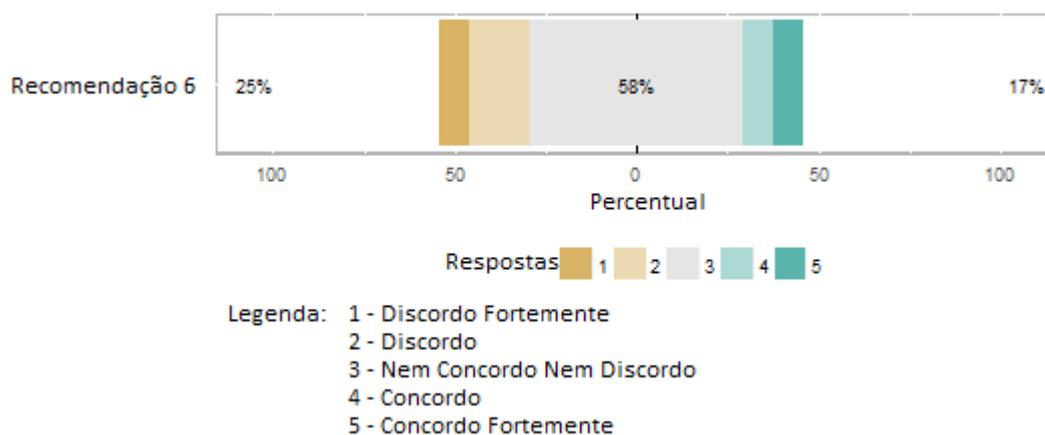


Figura 29: Nível de Concordância com a Rec#6

Enfim, pode existir uma relação entre a dificuldade em decompor a tarefa (Rec#5) e a e, por conseguinte, determinar a duração (tempo) para a execução dessa tarefa (Rec#6), pois as duas recomendações foram categorizadas como inconclusivas.

Rec#7: O trabalhador na multidão deve preparar o ambiente para a execução de cada tarefa.

O objetivo dessa recomendação, como aconteceu nas Rec3#, Rec#4, Rec#5 e Rec#6, é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa.**

Preparar o ambiente para a execução de determinados tipos de tarefa é apontado por Steinmacher *et al.* [108] como uma das principais barreiras enfrentadas por novatos em suas primeiras contribuições em projetos de software livre.

O guia Project Management Body of Knowledge (PMBOK), [148] é um conjunto de práticas na gestão de projetos onde, em seu item 6.4 – na 5ª edição – considera fundamental estimar os recursos humanos, equipamentos, suprimentos e quantidades de materiais necessários para a realização de cada atividade, ou seja, preparar o ambiente é algo já conhecido e relacionado a gestão de projetos. A plataforma para software crowdsourcing deve indicar um ambiente para o desenvolvimento da tarefa bem como,

quando possível, integrar a possibilidade da execução da tarefa com outras ferramentas de desenvolvimento, utilizando, por exemplo, repositórios colaborativos. Inclusive, [82] recomendam algumas melhores práticas para tal, como por exemplo, “Use common software without requiring admin installations!” (p. 5).

A seguir, alguns comentários extraídos dos participantes que auxiliaram no processo avaliativo da Rec#7. Para o participante E14, esse tipo de recomendação é válida para qualquer modelo de desenvolvimento de software, independentemente de ser crowdsourcing. Vale lembrar que práticas de desenvolvimento de software que utilizam o modelo do software crowdsourcing são potencializadas quando usam esse modelo.

E14: These are all sensible recommendations, which I guess apply beyond crowdwork. What makes them specific to crowdworking? I did not understand Rec 7, so I did neither agree nor disagree with it.

O entrevistado E10 comenta que não vê relação entre preparar o ambiente e gestão do tempo e questiona se entendeu a recomendação.

E10: I'm completely at a loss what "setup environment" and "timeline" are, so my answers to these questions are meaningless. I'm not even a 100% sure what you mean by "manage time", are you thinking about the available time and how to distribute it on solving multiple tasks? Or are you talking about estimating how long each task will take? Both are related, but not the same.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 7 (Rec #7) podem ser observados na Figura 30. Percebe-se ao analisar a Figura 30, que 42% (6) dos especialistas nem concordam nem discordam e, 33% (4) concordam (*Concordo ou Concordo fortemente*), e 25% (3) de discordância (*Discordo fortemente, Discordo*) com essa recomendação. Diante do exposto os especialistas **“Concordam”** com esta recomendação, isto é, o trabalhador na multidão deve preparar o ambiente para a execução de cada tarefa.

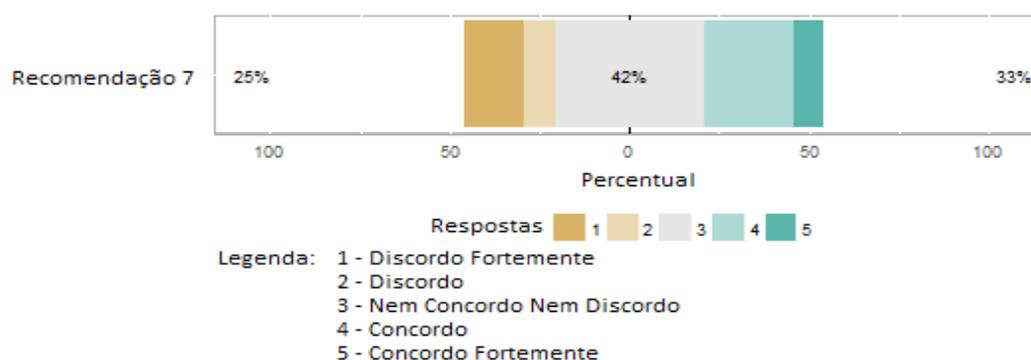


Figura 30: Nível de Concordância com a Rec#7

Rec#8: O trabalhador na multidão deve definir sua agenda para finalizar a tarefa

O objetivo dessa recomendação, como aconteceu nas Rec3#, Rec#4, Rec#5, Rec#6 e Rec#7, é auxiliar o trabalhador na multidão a superar a barreira **BF3: É difícil gerenciar o tempo disponível para executar a tarefa.**

Não foram identificados comentários dos participantes que pudessem auxiliar o processo avaliativo de que o trabalhador na multidão deve definir sua agenda para finalizar a tarefa. Assim, a definição da categorização desta recomendação, no caso a Rec#8, far-se-á apenas com os dados quantitativos.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 8 (Rec #8) podem ser observados na Figura 31. Percebe-se ao analisar a Figura 31, que cerca de 42% (6) dos especialistas concordam (*Concordo ou Concordo fortemente*), o mesmo índice de 50% (6) nem concordam nem discordam e, apenas 8% (1) com discordância (*Discordo fortemente, Discordo*) com essa recomendação. Diante do exposto os especialistas **“Concordam”** com esta recomendação, isto é, o trabalhador na multidão deve definir sua agenda para finalizar a tarefa.

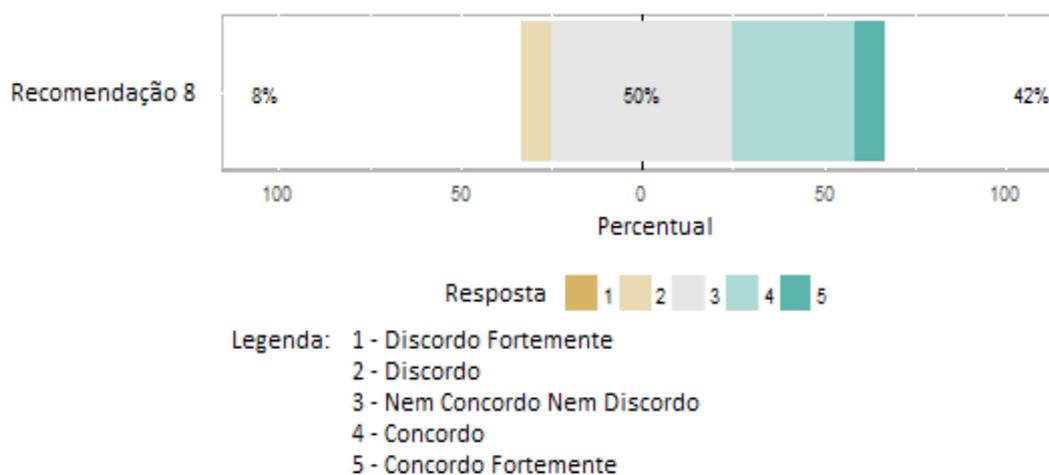


Figura 31: Nível de Concordância com a Rec#8

Enfim, o trabalhador deve gerenciar o seu tempo para encontrar a tarefa, entender a tarefa e possuir um ambiente computacional apto para fazer a tarefa em projetos de software crowdsourcing no modelo competitivo ao utilizar a plataforma Topcoder.

Rec#9: O trabalhador na multidão deve utilizar máquinas virtuais com Ambientes de Desenvolvimento Integrados (IDEs) para fazer a tarefa;

O objetivo dessa recomendação é auxiliar o trabalhador na multidão a superar a barreira **BF4: É difícil configurar o ambiente necessário para executar a tarefa.**

A seguir, um comentário extraído do participante que auxiliou no processo avaliativo da Rec#9. O entrevistado E1 concorda com a recomendação e sugere a utilização de *dockers* para auxílio na preparação do ambiente para o desenvolvimento da tarefa.

E1: I agree with the use of virtual machines or docker containers to overcome the barrier "It's hard to configure the necessary environment to perform the task".

Como identificou-se apenas um comentário sobre a Rec#9, a definição da categorização desta recomendação, far-se-á apenas com os dados quantitativos.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 9 (Rec #9) podem ser observados na Figura 32. Percebe-se ao analisar a Figura 32, cerca de 42% (6) dos especialistas concordam (*Concordo ou Concordo fortemente*), o mesmo índice de 33% (4) nem concordam nem discordam e, 25% (3) com discordância (*Discordo fortemente, Discordo*) com essa recomendação. Diante do exposto os especialistas **“Concordam”** com esta recomendação, isto é, o trabalhador na multidão deve utilizar máquinas virtuais com Ambientes de Desenvolvimento Integrados (IDEs) para fazer a tarefa.

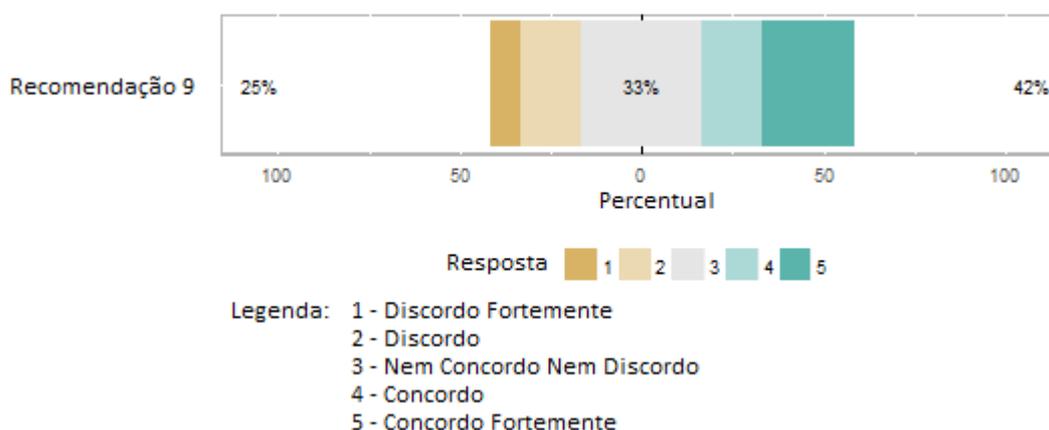


Figura 32: Nível de Concordância com a Rec#9

Rec#10: O trabalhador na multidão deve utilizar um ambiente de desenvolvimento integrados (IDEs) que possua uma Wiki, como apoio para fazer a tarefa;

O objetivo da recomendação, como aconteceu na Rec#9, é auxiliar o trabalhador na multidão a superar a barreira **BF4: É difícil configurar o ambiente necessário para executar a tarefa.**

Existem algumas ações que podem auxiliar no processo de preparação do ambiente para a execução das tarefas. A wikilIDE⁴³, por exemplo, oferece um ambiente de desenvolvimento colaborativo, com suporte a determinados serviços e aplicativos que podem auxiliar no desenvolvimento de tarefas em projetos de software crowdsourcing. A utilização de máquinas virtuais com ambiente de desenvolvimento integrado (IDE) também é uma opção.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 10 (Rec #10) podem ser observados na Figura 33. Percebe-se ao analisar a Figura 33, cerca de 25% (3) dos especialistas concordam (*Concordo ou Concordo fortemente*), o mesmo índice de 33% (4) nem concordam nem discordam e, 42% (6) com discordância (*Discordo fortemente, Discordo*) com essa recomendação. Diante do exposto os especialistas “**Discordam**” com esta recomendação, isto é, não é aconselhado ao trabalhador na multidão utilizar um ambiente de desenvolvimento integrados (IDEs) que possua uma Wiki, como apoio para fazer a tarefa. Não foram sugeridas pelos especialistas uma alternativa à essa recomendação.

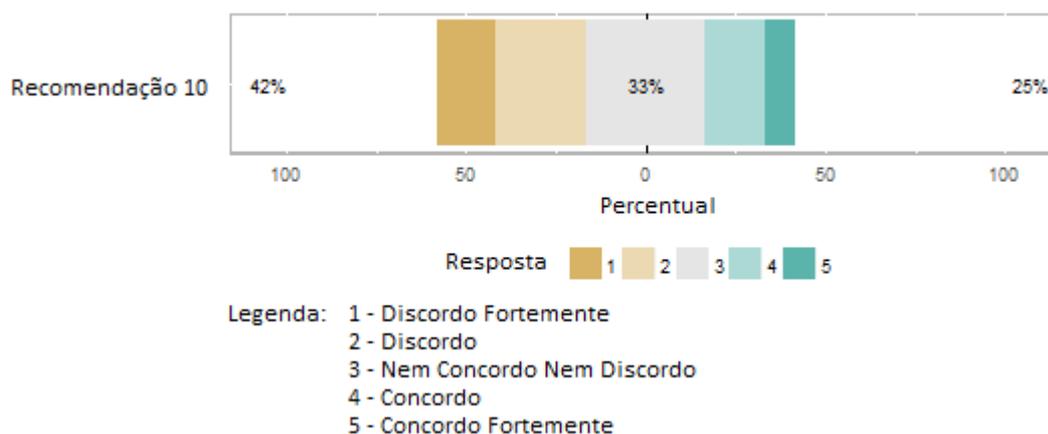


Figura 33: Nível de Concordância com a Rec#10

⁴³ <http://wiki.c2.com/?Wikilde>

Rec#11: O trabalhador na multidão deve utilizar ambientes de desenvolvimento integrados (IDEs) sob o conceito de computação em nuvem para fazer a tarefa

O objetivo dessa recomendação, como aconteceu na Rec#9 e Rec#10, é auxiliar o trabalhador na multidão a superar a barreira **BF4: É difícil configurar o ambiente necessário para executar a tarefa.**

Preparar uma estrutura computacional, com softwares, e com hardwares específicos, entre outras estruturas, são apontados pelos participantes, como uma das principais barreiras para a execução de determinadas macro tarefas em projetos de software crowdsourcing, tanto é, que essa barreira consta em duas categorias, pois requer tempo e competência para essa preparação. É uma barreira de difícil sobreposição por parte do trabalhador, visto que, é dependente das informações advindas do cliente, da plataforma, da documentação associada à plataforma, ou de outros trabalhadores na plataforma. Oferecer uma estrutura computacional anteriormente configurada para que, os novatos em ambiente OSS possam contribuir pela primeira vez, foi uma sugestão discutida por [166]. Em estudo similar de Steinmacher *et al.* [129], os autores também apresentam a barreira “montar o ambiente de configuração” como uma das mais importantes que os novatos enfrentam quando participam pela primeira vez em projetos *open source*, como ocorre, também, em projetos de software crowdsourcing apontados e discutidos por Zanatta *et al.* [131].

Outra possível solução é o uso de IDEs sob o conceito de computação em nuvem. São exemplos dessas IDEs, a Codeanywhere⁴⁴, a Cronapp⁴⁵, e, por fim a Onion Cloud⁴⁶.

Com o recente surgimento da plataforma Docker⁴⁷, a qual utiliza o conceito de *container*, cujo objetivo é facilitar a criação e administração de ambientes isolados para o desenvolvimento soluções computacionais, isso, pode auxiliar a preparação de ambientes para a execução de tarefas em software crowdsourcing. Segundo a plataforma Docker, contêiner é “*A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system*

⁴⁴ <https://codeanywhere.com/>

⁴⁵ <https://www.cronapp.io/>

⁴⁶ <https://cloud.onion.io/>

⁴⁷ <https://www.docker.com/>

libraries, settings". São exemplos de outras ferramentas de gerenciamento de container o Portainer (<http://portainer.io/>) e o Rancher (<http://rancher.com/>).

Seja com a utilização de máquinas virtuais integradas à IDEs ou contêineres, existem possibilidades para que o trabalhador na multidão prepare o seu ambiente para a execução de uma tarefa, entretanto, pode ser considerada de difícil solução dependendo, por óbvio, do tipo de macro tarefa a ser entendida, planejada e executada.

A constante demanda e diversidade dos tipos das tarefas (micro e macro), sua classificação (baixa complexidade, moderada ou complexa) limitações da plataforma, baixa demanda cognitiva do trabalhador aliadas a evolução tecnológica são fatores impeditivos para a montagem de um ambiente que sirva para a execução de qualquer macro tarefa.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 11 (Rec #11) podem ser observados na Figura 34. Percebe-se ao analisar a Figura 34, que para a maioria, cerca de 58% (7) dos especialistas concordam (*Concordo ou Concordo fortemente*), apenas 17% (2) nem concordam nem discordam e, 25% (4) com discordância (*Discordo fortemente, Discordo*) com essa recomendação.

Diante do exposto os especialistas "**Concordam**" com esta recomendação, isto é, o trabalhador na multidão deve utilizar ambientes de desenvolvimento integrados (IDEs) sob o conceito de computação em nuvem para fazer a tarefa.

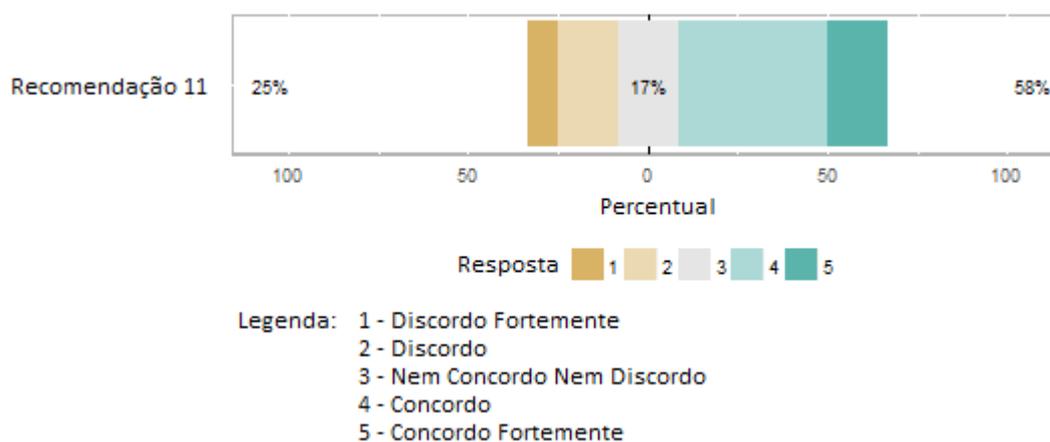


Figura 34: Nível de Concordância com a Rec#11

Rec#12: O trabalhador na multidão deve utilizar diferentes canais para comunicar-se sincronamente ou assincronamente (software crowdsourcing é baseado na interação).

O objetivo da recomendação R12 é auxiliar o trabalhador na multidão a superar a barreira **BF5: É difícil colaborar com outros membros da plataforma para auxiliar na execução da tarefa.**

A atividade de desenvolvimento de software é essencialmente colaborativa e de acordo com Howe [5], MacLean *et al.* [151] entendem que o crowdsourcing é uma “forma massiva de colaboração paralela”. Colaborar é o ato no qual duas ou mais pessoas trabalham juntas para alcançarem o mesmo objetivo⁴⁸. E, para isso, as pessoas precisam compartilhar informações, comunicarem-se, ajudarem-se mutuamente.

Não conseguir se expressar ou compreender um idioma não nativo contribui para minimizar a colaboração entre os membros da multidão durante o processo de participação em atividades de software crowdsourcing, como afirmam Codagnone *et al.* [76] e Kittur *et al.* [58]. A comunicação entre os membros dos projetos desempenha um papel primordial em qualquer projeto de software bem-sucedido como destacam Panichella *et al.* [152].

Os participantes dos diversos estudos realizados na presente tese comentaram que houve pouca colaboração entre os membros antes, durante e depois da execução da tarefa. Como visto, interagir com outros membros é fundamental em atividades de desenvolvimento de software e, no software crowdsourcing, essa necessidade de interação é potencializada [153], [135], [99], [136]. Barreiras de comunicação, diferenças culturais e diferentes fusos horários diminuem a colaboração entre os membros da multidão como relatam [137]. Saxton *et al.* [70] mostram que o nível de colaboração pode variar substancialmente conforme o modelo do crowdsourcing, o tipo e a classificação (complexidade) da tarefa adotados para o desenvolvimento do projeto. Assim, oferecer mecanismos de comunicação entre os membros da multidão, como conversas síncronas, como pôr o exemplo, o *chat*, e canais de *feedback* para que os membros da multidão possam interagir uns com outros é primordial como apontam Lu *et al.* [40].

Panichella *et al.* [154] perceberam que a comunicação entre os desenvolvedores, em equipes emergentes, em projetos de software livre surge naturalmente via canais assíncronos (por exemplo, e-mails), sendo isso, fundamental para o sucesso do projeto. Hoßfeld *et al.* [83] recomendam também a utilização e integração de canais de feedback

⁴⁸ <http://dictionary.cambridge.org/dictionary/english/collaboration>

pode melhorar na comunicação entre os trabalhadores. Kittur *et al.* [58] acrescentam e entendem que é essencial manter o trabalhador motivado, oferecendo-lhe constante retorno (feedback) das tarefas além, de utilizar um sistema eficiente de pagamento – quando for o caso – podem proporcionar uma melhor interação entre os trabalhadores na multidão.

Como acontece no DDS, as pessoas precisam de mecanismos de relacionamentos eficientes para criar uma visão compartilhada por meio de interações, assim, é fundamental que atores da plataforma, no modelo crowdsourcing desempenhem papel de facilitador nesse processo além de, oferecer formas de discutir e receber opiniões do trabalhador na multidão, função do fórum. No software crowdsourcing, quanto mais pessoas e papéis envolvidos, mais desafiadores serão os problemas relacionados à interação.

Como destacado por Bari *et al.* [155], a plataforma de testing crowdsourcing (uTest), utiliza e oferece aos participantes algumas ferramentas que auxiliam a colaboração. A utilização de elementos da Gamificação para fomentar e incentivar a colaboração em comunidades on-line como o crowdsourcing é comentada em Júnior [156]. A combinação de dois frameworks propostos pelos autores (denominados OMELETTE e DGS) constituem uma sólida fundação para colaboração em ambiente crowdsourcing como apontam Satzger *et al.* [137].

A seguir, alguns comentários extraídos dos participantes que auxiliaram no processo avaliativo da Rec#12. Os entrevistados E1 e E11 sugerem alguns mecanismos de comunicação como o Kanban e o Slack, e o E10 ressalta que é papel da plataforma oferecer canais de comunicação. O E12 comenta que, a recomendação deveria indicar qual canal de recomendação e a sua frequência.

E1: Kanban boards or slack channels would be a good example.

E10: Platform should support some async/sync collaboration channels.

E11: this shouldn't be necessary...there are tools like Slack that can be used both synchronously and asynchronously

E12: But the recommendation should suggest what channels and what frequency

O entrevistado E6 concorda, porém comenta que falta detalhamento na recomendação e questiona que tipo de mecanismos poderiam ser utilizados para a comunicação.

E6: Sure, but the recommendation isn't very detailed. What kind of communication mechanisms do you have in mind?

Interessante observar o comentário de E5. Para ele, atividades de micro tarefas em ambientes de crowdsourcing deve “*manter um mínimo de comunicação*” pois acredita que seria difícil gerenciar essa comunicação de uma multidão de trabalhadores. Porém, vale sublinhar que software crowdsourcing é baseado em macro e não em micro tarefas e, desenvolver software é uma atividade essencialmente colaborativa. O E13 questiona se os trabalhadores em ambientes de software crowdsourcing irão colaborar.

E5: Wouldn't it defeat the purpose of crowdsourcing? How would you manage communication of hundreds or thousands of workers? I believe a pure microtask crowdsourced project should keep a minimum of communication and rely on basic voting / score systems for certain decisions. Obviously this is a personal opinion and it would require further research to evaluate the viability of this approach.

*E13: I assume it is meant that in *competitive crowdsourcing*, workers within a group should collaborate (but they do not really need to collaborate with workers outside the group). Additionally, if a project requires much collaboration between workers, then probably the division in micro-tasks was not optimally performed (as it should have resulted in micro-tasks that can be solved independently of each other).*

Os entrevistados E7 e E8 comentam que depende do tipo da tarefa que o trabalhador na multidão deve utilizar diferentes canais para comunicar-se sincronamente ou assincronamente.

E7: this might also depend on the type of task, if there are isolated tasks which might not need any collaboration, others might exist where discussion is needed (e.g., team meetings, or knowledge exchange).

E8: Depends on the type of Crowdsourced platform and level of the issue raising, I agree that there is a need for different communication channel.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 12 (Rec #12) podem ser observados na Figura 35. Percebe-se ao analisar a Figura 35, que 42% (6) dos especialistas concordam (*Concordo ou Concordo fortemente*), 42% (6) nem concordam nem discordam, e 16,0% (1) discordam (*Discordo fortemente, Discordo*) com essa recomendação.

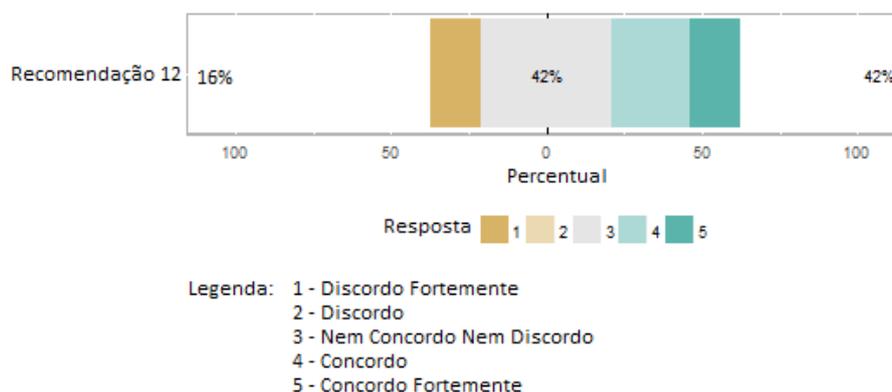


Figura 35: Nível de Concordância com a Rec#12

Assim e diante do exposto os especialistas “**Concordam**” com esta recomendação, isto é, como o software crowdsourcing é baseado fortemente em interação, os trabalhadores da multidão devem utilizar diferentes canais para comunicar-se sincronamente ou assincronamente como auxílio na realização da tarefa.

Rec#13 – O trabalhador na multidão deve utilizar algum mecanismo de tradução da tarefa para facilitar o entendimento da tarefa e a colaboração com outros membros da plataforma

O objetivo da recomendação Rec#13 é auxiliar o trabalhador na multidão a superar a barreira **BF6: Para trabalhadores não fluentes em inglês, e, como a plataforma está nesse idioma, é difícil entender a tarefas. Durante a execução da tarefa, é difícil colaborar com outros membros da plataforma porque eles apenas comunicam-se em inglês.**

Compreender o idioma não nativo ainda é um problema e uma importante barreira no desenvolvimento de tarefas em projetos de software crowdsourcing no modelo competitivo. Estudos apresentam soluções para a tradução simultânea entre diversos idiomas por meio de técnicas como a “*Machine translation*” (MT) e o aprendizado de línguas assistido por computador [160], [158], [162], [163], [161], [159]. Essas soluções podem auxiliar o trabalhador a realizar macro tarefas em projetos de software crowdsourcing.

A falta de especialistas em tradução representa um problema para os diferentes e crescentes mercados em todo o mundo [161]. Uma das soluções propostas para a falta de recursos humanos são as ferramentas de tradução automatizadas.

A seguir, alguns comentários extraídos dos participantes que auxiliaram no processo avaliativo da Rec#13. Apenas para o entrevistado E1 essa recomendação foi confirmada, e inclusive sugere a utilização de um mecanismo do Google.

E1: I think it is a good solution (Google translation integration).

Para os entrevistados E9, E11 e E13, pode ser um ponto de partida, porém não consideram suficiente entender a tarefa apenas com tradução automática

E9: I cannot answer this question generally. English is not my native language, yet, I feel confident enough to understand tasks described to me in English, so I wouldn't use auto-translation. Others might need some form of translation. Auto-translation is one option, but certainly not optimal, since it's prone to make mistakes, especially with respect to jargon. In any case, auto-translation is an assistance at best, not a solution to the language barrier.

E11: Machine translation can be a starting point, but is not a substitute for true understanding of task and program requirements.

E13: At first this is a good idea, but auto-translation might misinform the worker, especially because translation leaned from general language (web) corpora will not work very well on specialised texts such as those within a software project. Therefore, I am not that convinced about this recommendation. At the same time, I would not know what else to propose.

Para os entrevistados E6, E7, utilizar mecanismos de tradução automática pode ser considerado até um risco no entendimento da tarefa. E8 alerta que a legibilidade é um fator importante que uma máquina de tradução não pode ajudar e, nesse sentido E10 acrescenta a imprecisão na tradução automática.

E6: I think there's a risk here if the translation introduces mistakes, have to be careful with this one

E7: this might be difficult to manage; for supporting material "auto-translation" might be helpful. For deliverables this should be a uniform language such as English (again, depending on the task type)

E8: Readability is a very important factor that machine cannot help in translation

E10: Machine translation of tasks will introduce a lot of incorrect wording. Ideally worker should understand the original language of the task.

Os entrevistados E5 e E12 remetem à competência do trabalhador para entender a tarefa.

E5: Why not just localizing the microtask platform? Besides, as the worker is required to have certain skills to perform the task, IMO certain languages should be required as well.

E12: There must be a level of common understanding.

No processo de avaliação, os resultados do nível de concordância pelos especialistas com a Recomendação 13 (Rec #13) podem ser observados na Figura 36. Percebe-se ao analisar a Figura 36, que 59% (7) dos especialistas discordam (Discordo fortemente, Discordo), 33% (4) nem concordam nem discordam e, apenas 8% (2) concordam (*Concordo ou Concordo fortemente*), com essa recomendação. Ressalta-se que essa recomendação foi a que obteve o maior índice de discordância dentre todas as recomendações.

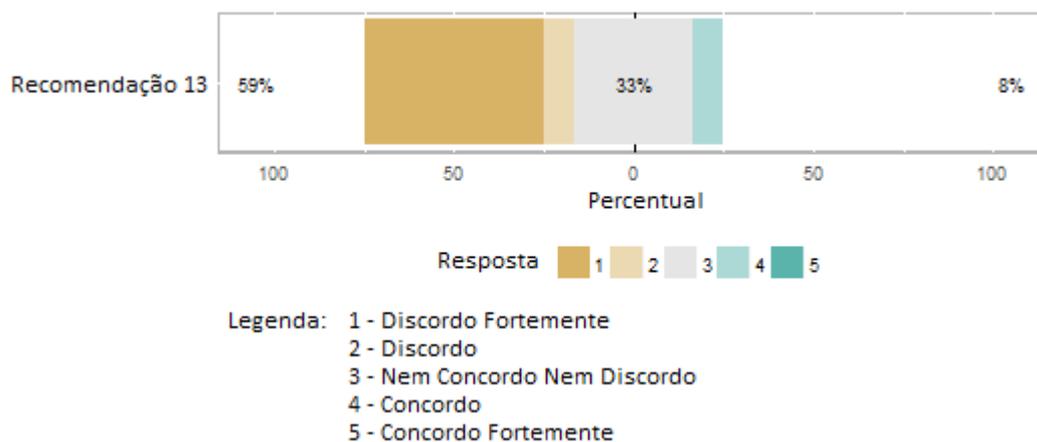


Figura 36: Nível de Concordância com a Rec#13

Assim e diante do exposto os especialistas “**Discordam**” com esta recomendação, isto é, mesmo que compreender o idioma não nativo é uma dificuldade no desenvolvimento de tarefas em projetos de software crowdsourcing no modelo competitivo, o trabalhador na multidão não deve utilizar algum mecanismo de tradução da tarefa para facilitar o entendimento da tarefa e a colaboração com outros membros da plataforma pois os mecanismos atuais não são suficientes para auxiliar o trabalhador a entender a tarefa.

Os resultados gerais dos graus de concordância realizados pelos especialistas com todas as recomendações podem ser observados na Figura 37.

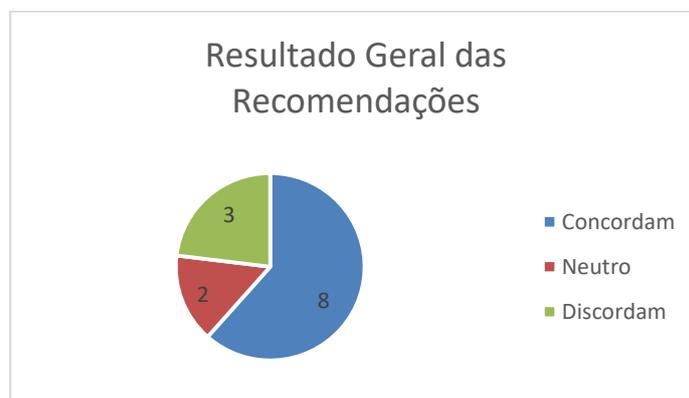


Figura 37: Resultado dos Graus de Concordância.

5.4. Reflexões sobre o capítulo

Esse capítulo teve como objetivo avaliar as recomendações relativas ao trabalhador na multidão. Os resultados mostram que utilizar algum mecanismo que auxilie o trabalhador a encontrar uma tarefa na plataforma de acordo com suas habilidades ou preferências e expectativas de recompensas financeiras pessoais foi considerada como a recomendação com maior nível de concordância dos especialistas. Merece destaque também que o trabalhador na multidão deve avaliar e gerenciar o tempo disponível para aprender como executar a tarefa, além de desenvolver seu próprio plano e gerenciar sua agenda para dispor de tempo suficiente para terminar a tarefa. No caso de tarefas que envolvam codificação os especialistas concordam que o trabalhador na multidão deve utilizar ambientes de desenvolvimento integrados (IDEs) sob o conceito de computação em nuvem para fazer a tarefa.

Por outro lado, os especialistas entenderam que é difícil para o trabalhador, utilizar algum mecanismo que o auxilie a decompor a tarefa para melhorar o entendimento da mesma. Na mesma linha, entendem de difícil aplicação a utilização de mecanismos de tradução da tarefa para facilitar o entendimento da tarefa e a colaboração com outros membros da plataforma.

Uma dificuldade nessa fase, foi o acesso aos especialistas para o processo de avaliação das recomendações, desta forma, poder ser considerada como uma restrição dessa etapa da pesquisa. Portanto, e para concluir, o trabalhador na multidão necessita de competência e de uma eficiente gestão no seu tempo para participar colaborativamente em tarefas de projetos de software crowdsourcing no modelo competitivo na plataforma Topcoder

6. CONCLUSÕES

Este capítulo sintetiza a pesquisa realizada nessa tese e apresenta algumas possibilidades de trabalhos futuros. O capítulo inicia com uma revisão das questões de pesquisa bem como com o objetivo geral e os objetivos específicos. Após apresenta as contribuições dessa tese, bem como, as limitações da pesquisa. O capítulo termina com recomendações para outros elementos do software crowdsourcing e os trabalhos futuros

Utilizar o conhecimento e a sabedoria da multidão online para contribuir na resolução de problemas é uma realidade. O crowdsourcing, mesmo que recente, é um modelo interessante e promissor, e necessita que os seus quatro elementos trabalhem como uma engrenagem sincronizada e em sinergia para o seu sucesso. A multidão é peça fundamental nesse sistema e, é preciso criar mecanismos e condições para que esses indivíduos, pertencentes a multidão, possam realizar as suas tarefas dentro do prazo estipulado, com a qualidade desejada a um custo acessível, razões essas do modelo crowdsourcing. Mas, no que se refere a software crowdsourcing o tema é recente e carente de estudos. O processo de desenvolvimento de software é uma atividade complexa e acrescentar a possibilidade de desenvolvê-lo no modelo crowdsourcing apresenta vários desafios. Atrair e manter os desenvolvedores motivados, atuantes e criativos potencializam esses desafios. Assim, identificar e compreender quais são as barreiras e dificuldades encontradas em projetos de software crowdsourcing e oferecer algum tipo de apoio para que ultrapassem essas barreiras foi o principal objetivo deste trabalho.

Foram identificadas 6 barreiras e sugeridas 13 recomendações específicas para os trabalhadores na multidão em projetos de software crowdsourcing no modelo competitivo. As barreiras emergiram por meio do uso rigoroso e sistemático de vários métodos científicos. Conclui-se que o trabalhador teve dificuldades à aspectos relacionados a gestão do pessoal, a colaboração e a falta de competência para fazer as tarefas propostas. As recomendações foram avaliadas por especialistas.

A difusão do conhecimento deu-se por meio de publicações científicas em importantes meios de comunicação e apresentações da temática da presente tese em eventos científicos reconhecidos internacionalmente.

O conhecimento gerado pela tese auxiliará também aos outros três elementos do modelo crowdsourcing, o cliente, a plataforma e a tarefa. O cliente para a criação de mecanismos de seleção e retenção dos talentos, bem como, no preparo das especificações

das tarefas. Já na plataforma, auxiliará com subsídios que fundamentem a inserção de recursos tecnológicos capazes de antever possíveis problemas de desistências dos trabalhadores na multidão, como, por exemplo, a utilização de sistemas de recomendação para as tarefas de acordo com o perfil do usuário que irá desenvolvê-la, tornar o layout do sistema de feedback mais integrado com o painel da plataforma, melhorar os padrões de interações com outros usuários e maior transparência e interação no fluxo de trabalho da tarefa.

Do ponto de vista de impacto para a indústria de software, os gestores das plataformas poderão implementar melhorias e realizar os ajustes sugeridos pelo presente trabalho como ao ofertar uma tarefa para multidão, levar em consideração se a documentação está consistente com o que foi solicitado, oferecer mecanismos que auxiliem este trabalhador a escolher uma tarefa, oferecer fortes canais de comunicação e suporte aos trabalhadores.

Sob a perspectiva acadêmica, a temática proposta, além do uso de métodos empíricos e a análise qualitativa, contribuirá para a formação de uma base conceitual e aplicada em projetos de software com força de trabalho alternativa, entre eles, o crowdsourcing. Espera-se, com isso, estimular a comunidade acadêmica em discutir a possibilidade de inclusão do tema em disciplinas relacionadas a engenharia de software em cursos de graduação e pós-graduação da área da informática. Portanto, contribuirá para a discussão e o ensino de novas forças e formas de desenvolver software como é o caso do software crowdsourcing. Novas pesquisas e trabalhos poderão ser originados dessa tese cuja atenção recai sobre os aspectos humanos e suas influências no desenvolvimento de software, em especial, nas dificuldades que enfrentam em suas contribuições em projetos de software crowdsourcing.

Sob a luz da sociedade, ao ampliar a discussão e socialização do modelo, abre-se o leque de opções para uma possibilidade, ou alternativa de trabalho e de aumento de renda do indivíduo em ambientes competitivos em projetos de software crowdsourcing.

6.1. Limitações do trabalho.

Como qualquer outro trabalho empírico, existem algumas limitações e tópicos que permanecem para o trabalho futuro. Como uma limitação, o estudo concentrou-se em apenas uma plataforma de crowdsourcing, no caso a Topcoder. Embora seja a plataforma mais utilizada pela comunidade de software crowdsourcing competitivo, não é possível afirmar que o comportamento identificado possa ser generalizado. Além disso, a

abordagem não considera outras competições da plataforma, como a especificação, arquitetura, projeto e testes.

Foram realizados vários estudos qualitativos utilizando entrevistas, questionários que, potencialmente, tem suas limitações, principalmente pela possibilidade de os participantes serem tendenciosos a encontrarem algumas barreiras. Talvez outra limitação é que os indivíduos que participaram desse estudo pertencem ou estão alocados em uma determinada região do estado do Rio Grande do Sul no Brasil e são brasileiros, não correspondendo, de certa forma, as características de uma típica multidão no crowdsourcing.

Para o processo de avaliação das recomendações o fato de não conseguir um maior número de especialistas bem como usuários da plataforma Topcoder pode ser considerado uma restrição. Embora não se tenha avaliado as recomendações para verificar até que ponto elas podem mitigar as barreiras evidenciadas, considera-se um resultado importante para plataformas de crowdsourcing e clientes. Observa-se que algumas dessas abordagens estão em uso em outros contextos, por exemplo, em projetos de software livre.

6.2. Trabalhos futuros

A análise qualitativa na engenharia de software é um tema recente e merece atenção. Aplicar outros métodos de pesquisa, incluindo a análise de registros e experimentos em plataformas de crowdsourcing podem auxiliar a entender as barreiras enfrentadas pelos trabalhadores na multidão em projetos de crowdsourcing de software. Também pretende-se trabalhar com diferentes tipos de desafios, plataformas e trabalhadores para identificar padrões, comportamentos e maneiras de melhorar a forma como os trabalhadores na multidão contribuem em projetos de crowdsourcing de software.

Software crowdsourcing é basicamente formado por tarefas complexas, assim, é interessante investigar o “quanto” uma tarefa possa ser decomposta até o nível de uma HIT. Fazer tarefas relacionadas ao desenvolvimento de software apenas por meio de HIT são difíceis como argumentam Stol e Fitzgerald [20] pois, em sua grande maioria, são complexas e necessitam de trabalhadores na multidão especializados, além de necessitarem tempo para a sua execução, muito diferente do que ocorre com as HITs.

Outra oportunidade futura é oferecer ao trabalhador na multidão um mecanismo, baseado em sistemas de recomendação, que o auxilie a encontrar uma tarefa na plataforma

de acordo com suas habilidades ou preferências e expectativas de recompensas financeiras pessoais.

Investigar se a utilização de máquinas virtuais com Ambientes de Desenvolvimento Integrados (IDEs) com as plataformas de software crowdsourcing contribuem para que o trabalhador da multidão possa realizar a tarefa.

Como o software crowdsourcing é fortemente baseado em interação, justifica-se analisar futuramente se a utilização de canais ou mecanismos de comunicação (síncrono ou assíncrono) auxiliam o trabalhador na multidão no processo de ganhar alguma competição em plataformas de software crowdsourcing.

Vislumbra-se também uma possibilidade de verificar, por meio de experimentos se as recomendações que foram confirmadas auxiliam o trabalhador na multidão, não apenas no registro e submissão da tarefa, mas sim, a ganhar uma competição.

REFERÊNCIAS

- [1] Castells, Manuel "A sociedade em Rede – a era da informação: economia, sociedade e cultura". Paz & Terra – Volume 1, 2002,
- [2] Levy, P. "A inteligência coletiva: por uma antropologia do ciberespaço". Edições Loyola, 2007, 216p.
- [3] Tapscott, D.; Williams, D. A. "Wikinomics: How Mass Collaboration Changes Everything". Penguin, 2007, 320p.
- [4] Deloitte, "Navigating the future of crowd", Relatório Técnico, Deloitte Review, 2017, 202p.
- [5] Howe, J. "The rise of crowdsourcing". *Wired magazine*, vol. 14-6, Jan 2006, pp. 1-4.
- [6] Leimeister, J.M.; Huber, M.; Bretschneider, U.; Krcmar, H. "Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition". *Journal of Management Information Systems*, vol. 26-1, Dez 2009, pp. 197–224.
- [7] Thuan, H. N.; Antunes, P.; Johnstone, D. "Factors influencing the decision to crowdsource: A systematic literature review". *Information Systems Frontiers*, vol. 18-1, Jun 2016, pp. 47-68.
- [8] Cooper, S; Khatib, F; Treuille, A; Barbero, J.; Lee, J.; Beenen, M. "Predicting protein structures with a multiplayer online game". *Nature*, vol. 466-1, Ago 2010, pp. 756-760.
- [9] Brabham, C. D. "Crowdsourcing as a model for problem solving an introduction and cases". *Convergence: the international journal of research into new media technologies*, vol. 14-1, Fev 2008, pp. 75–90.
- [10] Xiao, L.; Paik, H.Y. "Supporting Complex Work in Crowdsourcing Platforms: A View from Service-Oriented Computing". In: Proceedings of the 23rd Australian IEEE Software Engineering Conference, 2014, pp. 11-14.
- [11] Doan, A; Ramakrishnan, R; Halevy, Y. A. "Crowdsourcing systems on the world-wide web". *Communications of the ACM*, vol. 54-4, Abr 2011, pp. 86-96.
- [12] Hosseini, M.; Phalp, K.T.; Taylor, J.; Ali, R. "Towards crowdsourcing for requirements engineering". In: Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, 2014, pp. 43-69.
- [13] LaToza, T.; Towne, W.; van der Hoek, A.; Herbsleb, J. "Crowd development". In: Proceedings of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering, 2013, pp. 85-88.

- [14] Mao, K.; Yang, Y.; Li, M.; Harman, M. "Pricing crowdsourcing-based software development tasks". In: Proceedings of the International Conference on Software Engineering, 2013, pp. 1205-1208.
- [15] Wu, W.; Li, W.; Tsa, T. W. "An evaluation framework for software crowdsourcing". *Frontiers of Computer Science*, vol. 7-5, Out 2013, pp. 694-709.
- [16] Howe, J. "O Poder das Multidões". Campus, 2009, 296p.
- [17] Marjanovic, S.; Fry, C.; Chataway, J. "Crowdsourcing based business models: In search of evidence for innovation 2.0". *Science and public policy*, vol. 39-3, Jun 2012, pp. 318-332.
- [18] Kaganer, C. D.; Carmel, E.; Hirschheim, and Olsen, T. R. "Managing the human cloud". *Sloan Management Review*, vol. 54-1, Dez 2013, pp. 23-32.
- [19] Johns, T.; Laubscher, J. R.; Malone, W. T. "The age of hyper specialization". *Harvard Business Review*, vol. 89-1, Jul 2011, pp. 56-65.
- [20] Fitzgerald, B.; Stol, K. "Two's company, three's a crowd: a case study of crowdsourcing software development". In: Proceedings of the 36th International Conference on Software Engineering, 2014, p. 187.
- [21] LaToza, D. T.; Chen, M.; Jiang, L.; Zhao, M.; Van Der Hoek, A. "Borrowing from the crowd: A study of recombination in software design competitions". In: Proceedings of the 37th International Conference on Software Engineering, 2015, p. 551.
- [22] LaToza, D. T.; van der Hoek, A. "Crowdsourcing in Software Engineering: Models, Motivations, and Challenges". *IEEE Software*, vol. 33-1, Jan 2016, pp. 74-80.
- [23] Schmid-Druner, M. "The Situation of Workers in the Collaborative Economy", Relatório Técnico, European Parliament, 2016, 108p.
- [24] Felstiner, Alek. "Working the crowd: employment and labor law in the crowdsourcing industry". *Berkeley Journal of Employment & Labor Law*, vol. 32-1, Ago 2011, p. 92.
- [25] Mao, K.; Yang, Y.; Wang, Q.; Jia, Y.; Harman, M. "Developer Recommendation for Crowdsourced Software Development Tasks". In: Proceedings of the Service-Oriented System Engineering, 2015, pp. 347-356.
- [26] Lakhani, K; Garvin, E; Lonstein, D. "TopCoder (a): developing software through crowdsourcing". *Harvard Bussines School General Management Unit Case*, vol. 610-32, Mai 2010, p. 20.
- [27] Branquinho, L. C. "Crowdsourcing: uma forma de inovação aberta". Centro de Tecnologia Mineral / Ministério da Ciência, Tecnologia e Inovação, 2016, 194p.

- [28] Archak, N. "Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder.com". In: Proceedings of the 19th International World Wide Web Conference, 2010, pp. 21-30.
- [29] Hosseini, M.; Phalp, K.; Taylor, J.; Raian, A. "The four pillars of crowdsourcing: A reference model". In: Proceedings of the 8th International Conference on Research Challenges in Information Science, 2014, pp. 1-12.
- [30] Qureshi, I.; Fang, Y. "Socialization in open source software projects: A growth mixture modeling approach". *Organizational Research Methods*, vol. 14-1, Ago 2010, pp. 210-238.
- [31] Kraut, E R.; Resnick, P.; Kiesler, S.; Burke, M.; Chen, Y.; Kittur, N. "Building successful online communities: Evidence-based social design". Mit Press, 2012, 328p.
- [32] Steinmacher, F. I. "Supporting newcomers to overcome the barriers to contribute to open source software projects", Tese de Doutorado, Universidade de São Paulo, 2015, 183p.
- [33] Saengkhattiya, M.; Sevandersson, M.; Vallejo, U. "Quality in Crowdsourcing", Dissertação de Mestrado, Department of Informatics, Lund University, 2012, 75p.
- [34] Saremi, L R.; Yang, Y. "Dynamic simulation of software workers and task completion". In: Proceedings of the 2nd International Workshop on CrowdSourcing in Software Engineering, 2015, pp. 17-23.
- [35] Yang, Y.; Karim, R. M.; Saremi, R.; Ruhe, G. "Who Should Take This Task?: Dynamic Decision Support for Crowd Workers". In: Proceedings of the 10th International Symposium on Empirical Software Engineering and Measurement, 2016, p. 8.
- [36] Hill, C. W.; Hollan, D. J.; Wroblewski, D.; Mccandless, T. "Edit wear and read wear". In: Proceedings of the Conference on Human Factors in Computing Systems, 1992, pp. 3-9.
- [37] Stol, K.; Fitzgerald, B. "Researching crowdsourcing software development: perspectives and concerns". In: Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering, 2014, pp. 7-10.
- [38] Mehta, D. "An Insight into Software Crowd Sourcing: How Crowd can transform the Business Model for Technology Service Providers". *International Journal of Computer Applications*, vol. 101-12, Set 2014, pp. 34-40.
- [39] Zhao, Y.; Zhu, Q. "Evaluation on crowdsourcing research: Current status and future direction". *Information Systems Frontiers*, vol. 16-3, Jul 2014, pp. 417-434.
- [40] Lu, X.; Phang, W.; Yu, J. "Encouraging participation in virtual communities through usability and sociability development: An empirical investigation". *Special Interest Group on Management Information Systems*, vol. 42-3, Ago 2011, pp. 96-114.

- [41] Straub, T.; Gimpel, H.; Teschner, F.; Weinhardt, C. "How (not) to Incent Crowd Workers - Payment Schemes and Feedback in Crowdsourcing". *Business & Information Systems Engineering*, vol. 57-3, Jun 2015, pp. 167-179.
- [42] Brabham, C. D. "Crowdsourcing". John Wiley & Sons, 2013, 168p.
- [43] Yvonne, D.; Michael, J.; Janice, S. "For the Special Issue on Qualitative Software Engineering Research". *Information and Software Technology*, vol. 49-6, Jun 2007, pp. 531–539.
- [44] Strauss, A.; Corbin, J. "Basics of qualitative research: Techniques and procedures for developing grounded theory". Sage Publications, 1998, 456p.
- [45] Gil, Carlos Antonio "Como elaborar projetos de pesquisa". Atlas, 2010, 200p.
- [46] Yin, K. R. "Case Study Research and Applications: Design and Methods". Sage Publications, 2013, 320p.
- [47] Mao, K.; Licia, C.; Harman, M.; Yue, J. "A Survey of the Use of Crowdsourcing in Software Engineering". *Journal of Systems and Software*, vol. 126-1, Abr 2016, pp. 57-84.
- [48] Green, E. A.; De Hoyos, M.; Barnes, A. S.; Baldauf, Behle, H. B., "CrowdEmploy crowdsourcing case studies: an empirical investigation into the impact of crowdsourcing on employability", EUR - Scientific and Technical Research Reports, Publications Office of the European Union, 2013, 108p.
- [49] Herbsleb, J.; Moitra, D. "Global Software Development". In: Proceedings of the International Conference on Software Engineering, 2001, pp. 81-90.
- [50] Audy, J.; Prikladnicki, R. "Desenvolvimento Distribuído de Software". Elsevier, 2008, 220p.
- [51] Karolak, D. "Global Software Development – Managing Virtual Teams and Environments". Wiley-IEEE Computer Society, 1998, 159p.
- [52] Estellés-Arolas, E.; González-Ladrón-De-Guevara, F. "Towards an integrated crowdsourcing definition". *Journal of Information Science*, vol. 38-2, Abr 2012, pp. 189–200.
- [53] Allahbakhsh, M. "Quality control in crowdsourcing systems: issues and directions". *IEEE Internet Computing*, vol. 17-2, Mar 2013, pp. 76–81.
- [54] Peng, X.; Ali Babar, M.; Ebert, C. "Collaborative Software Development Platforms for Crowdsourcing". *IEEE Software*, vol. 31-2, Mar 2014, pp. 30–36.
- [55] O'Reilly, T. "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software". Capturado em: <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>, Maio 2005.

- [56] Keimel, C. "QualityCrowd — A framework for crowd-based quality evaluation". In: Proceedings of the Picture Coding Symposium, 2012, pp. 245–248.
- [57] Ramakrishnan, S.; Srinivasaraghavan, V. "Delivering software projects using captive university crowd". In: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, 2014, pp. 115-118.
- [58] Kittur, A.; Nickerson, V. J.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J; J. Horton, "The future of crowd work". In: Proceedings of the Conference on Computer Supported Cooperative Work, 2013, pp. 1301-1318.
- [59] Giles, J. "Internet encyclopaedias go head to head". *Nature*, vol. 438-1, Dez 2005, pp. 900–901.
- [60] Stewart, O.; Huerta, J.; Sader M., "Designing crowdsourcing community for the enterprise". In: Proceedings of the Workshop on Human Computation, 2009, pp. 50–53.
- [61] Whitla, P. "Crowdsourcing and its application in marketing activities". *Contemporary Management Research*, vol. 5-1, Mar 2009, pp. 15-28.
- [62] Vukovic, M. "Crowdsourcing for enterprises". In: Proceedings of the World Conference on Services, 2009, pp. 686–692.
- [63] Schenk, E.; Guittard, C. "Towards a characterization of crowdsourcing practices". *Journal of Innovation Economics & Management*, vol. 7-1, Mar 2011, pp. 93-107.
- [64] Kittur, A.; Chi, E.H.; Suh, B. "Crowdsourcing user studies with Mechanical Turk". In: Proceedings of the Human factors in computing systems, 2008, pp. 453– 456.
- [65] Prikladnicki, R.; Machado, L.; Carmel, E.; Souza, C. "Brazil software crowdsourcing: a first step in a multi-year study". In: Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering, 2014, pp. 1–4.
- [66] Li, K. "Analysis of the key gactors for software quality in crowdsourcing development: An empirical study on TopCoder.com". In: Proceedings of the 37th Annual Computer Software and Applications Conference, 2013, pp. 812–817.
- [67] Wu, W.; Huhns, N. M.; Tsai, W. "Cloud-Based Software Crowdsourcing". *IEEE Internet Computing*, vol. 18-3, Mai 2014, pp. 78–83.
- [68] Zanatta, L. A.; Machado, S. L.; Pereira, B. G.; Prikladnicki, R.; Carmel, E. "Software Crowdsourcing Platforms". *IEEE Software*, vol. 33-6, Nov 2016, pp. 112-116.
- [69] Schenk, E.; Guittard, C. "Crowdsourcing: What can be outsourced to the crowd, and why". In: Proceedings of the Workshop on Open Source Innovation, 2009, p. 72.

- [70] Saxton, D. G.; Oh, O.; Kishore, R. "Rules of crowdsourcing: Models, issues, and systems of control". In: Proceedings of the Information Systems Management, 2013, pp. 2-20.
- [71] Cullina, E.; Conboy, K.; Morgan, L. "Measuring the crowd: a preliminary taxonomy of crowdsourcing metrics". In: Proceedings of the 11th International Symposium on Open Collaboration, 2015, p. 7.
- [72] Olson, L. D.; Rosacker, K. "Crowdsourcing and open source software participation". *Service Business*, vol. 7-4, Dez 2013, pp. 499-511.
- [73] Usui, Y.; Morisaki, S. "An approach for crowdsourcing software development". In: Proceedings of the International Workshop on Software Measurement, 2011, pp. 32–33.
- [74] Bechmann, A.; Lomborg, S. "Mapping actor roles in social media: Different perspectives on value creation in theories of user participation". *New media & society*, vol. 15-5, Nov 2013, pp. 765-781.
- [75] Malinen, S. "Understanding user participation in online communities: A systematic literature review of empirical studies". *Computers in Human Behavior*, vol. 46-C, Mai 2015, pp. 228-238.
- [76] Codagnone, C.; Abadie, F.; Biagi, F. "The future of work in the 'sharing economy', market efficiency and equitable opportunities or unfair", JRC Science for Policy Report, Publications Office of the European Union, 2016, 100p.
- [77] Dubey, A.; Viridi, G.; Kuriakose, S.; Arora, V. "Towards adopting alternative workforce for software engineering." In: Proceedings of the 11th International Conference on Global Software Engineering, 2016, pp. 16-23.
- [78] Simperl, E. "How to use crowdsourcing effectively: Guidelines and examples". *Liber Quarterly*, vol. 25-1, Mar 2015, pp. 18-39.
- [79] Nielsen, J. "Participation inequality: Encouraging more users to contribute", Relatório Técnico, Nielsen Norman Group, 2006, 25p.
- [80] Martineu, E. "A typology of crowdsourcing participation styles", Dissertação de Mestrado, Concordia University, 2012, 82p.
- [81] Rohrmeier, P. "Social networks and online communities—managing user acquisition, activation and retention", Thesis dissertation, Technischen Universität München, 2012, 322p.
- [82] LaToza, D T.; Towne, B. W.; Adriano, M. C.; Van Der Hoek, A "Microtask programming: Building software with a crowd". In: Proceedings of the 27th annual ACM symposium on User interface software and technology, 2014, pp. 43-54.

- [83] Hoßfeld, T.; Hirth, M.; Redi, J.; Mazza, F.; Korshunov, P.; Naderi, B.; Keimel, C. "Best Practices and Recommendations for Crowdsourced QoE-Lessons learned from the Qualinet Task Force "Crowdsourcing"". In: Proceedings of the European Network on Quality of Experience in Multimedia Systems and Services, 2014, p. 10.
- [84] Fuller, J. "Why consumers engage in virtual new product developments initiated by producers". *Advances in Consumer Research*, vol. 33-1, Jan 2006, pp. 639–646.
- [85] Morgan, J.; Wang, R. "Tournaments for ideas". *California Management Review*, vol. 52-2, Dez 2010, pp. 77–97.
- [86] Martinez, G.; Walton, B. "The wisdom of crowds: The potential of online communities as a tool for data analysis". In: Proceedings of the Technovation, 2014, pp. 203-214.
- [87] Rouse, C. A. "A preliminary taxonomy of crowdsourcing". In: Proceedings of the 21st Australasian Conference on Information Systems, 2010, pp. 1-10.
- [88] Geiger, D.; Schader, M. "Personalized task recommendation in crowdsourcing information systems—Current state of the art". *Decision Support Systems*, vol. 65-C, Set 2014, pp. 3-16.
- [89] Suzuki, R.; Igarashi, T. "Collaborative 3D modeling by the Crowd". In: Proceedings of the 43rd International Conference on Graphics, Visualization and Human-Computer Interaction, 2017, pp. 124-131.
- [90] Ipeirotis, G. P. "Analyzing the amazon mechanical turk marketplace". *The ACM Magazine for Students*, vol. 17-2, Jan 2010, pp. 16-21.
- [91] Berinsky, J.; Huber, A.; Lenz, G. "Evaluating online labor markets for experimental research: Amazon.com's Mechanical Turk". *Political Analysis*, vol. 20-3, Ago 2012, pp. 351-368.
- [92] Agerfalk, P.J.; Fitzgerald, B.; Stol, K. "Software Outsourcing in the Age of Open: Leveraging the Unknown Workforce". Springer, 2015, 45-60p.
- [93] Figuerola, S. Ó; Azdic, V.; Shah, A.; Kalva, H. "Assessing internet video quality using crowdsourcing". In: Proceedings of the 2nd ACM international workshop on Crowdsourcing for Multimedia, 2013, pp. 23-28.
- [94] Ross, J.; Irani, L.; Silberman, M.S.; Zaldivar, A; Tomlison, B. "Who are the crowdsourcers? Shifting demographics in mechanical turk". In: Proceedings of the Conference on Human Factors in Computing Systems, 2010, pp. 2863-2872.
- [95] Paolacci, G.; Chandler, J.; Ipeirotis, P. "Running experiments on amazon mechanical turk". *Judgment and Decision Making*, vol. 5-5, Ago 2010, pp. 411-419.
- [96] Ipeirotis, P. "Crowdsourcing using mechanical turk: quality management and scalability". In: Proceedings of the 8th International Workshop on Information Integration on the Web, 2011, p. 8.

- [97] Silberman, S M.; L. Irani,; Ross, J. "Ethics and tactics of professional crowdwork". *XRDS Crossroads The ACM Magazine for Students*, vol. 17-2, Jan 2010, pp. 39-43.
- [98] Postigo, H. "From pongto planet quake: post-industrial transitions from leisure to work". *Information Communication and Society*, vol. 6-4, Dez 2003, pp. 593–607.
- [99] Wexler, N. M. "Reconfiguring the sociology of the crowd: exploring crowdsourcing". *International Journal of Sociology and Social Policy*, vol. 31-1/2, Jun 2011, pp. 6-20.
- [100] Busarovs, A. "Ethical Aspects of Crowdsourcing, or is it a Modern Form of Exploitation". *International Journal of Economics & Business Administration*, vol. 1-1, Jan 2013, pp. 3-14.
- [101] Sommerville, I. "Engenharia de Software". Pearson Prentice Hall, 2011, 544p.
- [102] Li, W; Tsai, T. W.; Wu, W. "Crowdsourcing for Large-Scale Software Development Crowdsourcing". *Springer Berlin Heidelberg*, vol. 10-1, Mai 2015, pp. 3-23.
- [103] Kazman, R.; Chen, M. "The metropolis model a new logic for development of crowdsourced systems". *Communication of the ACM*, vol. 52-7, Jul 2009, pp. 76–84.
- [104] Lacity, C. M.; Khan, S.; Yan, A.; Willcocks, P. L. "A review of the IT outsourcing empirical literature and future research directions". *Journal of Information technology*, vol. 25-4, Dez 2010, pp. 395-433.
- [105] Tajedin, H.; Nevo, D. "Determinants of success in crowdsourcing software development". In: *Proceedings of the Conference on Computers and people research*, 2013, pp. 173-178.
- [106] Xie, T.; Bishop, J.; Horspool, N. R.; Tillmann, N.; de Halleux, J. "Crowdsourcing code and process via code hunt". In: *Proceedings of the 2nd International Workshop on CrowdSourcing in Software Engineering*, 2015, pp. 15-16.
- [107] Verner, M.; Evanco, W. "In-house software development: what project management practices lead to success?" *IEEE Software*, vol. 22-1, Jan 2005, pp. 86-93.
- [108] Steinmacher, I.; Silva, A. G. M.; Gerosa, A. M.; Redmiles, F. D. "A systematic literature review on the barriers faced by newcomers to open source software projects". *Information and Software Technology*, vol. 59-C, Mar 2015, pp. 67-85.
- [109] Heizer, J.; Render, B. "Operations Management". Pearson/Prentice Hall, 2008, 912p.
- [110] Guaiani, F.; Muccini, H. "Crowd and laboratory testing can they co-exist?: an exploratory study". In: *Proceedings of the 2nd International Workshop on CrowdSourcing in Software Engineering*, 2015, pp. 32-37.
- [111] Boughzala, I; Vreede, T.; Nguyen, C.; Vreede, G.J. "Towards a maturity model for the assessment of ideation in crowdsourcing projects". In: *Proceedings of the 47th Hawaii International Conference*, 2014, pp. 483-490.

- [112] Lykourantzou, I.; Vergados, J. D.; Papadaki, K.; Naudet, Y. "Guided crowdsourcing for collective work coordination in corporate environments". In: Proceedings of the Computational Collective Intelligence. Technologies and Applications, 2013, pp. 90-99.
- [113] Bernstein, M.S; Little, G.; Miller, R.C. "Soylent: A Word Processor with a Crowd Inside". In: Proceedings of the 23rd annual ACM symposium on User interface software and technology, 2010, pp. 313-322.
- [114] Sharp, H.; Baddoo, N.; Beecham, S.; Hall, T.; Robinson, H. "Models of motivation in software engineering". *Information and Software Technology*, vol. 51-1, Jan 2009, pp. 219-233.
- [115] França, A.; Gouveia, T.; Santos, P.; Santana, C.; Silva, F. "Motivation in Software Engineering: A systematic literature review update". In: Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering, 2011, pp. 154-163.
- [116] Zhu, J.; Shen, B.; Hu, F. "A Learning to Rank Framework for Developer Recommendation in Software Crowdsourcing". In: Proceedings of the Asia-Pacific Software Engineering Conference, 2015, pp. 285-292.
- [117] Kinnaird, P; Dabbish, L; Kiesler, S; Faste, H "Co-worker transparency in a microtask marketplace". In: Proceedings of the Computer supported cooperative work, 2013, pp. 1285-1290.
- [118] Jayakanthan, R.; Sundararajan, D. "Enterprise crowdsourcing solutions for software development and ideation". In: Proceedings of the 2nd international workshop on Ubiquitous crowdsourcing, 2011, pp. 25-28.
- [119] Jayakanthan, R.; Sundararajan, D. "Enterprise crowdsourcing solution for software development in an outsourcing organization". In: Proceedings of the International Conference on Web Engineering, 2011, pp. 177-180.
- [120] Varshney, R. L. "Participation in crowd systems". In: Proceedings of the Communication, Control, and Computing, 2012, pp. 996-1001.
- [121] Sherief, N. "Software evaluation via users' feedback at runtime". In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014, p. 58.
- [122] Difallah, E. D.; Demartini, G.; Cudré-Mauroux, P. "Pick-a-crowd: tell me what you like, and i'll tell you what to do". In: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 367-374.
- [123] Kulkarni, Anand; Can, Matthew; Hartmann, Björn "Collaboratively crowdsourcing workflows with turkomatic". In: Proceedings of the Conference on Computer Supported Cooperative Work, 2012, pp. 1003-1012.

- [124] Begel, A.; Simon, B. "Novice software developers, all over again". In: Proceedings of the 4th International Workshop on Computing Education, 2008, pp. 3-14.
- [125] Berlin, M. L. "Beyond program understanding: A look at programming expertise in industry", Relatório Técnico, HP Software Technology Laboratory, 1993, 16p.
- [126] Sim, E. S.; Holt, C. R. "The ramp-up problem in software projects: A case study of how software immigrants naturalize". In: Proceedings of the 20th International Conference on Software Engineering, 1998, pp. 361-370.
- [127] German, Daniel M. "GNOME, a case of open source global software development". In: Proceedings of the International Workshop on Global Software Development, 2003, pp. 39-43.
- [128] Steinmacher, I; Chaves, P A.; Conte, T.; Gerosa, A. M. "Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects". In: Proceedings of the Software Engineering Brazilian Symposium IEEE, 2014, pp. 51-60.
- [129] Steinmacher, I.; Conte, T.; Gerosa, A. M.; Redmiles, D. "Social barriers faced by newcomers placing their first contribution in open source software projects". In: Proceedings of the Computer supported cooperative work & social computing, 2015, pp. 1379-1392.
- [130] Von Krogh, G.; Spaeth, S.; Lakhani, R. K. "Community, joining, and specialization in open source software innovation: a case study". In: Proceedings of the Research Policy, 2003, pp. 1217-1241.
- [131] Zanatta, L. A.; Steinmacher, I.; Machado, L.; Souza, R.B de C.; Prikladnicki, R. "Barriers Faced by Newcomers in Software Crowdsourcing Projects". *IEEE Software*, vol. 34-2, Mar 2017, pp. 37-43.
- [132] Cordeiro, M. A.; Oliveira, M G.; Rentería, M. J.; Guimarães, A. C. "Revisão sistemática: uma revisão narrativa". *Revista do Colégio Brasileiro de Cirurgiões*, vol. 6-34, Nov 2007, pp. 428-431.
- [133] Machado, L; Kroll, J; Prikladnicki, R.; de Souza, R C.; Carmel, E. "Software crowdsourcing challenges in the Brazilian IT Industry". In: Proceedings of the 18th International Conference on Enterprise Information Systems, 2016, pp. 482-489.
- [134] Machado, L.; Zanatta, A. L.; Marczak, S.; Prikladnicki, R. "The good, the bad and the ugly: an onboard journey in software crowdsourcing competitive model". In: Proceedings of the 4th International Workshop on CrowdSourcing in Software Engineering, 2017, pp. 2-8.
- [135] Bederson, B.; Quinn, J. A. "Web workers, unite! addressing challenges of online laborers". In: Proceedings of the Extended Abstracts on Human Factors in Computing Systems, 2011, pp. 97-106.

- [136] Wu, W.; Tsai, T. W.; Li, W. "Creative software crowdsourcing: from components and algorithm development to project concept formations". *International Journal of Creative Computing*, vol. 1-1, Mar 2013, pp. 57-91.
- [137] Satzger, B.; Zabolotnyi, R.; Dustdar, S.; Wild, S.; Gaedke, M.; Göbel, S.; Nestler, T "Toward Collaborative Software Engineering Leveraging the Crowd". In: Proceedings of the Architecture, Economics-Driven Software, 2014, pp. 159-181.
- [138] Dustdar, S.; Gaedke, M. "The social routing principle". *Internet Computing*, vol. 15-4, Jul 2011, pp. 80-83.
- [139] Luther, K.; Dow, S.; Kittur, A. "How can crowdsourcing help individuals learn?" In: Proceedings of the Workshop on Designing Futures for Peer-to-Peer Learning, 2014, pp. 1-4.
- [140] Preist, C.; Massung, E.; Coyle, D. "Competing or aiming to be average?: Normification as a means of engaging digital volunteers". In: Proceedings of the Computer-Supported Cooperative Work, 2013, pp. 1222-1233.
- [141] Chanal, V.; Caron-Fasan, M. "How to invent a new business model based on crowdsourcing: the Crowdspirit® case." In: Proceedings of the Conférence de l'Association Internationale de Management Stratégique, 2008, pp. 1-27.
- [142] Djelassi, S.; Decoopman, I. "Customers' participation in product development through crowdsourcing: Issues and implications". *Industrial Marketing Management*, vol. 42-5, Jul 2013, pp. 683-692.
- [143] Erickson, T. "Some thoughts on a framework for crowdsourcing". In: Proceedings of the Workshop on crowdsourcing and human computation, 2011, pp. 1-4.
- [144] De Souza, S. C.; Preece, J. "A framework for analyzing and understanding online communities". *Interacting with computers*, vol. 16-3, Jun 2004, pp. 579-610.
- [145] Horton, J. J.; Chilton, B. L. "The labor economics of paid crowdsourcing". In: Proceedings of the 11th ACM Conference on Electronic Commerce, 2010, pp. 209-218.
- [146] Fershtman, C.; Gneezy, U. "The tradeoff between performance and quitting in high power tournaments". *Journal of the European Economic Association*, vol. 9-2, Apr 2011, pp. 318-336.
- [147] Eriksson, T.; Poulsen, A.; Villeval, C. M. "Feedback and incentives: Experimental evidence", Discussion Paper, IZA - Institute for the Study, 2009, 42p.
- [148] Zhang, H.; Kitchenham, B.; Jeffery, R. D. "Toward trustworthy software process models: an exploratory study on transformable process modeling". *Journal of Software: Evolution and Process*, vol. 24-7, Ago 2012, pp. 741-763.

- [149] Park, Y.; Jensen, C. "Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers". In: Proceedings of the 5th IEEE International Workshop Visualizing Software for Understanding and Analysis, 2009, p. 3.
- [150] Khazankin, R.; Satzger, B.; Dustdar, S. "Optimized execution of business processes on crowdsourcing platforms". In: Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2012, pp. 443-451.
- [151] MacLean, D.; Yoshida, K.; Edwards, A.; Crossman, L.; Clavijo, B.; Clark, M.; Caccamo, M. "Crowdsourcing genomic analyses of ash and ash dieback—power to the people". *GigaScience*, vol. 2-2, Feb 2013, p. 10.
- [152] Panichella, S.; Bavota, G.; Penta, Di M.; Canfora, G.; Antoniol, G. "How developers' collaborations identified from different sources tell us about code changes". In: Proceedings of the IEEE international Conference on Software Maintenance and Evolution, 2014, pp. 251-260.
- [153] Alonso, O.; Rose, E. D.; Stewart, B. "Crowdsourcing for relevance evaluation". *Special Interest Group on Information Retrieval*, vol. 42-2, Dez 2008, pp. 9-15.
- [154] Panichella, S.; Canfora, G.; Penta, Di M.; Oliveto, R. "How the evolution of emerging collaborations relates to code changes: an empirical study". In: Proceedings of the 36th International Conference on Program Comprehension, 2014, pp. 177–188.
- [155] Bari, E.; Johnston, M.; Wu, W.; Tsai, T. W. "Software Crowdsourcing Practices and Research Directions". In: Proceedings of the Service-Oriented System Engineering , 2016, pp. 372-379.
- [156] Júnior, B. "Gamifying user interaction to increase collaboration: the game conceptual framework", Dissertação de Mestrado, Universidade Federal da Bahia, 2016, 151p.
- [157] O'Neill, J.; Roy, S.; Grasso, A.; Martin, D. "Form digitization in BPO: from outsourcing to crowdsourcing?" In: Proceedings of the Human factors in computing systems, 2013, pp. 197-206.
- [158] Dunne, J. K. "Computer-Assisted Translation. The Encyclopedia of Applied Linguistics", Relatório Técnico, Wiley Online Library , 2013, 220p.
- [159] Trujillo, A. "Translation engines: techniques for machine translation". Springer Science & Business Media, 2012, 303p.
- [160] Bowker, L. "Computer-Aided Translation Technology: A Practical Introduction". University of Ottawa Press, 2010, 220p.
- [161] Taravella, A.; Villeneuve, O. A. "Acknowledging the needs of computer-assisted translation tools users: the human perspective in human-machine translation". *The Journal of Specialised Translation*, vol. 19-1, Jan 2013, pp. 62-74.

- [162] Muegge, U. "Teaching computer-assisted translation in the 21st century". *Translatologische Interdependenzen*, vol. 59-1, Mar 2013, pp. 137-146.
- [163] Hu, C.; Bederson, B. B.; Resnik, P. "Translation by iterative collaboration between monolingual users". In: *Proceedings of the Graphics Interface - Canadian Information Processing Society*, 2010, pp. 39-46.
- [164] Yang, J.; Redi, J.; Demartini, G.; Bozzon, A. "Modeling task complexity in crowdsourcing". In: *Proceedings of the 4th Conference on Human Computation and Crowdsourcing*, 2016, pp. 1-10.
- [165] Bourque, P.; Fairley, R. "SWEBOK. Guide to the Software Engineering Body of Knowledge". IEEE Computer Society, 2014, 348p.
- [166] Hannebauer, C.; Book, M.; Gruhn, V. "An exploratory study of contribution barriers experienced by newcomers to open source software projects". In: *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering*, 2014, pp. 11-14.
- [167] Radatz, J.; Geraci, A.; Katki, F. "IEEE standard glossary of software engineering terminology", Relatório Técnico, IEEE Standards Board, 1990, 84p.

APÊNDICE A – E-mail enviado aos participantes Fase 1

Subject: Participants being sought for a Crowdsourcing in software engineering research study

Hi [name]

My name is Alexandre Lazaretti Zanatta and I am Ph.D. student at Pontifical Catholic University of Rio Grande do Sul (PUCRS) - Brazil, and I am looking for participants for my research study. You are receiving this email because you are a newcomer Topcoder member.

This study aims to know how to support new contributors during their first steps in the crowdsourcing projects. My first goal is to find out what are the barriers and difficulties that influence newcomers' to Crowdsourcing in software engineering projects. I'd like to hear Topcoder community, interviewing new contributors. If you take part in this study, I will conduct the interview via textual chat or Skype, and we can schedule it at the time that fits better for you. To be able to take part in this study, you must be at least 18 years old and have experience in software engineering and be able to communicate in English.

If you are interested in participating or have any questions about the study, please send me a private email to alexandre.zanatta@acad.pucrs.br or call me via Skype id:alexandrezanatta.

Regards

Alexandre

APÊNDICE B – Questões abertas – Fase 1

[script 1] newcomers – (adaptado de Steinmacher [32]) versão 3 de 31/05/2016

PROFILING

- How old are you?
- How many years of experience do you have as a software developer?
- How many years of experience do you have with crowdsourcing development?
- Do you know/use Crowdsourcing platforms (other than TopCoder)? Which are?
- how much time (hours/week) do you spend contributing on Crowdsourcing Software projects? [] Less than 5 [] From 5 to 10 [] From 10 to 20 [] More than 20

MOTIVATION

- What motivated you to start contributing to an Crowdsourcing Software project?
- How receptive was the “platform”?
- Why do you keep contributing? (Or “Why did you stop contributing?” or “What demotivated you?”)

JOINING

- What kind of tasks had you conducted when you started?
- And currently, what tasks do you perform?

PROBLEMS

- Why didn't you contribute more?
 - Apêndice A What do you see as the main drawbacks of the crowdsourcing model (based on you experience on TopCoder)?
- What are your main “frustrations” with the project (considering joining process)?
 - How did you handle them?

SUGGESTIONS

- What steps do you consider important for a newcomer to follow to become a contributor?
 - Were these steps clearly presented to you?
 - How had you learnt?
 - Which mechanisms would you propose to reduce the problems faced by newcomers when joining the work? Can you think about something?

If you use one or more platforms, do you believe that these are different? Can you compare it.

APÊNDICE C – Protocolo do estudo de caso

Etapa / Atividade	Realizado	Comentário
1. revisão dos principais artigos e publicações sobre o assunto abordado na pesquisa;	✓	realizado na fase 1
2. definição da unidade caso;	✓	realizado na fase 1
3. obtenção da autorização formal	X	
4. elaboração do plano de amostragem;		
5. estabelecimento dos instrumentos de coleta de dados;	✓	conforme Apendice
6. investigação sobre as técnicas de análise de dados;	✓	Teoria Fundamentada nos Dados
7. delineamento e formatação das questões propostas no questionário	✓	Apendice
8. realização do pré-teste dos instrumentos de coleta de dados;	✓	
9. construção de um banco de dados para armazenamento das respostas obtidas pelos sujeitos participantes;	✓	Ferramenta Nvivo
10. avaliação, análise, interpretação e discussão dos resultados (triangulação);	✓	
11. elaboração do relatório final	✓	Redação de um artigo

Legenda: V realizado

APÊNDICE D – Termo de consentimento



Faculdade de Informática /PUCRS

Programa de Pós-Graduação em Ciência da Computação

Avenida Ipiranga, 6681 – Prédio 32 - 90619-900 – Porto Alegre – RS

Termo de Consentimento Livre e Esclarecido

A PUCRS, através das equipes de Engenharia de Software da Faculdade de Informática, agradece a todos os participantes de entrevistas realizados sob sua responsabilidade, a inestimável contribuição que prestam para o avanço da pesquisa sobre Engenharia de Software. O objetivo desta pesquisa é compreender a colaboração e as barreiras enfrentadas por trabalhadores na multidão quando participam de atividades de software crowdsourcing no modelo competitivo. Com isto visamos identificar como e quando ocorrem essas barreiras. Além disso, visa-se descobrir possíveis recomendações para que esses trabalhadores possam submeter a atividade/tarefa. Espera-se que os resultados desta pesquisa possam contribuir com dados para a consolidação do entendimento sobre o uso alternativo no processo de desenvolvimento de software.

Lembramos que o objetivo deste estudo **não** é avaliar o entrevistado, **mas sim** entender como está sendo usado o BDD. O uso que se faz dos registros efetuados durante a entrevista é **estritamente** limitado às atividades de pesquisa e desenvolvimento, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências, periódicos, livros e assemelhados) ou pedagógicos (tais como apostilas de cursos, *slides* de apresentações, e assemelhados).
2. Todo entrevistado terá acesso às cópias destes documentos após a publicação dos mesmos.
3. Todo entrevistado que se sentir constrangido ou incomodado durante uma situação da entrevista pode interromper a entrevista e estará fazendo um favor à equipe se registrar por escrito as razões ou sensações que o levaram a esta atitude. A equipe fica obrigada a descartar a entrevista para fins da avaliação a que se destinaria.
4. Os entrevistados que forem menores de idade terão, obrigatoriamente, que apresentar o consentimento de seu responsável, para participação no estudo, o qual será declarado ciente do estudo a ser realizado através de sua assinatura no presente Termo de Consentimento.
5. Todo entrevistado tem direito de expressar por escrito, na data da entrevista, qualquer restrição ou condição adicional que lhe pareça aplicar-se aos itens acima enumerados (1, 2, 3 e 4). A equipe se compromete a observá-las com rigor e entende que, na ausência de tal manifestação, o participante concorda que rejam o comportamento ético da equipe somente as condições impressas no presente documento.
6. A equipe tem direito de utilizar os dados das entrevistas, mantidas as condições acima mencionadas, para quaisquer fins acadêmicos, pedagógicos e/ou de desenvolvimento contemplados por seus membros.

[a ser preenchido pelo entrevistador]

Forma: _____ Data: __ / __ / ____

Condições especiais (caso não haja condições especiais, escreva "nenhuma"):

continua no verso

Por favor, indique sua posição em relação aos termos acima:

Estou de pleno acordo com os termos acima.

Em anexo registro condições adicionais para este teste.

Assinatura do participante

Assinatura do responsável
(caso o participante seja menor de idade)

Assinatura do pesquisador

Nome do Participante:

Pesquisador Responsável:

APÊNDICE E – Descrição da Tarefa – Fase 2

PUCRS – Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profa.

Software Crowdsourcing: Entrega 1 – Descrição da Tarefa Realizada

Dados do aluno

1. Nome completo:
2. Username na plataforma TopCoder:
3. E-mail de contato na plataforma TopCoder

Dados da tarefa

4. Tipo de Development Challenge

- | | |
|---|---|
| a. <input type="checkbox"/> Architecture | i. <input type="checkbox"/> Component development |
| b. <input type="checkbox"/> Assembly | j. <input type="checkbox"/> F2F |
| c. <input type="checkbox"/> Bug hunt | k. <input type="checkbox"/> Marathon |
| d. <input type="checkbox"/> Code | l. <input type="checkbox"/> RIA |
| e. <input type="checkbox"/> Conceptualization | m. <input type="checkbox"/> Specification |
| f. <input type="checkbox"/> Content creation | n. <input type="checkbox"/> Test scenarios |
| g. <input type="checkbox"/> Copilot | o. <input type="checkbox"/> Test Suites |
| h. <input type="checkbox"/> Component design | p. <input type="checkbox"/> UI prototype |

5. Título e data de término da tarefa na plataforma TopCoder:
6. Breve descrição do objetivo da tarefa:
7. Breve descrição da solução provida (em Português)
8. Lista e descrição de passos realizados para desenvolver a solução provida
9. Procedimento de submissão da solução realizada
10. Requisitos para realização da tarefa

APÊNDICE F – Questões abertas – Fase 2

1. Explique como se deu o processo de seleção da tarefa que você realizou na plataforma. Por exemplo, como você decidiu por qual tarefa realizar? Você decidiu por uma tarefa e por algum motivo acabou realizando uma outra? Qual a razão? Nota: Toda e qualquer informação relevante para o entendimento do motivo da seleção da tarefa realizada é de interesse e deve ser relatado.
2. Que aspectos você considerou interessante na sua experiência em *software* crowdsourcing com a plataforma TopCoder?
3. Colaboração é considerado um aspecto importante durante o processo de desenvolvimento de software, conforme foi discutido na disciplina. Como você percebeu a colaboração em *software* crowdsourcing na plataforma TopCoder?
 - Entre os membros da plataforma
 - Entre a plataforma (ou cliente) e os membros da plataforma
4. Você acredita que a plataforma utilizada suporta a colaboração? Justifique sua resposta.
5. Na sua opinião, quais as dificuldades para colaboração você enfrentou na plataforma ou durante a participação da tarefa? Se você acredita que não encontrou nenhuma dificuldade, explique sua razão por ter esta crença.
6. Quais as suas sugestões para minimizar as dificuldades de colaboração encontradas na plataforma ou na participação de uma tarefa? Nota: A questão não precisa ser respondida caso você tenha respondido que não enfrentou dificuldades na Questão 5.
7. Quais são as atividades/etapas, na sua opinião, que mais necessitam de suporte durante a participação na plataforma?
8. Que outros tipos de dificuldades foram encontradas por você durante a participação na plataforma? As dificuldades podem ser de ordem pessoal e/ou técnica, tanto na utilização da plataforma como na realização da tarefa.
 - Na utilização da plataforma
 - Na realização da tarefa
9. Que sugestões você propõe para minimizar estas dificuldades? Nota: A questão não precisa ser respondida caso você tenha respondido que não enfrentou dificuldades na Questão 8

17 - Você venceu alguma competição? *

Considerar apenas para a tarefa enviada para a realização desse trabalho. *Mark only one oval.*

Sim Não Não tenho condições de responder

Aspectos sobre a Colaboração**18 - Você utilizou algum canal externo à plataforma (p.ex Github, stackoverflow) para a resolução da tarefa? Caso sim, indique qual o canal. * *Mark only one oval.***

Sim Não

19 - Você considera que houve algum tipo de colaboração para apoiar ou executar o processo de desenvolvimento da tarefa na plataforma? * *Mark only one oval.*

sim Não Não tenho condições de responder

20 - A plataforma apoia a troca de informação ou elucidação de dúvidas durante a execução da tarefa entre os participantes? * *Mark only one oval.*

sim Não Não tenho condições de responder

21 - Você percebeu a existência da figura (ou papel) de um intermediador durante a execução da tarefa? * *Mark only one oval.*

sim Não Não tenho condições de responder

SOBRE A PLATAFORMA**22 - Em linhas gerais, qual o seu grau de satisfação com a plataforma TopCoder? ***

Mark only one oval per row.

	Concordo Fortemente	Concordo	Nem concordo nem discordo	Discordo	Discordo Fortemente
Funcionalidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Confiabilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Usabilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eficiência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
No Geral	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

23 - Encontrou algum problema ao utilizar a plataforma TopCoder? (considere questões técnicas e não-técnicas) *

Sim Não

a - Quais foram os problemas? (considere questões técnicas e não-técnicas) ***SOBRE O SOFTWARE CROWDSOURCING****24 - Você acredita que é possível desenvolver software utilizando o modelo crowdsourcing? * *Mark only one oval.***

Sim Não Other:

24a - Porque você ACREDITA que é possível desenvolver software utilizando o modelo crowdsourcing? ***24b - Porque você NÃO acredita que é possível desenvolver software utilizando o modelo crowdsourcing**

APÊNDICE H – E-mail enviado – Fase 3

Modelo de primeiro e-mail enviado aos especialistas

“Dear,

I am Ph.D. student at PUCRS University – Brazil. My advisor is Prof. Dr. Rafael Prikladnicki, and my co-advisor is Prof. Dr. Igor Steinmacher. We are contacting you because you were member of the program committee of a specialized event or authored a paper related to crowdsourcing software development. This study aims to understand how to support crowdworkers during their participation (registration and submission) in Competition-Based Crowdsourcing Software Development.

So, I've invited you to fill out a form at <https://goo.gl/forms/Ys2HomFkT56oltW03>

Regards

Alexandre L. Zanatta”

Modelo do segundo e-mail enviado aos especialistas (cedido gentilmente pela Leticia)

E-mail foi enviado aos especialistas uma semana após o envio do primeiro email.

“Hello,

I send you this friendly reminder to answer my survey at <https://goo.gl/forms/Ys2HomFkT56oltW03>

Your answers are pretty critical for the success of my PhD, the reason why I kindly ask you to spend 15 min of your time.

Thank you so much for this.

Alexandre”

APÊNDICE I – Questões – Fase 3

Barriers and Recommendations for Crowdworkers

My name is Alexandre Lazaretti Zanatta and I am Ph.D. student at Pontifical Catholic University of Rio Grande do Sul (PUCRS) – Brazil. My advisor is Prof. Dr. Rafael Prikladnicki, and my co-advisor is Prof. Dr. Igor Steinmacher, and we are looking for experts to give us feedback on our results. We are contacting you because you were member of the program committee of a specialized event or authored a paper related to crowdsourcing software development.

This study aims to understand how to support crowdworkers during their participation (registration and submission) in software crowdsourcing projects. We found 6 barriers faced by crowdworkers, and we provide with some (13) recommendations on how to minimize them.

This survey will take 15 min of your time. Your participation is very valuable and we appreciate you taking the time to complete it. Some questions are optional and the responses will not contain any identifying information about participants, and you can opt-out at any time. All research data collected will be stored securely and confidentially. This work will be part of my Ph.D. thesis.

If you have any questions about the study, please send me a email to alexandre.zanatta@acad.pucrs.br

Please return this survey before May, 7. If you agree, please select "NEXT" button (see bellow).

* Required

Barrier #1: It is hard to find a task, to understand and execute it according to crowdworkers profiles.

In Crowdsourcing Software Development, crowdworkers needed specific skills to find, to understand and to perform the tasks. Some skills are attained by attending training programs; others can be acquired through experience learning on the some. How would you rate the LEVEL OF AGREEMENT with the recommendations made to support crowdworkers overcoming the barrier "It's hard to find a task, to understand and execute it according to crowdworkers' profiles".

Recommendation #1: The crowdworker should use some system that supports task selection according to his/her personal learning versus earning objectives. **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Barrier #2: Even finding a task according to crowdworker profile, it's very difficult to finish it.

How would you rate the LEVEL OF AGREEMENT with the recommendations made to support crowdworkers overcoming the barrier: Even finding a task according to crowdworker profile's, it's very difficult to execute it.

Recommendation #2: The crowdworker should decompose a task in many other activities (micro-tasks) to help the comprehension of that task

It's the requester's responsibility to decompose a task. If crowdworker perceives any issues with the task scope or requirements, he may follow up with recommendation. ***Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Barrier #3: It's hard to manage time available to execute the task.

Crowdworkers need to manage personal available time to work on tasks and deliver solutions within the time frame defined in the platform. For each item below, how would you rate the LEVEL OF AGREEMENT with the recommendations made to support crowdworkers overcoming the barrier "it's hard to manage time available to execute the task."

Recommendation #3: The crowdworker should evaluate and manage the time to learn how to do the task **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #4: The crowdworker should create his/her own plan and manage his/her timeline to have enough time to finish the task **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #5: The crowdworker should define small activities to finish the task **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #6: The crowdworker should to determine the activity durations. **Mark only one oval.*

Once the task has been divided into small activities, the next step is to determine the activity durations ((how long it will take to accomplish from beginning to finish).

The crowdworker can perform a basic PERT (Program, Evaluation, and Review Technique) analysis to estimate a activity duration. Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #7: The crowdworker should prepare setup environment for performing each task. **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #8: The crowdworker should define timeline to finish the task. **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Barrier #4: It is hard to configure the necessary environment to perform the task.

Prepare a computational structure, with specific software and hardware, among other structures is a problem for the execution of coding tasks in Competition-Based Crowdsourcing Software Development because since it depends on information from the client, the platform, the documentation associated to the platform, as well as from other workers in the platform. For each item below, how would you rate the LEVEL OF AGREEMENT with the recommendations made to support crowdworkers overcoming the barrier: "It´s hard to configure the necessary environment to perform the task."

Recommendation #9: The crowdworker should use virtual machines with integrated development environments (IDEs) to execute the task. **Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #10: The crowdworker should use a Wiki as a collaborative integrated development environment (IDE) with support to specific services and applications to perform the task *Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Recommendation #11: The crowdworker should use IDEs under the concept of cloud computing to execute the task. *Mark only one oval.

Some of the examples of these IDEs are Codeanywhere, Cronapp and Onion Cloud. Open platforms for developers like Docker may help the preparation of setup environments for the execution of tasks in Competition-Based Crowdsourcing Software Development. *Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Barrier #5: It is difficult to collaborate with the other members of the platform to perform the task.

The software development activity is essentially collaborative. Crowdsourcing is a "massive form of parallel collaboration", and the communication between members of project plays an important role in any well-succeeded software project. For each item below, how would you rate the LEVEL OF AGREEMENT with the recommendations made to support crowdworkers overcoming the barrier "It's difficult to collaborate with the other members of the platform to perform the task."

Recommendation #12: The crowdworker should use different channels to communicate asynchronous or synchronously (because in competitive crowdsourcing software there is need for interaction generally). *Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Barrier #6: "For non-speaking English, and as the platform is in English, it is difficulty in understanding the task, and when performing the task, it is difficulty collaborating with other members of the platform because they only communicate in English."

Lack of English skills to express or understand a non-native language affects negatively the collaboration among crowd members and understanding the task while performing their tasks and is still a problem and an important barrier in the development of tasks in

Competition-Based Crowdsourcing Software Development. How would you rate the LEVEL OF YOUR AGREEMENT with the following recommendation to crowdworker overcome this barrier?

Recommendation #13: The crowdworker should use auto-translation mechanisms (machine translation) in order to help him to perform and understanding the tasks. * *Mark only one oval.*

	1	2	3	4	5	
Strongly Disagree	<input type="checkbox"/>	Strongly Agree				

Please, comment your answer. *

Participant Information

How long have you been involved in software crowdsourcing projects? Industry or Academic experience. *Mark only one oval.*

- 1 year or less
- Between 2 and 5 years
- Between 6 and 10 years
- More than 11 years
- I do not have experience in software crowdsourcing projects
- Other:

Any overall feedback (optional) ?

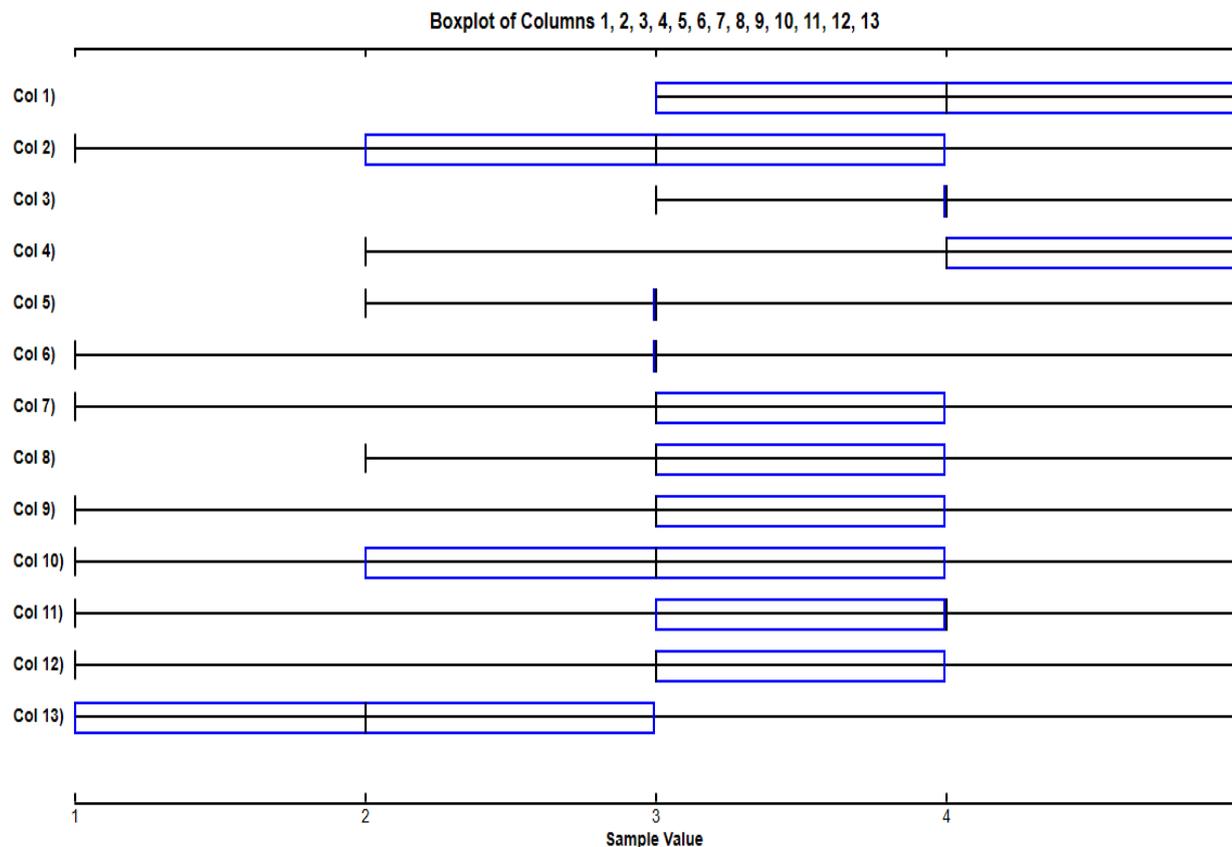
Could you please let me know of your availability for a interview via textual chat or hangout. (optional) *Mark only one oval.*

- Yes
- No

Email address (optional):

APÊNDICE J – Gráficos de caixa e de divergência

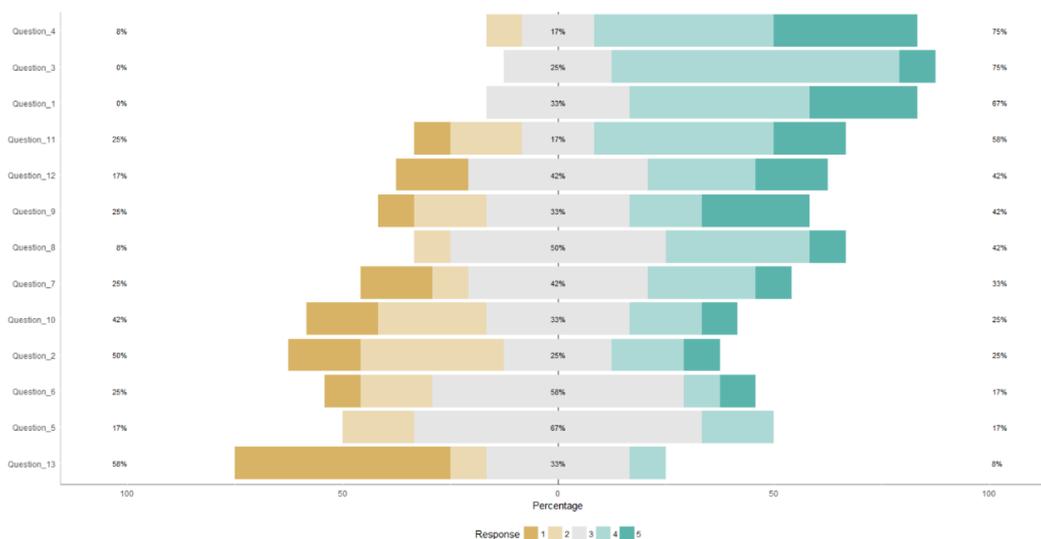
Gráfico de caixa dos graus de concordância dos especialistas com as recomendações



Legenda: Col1 = Recomendação 1 / Col2 = Recomendação 2 / Col3 = Recomendação 3 / Col4 = Recomendação 4 / Col5 = Recomendação 5 / Col6 = Recomendação 6 / Col7 = Recomendação 7 / Col8 = Recomendação 8 / Col9 = Recomendação 9 / Col10 = Recomendação 10 / Col11 = Recomendação 11 / Col12 = Recomendação 12 / Col13= Recomendação 13

1- Discordo Fortemente 2- Discordo 3- Nem concordo nem discordo 4-Concordo 5-Concordo fortemente

Gráfico de Divergência dos graus dos especialistas com as recomendações





Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br