Análise do Método HTA (Análise Hierárquica de Tarefas) para Decomposição de Tarefas em Crowdsourcing

Matheus Bianchi Godinho

Curso de Ciência da Computação – Universidade de Passo Fundo (UPF) – Campus I
Av. Brasil Leste, 285 – São José, Passo Fundo – RS, 99052-90

126886@upf.br

Resumo. *Trabalhadores de plataformas crowdsourcing encontram barreiras que dificultam a sua participação, tendo como exemplo as que estão relacionadas à tarefas. Tarefas consideradas complexas podem limitar o número de trabalhadores em potencial. A realização da decomposição de tarefas é necessária para que as mesmas se tornem de fácil resolução. Esse trabalho foi dividido em duas etapas, onde em um primeiro momento foram analisadas tarefas de testes funcionais de cinco plataformas crowdtesting, com objetivo de encontrar padrões entre elas. Com os padrões levantados, na segunda etapa, foi analisada a técnica HTA de decomposição, com o propósito de obter melhor entendimento sobre sua utilização em decomposição de tarefas de software crowdsourcing. Foram decompostas as tarefas das cinco plataformas estudadas, originando em um padrão de decomposição final, servindo como base para sua futura aplicação e avaliação.*

1. Introdução

Com o aumento da qualidade da mão de obra, redução de vida dos produtos e de um mercado de trabalho volátil, a informação acabou se tornando um bem valioso, ocasionando na criação e evolução da inteligência coletiva, como o c*rowdsourcing* [1].

Para Pierre Lévy [2], inteligência coletiva é "[...] uma inteligência distribuída por toda parte, incessantemente valorizada, coordenada em tempo real, que resulta em uma mobilização efetiva das competências." O saber está na humanidade e todos podem oferecer conhecimento, não há ninguém que seja nulo nesse contexto. Diante disso, o autor afirma que a inteligência coletiva deve ser valorizada, procurando encontrar o contexto em que o saber do indivíduo pode ser considerado valioso e importante para o desenvolvimento de um determinado grupo, além de defender que todos os indivíduos tem a sua própria inteligência acumulada em suas vivências pessoais.

Sendo assim, inteligência coletiva, é a capacidade de uma comunidade humana evoluir no sentido de se obter maior complexidade do pensamento, resolução de problemas, integração por meio da colaboração e da inovação, encontrando um ambiente fácil para prosperar, quando existe sinergia a partir da colaboração entre indivíduos. Isto traz forças complementares para uma comunidade, já que o limite cognitivo de um indivíduo ou de um pequeno grupo pode ser superado pela interação com outros e a criação da inteligência coletiva [3].

A função da inteligência coletiva no ambiente de trabalho é prover orientação à organização, frente a um crescente cenário de complexidade. Ela serve para sustentar a comunidade proporcionando a contínua argumentação e evidenciação de competências para uma resposta rápida às possibilidades e desafios vitais, à medida que estes aparecem [4].

A intensa competição global entre multidões e as rápidas mudanças tecnológicas impulsionam uma nova forma de colaborar e executar tarefas. Torna-se necessário compreender a ligação entre a colaboração e o conceito de multidão. Para apresentar essa nova forma de trabalho, conceitua-se um novo termo, o *crowdsourcing*, que pode ser aplicado em diversas situações e aperfeiçoado na proporção em que se pratica [1].

Howe [5], utiliza o termo *crowdsourcing* como o ato de distribuir a um grupo de pessoas, geralmente grande, um trabalho que antes teria que ser realizado por uma única pessoa. Dessa forma, trata-se de uma variação do conceito de inteligência coletiva, expandindo-o para um "esforço em equipe", onde todos trabalham em comum acordo, de forma estratégica, levando em consideração os conhecimentos e capacidades individuais [3].

Um exemplo de aplicação do *crowdsourcing* é o *software crowdsourcing*, voltado para o desenvolvimento de *software.* Conforme Ke Mao et al. [6], *software crowdsourcing*, é um método que busca fragmentar o processo de desenvolvimento de *software* a um grupo de trabalhadores *online*, onde cada indivíduo, de forma autônoma, escolhe as tarefas que consideram possíveis de serem realizadas. Ou seja, é um convite aberto a um grupo de indivíduos, com formação técnica, ou não, à participação de qualquer tarefa de desenvolvimento de *software*, incluindo documentação, projeto, codificação e teste.

Crowdsourcing está mudando a forma de desenvolvimento de *software*, através de modelos de organização de trabalho [7]. O modelo de processos de *software* é uma representação simplificada do processo de desenvolvimento de *software*.

Para Zanatta et al. [9], novos participantes em plataformas *crowdsourcing* encontram algumas barreiras que os impede de concluir devidas tarefas, como por exemplo, falta de documentação, gerenciamento inadequado de tarefas, problemas para entender estrutura de códigos, fraca usabilidade da plataforma e problemas com o idioma. Assim como, dificuldade em gerenciar seu tempo pessoal, atividades de lazer, estudo e trabalho também são barreiras que acabam impactando na experiência do participante, impactando na motivação do mesmo.

Tarefas consideradas complexas podem limitar o número de trabalhadores em potencial. Em seu estudo, Stol [10] apontou que a decomposição de tarefas está entre as principais preocupações dos modelos *crowdsourcing*. Quando decompostas em tarefas menores, sistemas *crowdsourcing* permitem que trabalhadores com diferentes habilidades concluam as tarefas com maior agilidade através da paralelização e larga escala de atividades. A realização da decomposição de uma tarefa é necessária para que as mesmas sejam paralelizadas, mesmo que em muitos casos essa atividade não seja trivial [6]. Para Morris [11], decomposição de tarefas aumenta a probabilidade de bons resultados.

O objetivo do presente trabalho é analisar como o método HTA pode ser utilizado para decompor tarefas em *software crowdsourcing*. Para isso, foram analisadas tarefas com foco em teste funcional de cinco plataformas *crowdsourcing*, visando encontrar padrões entre elas. Para, a partir destes padrões, ser aplicada a técnica de decomposição de tarefas, resultando em um padrão final de decomposição.

2. Fundamentação Teórica

Neste capítulo serão apresentados os principais conceitos que foram utilizados como base para o desenvolvimento deste trabalho.

2.1 Software Crowdsourcing

Criado por Jeff Howe [5], em 2006, o termo *crowdsourcing* descreve um conceito de interação social, baseado na construção coletiva de soluções com benefício a todos. Howe oferece a seguinte definição:

> O *crowdsourcing* representa o ato de uma empresa ou instituição terceirizar uma função realizada por seus funcionários para uma rede indefinida (e geralmente grande) de pessoas, na forma de uma chamada aberta. Isto pode assumir a forma de *peer-production* (quando o trabalho é realizado de forma colaborativa), mas também é frequentemente realizado individualmente. O pré-requisito crucial é o uso do formato de chamada aberta e da grande rede de trabalhadores.

Crowdsourcing é a fonte de força de trabalho, formada por grupo de pessoas de qualquer formação, que, por meio da internet, contribuem com os seus conhecimentos para desenvolver um projeto ou resolver problemas [5].

Para Brabham et al. [12], o *crowdsourcing* também pode ser definido como um modelo de produção *online*, que utiliza a inteligência coletiva da multidão que está interconectada em rede, gerando soluções para propósitos específicos. Sendo assim, um conjunto de produção tradicional adotado por empresas, com produção realizada pelos usuários. O modelo permite vantagem para empresas, aumentando o desenvolvimento e criatividade dos produtos, assim como o engajamento das pessoas envolvidas no projeto.

Dessa forma, uma empresa divulga um problema ou desafio em uma plataforma e então, os membros dessa comunidade *online* fornecem soluções para o problema, o qual a empresa processa e consolida em um produto, ou solução final [12].

Conforme ilustrado na Figura 1, Hosseini [13], apresenta os 4 pilares do processo do *crowdsourcing*, descritos assim: *crowd*, pessoas que desenvolvem as soluções dos desafios propostos; *requester*, é o responsável por fornecer o desafio à plataforma, buscando conhecimento da *crowd*; *tasks*, que é a atividade solicitada pelo *requester* no qual a *crowd* participa e *plataform,* que é a plataforma responsável pela comunicação entre o *requester* com a *crowd* onde a *task* é executada.

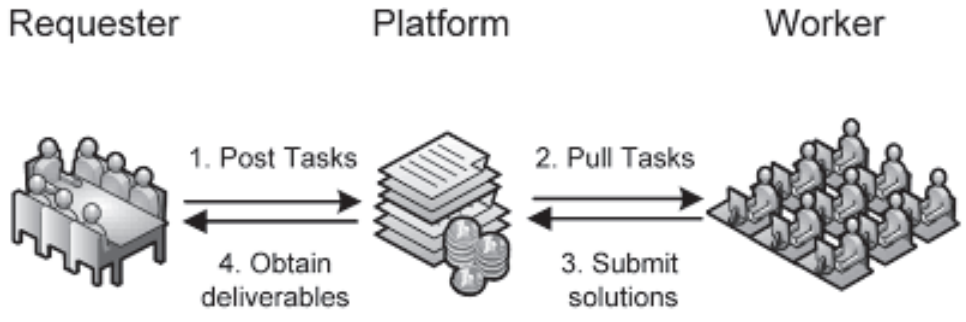

Figura 1. Atores em crowdsourcing.
Fonte: Ke Mao, 2015

Um exemplo de aplicação do *crowdsourcing* é o *crowdtesting.* Segundo Roodenrijs e Prins [14], *crowdtesting* é definido da seguinte forma: "É o uso de pessoas desconhecidas, a multidão, para testar *software*". Narayanan [15] define *crowdtesting* como um meio de assegurar a qualidade do *software* através de testes realizados por uma multidão.

Já em uma visão mais simplista, para Padmanaban [7], o termo *crowdtesting* caracteriza o teste de *software* quando este utiliza uma comunidade para testar um determinado produto. Esta comunidade abrange pessoas voluntárias de diversos locais, culturas, línguas e estilos de vida que testam o *software* fornecido utilizando cenários realistas, como se fossem os próprios usuários do sistema. Empresas que administram este tipo de teste normalmente pagam os testadores por defeitos encontrados.

Esse modelo obteve êxito em plataformas de teste de *software*, onde centenas e milhares de testadores participam. Esses serviços oferecem aos clientes o benefício de fluidez e velocidade no trabalho. Os clientes podem rapidamente localizar e contratar uma força de trabalho, onde por sua vez a plataforma fica responsável pela triagem e pagamento dos trabalhadores [16].

Crowdtesting fornece uma alternativa ao cenário de testes tradicional, visando reduzir os recursos necessários para conduzir estes testes utilizando *crowdsourcing* [17].

Para Ke Mao et al. [6], *software crowdsourcing,* por sua vez, é o ato de assumir, externamente, qualquer tarefa em um processo de desenvolvimento de *software* à um grande e indefinido grupo de trabalhadores online em formato de chamada aberta. *Software crowdsourcing* é mediado por meio de plataformas que conectam os *requesters* com os trabalhadores – a multidão. Essas plataformas surgiram como uma importante parte interessada no processo de desenvolvimento de *software* [18].

Diante deste modelo inovador de distribuição de solução de problemas, tornou-se necessário a criação de um elo entre o *requester* e a multidão. Ou seja, a criação de plataformas para auxílio no gerenciamento, não apenas de pequenas tarefas (*tasks*), mas projetos inteiros, partindo desde sua documentação, até a entrega final. Desta forma, possibilitando o *requester* encontrar talentos fora de seus limites físicos, além de vantagens como: redução nos custos, visibilidade de sua marca/empresa, soluções diversificadas, melhor qualidade, acesso à criatividade e da resolução de seus problemas [19].

A seguir far-se-á uma breve descrição das 5 plataformas analisadas. TopCoder[1], fundada em 2001 por Jack Hughes, é a plataforma pioneira na utilização do uso do poder da multidão para solução de problemas. A comunidade, atualmente com mais de 1 milhão de membros, é a principal força de trabalho por trás de todos os projetos da plataforma. TopCoder é uma plataforma que atua diretamente em todas as fases de desenvolvimento de *software*. Diferente de algumas plataformas que buscam o aprimoramento apenas em uma determinada fase desse processo, como por exemplo, a de validação do escopo de uma demanda, onde os usuários participam para encontrar defeitos em sistemas. Esses usuários não precisam, necessariamente, ter algum conhecimento em programação ou análise de requisitos, já que a plataforma é apenas para o levantamento de problemas que os usuários finais do produto daquele *requester* poderiam encontrar no sistema. [20].

A comunidade da uTest[2] é composta por um amplo grupo de testadores abrangendo diversos locais, idiomas, sistemas operacionais, dispositivos e navegadores. Os clientes da uTest especificam os requisitos de testes, que por sua vez a plataforma identifica e convida os testadores da comunidade para participarem da tarefa. Os testadores que aceitarem participar irão testar funcionalidades ou usabilidade (dependendo do caso de teste fornecido) de *websites* e aplicativos, e então os *requesters,*

[1] https://www.topcoder.com/

[2] https://www.utest.com/

por sua vez, aprovam ou rejeitam o relatório realizado pelos testadores baseados na qualidade dos mesmos [21].

Testbirds[3], fundada em 2011, atualmente com mais de 65 mil usuários registrados, oferece testes funcionais e de usabilidade para praticamente todos os tipos de software, porém possuindo um foco em aplicativos *web* e *mobile* [22].

Em 2010, foi lançada a plataforma brasileira, Crowdtest[4]. A Crowdtest recebe a demanda dos clientes, organiza o projeto e a equipe que irá participar, recebe falhas, faz a validação e disponibiliza o resultado para os clientes [23].

A plataforma test IO[5], fundada em 2011, possui um grande número de participantes, atuando diretamente em todas as partes do ciclo de um *software*, além de oferecer diversos tipos de testes de *softwares*, garantindo cobertura a todos os tipos de dispositivos, sistemas operacionais e idiomas [24].

2.2 Decomposição de Tarefas

As tarefas são definidas como processos em um nível elevado de detalhamento, contendo fluxos de trabalho suficientes para execução de uma atividade. Na TopCoder, cada tarefa é organizada com as informações necessárias para realização da mesma. São apresentados o título da tarefa, o valor do prêmio pago pela realização da tarefa, visão geral do desafio da tarefa, diretrizes para submissão da tarefa, informações sobre o pagamento e informações sobre a classificação por confiabilidade. Na visão geral dos desafios são detalhados os objetivos do desafio, as tecnologias, escopo da tarefa e os requisitos das tarefas [25].

As dimensões da complexidade de uma tarefa podem ser definidas como a estrutura, interdependência e o compromisso da tarefa. Tarefas estruturadas possuem soluções ou contribuições definidas, como, por exemplo, a criação de um site de uma empresa. Tarefas não estruturadas se caracterizam por estruturas que exigem algo novo, sem que uma abordagem específica seja definida. Neste tipo de tarefa geralmente são envolvidas soluções inovadoras, como por exemplo, o desenvolvimento de um algoritmo para um problema específico [26].

Para Rouse [27], a natureza das tarefas é definida como simples, moderadas e sofisticadas. As tarefas simples são aquelas consideradas de baixa complexidade e podem ser realizadas por alguém que possua educação e treinamento moderados, sendo avaliadas facilmente. Podem ser enquadradas como simples alguns aprimoramentos e ideias para novos produtos.

As tarefas moderadas são aquelas que possuem sua resolução minimamente complexa, porém em um nível superior comparadas com as simples. Tarefas moderadas podem envolver algumas de design como, por exemplo, a elaboração de logomarcas [27].

Já as tarefas sofisticadas, são aquelas que possuem a complexidade elevada, altamente qualificadas e de avaliação complexa, exigindo a compreensão substancial e uma visão do negócio altamente desenvolvida. Devem ser desenvolvidas por alguém que possua um conhecimento técnico considerável, como por exemplo, a elaboração de um plano de negócio e diversas tarefas de design [27].

---

[3] https://www.testbirds.com/

[4] https://app.crowdtest.me/

[5] https://test.io/

Tarefas consideradas complexas acabam limitando o número de trabalhadores em potencial. A decomposição se torna essencial para que essas tarefas sejam paralelizadas, mesmo que em muitos casos essa atividade não seja trivial [6].

Um estudo realizado por Morris [11], relata que decomposição de tarefas aumenta a probabilidade de bons resultados. Porém, estudo realizado na TopCoder indicou que decomposição de tarefas é um dos principais desafios da plataforma.

Latoza [16], pontua que um dos principais desafios é encontrar uma decomposição que seja adequada para que o *software* se torne efetivo. Tajedin [28], afirma que os projetos que são decompostos em módulos menores tem mais chances de obter sucesso.

Ao serem decompostas em tarefas menores, as plataformas *crowdsourcing* permitem que os trabalhadores com diferentes habilidades possam concluir as tarefas com maior agilidade através da paralelização e larga escala das atividades. Ao ser aplicada a abordagem ao desenvolvimento de *software*, a participação tende a aumentar e com isso a redução do tempo de execução das tarefas, assim como de suas barreiras [16].

A Análise Hierárquica de Tarefas – HTA (*Hierarchical Task Analysis)*, tem como objetivo entender competências e habilidades associadas às tarefas complexas. O ponto de partida desta análise são os objetivos do usuário. A partir de um objetivo, as principais tarefas associadas ao alcance deste objetivo são identificadas [31] [32].

Tarefas complexas podem ser divididas em sub tarefas e assim sucessivamente, de acordo com o nível de detalhamento de cada tarefa sendo analisada, dessa forma, realizando a sua decomposição [32]. A estrutura foca em um determinado problema, os critérios/subcritérios, os quais podem ser determinados em tantos níveis quanto necessários e as alternativas viáveis para a resolução do problema [33].

A análise é composta em etapas de atuação com o objetivo de facilitar a execução pelo usuário indicando caminhos para a sua aplicação. Inicia-se com uma especificação do motivo da análise, seguindo pela determinação das metas da tarefa. Feito isso, identificam-se as fontes de informação da tarefa, sendo realizada a obtenção de dados e a decomposição, gerando diagramas [33].

No nível mais alto do diagrama, considera-se que uma tarefa consiste em uma operação, sendo definidas em termos de seus objetivos. Pode ser "quebrada" em sub operações definidas por seu sub objetivos. A relação entre as operações e as sub operações é hierárquica, embora não seja necessário que os objetivos devem ser alcançados seguindo uma sequência [34].

3. Trabalhos Relacionados

Neste capítulo serão apresentados os trabalhos realizados por outros pesquisadores que serviram como base para a pesquisa realizada.

3.1 Microtask Programming: Building Software with a Crowd

No estudo realizado por Latoza [29], foi implementado uma abordagem que realiza a decomposição de tarefas. No trabalho é realizando um rastreamento das alterações de um gráfico de artefatos, gerando as micro tarefas de maneira apropriada. A abordagem é implementada na IDE Crowdcode[6], onde apresentou um paradigma (*map-reduce*), que divide o problema em vários problemas menores (sub problemas). Logo após, é realizada

[6] https://www.crowdcode.io/

a resolução desses sub problemas (*map*) e então os une (*reduce*). Foi realizado um experimento para avaliação do método. No resultado do trabalho, verificou-se que os participantes estariam mais propensos a contribuir com esses projetos utilizando o Crowdcode. A relação com o presente estudo, é com o objetivo da redução das barreiras na contribuição dos trabalhadores e na proposta de decomposição de tarefas.

3.2 The Future of Crowd Work

No trabalho de Kittur [30], é apresentada a estrutura de propósito geral para realização de tarefas complexas com a utilização da abordagem de micro tarefas. O estudo foi realizado na plataforma CrowdForge[7] com estrutura de MapReduce na realização de processamento distribuído. Foi gerado um protótipo e estudos de caso para comprovar a abordagem que foi proposta. O estudo conclui que os estudos de caso mostraram que as tarefas produzidas pela plataforma CrowdForge eram melhores classificados que produzidos individualmente. A relação com o trabalho é apresentada na proposta de decomposição de tarefas.

3.3 Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development

O estudo realizado por Stol [10], apontou diversos desafios emergentes em modelos *crowdsourcing*. O objetivo do estudo foi obter melhor entendimento do processo e desafios relacionados ao modelo, a partir da realização de um estudo de caso. Como resultado foi apontado que decomposição de tarefas é um dos diversos desafios, servindo como relação para a execução deste trabalho.

4. Metodologia

Este trabalho se enquadra, quanto à metodologia, como sendo uma pesquisa descritiva[8], um tipo de pesquisa cientifica, onde seu objetivo é descobrir com que frequência um fato ocorre baseada na técnica de observação. Quanto a forma de abordagem, esta pesquisa é de abordagem qualitativa, visando adquirir um melhor entendimento sobre tarefas das plataformas abordadas neste estudo.

Inicialmente, na etapa 1, foi realizada uma revisão bibliográfica de trabalhos que serviram como base para a pesquisa desenvolvida, obtendo conhecimento, conceitos e funcionamento do *crowdsourcing*, apresentados no Capítulo 2.

Foi realizado o estudo de 5 plataformas de *crowdtesting*, citadas na sessão 2.1. Uma vez que o critério de escolha das plataformas TopCoder, uTest e Tesbirds foi baseado em estudos de Ke Mao [6], e a escolha das plataformas Crowdtest e test IO foi a partir de pesquisa do autor do trabalho. A escolha das plataformas compreendeu o cadastro nas 5 plataformas. Após a realização do cadastro, iniciou-se um estudo sobre tarefas de cada plataforma, compreendendo na seleção de tarefas que possuíam abordagem em testes funcionais[9] de *software*.

---

[7] https://crowdforge.io/

[8] MÉTODOS DE PESQUISA por TATIANA GERHARDT e DENISE SILVEIRA

[9] INTRODUÇÃO AO TESTE DE SOFTWARE por MARCIO EDUARDO DELAMARO

Com as tarefas já selecionadas, foi realizada uma análise baseada em análise de conteúdo[10] entre as tarefas de cada plataforma individualmente. Na primeira parte da análise das tarefas, foi avaliada a forma como os dados eram dispostos e estruturados, analisando as características contidas nas tarefas, como por exemplo, objetivos gerais, deadlines, detalhes do projeto, e assim por diante. Após o levantamento desses pontos, foi criado um primeiro esboço de um diagrama estrutural de cada plataforma e de suas características para melhorar a compreensão e organização dos dados levantados.

Em um segundo momento da etapa 1, realizou-se a análise das descrições das tarefas, com objetivo de encontrar, no texto, características que também pudessem ser consideradas comuns entre as tarefas estudadas. Com a análise das descrições das tarefas concluída, foi elaborado um diagrama final para cada plataforma, visando apresentar as características semelhantes em todas as tarefas, dessa forma, identificando e avaliando os padrões entre elas.

Na etapa 2 da pesquisa, foi realizado um estudo sobre o método de Análise Hierárquica de Tarefas (HTA), compreendendo em um entendimento sobre a sua aplicação. Após a análise do método, o mesmo foi aplicado nas tarefas de cada plataforma individualmente, produzindo diagramas individuais para sua representação.

No final da segunda etapa do trabalho, com a HTA aplicada em cada tarefa de cada plataforma, foi elaborado um padrão geral de decomposição de tarefas para todas as plataformas estudadas. Para ser considerado um padrão, o critério de avaliação foi que o passo decomposto deveria estar contido em no mínimo 3 das 5 plataformas estudadas. A Figura 2 ilustra as duas etapas que compõe os passos metodológicos para execução deste trabalho.

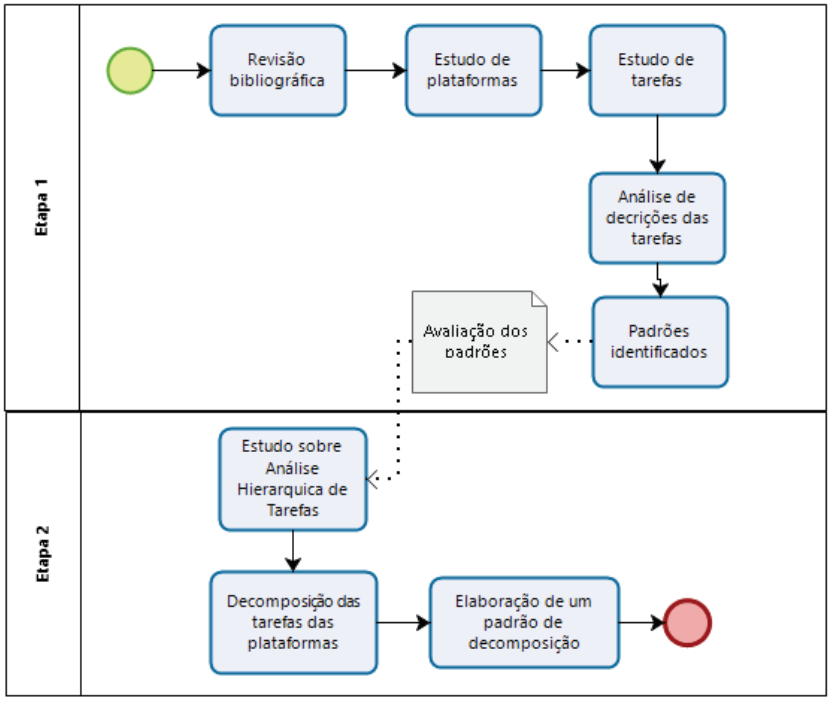

Figura 2. Etapas de execução do trabalho.
Fonte: Elaborada pelo Autor

[10] ANÁLISE DE CONTEÚDO: A VISÃO DE LAURENCE BARDIN por FERNANDA MARSARO publicado na REVISTA ELETRÔNICA DE EDUCAÇÃO

5. Resultados

Neste capítulo serão apresentados os resultados das duas etapas desenvolvidas durante o trabalho.

5.1 Resultados da Etapa 1

A partir da análise das tarefas de teste funcional de *software* nas plataformas analisadas, foi possível analisar as diversas informações que são disponibilizadas pelas plataformas para os usuários, desde valores da premiação para cada tipo de defeito encontrado nos *softwares* a etapas de como participar do projeto em questão. É a partir das descrições dessas tarefas que estão englobados os tópicos que foram considerados padrões entre elas.

5.1.1 Padrões de tarefas na plataforma uTest

Na plataforma uTest as características analisadas, contidas em toda tarefa são: objetivo geral da tarefa que será executada; etapas para participação do projeto; o que deverá ser testado; além de requisitos básicos para participação da mesma, como, *softwares*, dispositivos, navegadores e sistemas operacionais que o usuário deve possuir para execução da mesma, além de *deadlines* e tópicos de ajuda possuindo instruções especiais, dicas, FAQs e e-mail para comunicação direta com a equipe responsável pelo gerenciamento do projeto.

A Figura 3 ilustra os pontos definidos como padrões entre as tarefas da plataforma.

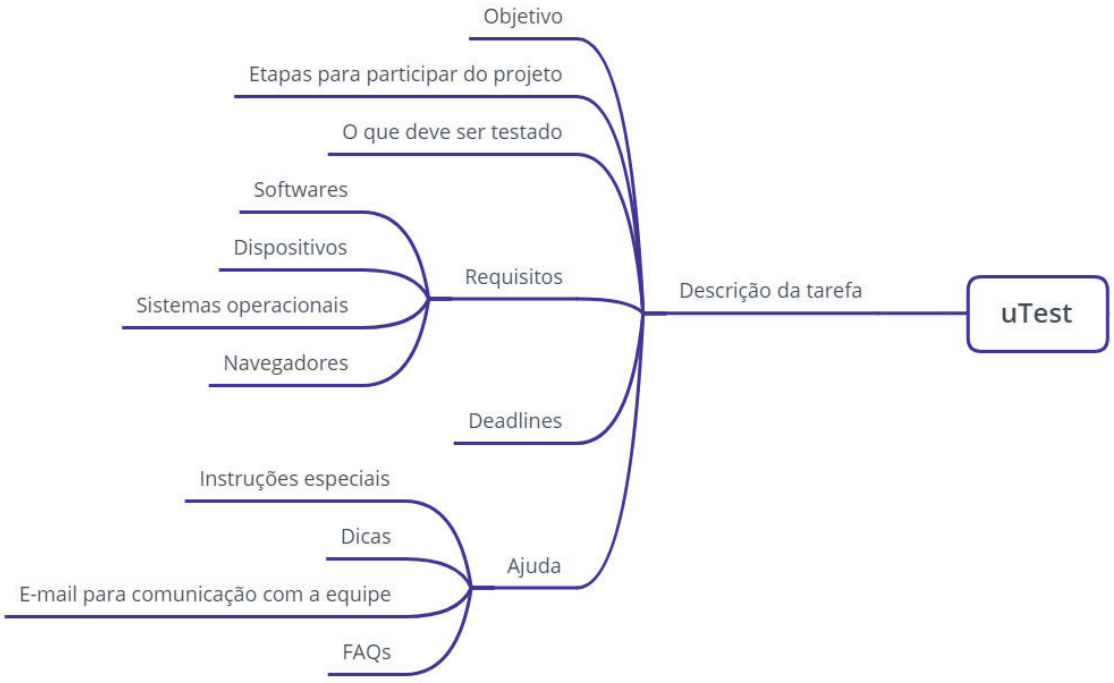

Figura 3. Padrões de tarefas na plataforma uTest.
Fonte: Elaborada pelo Autor

5.1.2 Padrões de tarefas na plataforma Crowdtest

Na plataforma Crowdtest, que pode ser observado na Figura 4, os padrões compreendidos a partir da descrição de cada tarefa analisada e comparada são: objetivos gerais da tarefa, contendo os ambientes em que os testes serão realizados sendo eles de homologação ou de produção, o que deverá ser testado pelo testador; *deadlines* da tarefa; instruções com os passos a serem seguidos para realização da tarefa e suas funcionalidades testadas; os

requisitos para participação da tarefa, dispositivos, sistemas operacionais e navegadores exigidos para realização da tarefa, assim como o seu tipo se é *web* ou *mobile*; tópicos de ajuda contendo observações importantes, dicas, orientações para reportar os defeitos encontrados, FAQs e guias para realizações das ocorrências.

Figura 4. Padrões de tarefas na plataforma Crowdtest.
Fonte: Elaborada pelo Autor

5.1.3 Padrões de tarefas na plataforma Testbirds

Na Testsbirds, contidos nas descrições das tarefas, os padrões identificados foram: objetivo geral da tarefa; *deadlines*; tópico de ajuda contendo dicas, orientações para execução da tarefa, FAQs, possibilidade de enviar mensagens diretas para o administrador da tarefa (conhecido como Birdmaster), além de instruções para realização da tarefa e requisitos como, *software,* idioma em que a tarefa será realizada, navegadores e sistemas operacionais. A Figura 5 demonstra esses padrões.

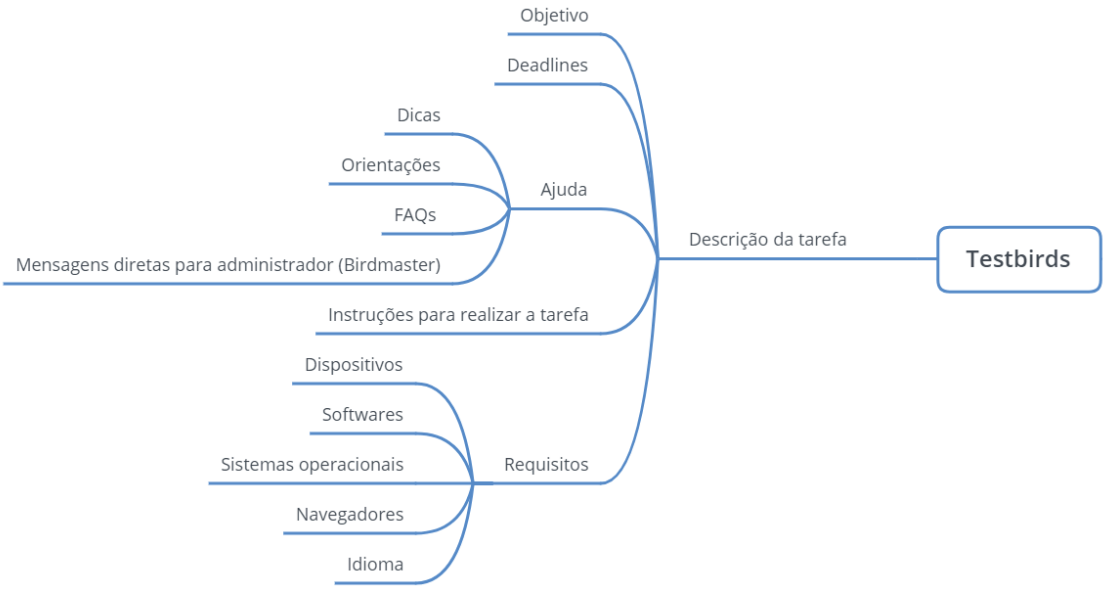

Figura 5. Padrões de tarefas na plataforma Testbirds.
Fonte: Elaborada pelo Autor

5.1.4 Padrões de tarefas na plataforma test IO

A plataforma test IO, por sua vez, apresenta os seguintes tópicos em comum para toda tarefa de teste: objetivo geral da tarefa em questão; dispositivos, *softwares*, sistemas operacionais e navegadores como requisitos para execução da tarefa; *deadlines*; restrições, contendo detalhes do que não deve ser testado durante execução da tarefa; e tópicos de ajuda para o usuário contendo instruções para execução da tarefa, dicas e FAQs. Os padrões da test IO estão representados na Figura 6.

Figura 6. Padrões de tarefas na plataforma test IO.
Fonte: Elaborada pelo Autor

5.1.5 Padrões de tarefas na plataforma TopCoder

O estudo da última plataforma, TopCoder, acarretou na análise e identificação dos padrões das tarefas representados na Figura 7. A partir da descrição das tarefas da TopCoder, foram identificados como padrões entre as tarefas os seguintes tópicos: objetivo geral da tarefa; como participar da tarefa; o que deverá ser entregue pelo usuário; *deadlines*; regras a serem seguidas pelo usuário; requisitos para executar a tarefa como *softwares*, sistemas operacionais, navegadores e dispositivos.

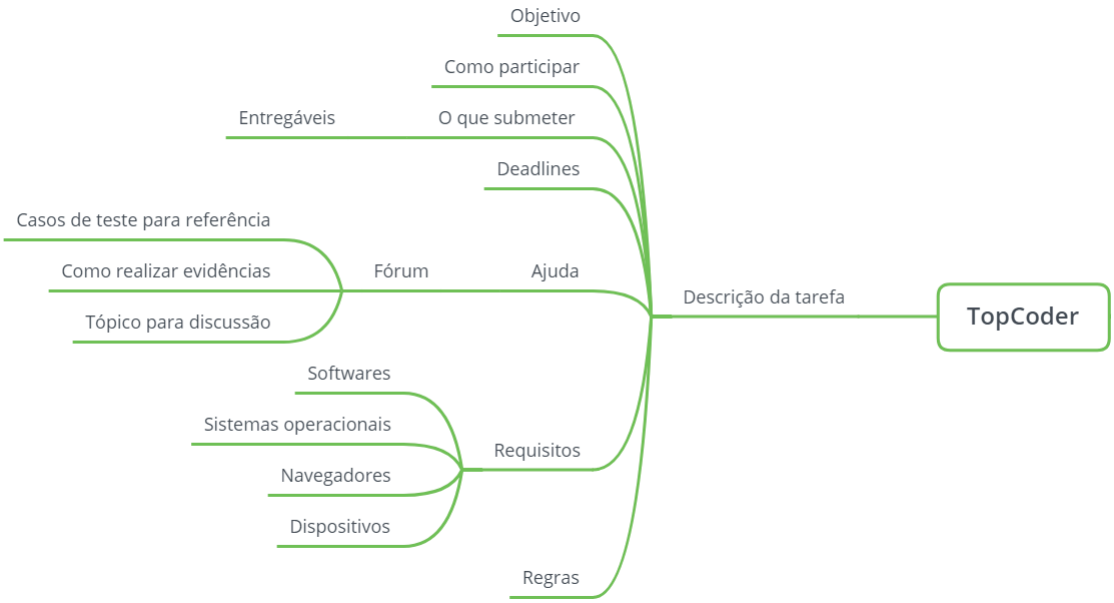

Figura 7. Padrões de tarefas na plataforma TopCoder.
Fonte: Elaborada pelo Autor

5.2 Resultados da Etapa 2

O levantamento dos padrões das tarefas apresentados na sessão 5.1, possibilitou identificar, em um contexto geral, a maneira como as tarefas estão dispostas em cada plataforma. Essa sessão tem como objetivo apresentar os resultados a partir da análise do método HTA e sua aplicação para cada uma das plataformas estudadas, originando um padrão final de decomposição de tarefas.

Tendo em vista como objetivo principal a resolução da tarefa, foram decompostas as tarefas em possíveis passos a serem seguidos pelo usuário. O padrão final de decomposição de tarefas pode ser observado na Figura 8.

Como citado anteriormente, para ser considerado um padrão de decomposição, foi julgado como critério que o passo decomposto estivesse contido em no mínimo 3 das 5 plataformas estudadas. Para aquelas plataformas que não dispõe o passo, foi incluso ao lado da descrição do passo a ser seguido o caractere hífen (-), juntamente com a abreviação do nome da plataforma, uma vez que, CT equivale à Crowdtest; UT à uTest e TIO à test IO.

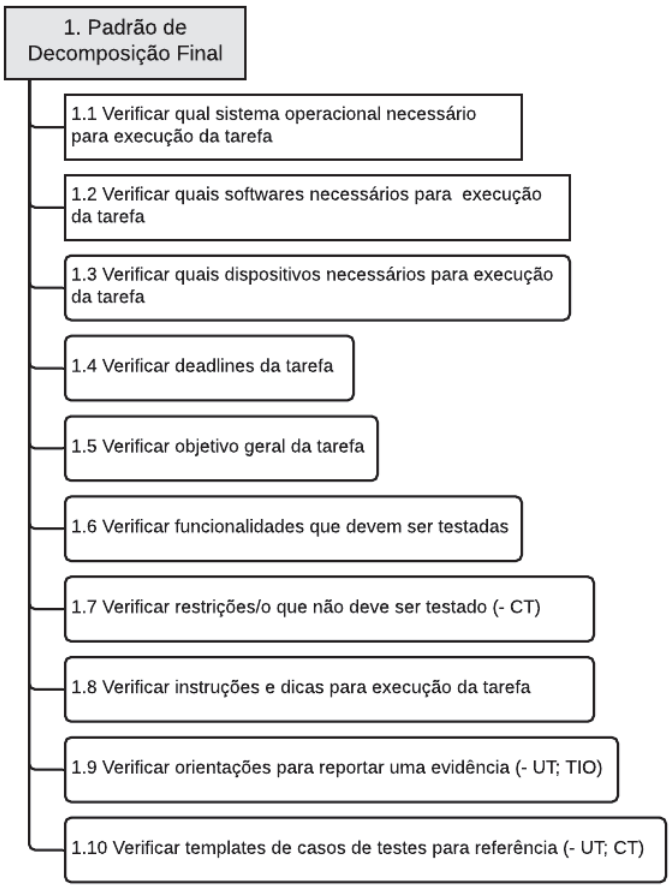

Figura 8. Padrão final de decomposição de tarefas.
Fonte: Elaborada pelo Autor

6. Discussão e Conclusão

Neste capítulo será apresentado a discussão dos resultados e a conclusão sobre o trabalho realizado.

6.1 Discussão dos resultados

Neste trabalho, a partir da análise das descrições das tarefas estudadas de cada plataforma, pode-se perceber que a maioria das plataformas apresentam características e informações em comum, que são disponibilizadas aos usuários, dessa forma, julgados como padrões nas respectivas plataformas.

A identificação de padrões nas plataformas auxiliou na escolha da técnica mais adequada para realizar decomposição de tarefas. Analisando os resultados identificados, pode-se verificar que os padrões encontrados vão ao encontro as etapas da HTA, uma técnica de decomposição de tarefas hierárquica. A HTA tem por objetivo analisar as metas na realização das tarefas pelo processo de descrição das mesmas, utilizando submetas e planos. Essa técnica analisa e representa aspectos comportamentais de tarefas complexas [34].

Conforme Padilha [3], a limitação de um indivíduo pode ser superada pela interação com a multidão. Abordando uma técnica de decomposição de tarefas, essa

limitação pode reduzir, sendo que a complexidade da tarefa reduziria. Nesse cenário, a decomposição de tarefas aumenta a probabilidade de bons resultados [11].

Para Ke Mao [6], o *software crowdsourcing* é um procedimento que busca fragmentar os processos de desenvolvimento de *software*. Autor ressalta que o *software crowdsourcing* busca fragmentar processos de desenvolvimento de *software*. A fragmentação pode ser realizada nessas tarefas utilizando técnica proposta e com isso reduzir a complexidade da mesma.

Nas tarefas de desenvolvimento de *software*, o participante realizará a mesma seguindo metas especificadas pelo solicitante. A identificação do padrão nas tarefas auxilia na identificação das metas e submetas, sendo que a decomposição hierárquica possui uma estrutura orientada a metas [6].

Seguindo a análise da técnica HTA, Stanton [34], por sua vez, aponta que a técnica pode ser usada para descrever como as tarefas podem ser realizadas pelo usuário. O propósito da HTA é dividir as operações a serem realizadas hierarquicamente, relacionando-se com o padrão de decomposição gerado, apontando os passos que o usuário deve seguir para realização da tarefa em questão na plataforma.

Para Ellwager et al. [33], o objetivo da utilização da técnica de decomposição HTA, é tornar prático e fácil a execução das tarefas pelos usuários, indicando os caminhos para a execução das mesmas. Caminhos esses que podem ser observados no padrão final de decomposição de tarefas nas plataformas de *crowdtesting*, auxiliando o usuário desde o momento da validação dos dispositivos necessários para realização da tarefa até a verificação de dicas e instruções para a execução do teste em questão.

O objetivo geral da análise é a resolução da tarefa por parte do usuário. Para essa resolução, diversos foram os pontos decompostos em sub operações para realização da mesma, baseados nos padrões em comum levantados de cada plataforma.

6.2 Conclusão

O *crowdsourcing* é um modelo que surge com uma abordagem distinta para o desenvolvimento de *software*. Ao contrário de estratégias tradicionais de terceirização, o *crowdsourcing* introduz um modelo voltado para pessoas aptas a cooperarem, ou seja, a multidão. Ao invés de um único fornecedor, pode haver qualquer nível de colaboradores envolvidos desde o início do ciclo de desenvolvimento do produto.

Muitas vezes, as tarefas disponibilizadas à multidão podem ser complexas, limitando o número de trabalhadores em potencial. A decomposição de uma tarefa surge como uma necessidade para que as mesmas tornem-se eficientes.

Este trabalho teve como principal objetivo realizar uma análise de um método de decomposição de tarefas, a HTA, em 5 plataformas de *crowdtesting*. Em um primeiro momento foram analisadas as características de tarefas com abordagem em testes funcionais das plataformas, com intuito de encontrar características que pudessem ser consideradas similares entre elas, levantando padrões. Com os padrões apontados, aplicou-se a HTA nas tarefas de cada plataforma individualmente, decompondo-as. Em um segundo momento, foram analisadas as decomposições de cada plataforma, originando em um padrão de decomposição final.

Dessa forma, este artigo disponibiliza a aplicação de uma técnica de decomposição de tarefas, que auxilie usuários na resolução das mesmas, servindo como

base para trabalhos futuros, sugerindo que sejam realizados experimentos aplicando a técnica apresentada, realizando a avaliação dos resultados.

Referências

[1] C. L. Branquinho, *Crowdsourcing: uma forma de inovação aberta*, no. 1. Rio de Janeiro: CETEM/MCTI, 2016.

[2] P. Lévy, *A inteligência coletiva: por uma antropologia do ciberespaço*, 5ª. São Paulo: Edições Loyola, 2007.

[3] M. A. O. Padilha e A. R. Graeml, "Inteligência Coletiva e Gestão do Conhecimento : Quem é Meio e Quem é Fim?," 2015.

[4] G. Pór, "Questing for Collective Intelligence," in *Community Building in Organizations: Renewing Spirit and Learning in Business*, New Leaders Press, 1995, pp. 1–16.

[5] J. Howe, "The Rise of Crowdsourcing," no. 14, pp. 1–5, 2006.

[6] K. Mao, L. Capra, M. Harman, e Y. Jia, "A Survey of the Use of Crowdsourcing in Software Engineering," *J. Syst. Softw.*, vol. 126, pp. 1–35, 2016.

[7] R. Padmanaban, "Crowd Sourced Testing - is it really for you?," 2012. [Online]. Disponível em: https://qainfotech.com/crowd-sourced-testing-is-it-really-for-you/. [Acesso em: 07-Mai-2019].

[8] I. Sommerville, *Engenharia de Software*, 9th ed. São Paulo: Pearson Prentice Hall, 2003.

[9] A. L. Zanatta, "Barriers Faced by Newcomers to Software-Crowdsourcing Projects," no. April, pp. 37–43, 2017.

[10] K. Stol e B. Fitzgerald, "Two ' s Company , Three ' s a Crowd : A Case Study of Crowdsourcing Software Development," pp. 187–198, 2014.

[11] R. R. Morris, M. Dontcheva, e E. M. Gerber, "Priming for better performance in microtask crowdsourcing environments," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 13–19, 2012.

[12] D. C. Brabham, "Crowdsourcing as a Model for Problem Solving: An Introduction and Cases," vol. 14, no. 1, pp. 75–90, 2008.

[13] M. Hosseini, K. Phalp, J. Taylor, e R. Ali, "The Four Pillars of Crowdsourcing: a Reference Model," 2014.

[14] E. Roodenrijs e A. Prins, "Join the crowd: Worldwide testing.," *Revista Test Focus*, África do Sul, pp. 8–11, 2009.

[15] M. Narayanan, "Crowd sourced testing - An emerging business model," 2011. [Online]. Disponível em: https://www.slideshare.net/manoj7698/star-west-2011-manoj-narayanan-presentation-10. [Acesso em: 07-Mai-2019].

[16] T. D. Latoza e A. Hoek, "Crowdsourcing in Software Engineering: Models, Motivations and Challenges," *IEEE Software*, vol. 374, pp. 74–80, 2006.

[17] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, e K. Diepold, "Best Practices for QoE Crowdtesting : QoE Assessment With Crowdsourcing," *IEEE Trans. Multimed.*, vol. 16, no. 2, pp. 541–558, 2014.

[18] A. L. Zanatta, L. S. Machado, G. B. Pereira, R. Prikladnicki, e E. Carmel, “Software Crowdsourcing Platforms,” 2016.

[19] B. S. Noveck, “Wiki Government: How Technology Can Make Government Better, Democracy Stronger, and Citizens More Powerful,” *Brookings Inst. Press*, 2009.

[20] A. M. Boff, “Barreiras e Recomendações em Software Crowdsourcing,” 2018.

[21] D. Liu, R. G. Bias, M. Lease, e R. Kuipers, “Crowdsourcing for usability testing,” *Proc. ASIST Annu. Meet.*, vol. 49, no. 1, 2012.

[22] S. Zogaj, N. Leicht, I. Blohm, U. Bretschneider, e J. M. Leimeister, “Towards Successful Crowdsourcing Projects: Evaluating the Implementation of Governance Mechanisms,” *Ssrn*, no. 2015, 2018.

[23] “Como Funciona o Crowdtest.” [Online]. Disponível em: https://app.crowdtest.me/como-funciona/. [Acesso em: 30-Mai-2019].

[24] “About testIO.” [Online]. Disponível em: https://test-io.workable.com/. [Acesso em: 30-Mai-2019].

[25] F. L. Costella, “Proposta de técnica para decomposição de tarefas para testes funcionais em softwares crowdsourcing,” 2018.

[26] H. Jiang and S. Matsubara, “Efficient Task Decomposition in Crowdsourcing,” pp. 65–73, 2014.

[27] A. C. Rouse, “A Preliminary Taxonomy of Crowdsourcing,” 2010.

[28] H. Tajedin, “Determinants of Success in Crowdsourcing Software Development,” pp. 173–178, 2013.

[29] T. D. LaToza, W. Ben Towne, C. M. Adriano, e A. van der Hoek, “Microtask Programming: Building Software with a Crowd,” *Proc. 27th Annu. ACM Symp. User interface Softw. Technol. - UIST ’14*, pp. 43–54, 2014.

[30] A. Kittur *et al.*, “The Future of Crowd Work,” *Proc. 2013 Conf. Comput. Support. Coop. Work - CSCW ’13*, p. 1301, 2013.

[31] S. Barbosa e B. Silva, “Interação Humano-Computador,” pp. 1–19, 2010.

[32] J. Preece, Y. Rogers, and H. Sharp, “Design de Interação,” 2005.

[33] C. Santos e C. Ellwager, “Avaliação de usabilidade de sistemas sob a ótica da estruturação conceitual hierárquica.”

[34] N. Stanton, “Hierarchical task analysis : Developments , applications , and extensions,” vol. 37, pp. 55–79, 2006.