

PROPOSTA DE SOFTWARE COLABORATIVO PARA AUXÍLIO DE ESTIMATIVA DE ESFORÇO NA EVOLUÇÃO DE SOFTWARE

Rodrigo Machado¹

Victor dos Santos Costa²

Andrea Sabedra Bordin³

Maicon Bernardino da Silveira⁴

Resumo:

Na Engenharia de Software, ainda existe uma grande dificuldade em se realizar estimativas de esforço para o desenvolvimento de softwares com precisão, e esse problema se torna ainda maior na fase de evolução de software. Mesmo com as estratégias existentes que auxiliam a obtenção de maior precisão, ainda é possível observar muitos equívocos pela falta de utilização das mesmas para a estimativa de esforço dentro desse contexto, o que causa insegurança e imprecisão na hora de estimar as tarefas de manutenção e evolução do software. Esse artigo tem como finalidade propor uma ferramenta colaborativa de auxílio à estimativa de esforço de software, baseada em métricas de desenvolvimento armazenadas por experiências passadas de outros projetos em determinados contextos, apresentando suas funcionalidades e demonstrando um exemplo de uso da mesma, mostrando como o desenvolvimento da mesma ajudaria os engenheiros de software a obter maior precisão na estimativa do esforço nas tarefas de evolução de software.

Palavras-chave: evolução de software; estimativa de esforço; engenharia de software;

Modalidade de Participação: Iniciação Científica

PROPOSTA DE SOFTWARE COLABORATIVO PARA AUXÍLIO DE ESTIMATIVA DE ESFORÇO NA EVOLUÇÃO DE SOFTWARE

¹ Aluno de graduação. rodrigo.blizzard92@gmail.com. Autor principal

² Aluno de graduação. victorsc.rs@gmail.com. Co-autor

³ Docente. andreabordin@unipampa.edu.br. Orientador

⁴ Docente. bernardino@unipampa.edu.br. Co-orientador

PROPOSTA DE SOFTWARE COLABORATIVO PARA AUXÍLIO DE ESTIMATIVA DE ESFORÇO NA EVOLUÇÃO DE SOFTWARE

1 INTRODUÇÃO

Os processos de evolução de software tornaram-se parte muito importante dentro das organizações desenvolvedoras de software devido a demandas por mudanças rápidas e constantes. Nesse contexto, para manter a empresa competitiva, é necessário que o atendimento ao cliente em relação a orçamentos de pedidos de evolução de software solicitados sejam rápidos e precisos.

A estimativa de esforço é um aspecto muito importante a ser analisado para o desenvolvimento e evolução de um software. Segundo Molokken e Jorgensen et al. (2003), a incapacidade do setor de software de fornecer estimativas precisas de custo, esforço e/ou tempo de desenvolvimento é bem conhecida. Os especialistas em software não esperam que tais estimativas sejam absolutamente precisas, mas as consequências de previsões imprecisas podem ser graves e até fatais para uma empresa.

Para tentar solucionar esses problemas, foram desenvolvidas diversas técnicas para estimar esforço no desenvolvimento de software que também podem ser aplicadas às atividades de evolução. Muitas dessas técnicas, além de terem como requisito um certo nível de conhecimento para serem executadas, também acabam consumindo muitos recursos de tempo, às vezes não atendendo a necessidade de resposta rápida esperada pelo cliente.

Dentre essas técnicas são muito utilizadas as baseadas em experiência, que incluem o julgamento de especialistas, que estimam cada item baseado em comparações com projetos passados. A grande dificuldade neste caso é quando na organização desenvolvedora de software, não existem documentados nenhum projeto semelhante, ou até mesmo a falta de um especialista para realizar tal julgamento.

Um aspecto a ser considerado é a colaboratividade como solução para a ferramenta proposta, pois o compartilhamento nos conecta a outras pessoas, criando um senso de comunidade, economizando recursos e desenvolvendo sentimentos como a solidariedade e a proximidade. É uma forma alternativa de distribuição diferente da troca de bens e se caracteriza como uma forma alternativa ao mercado como conhecemos (Belk, 2009).

Através de uma revisão na literatura, buscando observar trabalhos existentes sobre softwares colaborativos que apoiam em atividades da Engenharia de Software, foi possível identificar uma lacuna referente a trabalhos que abordam a estimativa de esforço nas atividades de manutenção e evolução de software.

Buscando facilitar o processo de estimar o esforço de atividades de manutenção de software, este artigo propõe o desenvolvimento de uma ferramenta web colaborativa, que centralizará dados dos projetos finalizados de várias organizações de forma a auxiliar no processo de medição do esforço de novos projetos.

2 FUNDAMENTAÇÃO TEÓRICA

Lanubile et al., (2010), realizou um levantamento das ferramentas colaborativas que apoiam atividades de engenharia de software, onde foram mapeadas uma grande quantidade de ferramentas em diversas áreas, como por exemplo: ferramentas de projeto, modelagem, comunicação e desenvolvimento. Embora diversas ferramentas relevantes tenham sido encontradas, nenhuma delas tinha como propósito a estimativa de esforço de tarefas de desenvolvimento e manutenção.

Na realização de uma tarefa de evolução de software, várias características podem ter relevância na hora de executar uma estimativa de esforço. Através da pesquisa amplamente citada de Lientz et al., (1980), foi exposto que uma fração muito alta dos custos do ciclo de vida do software que estavam sendo gastos na manutenção. O estudo categorizou as atividades de manutenção em quatro classes: adaptativa: mudanças no ambiente do software; perfectiva: novos requisitos de usuário; corretiva: correção de erros; e preventiva: prevenção de problemas futuros. Destes, a pesquisa mostrou que cerca de 75% do esforço de manutenção estava nos dois primeiros tipos de manutenção, e a correção de erros consumia cerca de 21%. A partir destes dados, pode-se perceber uma importância de reconhecimento do tipo de manutenção solicitada.

Foi usada a estratégia de estimativa de esforço por analogia para determinar o funcionamento da solução proposta. A estimativa de esforço por analogia é um exemplo de estratégia para se conseguir uma estimativa confiável de forma mais rápida. Segundo Walkerden et al. (1999), estimar o esforço do projeto de software por analogia geralmente envolve uma série de etapas: Medir ou estimar os valores das métricas do projeto para o projeto alvo; Procurar um repositório de projetos concluídos para projetos semelhantes ao alvo e selecionar um ou mais projetos como análogos de origem; Usar o valor de esforço dos análogos de origem como uma estimativa inicial para o projeto alvo; Comparar os valores de métrica conhecidos para os projetos alvo e fonte; Ajustando a estimativa do esforço à luz das diferenças entre os projetos alvo e fonte.

Assim como proposto por Angelis et al. (2000), Walkerden et al.(1999), Shepperd et al.(1996) e Menzies et al. (2006), também propuseram *frameworks* e metodologias de estimativa de esforço fazendo uso de analogias, e muitos destes enfatizam o ajuste dos dados utilizados para se obter maior precisão e menor taxa de erros, porém, nem os trabalhos mais recentes idealizam uma ferramenta colaborativa que colete e compartilhe os dados entre diferentes organizações e pessoas.

3 SOLUÇÃO PROPOSTA

Com o objetivo de definir o conjunto de requisitos para a ferramenta em questão, pode-se observar alguns aspectos a serem levados em consideração para a estimativa de esforço na evolução de software. Com base nas necessidades observadas, alguns requisitos básicos foram identificados:

RF01) Manter Usuário: A ferramenta deve prover meios para inserir, editar, ler e deletar os usuários do sistema. Os usuários deverão poder realizar cadastro tanto como pessoa física, quanto como pessoa jurídica. O cadastro deve ser bem completo e rigoroso, para evitar de que a ferramenta seja poluída com falsos mantenedores, que podem inserir dados inconsistentes, levando com que a ferramenta indique estimativas errôneas.

RF02) Manter Manutenções Realizadas: A ferramenta deve prover meios para que os usuários possam inserir, editar, ler e deletar dados de um pedido de manutenção realizado pelo mesmo. Esse será o mecanismo de coleta de dados, para que a ferramenta possa utilizá-los de modo a auxiliar outros usuários em suas estimativas. Os aspectos considerados para essa coleta de dados, são: O tipo de manutenção: corretiva, adaptativa, perfectiva e preventiva; A linguagem de programação no qual o sistema foi desenvolvido; A quantidade de pessoas envolvidas; O nível de conhecimento das pessoas envolvidas; Nível de impacto: o número de classes atingidas pela modificação; o número de artefatos a serem modificados, o número de LOCs (*Line of Code*) que sofreram alterações; Aspectos organizacionais como por exemplo, se a evolução foi desenvolvida pelos desenvolvedores originais do produto, ou por

equipe de mantenedores externas ao projeto; O esforço que foi gasto em horas para o desenvolvimento da evolução/manutenção; Descrição do pedido de manutenção com palavras chave.

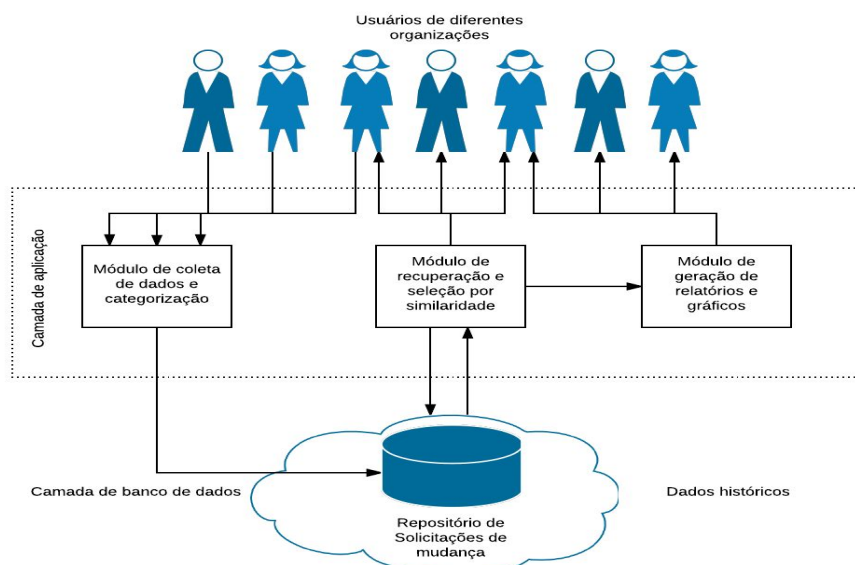
RF03) Filtrar Manutenções: A ferramenta deve prover meios para que os usuários possam buscar por dados históricos, dando como entrada um conjunto de características do pedido de manutenção, para que possa ser realizada estimativa análoga com os resultados obtidos.

RF04) Realizar Estimativa de Esforço: A ferramenta deve prover meios para que os usuários possam realizar a estimativa de esforço com base nas buscas de dados históricos. O usuário deverá fornecer ao sistema como entrada características referentes a estimativa que deseja realizar. A partir disso, a ferramenta deverá fornecer um conjunto primário de projetos de manutenção similares que podem auxiliar em sua estimativa de esforço. Após obter esse conjunto primário de projetos de manutenção, o usuário poderá observar os resultados, e excluir projetos indesejados para o contexto de sua estimativa. Com a escolha do conjunto das manutenções similares, o sistema deverá calcular a média de esforço gasto pelas atividades e apresentar relatórios e gráficos sobre o mesmo.

3.2 ARQUITETURA

A Figura 1, apresenta uma visão geral e simplificada da arquitetura da ferramenta proposta.

Figura 1 - Arquitetura do ambiente.



O **módulo de coleta de dados** deve se encarregar do RF01 e RF02, possibilitando inserção de novos cadastros de usuários e de dados de manutenções realizadas.

O **módulo de recuperação e seleção por similaridade** deve se encarregar do RF04 apoiando o planejamento das atividades, sugerindo estimativas de esforço de evoluções, levando em consideração a base de dados históricos mantida pela ferramenta e alimentada ao longo do tempo. Os dados apresentados no RF03 devem ser analisados quanto à sua semelhança, tornando possível obter visualizações de esforços reais já executados que podem ser úteis no processo de tomada de decisão e planejamento da manutenção/evolução.

A **Camada de banco de dados** é uma base de dados mantida publicamente na nuvem e composta pelo repositório de solicitações de mudança que é onde ficarão salvas as solicitações e o esforço respectivo a cada uma, assim como o usuário que a realizou.

4 EXEMPLO DE USO

Nessa seção será apresentado uma visão geral do funcionamento da ferramenta proposta, bem como um exemplo de uso para as suas principais funcionalidades.

O software deverá disponibilizar as suas principais funcionalidades mediante a autenticação do usuário. Caso o usuário não possua um cadastro na ferramenta, poderá realizar o cadastramento preenchendo devidamente os dados requisitados. Além disso, o usuário também poderá editar informações sobre o seu perfil de usuário, com a finalidade de mantê-lo sempre atualizado. Aspectos relevantes para manter o cadastro de um usuário são por exemplo: *e-mail*, senha, organização que o usuário está vinculado e documentos pessoais para que o cadastro não se torne tão trivial, dessa maneira reduzindo o envio de falsas manutenções com estimativas errôneas.

A adição de novos históricos de manutenção se dará a partir do preenchimento de alguns campos, contendo características pertinentes para a realização de uma estimativa de esforço. A medida que o sistema vai sendo alimentado com informações dos históricos de manutenção de diversos usuários diferentes, o acervo de resultados encontrados na hora de realizar uma estimativa de manutenção se torna mais completo, e por consequência, faz com que as próximas estimativas tenham mais precisão.

O processo de realização de uma estimativa de esforço levará em consideração aspectos similares aos de adição de dados históricos de manutenção, com a principal diferença de não conter o esforço em horas, que será obtido após a escolha dos projetos selecionados para a realização da estimativa.

5 CONSIDERAÇÕES FINAIS

A ideia de economia colaborativa está cada vez mais presente no cotidiano. Utilizar essa ideia para o desenvolvimento de ferramentas de apoio a engenharia de software, observando principalmente as áreas críticas, pode transformar o entendimento sobre a mesma. Verificando que algumas organizações já utilizam estimativa de esforço baseada em dados históricos coletados por elas, tornar esses dados públicos e realizar a centralização dos mesmos, poderia beneficiar de maneira sistemática diversas organizações e indivíduos. Por este motivo, a ferramenta proposta no presente trabalho tem grande potencial para apoiar o avanço na qualidade e precisão de estimativa de esforço em evoluções de software.

Em contrapartida, estimular os usuários a compartilharem dados pertinentes a seus projetos de manutenção pode ser algo difícil, pois enquanto a ferramenta não for popularizada, existirão poucos dados para que a mesma realize uma estimativa análoga de esforço, o que a tornaria pouco atraente, até que fosse alimentada com uma quantidade relevante de informações.

Por fim, através das funcionalidades apresentadas, deseja-se que o desenvolvimento da ferramenta proposta possa apoiar equipes de desenvolvimento no planejamento da manutenção e evolução de software, auxiliando no dimensionamento do esforço gasto para cada atividade de evolução, utilizando técnicas de rastreabilidade e similaridade.

É possível considerar a proposta da ferramenta algo viável, que poderia além de auxiliar na estimativa de esforço na evolução de software, aproximar desenvolvedores e

mantenedores de todas as partes do mundo para resolverem juntos uma das grandes lacunas da engenharia de software.

REFERÊNCIAS

ANGELIS, Lefteris; STAMELOS, Ioannis. A simulation tool for efficient analogy based cost estimation. **Empirical software engineering**, v. 5, n. 1, p. 35-68, 2000.

BELK, Russell. Sharing. **Journal of consumer research**, v. 36, n. 5, p. 715-734, 2009.

LANUBILE, Filippo et al. Collaboration tools for global software engineering. **IEEE software**, v. 27, n. 2, 2010.

LIENTZ, Bennet P.; SWANSON, E. Burton. Software maintenance management. 1980.

MENZIES, Tim et al. Selecting best practices for effort estimation. **IEEE Transactions on Software Engineering**, v. 32, n. 11, p. 883-895, 2006.

MOLOKKEN, Kjetil; JORGENSEN, Magne. A review of software surveys on software effort estimation. In: **Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on**. IEEE, 2003. p. 223-230.

PAREDES, Julia; ANSLOW, Craig; MAURER, Frank. Information visualization for agile software development. In: **Software Visualization (VISSEFT), 2014 Second IEEE Working Conference on**. IEEE, 2014. p. 157-166.

SHEPPERD, Martin; SCHOFIELD, Chris; KITCHENHAM, Barbara. Effort estimation using analogy. In: **Proceedings of the 18th international conference on Software engineering**. IEEE Computer Society, 1996. p. 170-178.

WALKERDEN, Fiona; JEFFERY, Ross. An empirical study of analogy-based software effort estimation. **Empirical software engineering**, v. 4, n. 2, p. 135-158, 1999.