

Capítulo 1

Introdução

Dentro do ciclo de vida de um produto de software o processo de manutenção tem papel fundamental. Devido ao seu alto custo, que segundo alguns estudos varia de 60% a 90% do preço final do software [Kaur & Singh, 2015], as atividades relacionadas a manter e evoluir têm sua importância considerada tanto pela comunidade científica quanto pela indústria. Desde o final da década de 1970 [Zelkowitz et al., 1979] uma atenção tem sido dada para os custos da Manutenção de Software. Nas décadas de 1980 e 1990 alguns estudos propuseram modelos de mensuração do valor necessário para manter o software [Herrin, 1985, Hirota et al., 1994]. Apesar da evolução das metodologias para se manter um software a estimativa é que nas últimas duas décadas o custo de manutenção tenha aumentado em 50% [Koskinen, 2010]. Esta tendência pode ser observada na Figura 1.1 onde é possível verificar a evolução dos gastos com manutenção de software como fração do preço final do produto.

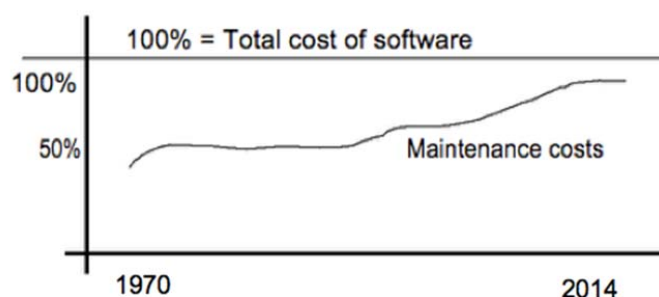


Figura 1.1: Evolução da manutenção de software como percentual do custo total. Extraído de [Engelbertink & Vogt, 2010]

Uma vez que o software entra em operação, anomalias são descobertas, mudanças ocorrem no ambiente de operação e novos requisitos são solicitados pelo usuário. Todas

estas demandas devem ser solucionadas pela Manutenção de Software que se inicia com entrega do sistema. Contudo, alguns autores defendem que certas atividades, como aquelas relativas à análise da qualidade, começam bem antes da entrega do produto.

A *Manutenção*, dentre outros aspectos, corresponde ao processo de modificar um componente ou sistema de software após a sua entrega com o objetivo de *corrigir falhas, melhorar o desempenho ou adaptá-lo devido à mudanças ambientais* [IEEE, 1990]. De maneira relacionada, *Manutenibilidade* é a propriedade de um sistema ou componente de software em relação ao grau de *facilidade* que ele pode ser corrigido, melhorado ou adaptado [IEEE, 1990].

As manutenções em software podem ser divididas em *Corretiva, Adaptativa, Perfeccionista e Preventiva* [Lientz & Swanson, 1980, IEEE, 1990]. A ISO 14764 discute os quatro tipos e propõe que exista um elemento comum denominado *Requisição de Mudança (RM)* que representa as características compartilhadas pelos demais tipos.

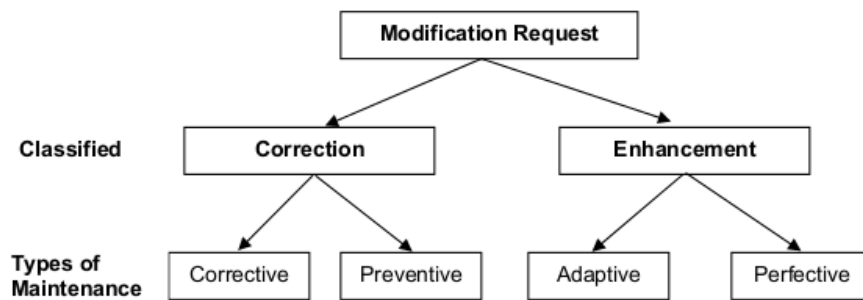


Figura 1.2: Tipos de manutenção segundo a norma ISO/IEC 14764. Extraído de [ISO/IEC, 2006]

Por conta do volume das RM é importante a utilização de um software com o objetivo de gerenciá-las. Esse controle é geralmente realizado por *Ferramentas de Gerenciamento de Requisição de Mudança - FGRM*, que auxiliam os desenvolvedores na correção de forma individual ou colaborativa de falhas (bugs) e no desenvolvimento de novas funcionalidades, por exemplo. Estas ferramentas podem ainda ser utilizadas por gestores, analistas de qualidade e usuários finais para atividades como gerenciamento de projetos, comunicação, discussão e revisões de código. A literatura em Manutenção de Software não define uma nomenclatura comum para este tipo de ferramenta. Em alguns estudos é utilizado nomes como Sistema de Controle de Defeito - Bug Tracking Systems, Sistema de Gerenciamento da Requisição - Request Management System, Sistemas de Controle de Demandas (SCD)- Issue Tracking Systems. Todavia, de modo geral, o termo se refere as ferramentas utilizadas pelas organizações para *gerir as Requisições de Mudança*. Neste trabalho utilizaremos o termo **Ferramentas**

de Gerenciamento de Requisições de Mudança (FGRM) ao referimos a este tipo de software. A Tabela 1.1 apresenta alguns exemplos de software que podem ser classificados como FGRMs. Também são listados serviços da Internet que oferecem funcionalidades presentes nas FGRMs na forma de Software como Serviço [Fox et al., 2013].

| Ferramentas | | Serviços da Internet | |
|-------------|---|----------------------|---|
| Bugzilla | https://www.bugzilla.org/ | SourceForge | https://sourceforge.net/ |
| MantisBT | https://www.mantisbt.org/ | Launchpad | https://launchpad.net/ |
| Trac | https://trac.edgewall.org/ | Code Plex | https://www.codeplex.com/ |
| Redmine | www.redmine.org/ | Google Code | https://code.google.com/ |
| Jira | https://www.atlassian.com/software/jira | GitHub | https://github.com/ |

Tabela 1.1: Exemplos de ferramentas e serviços da Internet. Extraído de [Cavalcanti et al., 2014]

1.1 Motivação

Diante da maior presença de software em todos os setores da sociedade existe um interesse por parte da academia e da indústria no desenvolvimento de processos, técnicas e *ferramentas* que reduzam o esforço e o custo do desenvolvimento e manutenção de software. Um exemplo é o trabalho de Yong & Mookerjee [Tan & Mookerjee, 2005] que propõe um modelo que reduz os custos de manutenção e reposição durante a vida útil de um sistema de software. O modelo demonstrou que em algumas situações é *melhor substituir um sistema do que mantê-lo*. Este problema é agravado tendo em vista que em alguns casos são necessários 60% dos desenvolvedores dedicados à tarefas de manutenção de sistemas [Zhang, 2003].

Dependendo do tamanho de um projeto de software é necessário a utilização de uma FGRM para gerenciar as Requisições de Mudança por conta do volume e das diferentes partes interessadas que necessitam de um local centralizado onde possa registrar as falhas encontradas e as melhorias que necessitam [Serrano & Ciordia, 2005]. Neste sentido, podemos afirmar que as RMs suportam projetos de diferentes tipos e tamanhos, sejam eles de código aberto (Apache, Linux, Open Office) ou em organizações públicas e privadas (NASA e IBM). Dando suporte à software de diferentes plataformas, seja aplicativos de área de trabalho (desktop), web ou móvel (mobile).

A literatura da área nos mostra que as FGRMs desempenham um papel além do gerenciamento dos pedidos de manutenção software. Avaliando o controle de demandas como um processo social, Bertram e outros [Bertram et al., 2010] realizaram um estudo

qualitativo sobre FGRMs que são utilizadas por equipes pequenas de desenvolvimento de software. Os resultados demonstraram que a ferramenta não é apenas um banco de dados de rastreamento de falhas, mas atua como um ponto central para a comunicação e coordenação de diversas partes interessadas (stakeholders) dentro e fora da equipe de manutenção. Os clientes, gerentes de projeto, analistas de qualidade e programadores contribuem em conjunto para o compartilhamento de conhecimento do projeto por meio da utilização das FGRMs.

No trabalho de Breu e outros [Breu et al., 2010a] o foco é analisar o papel dos FGRMs no suporte à colaboração entre desenvolvedores e usuários de um software. Com base nos resultados foi possível verificar que o uso da ferramenta possibilitou que os usuários desempenhassem um papel além de simplesmente reportar uma falha: a participação ativa e permanente foi importante no progresso da resolução das falhas que eles descreveram.

Um outro benefício da utilização das FGRM é que as mudanças no software podem ser rapidamente identificadas e reportadas para os desenvolvedores [Anvik et al., 2005]. Além disso, eles podem ajudar a estimar o custo do software, na análise de impacto, planejamento, rastreabilidade, descoberta do conhecimento [Cavalcanti et al., 2013].

Conforme exposto as FGRMs desempenham um papel fundamental no contexto do desenvolvimento e manutenção de software. Contudo, no escopo de utilização das FGRMs diversos desafios se apresentam: duplicação de RMs, pedidos de modificação abertos inadvertidamente, grande volume de RMs que devem ser atribuídas aos desenvolvedores mais aptos, erros descrito de forma incompleta, análise de impacto das RMs e atribuídas de maneira incorreta [Cavalcanti et al., 2014]. Diante destes problemas e desafios é importante entender como estas ferramentas estão sendo utilizadas. Com base neste conhecimento, e com o que está sendo proposto na literatura, é possível avaliar e entender necessidades dos profissionais envolvidos com Manutenção de Software com o objetivo de melhorar as funcionalidades oferecidas por este tipo de software.

1.2 Problema

O desenvolvimento e a manutenção de software envolvem diversos tipos de métodos, técnicas e ferramentas. Em especial no processo de Manutenção, um importante aspecto são as diversas RMs que devem ser gerenciadas. Este controle é realizado pelas FGRMs cujo o uso vem crescendo em importância, sobretudo, por sua utilização por ge-

stores, analistas da qualidade e usuários finais para atividades como tomada de decisão e comunicação. Contudo, em alguns casos, as ferramentas são meramente interfaces melhores para o banco de dados que armazena as RMs [Zimmermann et al., 2009a]. Apesar da inegável importância das FGRMs, percebe-se um aparente desacoplamento deste tipo de ferramenta com as necessidades das diversas partes interessadas (stakeholders) na manutenção e evolução de software. A utilização de “*demanda*” como conceito central para as FGRMs parece ser distante das necessidades práticas dos projetos de software, especialmente no ponto de vista dos desenvolvedores [Baysal et al., 2013].

Um exemplo deste desacoplamento pode ser visto no trabalho proposto por Baysal & Holme [Baysal & Holmes, 2012] no qual desenvolvedores que utilizam o Bugzilla¹ relataram dificuldade em manter uma compreensão do escopo que as RMs atribuídas a eles possuem. Segundo os participantes do estudo seria importante que a ferramenta tivesse um suporte melhorado para o conceito de Consciência Situacional - Situational Awareness. Em síntese, eles gostariam de estar cientes da situação global do projeto bem como das atividades que outras pessoas estão realizando.

Existem outros prolemas que são potencializados pela ausência de certas funcionalidades nas FGRMs. Um exemplo são as RMs que acabam sendo relatadas com baixa qualidade. Nesta situação os usuários acabam sendo questionados a inserir mais detalhes que muitas vezes eles não tem conhecimento. Por outro lado, verifica-se uma frustração por parte dos desenvolvedores que acabam desapontados sobre a qualidade do que foi reportado [Just et al., 2008].

Corroborando com a necessidade de evolução das FGRMs, o estudo realizado por Zimmermann e outros [Zimmermann et al., 2009a] propõem quatro dimensões de melhorias para este tipo de software. Estas dimensões estão apresentadas na Figura 1.3 e serão detalhadas no Capítulo 3 onde foram utilizadas na classificação de estudos no Mapeamento Sistemático realizado.

Neste estudo estamos especialmente interessados em analisar e propor melhorias relativas ao domínio da *Ferramenta*. Ao bem do nosso conhecimento é reduzido o número de trabalhos que avaliem de forma sistemática as funcionalidades oferecidas pelas FGRMs ao mesmo tempo que faça relação com que vem sendo proposto na literatura sobre o assunto.

É importante ressaltar que os estudos sobre melhorias das FGRMs a crescente adoção de técnicas propostas pelos agilistas na Manutenção de Software [Soltan & Mostafa, 2016, Devulapally, 2015, Heeager & Rose, 2015]. Neste contexto, seria importante que ferramentas que dão suporte à manutenção, tal como

¹<https://www.bugzilla.org>

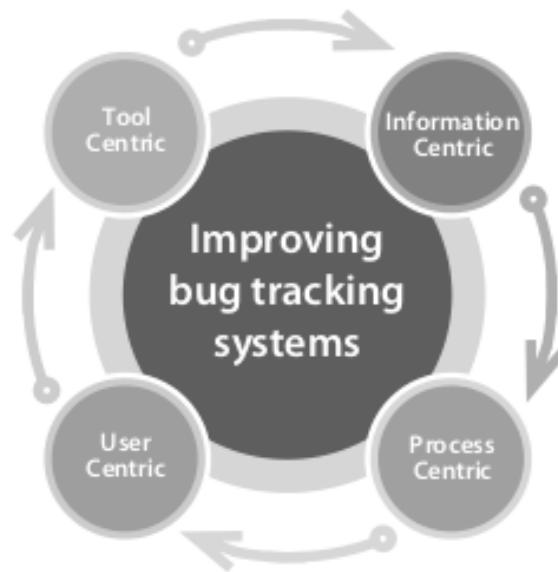


Figura 1.3: Dimensões de melhoria das FGRMs. Extraído de [Zimmermann et al., 2005]

as FGRMs, evoluíssem para se adaptar a esta nova forma de trabalhar. Um outro fator que agrega sobre a necessidade de adequação das FGRMs são as diversas extensões (plugins) propostas na literatura [Rocha et al., 2015, Thung et al., 2014c, Kononenko et al., 2014].

1.3 Objetivos

Segundo o nosso entendimento existe um distanciamento entre as necessidades dos profissionais envolvidos em Manutenção de Software e as funcionalidades oferecidas pelas FGRM. Por esta razão este trabalho de dissertação investiga e contribui no entendimento de como as Ferramentas de Gerenciamento de Requisição de Mudança estão sendo melhoradas ou estendidas no contexto da transformação do processo de desenvolvimento e manutenção de software de um modelo tradicional para outro que incorpora cada vez mais as práticas propostas pelos agilistas. O intuito é analisar como as FGRM estão sendo modificadas com base na literatura da área em contraste com o ponto de vista dos profissionais envolvidos em manutenção de software.

Neste contexto, elaboramos um estudo sobre as Ferramentas de Gerenciamento de Requisição de Mudança (FGRM) com os seguintes objetivos:

- (i) entender os requisitos comuns deste tipo de ferramenta;

- (ii) mapear as melhorias para as FGRM que estão sendo propostas na literatura;
- (iii) avaliar sobre o ponto de vista dos profissionais a situação atual dos FGRM;
- (iv) propor melhorias ou novas funcionalidades para as FGRM.

1.4 Visão Geral da Dissertação

A fim de alcançarmos os objetivos descritos foi proposto um conjunto de melhorias para as funcionalidades das FGRMs. As melhorias foram realizadas com base em três estudos empíricos: um mapeamento sistemático da literatura, apresentado no Capítulo 3; e uma pesquisa com profissionais, apresentada no Capítulo 4; um levantamento com questionário validando sugestões de melhorias descrito no Capítulo 5.

Mediante o mapeamento sistemático obtivemos e avaliamos o estado da arte sobre novas funcionalidades que estão propostas na literatura. A partir do estudo foi possível propor dois esquemas de classificação: por dimensão de melhoria e suporte ao papel desempenhado na manutenção de software. De maneira similar, através da caracterização das funcionalidade de algumas FGRMs código aberto ou disponíveis comercialmente e escolhidas mediante uma pesquisa com profissionais identificamos o estado da prática deste tipo de ferramenta.

Com base em dois estudos anteriores conduzimos uma pesquisa com profissionais envolvidos em manutenção de software onde pedimos que avaliassem os requisitos funcionais e não funcionais que poderiam melhorar as FGRM já existentes. O questionário também quis saber a opinião dos profissionais sobre a relevância das propostas de melhorias existente na literatura em sua rotina de trabalho. Fundamentado nos estudos descritos foi proposto um conjunto de melhorias. Como prova de conceito, uma destas melhorias foi implementada como uma extensão de uma FGRM.

1.5 Metodologia de Pesquisa

A metodologia de pesquisa utilizada neste estudo é baseada em uma abordagem multi-método [Hesse-Biber, 2010]. Este tipo de desenho combina dois ou mais métodos quantitativo (ou qualitativo) em um único estudo. Um estudo que faça uso de um survey e um experimento é um exemplo deste tipo de enfoque [Hesse-Biber, 2010].

As etapas do trabalho, que compõem a abordagem multi-método estão listadas a seguir:

- (i) Mapeamento Sistemático da Literatura [Petersen et al., 2008]

- (i) Caracterização das Ferramentas de Gerenciamento de Requisição de Mudança (FGRM)
- (i) Pesquisa (Survey) com os desenvolvedores [Wohlin et al., 2012]

1.6 Contribuições do Estudo

Este estudo sistematiza a literatura sobre melhorias das funcionalidades das FGRMs ao mesmo tempo que avalia junto aos profissionais as relevâncias de tais alterações. Este estudo ainda se propõe este tipo de software por meio de uma caracterização de suas funcionalidades. Ao final deste trabalho teremos um conjunto de funcionalidades que podem ser implementadas neste tipo de software visando a sua melhoria.

1.7 Organização do Trabalho

Este trabalho de dissertação está organizado conforme descrito a seguir. No Capítulo 2 apresentamos e discutimos os principais conceitos neste estudo. Neste mesmo capítulo é descrito um estudo onde coletamos as principais funcionalidades de um conjunto de FGRMs que foram definidas como relevantes na visão de profissionais envolvidos em Manutenção de Software.

Um Mapeamento Sistemático da Literatura foi conduzido no Capítulo 3 com o objetivo de levantar as melhorias nas funcionalidades das FGRMs que estão sendo propostas na literatura. Os estudos foram classificados em dimensões de melhorias e pelo papel desempenhado na Manutenção de Software que a melhoria visa dar suporte.

No Capítulo 4 reunimos a opinião de profissionais envolvidos em Manutenção de Software sobre as funcionalidades oferecidas pelas FGRMs que eles utilizam. Estes profissionais exercem suas atividades em projetos de código aberto e empresas no setor público e privado. Foi possível identificar que grande parte dos profissionais estão satisfeitos com a ferramenta que utiliza, contudo, eles estão bastante interessados em novos tipos de comportamento no software.

Tomando como base a literatura sobre melhorias nas FGRMs e os resultados obtidos nos estudos descritos nos capítulos anteriores, apresentamos e discutimos um conjunto de sugestões de melhorias no Capítulo 5. As recomendações propostas foram avaliadas por profissionais que contribuem no desenvolvimento de FGRM. Em geral, as recomendações tiveram boa aceitação tanto sobre sua necessidade quanto por sua facilidade de implementação. No Capítulo 6 foi realizada uma prova de conceito uma

das sugestões propostas foram implementadas com uma extensão de uma FGRM. As conclusões e trabalhos futuros estão descritos no Capítulo 7.

