Um Estudo de Ferramentas de Gerenciamento de Requisição de Mudança

Julho de 2017

Vagner Clementino
Rodolfo Resende - Orientador

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

Agenda

Contexto

Agenda

Contexto

Problema

Agenda

Contexto

Problema

Objetivos

Agenda

Contexto

Problema

Objetivos

Metodologia

Agenda

Contexto

Problema

Objetivos

Metodologia

Resultados

Agenda

Contexto

Problema

Objetivos

Metodologia

Resultados

Discussão

Agenda

Contexto

Problema

Objetivos

Metodologia

Resultados

Discussão

Ameças à Validade

Agenda

Contexto

Problema

Objetivos

Metodologia

Resultados

Discussão

Ameças à Validade

Conclusão e Trabalhos Futuros

Importância da Manutenção de Software

- *“Another flaw in the human character is that everybody wants to build and nobody wants to do maintenance”.* Kurt Vonnegut, Jr.

Importância da Manutenção de Software

- Dentro do ciclo de vida do software o processo de Manutenção de Software tem papel fundamental.
- Devido ao seu alto custo, que pode variar entre 60% e 90% do preço final do sistema [12], sua importância vêm sendo considerada tanto pela comunidade científica quanto pela indústria.

Conceito de Manutenção de Software

- *Manutenção de Software:* 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* [9].
- *Manutenibilidade:* propriedade de um sistema ou componente de software em relação ao grau de *facilidade* que ele pode ser corrigido, melhorado ou adaptado [9].

Tipos de Manutenção em Software

- A manutenção de software pode ser dividida em *Corretiva, Adaptativa, Perfectiva e Preventiva* [15, 9].
- A *ISO 14764* [10] propõe que exista um elemento denominado Requisição de Mudança (RM) que corresponde a uma agregação de características que representam uma solicitação de manutenção de qualquer das quatro categorias.

Tipos de Manutenção em Software

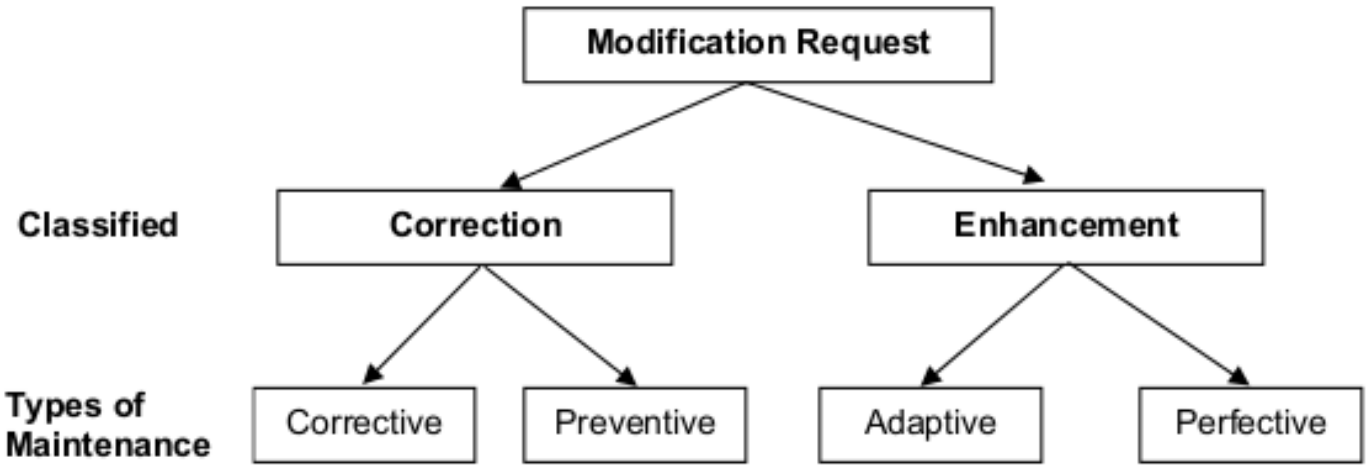


Figura 1: Tipos de manutenção segundo a norma ISO/IEC 14764 [10]

Papéis na Manutenção de Software

- Nesta dissertação consideramos os seguintes papéis desempenhados no processo de manter e evoluir software:
  - *Usuário Afetado:* Indivíduo que utiliza o software correspondente à Requisição de Mudanças (RM) que será relatada. O defeito, a melhoria ou evolução no software, representada pela RM, estão relacionadas com os desejos e necessidades deste papel.
  - *Reportador:* Responsável por registrar a RM. Em certas situações este papel é desempenhado tanto pelo usuário do sistema quanto pela equipe de manutenção.

Papéis na Manutenção de Software

- *Gerente de Requisição de Mudança (Maintenance-request manager):* Responsável por decidir se uma RM será aceita ou rejeitada. Além disso, ele define qual tipo de manutenção deverá ser aplicada.
- *Agente de Triagem (Scheduler)*: Deve planejar a fila de RMs e atribuí-las para o desenvolvedor mais apto. A decisão pode considerar a carga de trabalho existente para cada possível destinatário da RM.

Papéis na Manutenção de Software

- *Desenvolvedor:* Responsável por realizar as ações que irão solucionar a RM.
- *Analista de Qualidade:* Tem por responsabilidade avaliar se uma RM solucionada por um Desenvolvedor foi resolvida de forma correta e dentro dos padrões de qualidade exigidos pelo projeto.
- *Chefe da Manutenção (Head of Maintenance):* Este papel é responsável por definir os padrões e procedimentos que compõem o processo de manutenção que será utilizado.

Requisição de Mudança

- Requisição de Mudança (RM) corresponde ao registro da informação sobre o defeito, evolução ou melhoria de um sistema [25]

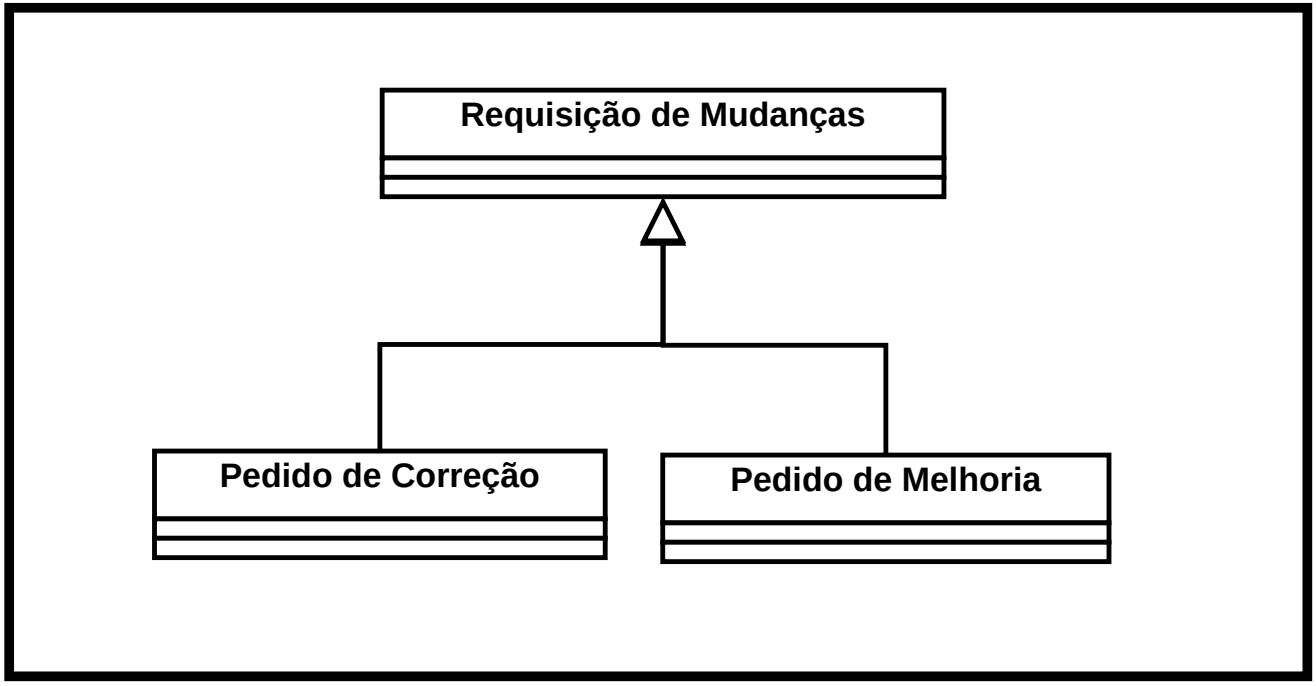


Figura 2: Modelo conceitual de uma Requisição de Mudanças

Atributos de uma RM

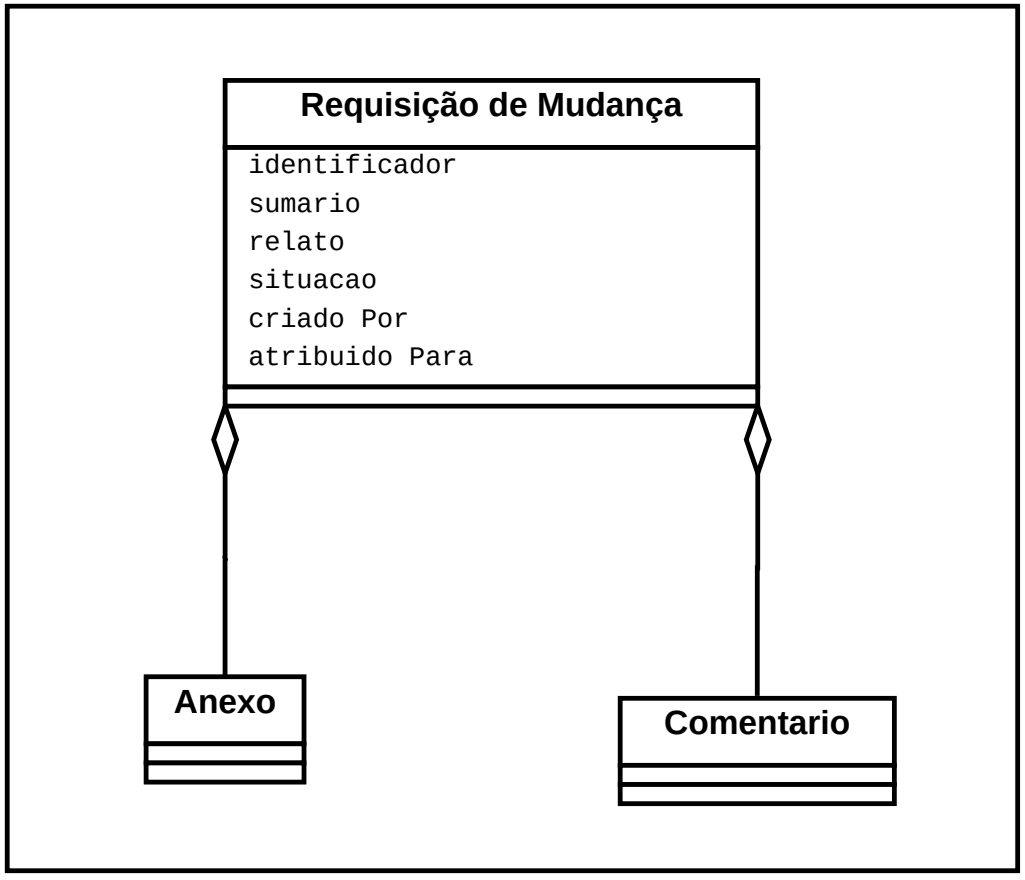
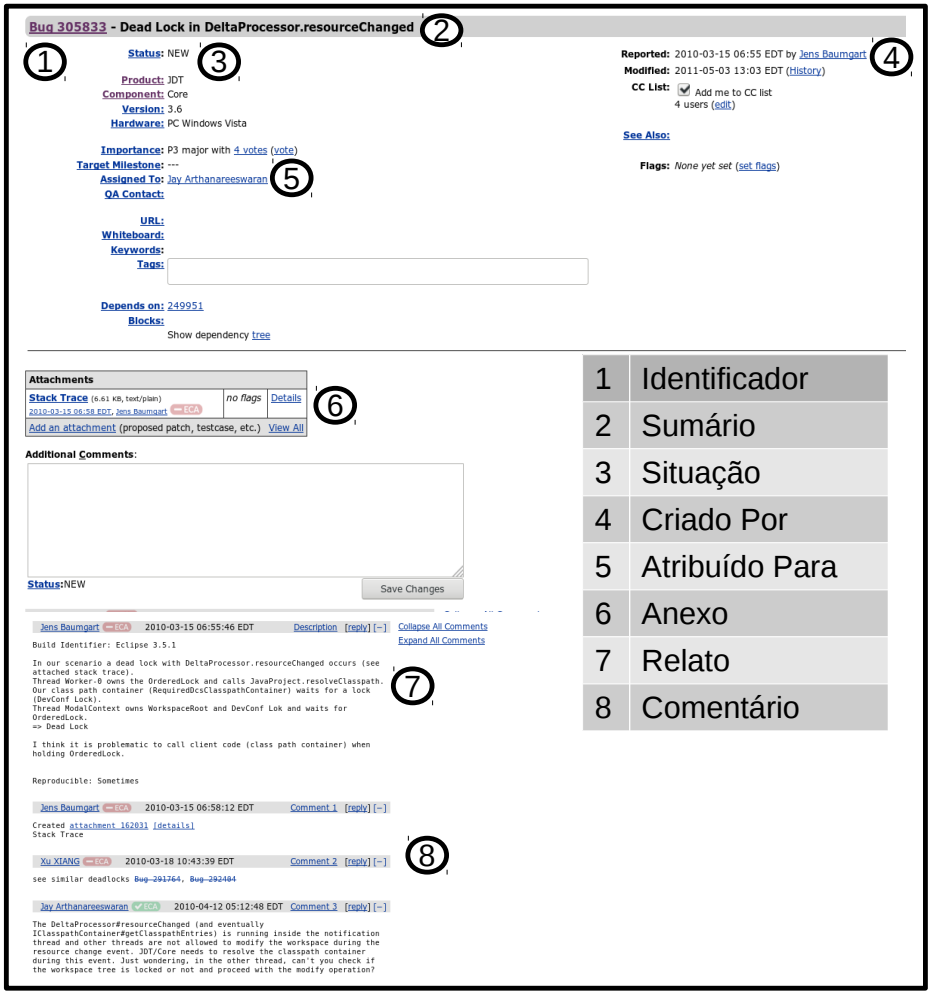


Figura 3: Informações que compõem uma RM. Baseado em trabalho de Singh & Chaturvedi [23]

Exemplo de uma RM



1	Identificador
2	Sumário
3	Situação
4	Criado Por
5	Atribuído Para
6	Anexo
7	Relato
8	Comentário

Figura 4: RM do Projeto Eclipse

Ciclo de Vida da RM

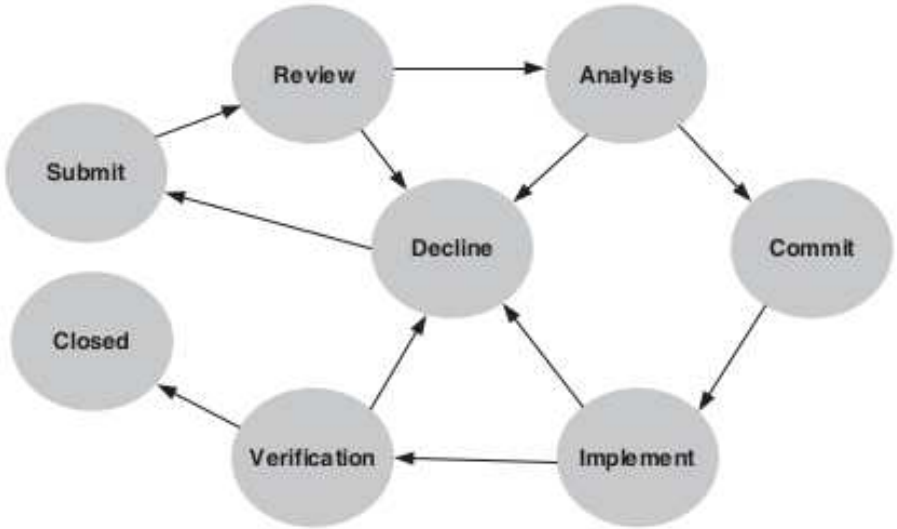


Figura 5: Diagrama de estados de uma RM. Extraído de [25]

Problemas e Desafios da Gestão das RMs

- Localização do Problema
- Dificuldade na Visualização das Informações das RMs
- Baixa Qualidade do Relato
- Identificação de RMs Duplicadas
- Atribuição (Triagem) de RM
- Classificação da RM
- Estimativa de Esforço da RM
- Recomendação de RMs

Volume de RMs do Projeto . . .

- Incluir uma figura ou tabela com o volume de RMs de um projeto

Ferramentas de Gerenciamento de Requisição de Mudança (FGRM)

- Gerenciar as atividades de manutenção e seus artefatos possui um alto custo.
- Dependendo do tamanho do projeto de software é necessário a utilização de uma FGRM para gerenciar as suas requisições de mudança.
- As partes interessadas (stakeholders) necessitam de um espaço único onde possam registrar as falhas encontradas e as melhorias que necessitam [22].

Ferramentas de Gerenciamento de Requisição de Mudança

Além do que Gerenciar RMs

- Ponto central para a comunicação e coordenação das diversas partes interessadas [3].
- Possibilita que os usuários participem do processo de solução das RMs [5].
- Suporte para atividades como [6]:
  - estimativa do custo do software
  - análise do impacto de uma modificação
  - planejamento do projeto
  - rastreabilidade de uma falha
  - extração de conhecimento

Problema

- Apesar da inegável importância das FGRMs percebe-se um aparente desacoplamento de suas funcionalidades com as necessidades de seus usuários [1, 11].
- A utilização de “demanda” parece estar distante das necessidades práticas dos projetos, especialmente no ponto de vista dos desenvolvedores [2].
- Diversas extensões (plugins) estão sendo propostas na literatura [21, 24, 14].

Objetivos

- Elaboramos um estudo sobre as FGRMs com os seguintes objetivos:

(i) entender os requisitos e funcionalidades oferecidas por este tipo de ferramenta;

(ii) mapear as melhorias para as FGRMs que estão sendo propostas na literatura;

(iii) avaliar sobre o ponto de vista dos profissionais a situação atual funcionalidades oferecidas pelas FGRMs;

(iv) propor melhorias para as funcionalidades das FGRMs.

Metodologia

- Estudo sobre as funcionalidades das FGRMs
- Mapeamento Sistemático da Literatura [19]
- Levantamento (Survey) com desenvolvedores [26]
- Sugestões de melhorias para as FGRMs
- Implementação de extensão para FGRM

Estudo sobre as funcionalidades das FGRMs

- Análise das funcionalidades oferecidas pelas FGRMs
- Inspeção inicial resulto em aproximadamente 50 ferramentas[1].
- Optamos por conduzir o estudo em um conjunto menor

[1] https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems

Estudo sobre as funcionalidades das FGRMs

- Etapas do estudo
  - (i) Seleção das Ferramentas
  - (ii) Inspeção da Documentação
  - (iii) Agrupamento das Funcionalidades

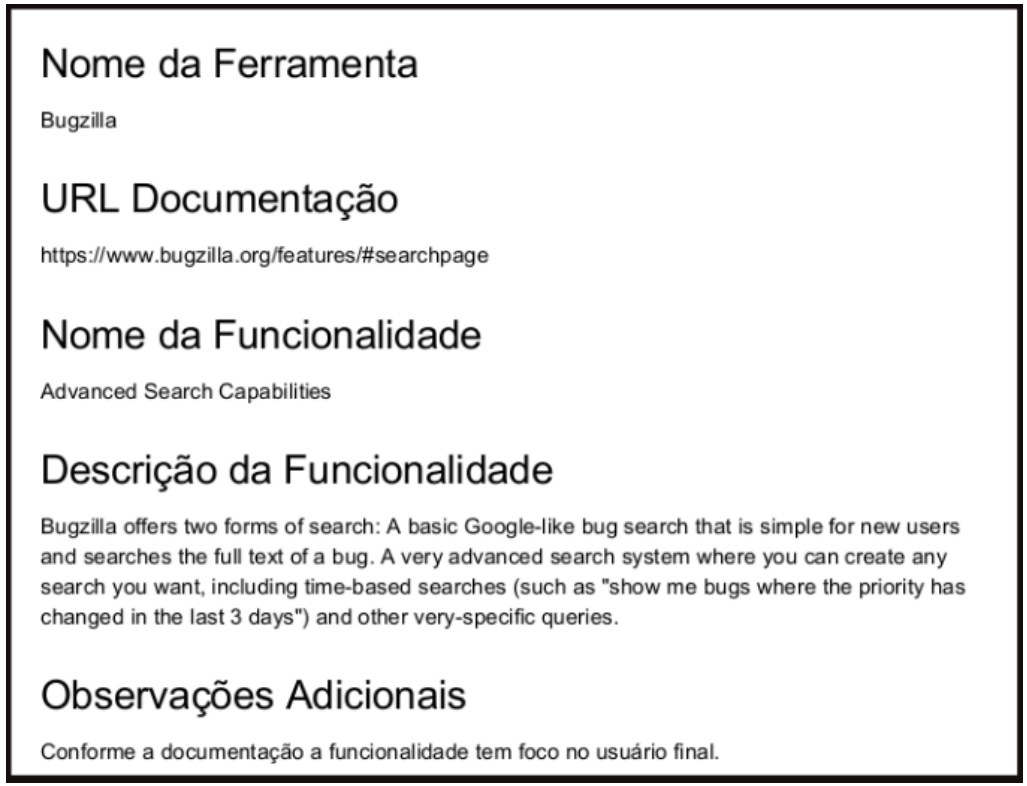
Estudo sobre as funcionalidades das FGRMs

- Seleção das Ferramentas
  - Levantamento por Questionário
  - Dois grupos de participantes
  - 52 participações
  - 06 ferramentas escolhidas

Estudo sobre as funcionalidades das FGRMs

- Inspeção da Documentação
  - Leitura do material disponível na Internet
  - As funcionalidades foram classificadas através da técnica de *Cartões de Classificação - Sorting Cards* [11, 18, 16].

Estudo sobre as funcionalidades das FGRMs



Nome da Ferramenta

Bugzilla

URL Documentação

https://www.bugzilla.org/features/#searchpage

Nome da Funcionalidade

Advanced Search Capabilities

Descrição da Funcionalidade

Bugzilla offers two forms of search: A basic Google-like bug search that is simple for new users and searches the full text of a bug. A very advanced search system where you can create any search you want, including time-based searches (such as "show me bugs where the priority has changed in the last 3 days") and other very-specific queries.

Observações Adicionais

Conforme a documentação a funcionalidade tem foco no usuário final.

Figura 6: Exemplo de um cartão ordenado para uma funcionalidade da FGRM Bugzilla

Estudo sobre as funcionalidades das FGRMs

- Agrupamento das Funcionalidades
  - Análise Individual: O autor e um outro especialista realizam de forma separada os agrupamentos.
  - Anaĺise Compartilhada: Em um segundo momento tanto o autor quanto o especialista discutem as possíveis divergências até que um consenso seja obtido.

Mapeamento Sistemático da Literatura

- Mapeamento com base nas diretrizes propostas por Petersen e outros [19].
- Questões de Pesquisa
  - *Questão 01*: Quais as melhorias e novas funcionalidades estão sendo propostas para as FGRM?
  - *Questão 02*: Quais papéis envolvidos no processo de manutenção de software as melhorias das funcionalidades visam dar suporte?

Mapeamento Sistemático da Literatura

- Os estudos primários coletados das bases de pesquisa *IEEE Explore, ACM Digital Library, Scopus, e Inspec/Compendex*.
- As sentenças de buscas foram produzidas com base na metodologia PICO (Population, Intervention, Comparison and Outcomes) [13].

Mapeamento Sistemático da Literatura

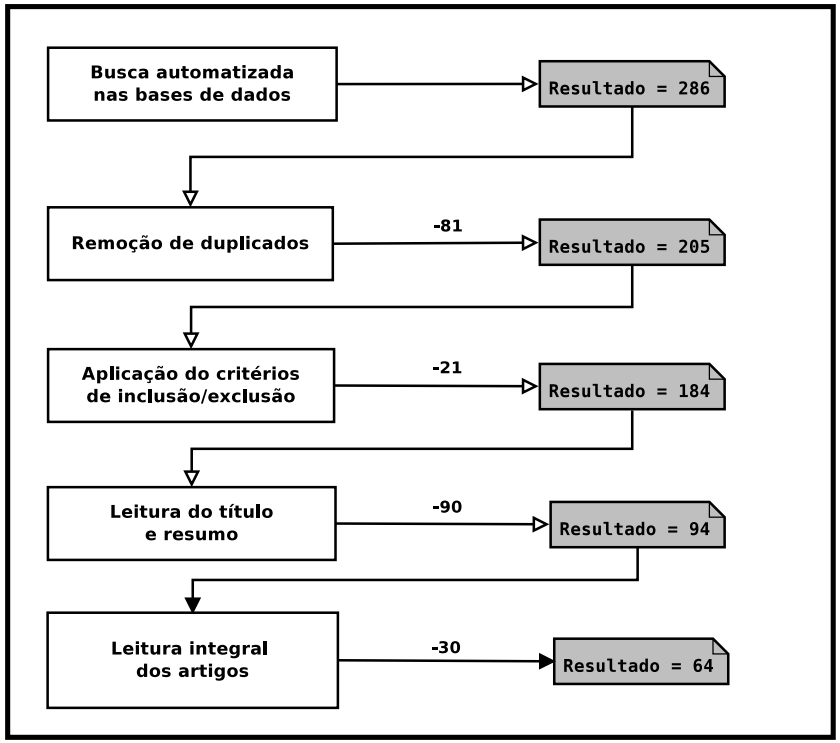


Figura 7: Número de artigos incluídos durante o processo de seleção dos estudos. Figura baseada em [20]

Levantamento com Desenvolvedores

- Questão 01: Qual a opinião dos profissionais envolvidos em manutenção de software com relação as funcionalidades oferecidas pelas FGRM?
- Questão 02: Na visão dos profissionais envolvidos em manutenção de software quais das melhorias nas funcionalidades das FGRMs propostas na literatura teriam maior relevância em suas atividades?

Levantamento com Desenvolvedores

- Questão 03: As práticas propostas pelos agilistas estão sendo utilizadas no processo de manutenção de software?
- Questão 04: Como as FGRMs podem ajudar as equipes de manutenção na adoção das práticas propostas pelos agilistas?

Levantamento com Desenvolvedores

- Fonte de Amostragem corresponde a um banco de dados, não necessariamente automatizado, em que um subconjunto válido da população pode ser recuperado. Outra característica é permitir a extração aleatória de amostras da população de interesse [7].

Identificador	Fonte de Amostragem	URL
FA01	Python	https://bugs.python.org/
FA02	Stack Overflow	https://stackoverflow.com

Tabela 1: Fontes de Amostragem utilizadas no estudo

Levantamento com Desenvolvedores

- Formulário preenchido por 85 participantes

Função Desempenhada	Total
Desenvolvedor	23
Engenheiro de Software	17
Gerente	12
Arquiteto de Software	5
Pesquisador	5
Consultor	4
Estudante	3
Analista de Qualidade	1
Designer	1

Tabela 2: Função desempenhada pelos participantes

Sugestões de Melhorias

- Sugestões foram compiladas utilizando a literatura da área e os levantamentos realizados nesta dissertação, especialmente com Mapeamento Sistemático e Levantamento com Profissionais;
- E nos estudos que propõem melhorias para as FGRM [27, 4, 23].

Sugestões de Melhorias

- Propostas 08 sugestões de melhorias
- Avaliadas através de um levantamento mediante questionário com profissionais que contribuem em projetos de código aberto hospedados no Github.

Projeto	Participantes
DEBBUGS	4
MANTISBT	4
TRAC	4
FOSSIL	3
BUGZILLA	2
REDMINE	2
OUTROS	6

Tabela 3: Projetos que os participantes contribuem.

Implementação de Extensão

- Implementação da Sugestão #1 na plataforma Github.
- Cliente para API do Github[2] que possibilita analisar a qualidade da informação fornecida no relato.
- Batizada de *IssueQuality*

Sugestão #01: As FGRMs devem fornecer realimentação (feedback) relacionado com a qualidade do texto relatado.

[2] `https://api.github.com/`

Implementação de Extensão

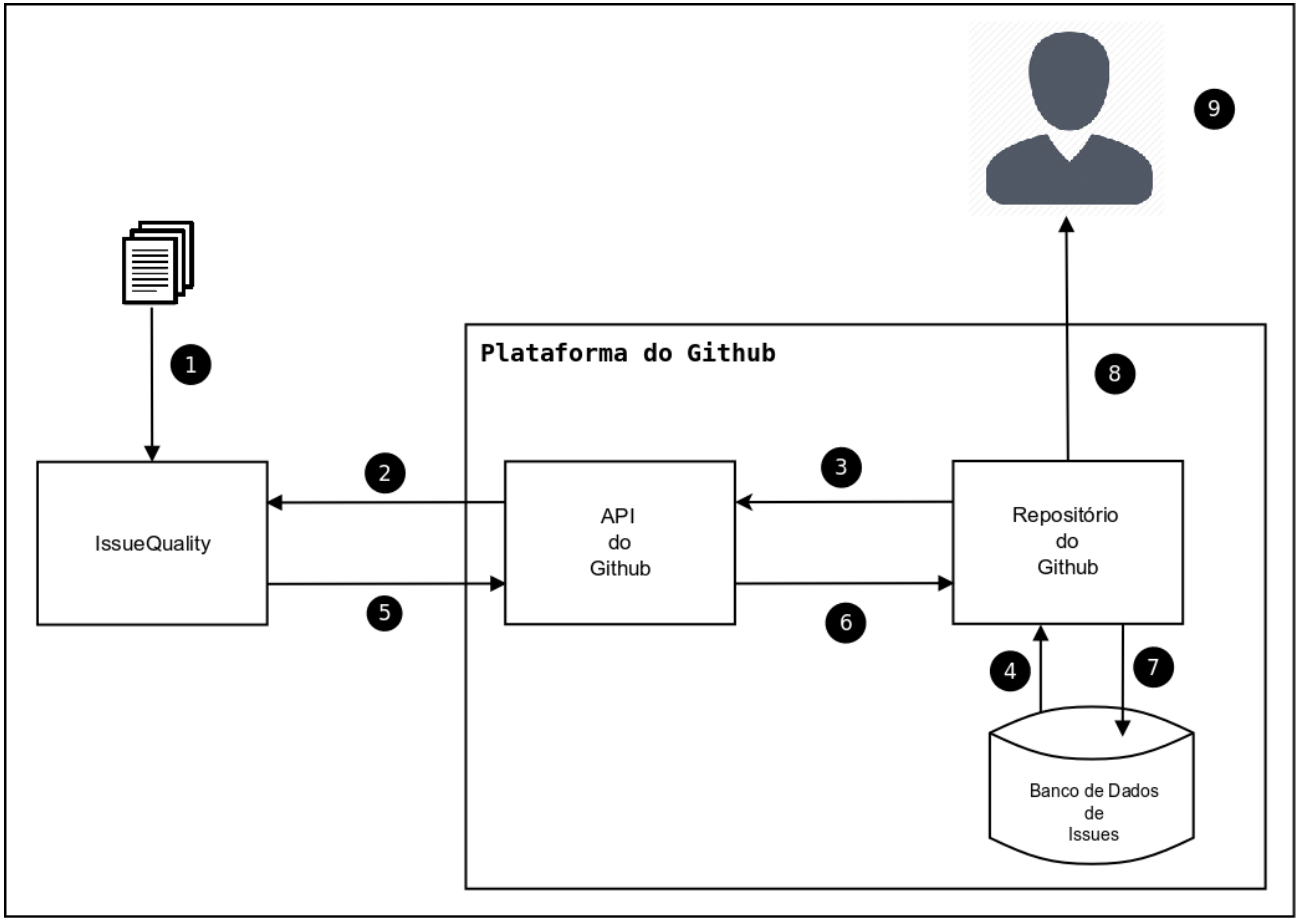


Figura 8: Visão geral do funcionamento da extensão *IssueQuality*

Estudo sobre as funcionalidades das FGRMs

Ferramenta	**Classificação**	**Versão**	URL
Bugzilla	Ferramenta	5.0.3	https://www.bugzilla.org
Mantis Bug Tracker	Ferramenta	1.3.2	https://www.mantisbt.org
Redmine	Ferramenta	3.3.1	http://www.redmine.org/
JIRA Software	Serviço	7.2.4	https://br.atlassian.com/software/jira
Github Issue System	Serviço	-	https://github.com/
Gitlab Issue Tracking System	Serviço	-	https://gitlab.com/

Tabela 4: Ferramentas utilizados no estudo

Estudo sobre as funcionalidades das FGRMs

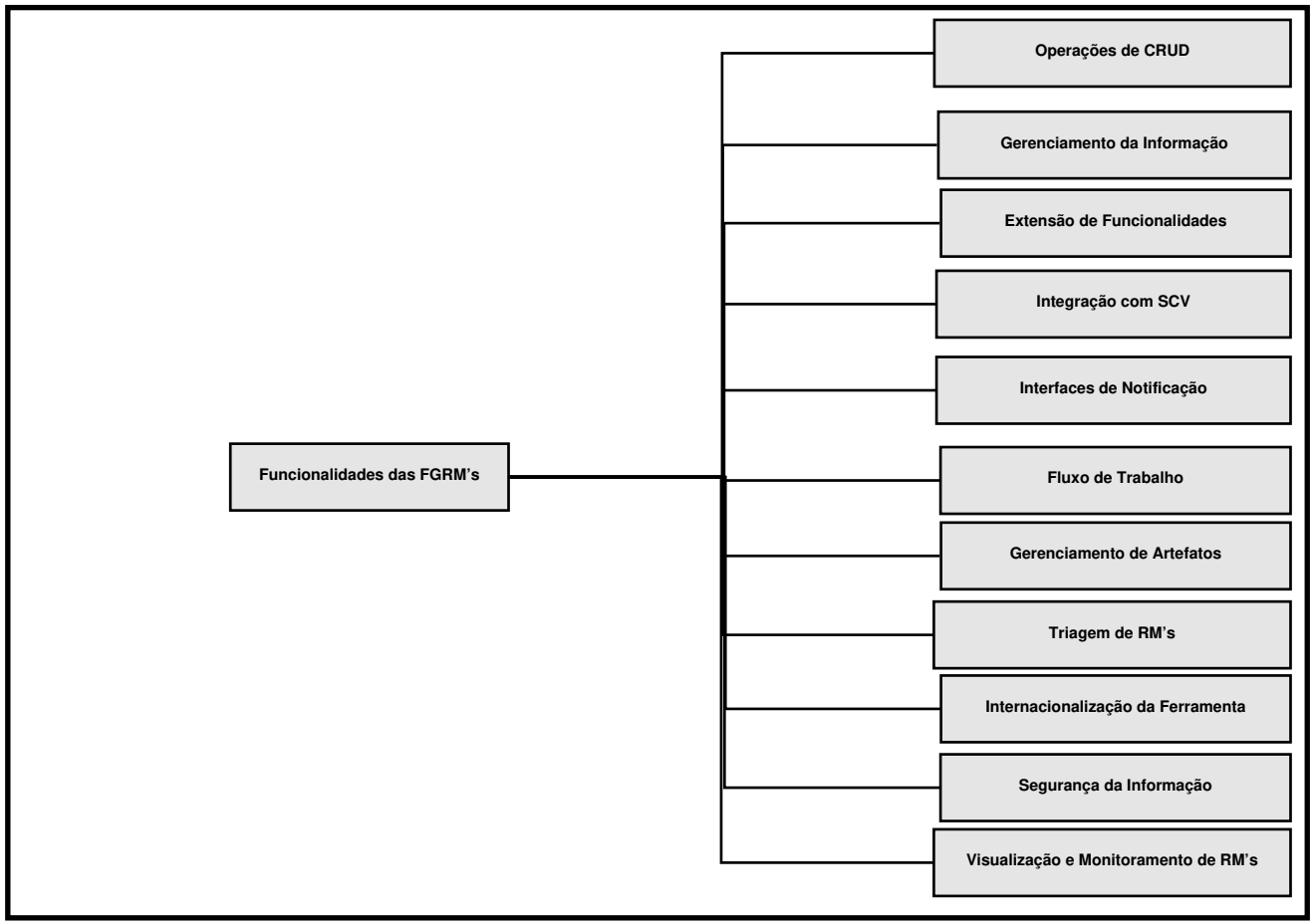


Figura 9: Modelo de funcionalidades básicas das FGRMs

Mapeamento Sistemático da Literatura

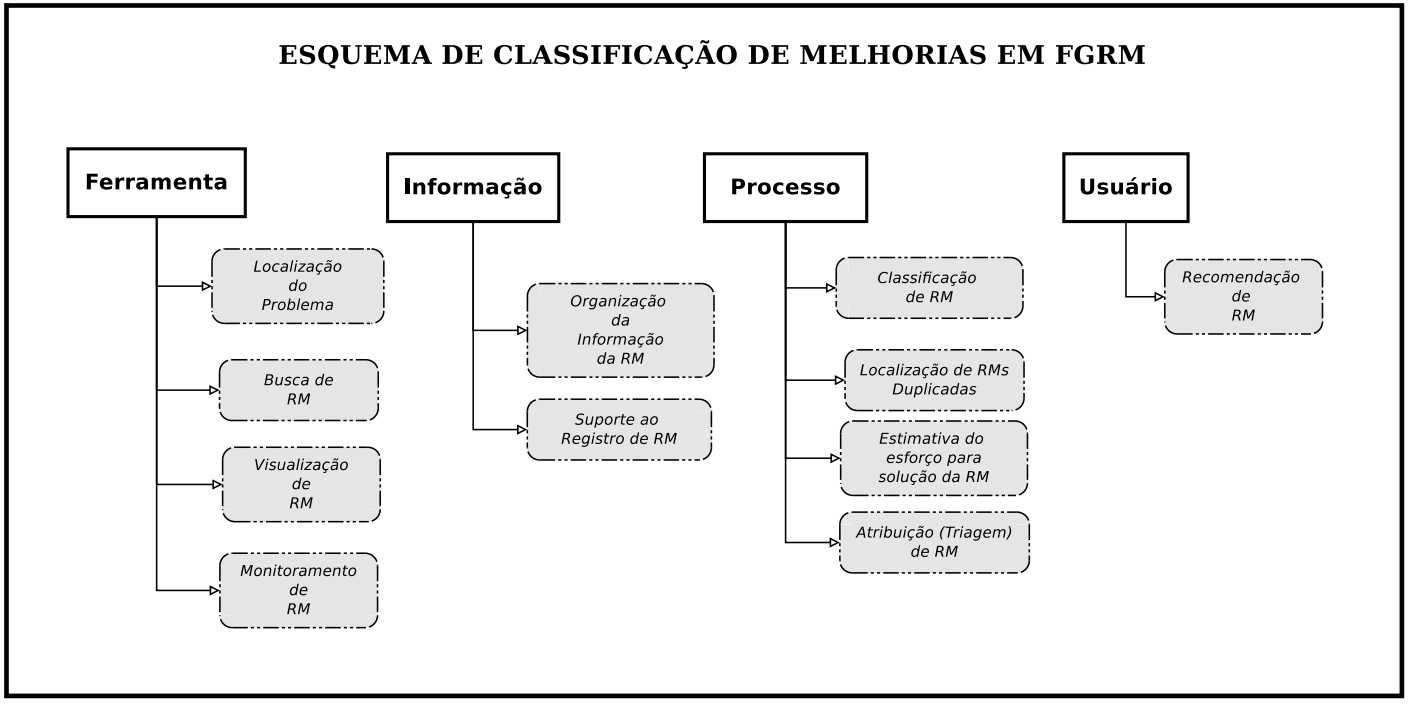


Figura 10: Esquema de classificação das melhorias propostas na literatura. Os retângulos representam as dimensões de melhorias e os polígonos de cantos arredondados representam tópicos de problemas do gerenciamento das RMs.

Mapeamento Sistemático da Literatura

Papel	Total de Artigos
Agente de Triagem	37
Desenvolvedor	26
Analista de Qualidade	13
Gerente de Requisição de Mudança	11
Reportador	6
Líder da Manutenção	4
Todos	3

Tabela 5: Total de artigos por papel na manutenção de software

Levantamento com Desenvolvedores

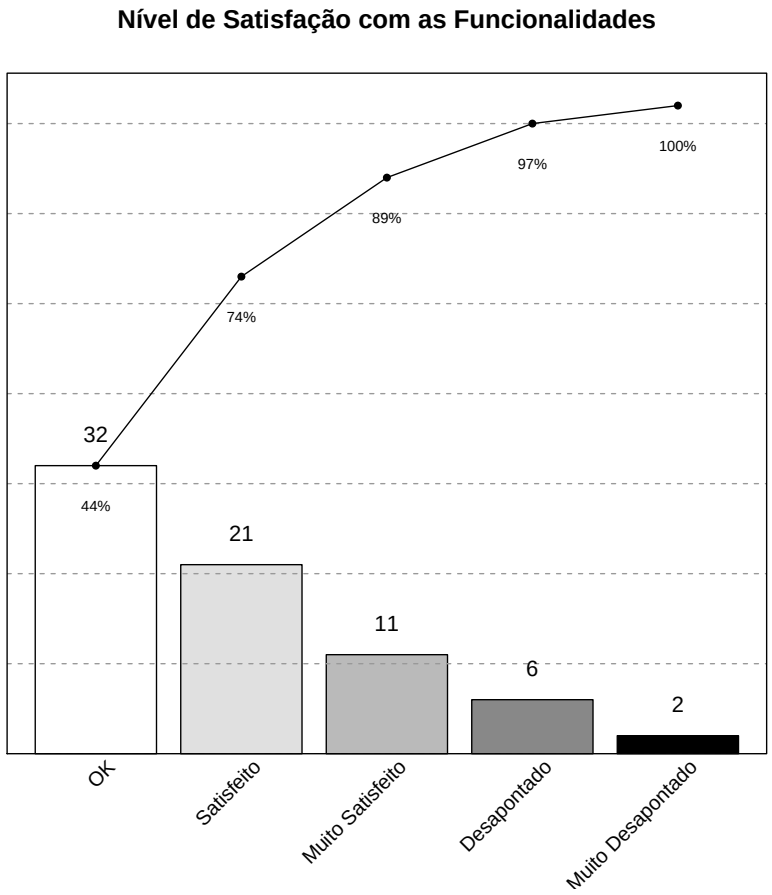


Figura 11: Nível de satisfação com as Ferramentas

Levantamento com Desenvolvedores

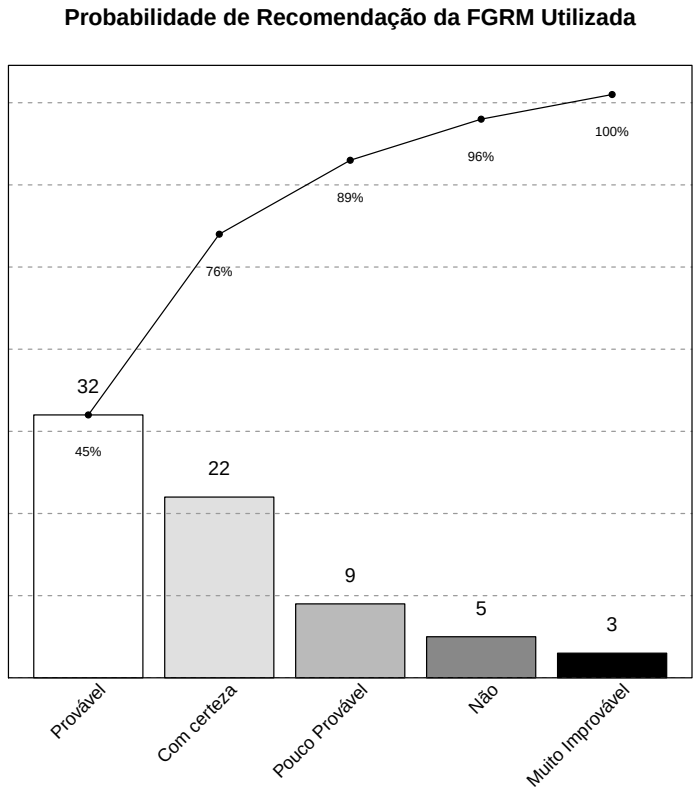


Figura 12: Probabilidade de Recomendação da Ferramenta Utilizada

Levantamento com Desenvolvedores

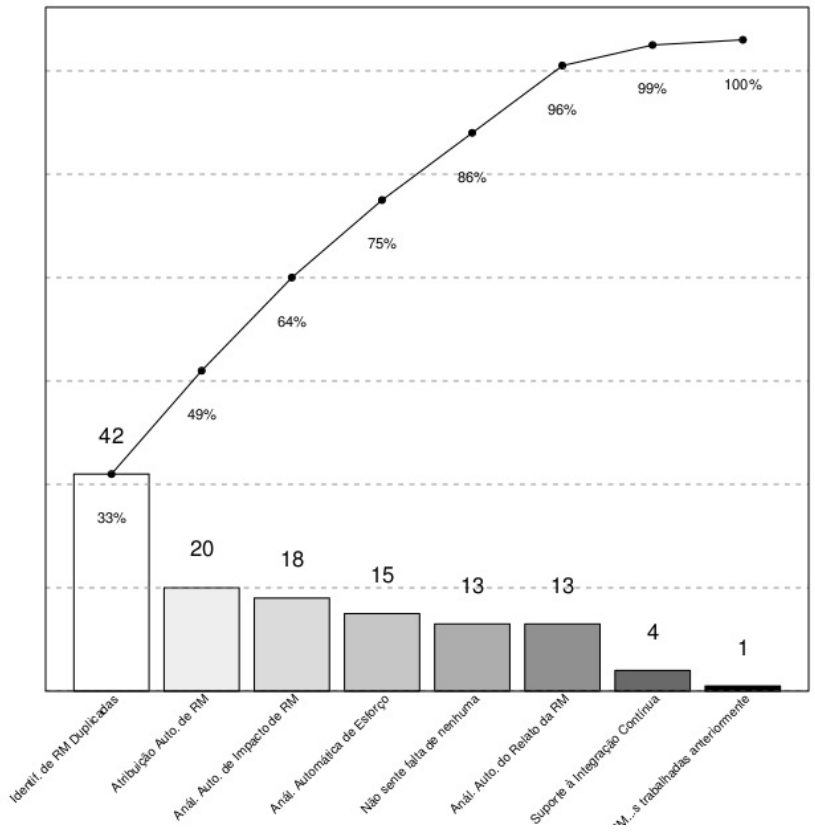


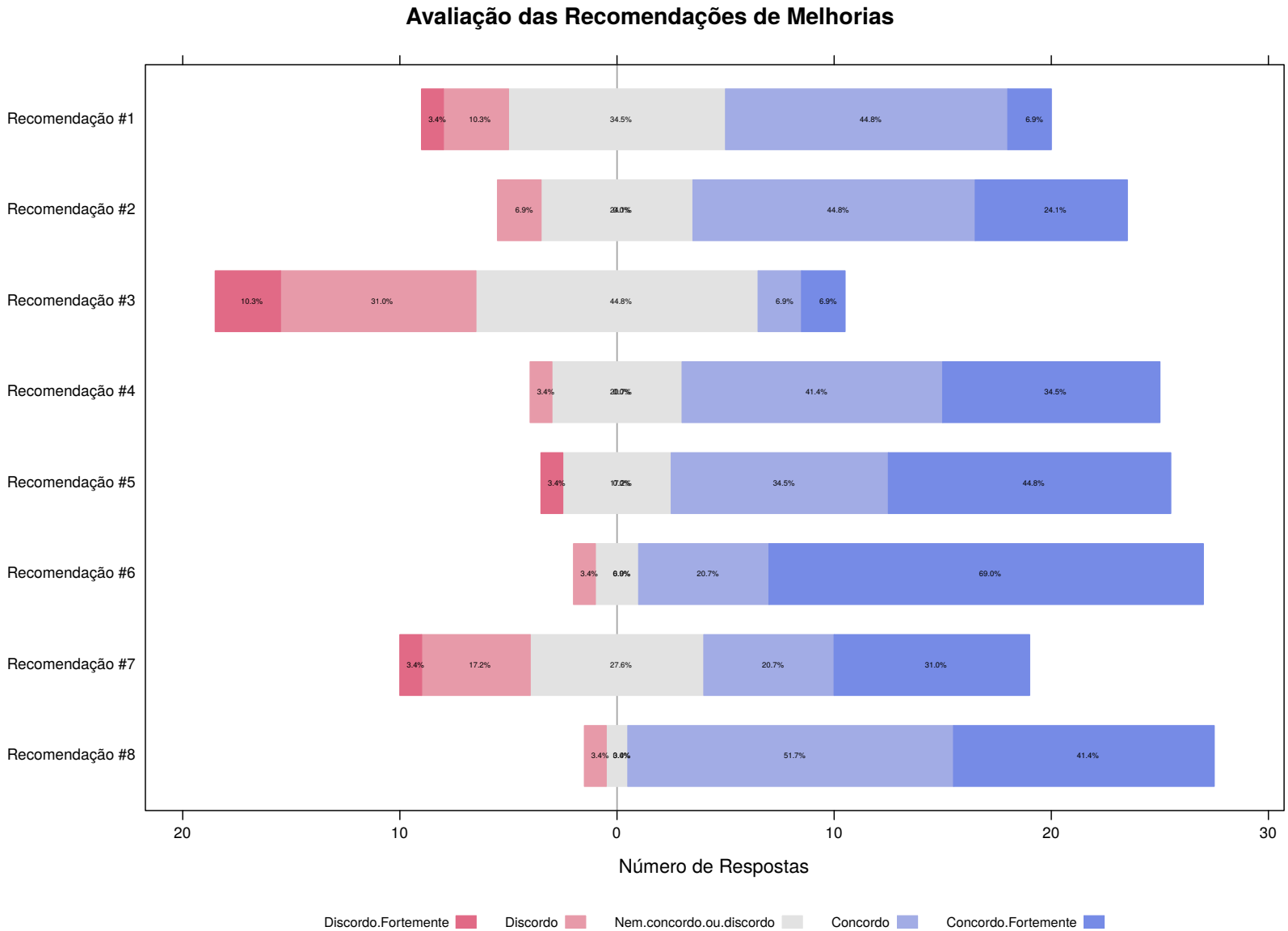
Figura 13: Funcionalidades que o participantes sentem falta.

Levantamento com Desenvolvedores

Melhorias Propostas	**Classificação**
Priorização automatizada de RMs urgentes e inesperadas	1
Sugestão automatizada das RMs que farão parte da iteração.	2
Suporte aos desenvolvedores na preparação para reunião diária	3
Suporte à divisão de tarefas de forma compartilhada	4
Facilitar a propriedade compartilhada de código	5

Tabela 6: Classificação das funcionalidades que possam dar suporte ao uso das metodologias dos agilistas.

Sugestões de Melhorias

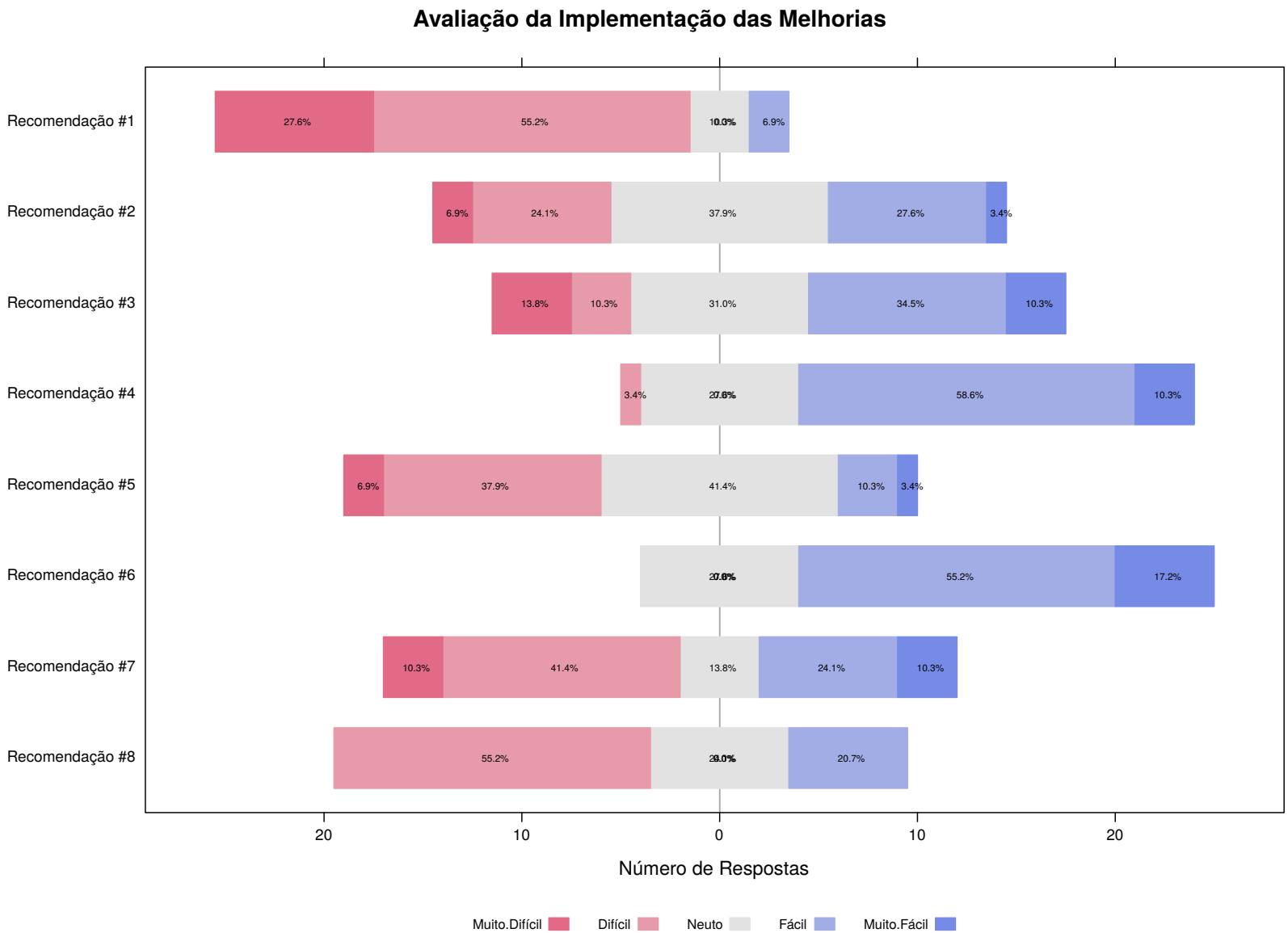


Sugestões de Melhorias

Recomendações	Discordo Fortemente	Discordo	Não concordo e nem discordo	Concordo	Concordo Fortemente	Ranking
Sugestão #6	0	1	2	6	20	45
Sugestão #8	0	1	1	15	12	38
Sugestão #5	1	0	5	10	13	34
Sugestão #4	0	1	6	12	10	31
Sugestão #2	0	2	7	13	7	25
Sugestão #7	1	5	8	6	9	17
Sugestão #1	1	3	10	13	2	12
Sugestão #3	3	9	13	2	2	-9

Tabela 7: Ranking das sugestões propostas

Sugestões de Melhorias



Sugestões de Melhorias

Recomendações	Muito Difícil	Difícil	Neutro	Fácil	Muito Fácil	Ranking
Sugestão #6	0	0	8	16	5	26
Sugestão #4	0	1	8	17	3	22
Sugestão #3	4	3	9	10	3	5
Sugestão #2	2	7	11	8	1	-1
Sugestão #7	3	12	4	7	3	-5
Sugestão #5	2	11	12	3	1	-10
Sugestão #8	0	16	7	6	0	-10
Sugestão #1	8	16	3	2	0	-30

Tabela 8: Ordenamento das sugestões pelo grau de dificuldade.

Estudo sobre as Funcionalidades

- As FGRMs evoluíram da gerência simples de RMs para colaborar no processo de desenvolvimento e manutenção do software. Entretanto, esta evolução não abrange todo tipo de necessidade.
- Seria importante que as FGRMs incorporassem outros comportamentos que ajudem no processo de desenvolvimento e manutenção de software, especialmente em temas como busca de duplicados, melhoria da qualidade do relato e atribuição e classificação automatizadas das RMs.

Mapeamento Sistemático da Literatura

- Prevalência de estudos na dimensão *Processo* especialmente para os tópicos de *Localização de RMs Duplicadas, Atribuição (Triagem) de RMs e Classificação de RMs*, respectivamente.
- Dos estudos que fizeram parte do mapeamento um total de 10 foram implementados como extensões ou protótipos, este número poderia ser maior.

Mapeamento Sistemático da Literatura

- Prevalência de estudos com foco no papel de *Agente de Triagem*. Existe possivelmente uma crença de que é possível melhorar a produtividade do processo de manutenção de software reduzindo o esforço de encontrar o desenvolvedor mais apto.
- As FGRMs deveriam dar suporte ao Reportador que, na maioria da vezes, é o primeiro a registrar as informações que serão necessárias à solução da RM.

Levantamento com Desenvolvedores

- Em geral, o nível de satisfação com as funcionalidades oferecidas pelas FGRMs é alto.
- As funcionalidades que os participantes mais sentiram falta, também representam a maior quantidade de estudos na literatura.
- As FGRMs poderiam oferecer suporte às praticas propostas pelos agilistas.

Sugestões de Melhorias

- Em geral podemos considerar que as sugestões propostas tiveram uma boa aceitação dos participantes.
- Com relação às recomendações propostas, verificamos que a utilização de uma linguagem além do texto simples, como por exemplo o Markdown, foi muito bem aceita.

Sugestões de Melhorias

- O suporte à tarefas compartilhadas (sugestão #8) também foi muito bem aceita.
- Por outro lado, as sugestões que têm algum tipo de relação com a interface das FGRMs (sugestões #6, #4, #3 e #2) foram consideradas como mais “fácil” de implementar.

Estudo sobre as Funcionalidades

- Uma ameaça à validade do trabalho está no processo de seleção das ferramentas.
- Como a extração dos dados dos Cartões foi realizada de forma manual pode ter ocorrido algum tipo de equívoco no processo, como por exemplo a não coleta de algum dado de determinada ferramenta por algum descuido.
- A classificação dos cartões pode ter ocorrido uma classificação de forma incorreta o que pode acarretar em limitação dos resultados apresentados.

Mapeamento Sistemático da Literatura

- É possível que as perguntas de discussões com membros do projeto e especialistas em Manutenção de Software foram realizadas para validar as perguntas.
- A forma que as sentenças de busca foram estruturadas pode não ser a mais otimizada para pesquisa do maior número de documentos relevantes.

Mapeamento Sistemático da Literatura

- É possível que as perguntas de discussões com membros do projeto e especialistas em Manutenção de Software foram realizadas para validar as perguntas.
- A forma que as sentenças de busca foram estruturadas pode não ser a mais otimizada para pesquisa do maior número de documentos relevantes.

Levantamento com Desenvolvedores

- Uma ameaça à validade deste trabalho está no número de respondentes da pesquisa.
- A amostragem de conveniência implica que as generalizações são limitadas já que a amostra pode não representar a população.
- Não temos garantias que as regras para seleção de participantes resultaram em um conjunto bem representativo da população.

Sugestões de Melhorias

- O total de participantes não nos permite extrapolar os resultados para todos os contextos em que as FGRMs estão inseridas.
- A utilização de apenas projetos públicos hospedados no Github pode ter causado algum tipo de direcionamento, como por exemplo foco em projetos de código aberto.
- A estrutura das perguntas do formulário pode ter causado impacto na quantidade de respostas ou na opção escolhida pelos participantes.

Conclusão e Trabalhos Futuros

- A contribuição deste trabalho de dissertação está na proposição de melhorias para as FGRMs tomando como base a literatura da área e a opinião de profissionais envolvidos em Manutenção de Software.
- Em algumas plataformas, tais como o Github e o Gitlab, foi possível perceber a tendência em que não existe uma clara separação entre o gerenciamento das RMs e o controle de versão do código.

Conclusão e Trabalhos Futuros

- Foi possível verificar um desacoplamento entre as necessidades dos desenvolvedores e o que está sendo proposto na literatura.
- Percebemos que os profissionais consultados estão satisfeitos com as funcionalidades oferecidas. Todavia, a nossa visão é que existem muitos outros comportamentos que poderiam ser acoplados a este tipo de software de modo a melhorar as atividades de manter e evoluir um software.

Conclusão e Trabalhos Futuros

- As metodologias propostas pelos agilistas vêm sendo adotadas por algumas equipes de manutenção de software. As FGRMs poderiam implantar funcionalidades com o objetivo de suportar algumas destas práticas.
- Entendemos que seria importante a condução de um novo trabalho com o objetivo de descrever e avaliar os papéis realizados no processo de manter um software.

Conclusão e Trabalhos Futuros

- Entendemos que seria importante a realização de um estudo com o objetivo de melhorar a organização dos conceitos da área de Manutenção de Software, em especial sobre as RMs e FGRMs.
- O processo de criação de RMs poderia ser melhorado com a utilização de uma interface que utilize um *chatbot* [17, 8], permitindo a criação iterativa de uma RM.

Dúvidas?

References I

[1] O. Baysal and R. Holmes, “A Qualitative Study of Mozillas Process Management Practices,” *David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada, Tech. Rep. CS-2012-10*, 2012.

[2] O. Baysal, R. Holmes, and M. W. Godfrey, “Situational awareness: Personalizing issue tracking systems,” in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE ’13. Piscataway, NJ,

References II

USA: IEEE Press, 2013, pp. 1185–1188. [Online]. Available: http://dl.acm.org.ez27.periodicos.capes.gov.br/citation.cfm?id=2486788.2486957

[3] D. Bertram, A. Voida, S. Greenberg, and R. Walker, “Communication, collaboration, and bugs: The social nature of issue tracking in small, collocated teams,” in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW ’10. New York, NY, USA: ACM, 2010, pp. 291–300. [Online].

References III

Available: http://doi.acm.org/10.1145/1718918.1718972

[4] N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann, “What makes a good bug report?” in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*. ACM, 2008, pp. 308–318.

References IV

[5] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: Improving cooperation between developers and users," in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '10. New York, NY, USA: ACM, 2010, pp. 301–310. [Online]. Available: http://doi.acm.org/10.1145/1718918.1718973

References V

[6] Y. C. Cavalcanti, P. A. d. M. S. Neto, D. Lucrédio, T. Vale, E. S. de Almeida, and S. R. de Lemos Meira, "The bug report duplication problem: an exploratory study," *Software Quality Journal*, vol. 21, no. 1, pp. 39–66, 2013.

[7] R. M. de Mello, P. C. da Silva, P. Runeson, and G. H. Travassos, "Towards a framework to support large scale sampling in software engineering surveys," in *Proceedings of the 8th ACM/IEEE International Symposium on*

References VI

Empirical Software Engineering and Measurement. ACM, 2014, p. 48.

[8] J. Huang, M. Zhou, and D. Yang, “Extracting chatbot knowledge from online discussion forums.” in *IJCAI*, vol. 7, 2007, pp. 423–428.

[9] IEEE, “IEEE Standard Glossary of Software Engineering Terminology,” *IEEE Std 610.12-1990*, pp. 1–84, Dec 1990.

References VII

[10] ISO/IEC, “International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering 2013; Software Life Cycle Processes 2013; Maintenance,” *ISO/IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998)*, pp. 01–46, 2006.

References VIII

[11] S. Just, R. Premraj, and T. Zimmermann, "Towards the next generation of bug tracking systems," in *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2008, pp. 82–85.

[12] U. Kaur and G. Singh, "A review on software maintenance issues and how to reduce maintenance efforts," *International Journal of Computer Applications*, vol. 118, no. 1, 2015.

References IX

[13] S. Keele, “Guidelines for performing systematic literature reviews in software engineering,” in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, 2007.

[14] O. Kononenko, O. Baysal, R. Holmes, and M. W. Godfrey, “Dashboards: Enhancing developer situational awareness,” in *Companion Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE Companion 2014. New York, NY, USA: ACM,

References X

2014, pp. 552–555. [Online]. Available: http://doi.acm.org.ez27.periodicos.capes.gov.br/10.1145/2591062.2591075

[15] B. P. Lientz and E. B. Swanson, *Software Maintenance Management*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1980.

References XI

[16] N. A. Maiden and G. Rugg, “Acre: selecting methods for requirements acquisition,” *Software Engineering Journal*, vol. 11, no. 3, pp. 183–192, 1996.

[17] M. L. Mauldin, “Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition,” in *AAAI*, vol. 94, 1994, pp. 16–21.

References XII

[18] S. McGee and D. Greer, “A software requirements change source taxonomy,” in *Software Engineering Advances, 2009. ICSEA’09. Fourth International Conference on*. IEEE, 2009, pp. 51–58.

[19] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” *EASE’08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pp.

References XIII

68–77, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=2227115.2227123

[20] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2015.03.007

References XIV

[21] H. Rocha, G. Oliveira, H. Marques-Neto, and M. T. Valente, “Nextbug: a bugzilla extension for recommending similar bugs,” *Journal of Software Engineering Research and Development*, vol. 3, no. 1, 2015. [Online]. Available: http://dx.doi.org/10.1186/s40411-015-0018-x

[22] N. Serrano and I. Ciordia, “Bugzilla, itracker, and other bug trackers,” *IEEE Software*, vol. 22, no. 2, pp. 11–13, March 2005.

References XV

[23] V. Singh and K. K. Chaturvedi, “Bug tracking and reliability assessment system (btras),” *International Journal of Software Engineering and Its Applications*, vol. 5, no. 4, pp. 1–14, 2011.

[24] F. Thung, T.-D. B. Le, P. S. Kochhar, and D. Lo, “Buglocalizer: Integrated tool support for bug localization,” in *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE

References XVI

2014. New York, NY, USA: ACM, 2014, pp. 767–770. [Online]. Available: http://doi.acm.org.ez27.periodicos.capes.gov.br/10.1145/2635868.2661678

[25] P. Tripathy and K. Naik, *Software Evolution and Maintenance*. Wiley, 2015. [Online]. Available: https://books.google.com.br/books?id=0UXxBQAAQBAJ

References XVII

[26] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[27] T. Zimmermann, R. Premraj, J. Sillito, and S. Breu, "Improving bug tracking systems." in *ICSE Companion*. Citeseer, 2009, pp. 247–250.