

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

- ▶ 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 [7], 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* [5].
- ▶ *Manutenibilidade*: propriedade de um sistema ou componente de software em relação ao grau de *facilidade* que ele pode ser corrigido, melhorado ou adaptado [5].

Tipos de Manutenção em Software

- ▶ A manutenção de software pode ser dividida em *Corretiva*, *Adaptativa*, *Perfectiva* e *Preventiva* [9, 5].
- ▶ A *ISO 14764* [6] propõe que exista um elemento denominado Requisição de Mudança (RM) que corresponde a uma agregação de características que representem uma solicitação de manutenção de qualquer das quatro categorias.

Tipos de Manutenção em Software

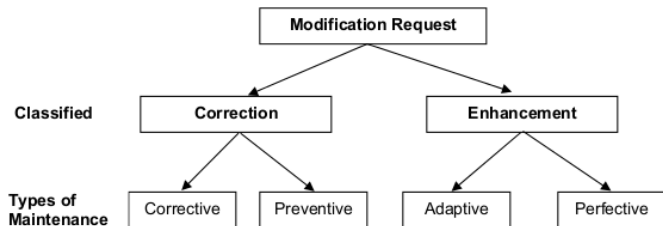


Figura 1: Tipos de manutenção segundo a norma ISO/IEC 14764 [6]

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. Posteriormente cabe ao profissional que cumpre este papel encaminhar a RM para o Agente de Triage.
- ▶ *Agente de Triage (Scheduler)*: Deve planejar a fila de RMs e atribuí-las para o desenvolvedor mais apto. A decisão pode considerar a carga de trabalho existente

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 [14]

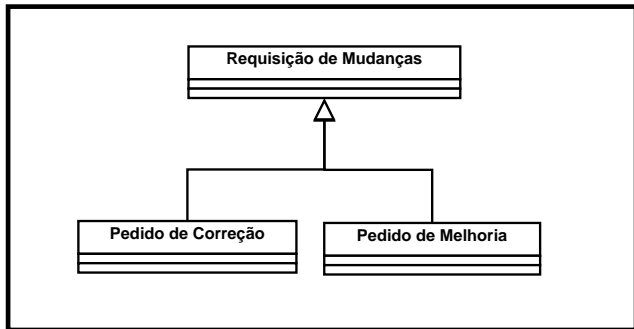


Figura 2: Modelo conceitual de uma Requisição de Mudanças

Atributos de uma RM

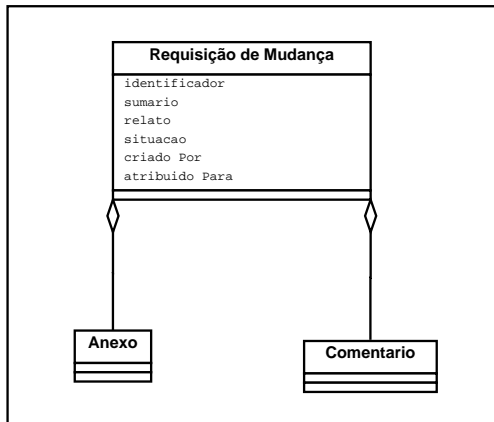


Figura 3: Informações que compõem uma RM. Baseado em trabalho de Singh & Chaturvedi [12]

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

12

Ciclo de Vida da RM

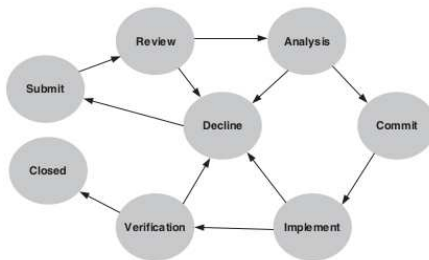


Figura 5: Diagrama de estados de uma RM. Extraído de [14]

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 [11].

Ferramentas de Gerenciamento de Requisição de Mudança



Mais do que Gerenciar RMs

- ▶ Ponto central para a comunicação e coordenação das diversas partes interessadas [2].
- ▶ Possibilita que os usuários participem do processo de solução das RMs [3].
- ▶ Suporte para atividades como [4]:
 - ▶ estimativa do custo do software
 - ▶ análise do impacto de uma modificação
 - ▶ planejamento do projeto
 - ▶ rastreabilidade de uma falha
 - ▶ extração de conhecimento

Problema

- ▶ A utilização de “demanda” parece estar distante das necessidades práticas dos projetos, especialmente no ponto de vista dos desenvolvedores [1].
- ▶ Apesar da crescente importância dos SCD's estes sistemas aparentemente não estão atendendo as necessidades dos diversos interessados (stakeholders)
- ▶ Diversas extensões (plugins) estão sendo propostas na literatura [10, 13, 8].

Objetivos

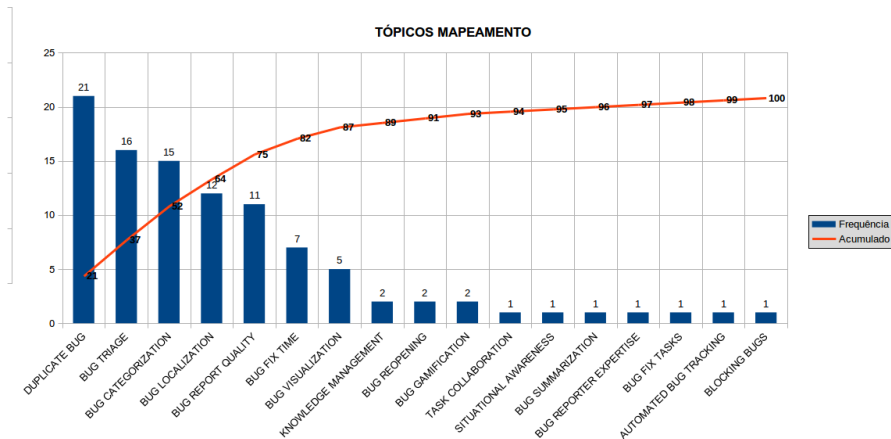
- ▶ Estudar as Ferramentas de Suporte de Problemas de Software visando:
 - ▶ *Entender* os requisitos básicos deste tipo de ferramenta
 - ▶ *Mapear* as extensões para FSPS que estão sendo propostas na literatura
 - ▶ *Avaliar* sobre o ponto de vista dos profissionais envolvidos em manutenção a situação atual dos FSPS
 - ▶ *Propor* novas extensões para os FSPS

Metodologia

- ▶ Mapeamento Sistemático da Literatura [?]
- ▶ Caracterização de Requisitos das Ferramentas de Suporte de Problemas de Software (FSPS)
- ▶ Pesquisa (Survey) com os desenvolvedores [15]
- ▶ Desenvolvimento de extensões para as FSPS's

Mapeamento Sistemático da Literatura

► Situação: Feito



Caracterização de Requisitos das FSPS

- ▶ Situação: Em andamento
- ▶ Resultados Parciais:
 - ▶ CRUD dos problemas
 - ▶ CRUD de interessados (desenvolvedores, usuários, gerentes de projeto e etc)
 - ▶ Regras relacionando Problemas vs Interessados
 - ▶ Classificação (Prioridade e Severidade)

Survey com os desenvolvedores

- ▶ Situação: Em andamento
- ▶ Atividades:
 - ▶ Planejamento do Survey (Feito)
 - ▶ Ferramenta de Coleta (Em andamento)
 - ▶ “Survey Piloto” (Em andamento)

Discussão

Ameças à Validade

Conclusão e Trabalhos Futuros

Dúvidas?



References I

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

References II

- [2] D. Bertram, A. Voids, S. Greenberg, and R. Walker, “Communication, collaboration, and bugs: The social nature of issue tracking in small, colocated 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]. Available: <http://doi.acm.org/10.1145/1718918.1718972>

References III

- [3] 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 IV

- [4] 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.
- [5] IEEE, “IEEE Standard Glossary of Software Engineering Terminology,” *IEEE Std 610.12-1990*, pp. 1–84, Dec 1990.

References V

- [6] 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.
- [7] 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 VI

- [8] 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, 2014, pp. 552–555. [Online]. Available: <http://doi.acm.org.ez27.periodicos.capes.gov.br/10.1145/2591062.2591075>

References VII

- [9] B. P. Lientz and E. B. Swanson, *Software Maintenance Management*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1980.
- [10] 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].

References VIII

Available:

<http://dx.doi.org/10.1186/s40411-015-0018-x>

- [11] N. Serrano and I. Ciordia, “Bugzilla, itracker, and other bug trackers,” *IEEE Software*, vol. 22, no. 2, pp. 11–13, March 2005.
- [12] 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.

References IX

- [13] 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 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>

References X

- [14] P. Tripathy and K. Naik, *Software Evolution and Maintenance*. Wiley, 2015. [Online]. Available: <https://books.google.com.br/books?id=0UXxBQAAQBAJ>
- [15] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.