

Um estudo de ferramentas de Suporte de Problemas de Software

Vagner Clementino¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)

vagnercs@dcc.ufmg.br

Dentro do ciclo de vida do software o processo de manutenção tem papel fundamental. Devido ao seu alto custo, em alguns casos chegando a 60% do total [Kaur and Singh 2015], sua importância vêm sendo considerada tanto pela comunidade científica quanto pela indústria. A manutenção em software podem ser dividida em *Corretiva, Adaptativa, Perfectiva e Preventiva* [Lientz and Swanson 1980, IEEE 1990]. A *ISO 14764* [ISO/IEC 2006] agrupa todos estes nomes em único termo denominado *Requisição de Mudança - Modification Request (RM)*.

Em um ambiente real de manutenção de software existe a necessidade de gerenciar as Requisição de Mudança (RM), especialmente por conta do seu volume. Esse controle é realizado por Sistemas de Controle de Demandas (SCD)- Issue Tracking Systems que ajudam os desenvolvedores na correção de forma individual ou colaborativa de defeitos (bugs) e no suporte à implementação de novas funcionalidades. Verifica-se na literatura diversos sinônimos para os Sistemas de Controle de Demanda (Sistema de Controle de Defeitos - Bug Tracking Systems, Sistema de Gerenciamento da Requisição - Request Management System e outros), todavia, de modo geral, o termo se refere as ferramentas utilizadas pelas organizações para *suporte de problemas de software*.

Diante da maior presença de software em todos os setores da sociedade se faz necessário o desenvolvimento de processos, técnicas e ferramentas que reduzam o esforço e o custo do desenvolvimento manutenção do software. Neste linha o trabalho de Yong & Mookerjee [Tan and Mookerjee 2005] propõe reduzir o custos de manutenção e reposição durante a vida útil de um sistema de software. O modelo proposto demonstrou quem em algumas situações é *melhor substituir um sistema do que mantê-lo*.

Neste contexto, os Sistemas de Controle de Demandas(SCD) vêm crescendo em importância tendo em vista sua utilização por gestores, analistas da qualidade e usuários finais para atividades tais como tomada de decisão e comunicação, dentre outras. Não obstante, a utilização de “demanda” como conceito central para as ferramentas de suporte de problemas de software 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 do SCD’s com a necessidade de seus usuários pode ser visto no trabalho proposto por Baysal & Holme [Baysal and Holmes 2012] no qual desenvolvedores que utilizam o Bugzilla¹ relatam a dificuldade em manter uma compreensão global das RM’s em que eles estão envolvidos. Segundo os desenvolvedores seria interessante que a ferramenta tivesse um suporte melhorado para a Consciência Situacio-

¹<https://www.bugzilla.org>

nal - Situational Awareness, ou seja, eles gostariam de estar cientes da situação global do projeto bem como das atividades que outras pessoas estão realizando.

Neste ponto temos a seguinte situação: apesar da crescente utilização e importância dos SCD's estes sistemas aparentemente não estão atendendo as necessidades dos diversos interessados(stakeholders) do processo de manutenção de software. Um sinal da necessidade de evolução do SCD's pode ser observado com base nas diversas extensões (plugins) propostos na literatura [Rocha et al. 2015, Thung et al. 2014b, Kononenko et al. 2014].

Neste contexto, é proposto um estudo sobre os Sistemas de Controle de Demandas no qual se discutirá os aspectos que são considerados mais importantes do ponto de vista da literatura da área bem como a partir do ponto de vista de profissionais envolvidos com tarefas de manutenção de software. De forma particular vamos estudar os mecanismos de personalização que algumas destas ferramentas permitem e tentaremos ainda criar exemplos de personalização para alguma possível extensão a ser identificada ao longo do trabalho.

O desenvolvimento de novas funcionalidades em SCD's, mediante a capacidade de extensão propiciada por alguns deles vêm sendo explorada na literatura. *Buglocalizer* [Thung et al. 2014b] é uma extensão para o Bugzilla que possibilita a localização dos arquivos do código fonte que estão relacionados ao defeito relatado. *NextBug* [Rocha et al. 2015] é uma extensão para o Bugzilla que recomenda novos bugs para um desenvolvedor baseado no defeito que ele esteja tratando atualmente. No trabalho de Kononenko et al. [Kononenko et al. 2014] é apresentada uma ferramenta denominada *DASH* cujo objetivo é agrupar as demandas que são relevantes para as atividades de um desenvolvedor. Na ferramenta proposta por Thung et al. [Thung et al. 2014a] o foco é na determinação de defeitos duplicados. A contribuição deste trabalho é a integração do estado da arte das técnicas não supervisionadas para detecção de falhas duplicadas conforme proposto por Runeson et al.[Runeson et al. 2007].

A Manutenção de Software é um processo complexo e caro que merece atenção da comunidade acadêmica e da indústria no desenvolvimento de técnicas, processo e ferramentas que reduzam o seu custo e o esforço envolvido. Neste contexto, os Sistemas de Controle de Demanda desempenha um papel fundamental que ultrapassa a simples função de registrar falhas em software. Neste sentido é proposto um estudo sobre os Sistemas de Controle de Demandas no qual se discutirá os aspectos que são considerados mais importantes do ponto de vista da literatura da área bem como a partir do ponto de vista de profissionais envolvidos com tarefas de manutenção de software. Este estudo pretende obter os seguintes resultados:

- (i) Caracterização dos Sistemas de Controle de Demandas com base nos requisitos comum a todos eles;
- (ii) Mapeamento das propostas de extensões existente na literatura;
- (iii) Coleta da opinião dos profissionais envolvidos em tarefas de manutenção sobre os problemas e soluções oferecidas por este tipo de ferramenta
- (iv) Desenvolvimento de uma extensão a ser identificada ao longo do trabalho

Referências

- Baysal, O. and Holmes, R. (2012). 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*.
- Baysal, O., Holmes, R., and Godfrey, M. W. (2013). Situational awareness: Personalizing issue tracking systems. In *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, pages 1185–1188, Piscataway, NJ, USA. IEEE Press.
- IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, pages 1–84.
- ISO/IEC (2006). 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*, pages 01–46.
- Kaur, U. and Singh, G. (2015). A review on software maintenance issues and how to reduce maintenance efforts. *International Journal of Computer Applications*, 118(1).
- Kononenko, O., Baysal, O., Holmes, R., and Godfrey, M. W. (2014). Dashboards: Enhancing developer situational awareness. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014*, pages 552–555, New York, NY, USA. ACM.
- Lientz, B. P. and Swanson, E. B. (1980). *Software Maintenance Management*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Rocha, H., Oliveira, G., Marques-Neto, H., and Valente, M. (2015). Nextbug: a bugzilla extension for recommending similar bugs. *Journal of Software Engineering Research and Development*, 3(1).
- Runeson, P., Alexandersson, M., and Nyholm, O. (2007). Detection of duplicate defect reports using natural language processing. In *Proceedings of the 29th International Conference on Software Engineering, ICSE '07*, pages 499–510, Washington, DC, USA. IEEE Computer Society.
- Tan, Y. and Mookerjee, V. (2005). Comparing uniform and flexible policies for software maintenance and replacement. *Software Engineering, IEEE Transactions on*, 31(3):238–255.
- Thung, F., Kochhar, P. S., and Lo, D. (2014a). Dupfinder: Integrated tool support for duplicate bug report detection. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, ASE '14*, pages 871–874, New York, NY, USA. ACM.
- Thung, F., Le, T.-D. B., Kochhar, P. S., and Lo, D. (2014b). Buglocalizer: Integrated tool support for bug localization. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2014*, pages 767–770, New York, NY, USA. ACM.