

Um Estudo Exploratório Sobre Falhas de Softwares

Patrick Helder Alvarenga Belém¹, Adriana S Vivacqua¹

¹Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

patrickbelem@ufrj.com ; avivacqua@dcc.ufrj.br

Abstract. *The society has become increasingly dependent on technology, especially software. Thus, it's natural that sometimes there is a need for "maintenance", either by adapting, expanding or even correcting failures due to project development. This paper aims to investigate software failures and how developer teams have dealt with them. It's intended to answer some questions that allow us to understand how teams deal with failures in real environments, if there is learning from failures and how it's propagated to other team members. An exploratory study will be conducted on the topic with the following steps: defining the scope and group of developers for research, locally observing how work in groups takes place, conducting interviews with developers, analyzing their responses and comparing results with other related works.*

Resumo. *A sociedade tem se tornado cada vez mais dependente da tecnologia, sobretudo de softwares. Dessa forma, é natural que em algum momento haja necessidade de "manutenção", seja por adaptação, ampliação ou até mesmo correção de falhas decorridas do desenvolvimento do projeto. O objetivo do presente trabalho é investigar as falhas de software e como as equipes de desenvolvedores têm lidado com as mesmas. Pretende-se responder algumas questões que seja permitido entender como as equipes lidam com as falhas em ambientes reais, se existe aprendizado a partir das falhas e, ainda como o mesmo é propagado para os demais membros da equipe. Será realizado um estudo exploratório sobre o tema com as seguintes etapas: definir o escopo e grupo de desenvolvedores para investigação, observar localmente como ocorre o trabalho nos grupos, conduzir entrevistas com os desenvolvedores, analisar as respostas dos mesmos e comparar os resultados com outros trabalhos relacionados.*

1. Contexto do Trabalho

Atualmente é praticamente impossível realizar tarefas cotidianas sem lidar com algum tipo software. Estamos rodeados de artefatos tecnológicos diariamente: negócios, entretenimento, lazer, comunicação, etc., onde os mesmos estão calcados em sistemas computacionais na grande maioria dos casos. Infraestruturas e serviços nacionais são controlados por sistemas computacionais, e a maioria dos produtos elétricos inclui um computador e um software que o controla. A manufatura e a distribuição industrial são totalmente informatizadas, assim como o sistema financeiro. A área de entretenimento,

incluindo a indústria da música, jogos de computador, cinema e televisão, faz uso intensivo de software (SOMMERVILLE, 2011).

Segundo Pressman e Maxin (2016) o software é a tecnologia mais importante no cenário mundial e também um ótimo exemplo da lei das consequências não intencionais, pois há 60 anos ninguém poderia prever que o softwares se tornariam uma tecnologia indispensável para negócios, ciências e engenharia e etc. E que ainda, o software evoluiria a ponto de modificar tudo ao seu redor: de pesquisa em bibliotecas a compras feitas pelos consumidores, de discursos políticos a comportamentos de namoro entre jovens e adultos não tão jovens. O autor ainda afirma que juntamente a dependência dos softwares, não era possível prever que milhões de softwares teriam que ser corrigidos, adaptados e ampliados e que a realização dessas atividades de “manutenção” absorveriam mais pessoas e recursos do que todo o esforço aplicado na criação de um novo software. Diante disso, a sociedade tem se tornado cada vez mais dependente de softwares, onde uma pequena falha de funcionamento pode acarretar numa grande catástrofe.

Falhas em softwares ocorrem com frequência, em diversas situações, domínios, contextos. Dwivedi et al.(2015) cita alguns exemplos de falhas de softwares, contidas no trabalho de Nelson (2007) e (Koch 2007;2002), que levaram a consequências terríveis para as organizações em termos financeiros, tais como o fracasso da Hewlett-Packard (HP) em 2004 que teve um impacto financeiro de US\$ 160 milhões e a falha da Hershey Foods que fez com que os valores de suas ações caíssem em 8%.

Por outro lado, uma falha é uma grande oportunidade de aprendizado e pode fazer com que casos semelhantes não ocorram no futuro. Segundo Nelson (2007), estatisticamente dois em cada três projetos de TI falham e, como fracasso ensina mais do que o sucesso, o estudo destas falhas poderia levar a melhorias nos processos e na gerência destes.

2. Problema Investigado

O objetivo deste trabalho é realizar um estudo exploratório sobre falhas de software e como as equipes de desenvolvimento de software têm lidado com as mesmas. Partimos do pressuposto que não existe uma cultura de “aproveitamento de falhas” ou que o aprendizado originado delas não seja documentado ou, ainda, divulgado durante ou depois da finalização dos projetos. Acreditamos ainda que as empresas/organizações não valorizam o conhecimento que as falhas podem fornecer.

Com isso em mente, pretendemos investigar algumas questões:

- Como as equipes de desenvolvimento de softwares lidam com as falhas/erros em ambientes reais?
- Será que os desenvolvedores aprendem com as falhas uns dos outros?
- Os desenvolvedores aprendem com falhas globais nos projetos em que estão envolvidos? E em outros projetos?
- O aprendizado a partir das falhas é documentado? Ou ainda propagado para todos os desenvolvedores?

3. Fundamentação e Trabalhos Relacionados

As falhas de software têm sido objeto de estudo de vários autores, onde cada um deles destaca a importância de investigá-las e adquirir conhecimento a partir delas. De acordo com o trabalho de Dwivedi et al.(2015) estudos focados nas falhas de sistemas de informação, em vez de sucesso, também têm se destacado nas últimas quatro décadas. Uma falha de SI pode ser definida, por exemplo, como “o sistema implementado não atende às expectativas do usuário” (Ewusi-Mensah, 2003). Nesse contexto, a necessidade de aprender com as falhas foi discutida ao estudar o porquê das implantações de sistemas ERP falham (Scott e Vessey, 2000).

Já no trabalho de Matthiesen e Bjørn (2017) os autores afirmam, por exemplo, baseado nos estudos de Cerpa e Verner (2009), Ewusi-Mensah(2003) que as falhas de projetos de software não são excepcionais e, que fatores que podem aumentar o risco de falha do projeto pode incluir várias dimensões, como a falta de habilidades especializadas dos membros da equipe, mudanças constantes de requisitos, complexidades técnicas, política corporativa, resistência do usuário ou planejamento de projeto insatisfatório (HAN; HUANG, 2007).

Estes e outros trabalhos serão buscados como base para o desenvolvimento do estudo.

4. Métodos de Investigação e Planejamento dos Estudos

Segundo Prodanov e Freitas (2013) uma pesquisa exploratória é usada quando a pesquisa se encontra na fase preliminar, tem como finalidade proporcionar mais informações sobre o assunto que será investigado, possibilitando sua definição e seu delineamento, isto é, facilitando a delimitação do tema da pesquisa; orientando a fixação dos objetivos e a formulação das hipóteses ou descobrindo um novo tipo de enfoque para o assunto.

O autor ainda enfatiza que é importante ressaltar que a base da pesquisa exploratória é levantamento bibliográfico. Como o objetivo é levantar informações, as pesquisas exploratórias acabam tendo métodos mais maleáveis. Importante salientar que ainda não há definido quais métodos serão utilizados no presente trabalho, pretende-se definir a medida que tenhamos mais informações a partir dos trabalhos relacionados.

Inicialmente pretende seguir o planejamento de estudos conforme tabela 1, a seguir. O objetivo de um planejamento de estudos e poder contar com um roteiro inicial, que serve de condução ao andamento da pesquisa.

Tabela 1. Detalhamento do Planejamento de Estudos

| Item | Ação |
|--------------|--|
| Escopo | Definir o escopo da pesquisa, ou seja, os objetivos específicos, entregas, tarefas, custos e prazos e etc; |
| Investigação | Observar localmente como ocorre o trabalho nos grupos, de modo a tentar colher dados importantes para a pesquisa; |
| Métodos | Definir os métodos a serem utilizados na pesquisa exploratória. |
| Análise | Analisar os dados coletados a partir dos métodos definidos e comparar os resultados com possíveis trabalhos relacionados. |
| Resultados | Comparar os resultados obtidos com outros trabalhos que tenham o mesmo tema, a fim de buscar <i>insights</i> para uma melhor contribuição acadêmica. |

5. Referências

Sommerville, I. (2011). Engenharia de Software, 9 edição. Pearson, Addison Wesley.

Pressman, R., & Maxim, B. (2016). Engenharia de Software-8ª Edição. McGraw Hill Brasil.

Dwivedi, Y. K., Wastell, D., Laumer, S., Henriksen, H. Z., Myers, M. D., Bunker, D., ... & Srivastava, S. C. (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1), 143-157.

Matthiesen, S., & Bjørn, P. (2017). When distribution of tasks and skills are fundamentally problematic: A failure story from global software outsourcing. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW), 74.

Nelson, R. R. (2007). IT project management: Infamous failures, classic mistakes, and best practices. *MIS Quarterly executive*, 6(2).

Koch, C. (2007). When Bad Things Happen to Good Projects. *CIO Magazin*, April, 2, online available: http://www.cio.com/article/101505/When_Bad_Things_Happen_to_Good_Projects (Acesso em 09/08/2019)

Koch, C. (2002). Supply Chain: Hershey's Bittersweet Lesson. *CIO Magazin*, November, 15, online available: http://www.cio.com/article/31518/Supply_Chain_Hershey_s_Bittersweet_Lesson (Acesso em 09/08/2019)

Scott, J. E., & Vessey, I. (2000). Implementing enterprise resource planning systems: the role of learning from failure. *Information systems frontiers*, 2(2), 213-232.

Ewusi-Mensah, K. (2003). *Software development failures: anatomy of abandoned projects*: The MIT press.

Cerpa, N., & Verner, J. M. (2009). Why did your project fail?. *Communications of the ACM*, 52(12), 130-134.

Ewusi-Mensah, K. (2003). *Software development failures*. Mit Press.

Han, W. M., & Huang, S. J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software*, 80(1), 42-50.

Prodanov, C. C., & de Freitas, E. C. (2013). *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale.