

Anteprojeto de Tese de Doutorado

Sobre a Adoção de Boas Práticas de Desenvolvimento de Software em Projetos de Software Livre

Rodrigo Rocha Gomes e Souza
rodrigo@dcc.ufba.br

Dezembro de 2009

1 Motivação

No contexto de desenvolvimento de software, boas práticas são estratégias ou atividades que, de acordo com experiências e pesquisas, contribuem para o sucesso de um projeto de software.

A transferência de boas práticas entre desenvolvedores de um projeto é importante para equilibrar diferenças de desempenho entre eles [6]. Apesar disso, a transferência pode ser prejudicada por fatores como falta de motivação ou a falta de experiência.

Existe, no entanto, uma lacuna entre a teoria e a prática do desenvolvimento de software [2]. Em particular, a difusão de uma boa prática depende não apenas dos benefícios preconizados por seus entusiastas, mas também de questões ligadas ao seu uso efetivo, como o custo de implantação e a escassez de boas ferramentas de apoio.

Por melhor que seja a argumentação, uma prática só é útil quando é efetivamente adotada no desenvolvimento de um software. Por isso, a compreensão dos fatores que influenciam essa adoção é uma parte importante do estudo de uma prática de desenvolvimento. Infelizmente, a maior parte do conhecimento disponível sobre tais fatores é proveniente de observações isoladas e estudos de caso [1]. Essas observações, embora de grande valia, são essencialmente tendenciosas. Uma compreensão mais abrangente dos fatores contribuiria para a proposta de práticas de desenvolvimento mais realistas.

A escolha feita nesta pesquisa de estudar projetos de software livre certamente limita a generalidade dos resultados, uma vez que a organização desses projetos em geral é diferente da organização de projetos comerciais [3]. Essa opção se justifica, no entanto, pela importância que tais softwares adquiriram em diversos segmentos. Um estudo mostrou que, em 2004, mais de 1 milhão de

desenvolvedores no Estados Unidos estavam trabalhando em projetos de software livre. Além disso, projetos de software livre dominam o mercado mundial de servidores web, servidores de e-mail e servidores DNS [7].

Crowston e Howison [1] recentemente mostraram que os padrões de interação entre desenvolvedores variam bastante de projeto para projeto. Alguns projetos são bastante centralizados em um conjunto pequeno de desenvolvedores que interagem com boa parte dos demais; outros possuem uma organização mais descentralizada, nas quais os desenvolvedores se organizam em grupos bem definidos. Não é claro, no entanto, como práticas de desenvolvimento se distribuem nessa rede social.

Sabe-se há muito tempo que é alta a rotatividade de desenvolvedores em empresas de software, e pesquisas recentes mostram que o cenário também ocorre em projetos de software livre [4]. Enquanto os processos praticados por uma empresa são mantidos graças à organização hierárquica de seus funcionários, vale a pena investigar como se dá essa manutenção em projetos de software livre, especialmente os mais descentralizados, face à saída e à entrada de desenvolvedores.

2 Objetivo

O principal objetivo desta pesquisa é entender como a adoção de boas práticas em um projeto de software é afetada pela evolução de aspectos ligados ao processo, ao produto e à organização de desenvolvedores do projeto. Pretende-se assim identificar em quais contextos determinadas práticas são efetivamente aplicadas e entender os mecanismos que reforçam a adoção de práticas de desenvolvimento de software.

Os seguintes aspectos do processo, do produto e da organização social de cada projeto serão investigados:

- idade do projeto;
- tamanho do software;
- número de desenvolvedores;
- número de downloads do software;
- atividade em listas de discussão;
- concentração do esforço de codificação em um grupo pequeno de desenvolvedores.

A análise será focada nas atividades de implementação e controle de qualidade. Serão investigadas, entre outras, as seguintes práticas:

- implementação de testes de unidade automatizados antes da implementação de funcionalidades;

- implementação dos testes por pessoas diferentes daquelas que implementam funcionalidades;
- correção de *bugs* antes da implementação de novas funcionalidades;
- integração contínua das mudanças com o repositório de código-fonte;
- posse coletiva de código.

Espera-se, com este trabalho, chegar a respostas para questões de pesquisa como as que se seguem:

- A adoção de boas práticas é afetada pelo tamanho da equipe? Quais práticas são mais afetadas?
- A adoção de práticas é mais uniforme em projetos mais centralizados?
- Existe relação entre a adoção de práticas e o perfil de um desenvolvedor? Por exemplo, será que desenvolvedores que participam ativamente de listas de discussão do projeto tendem a seguir mais boas práticas?
- A saída de desenvolvedores de um projeto prejudica a adoção de boas práticas?

3 Metodologia

Muitos projetos de software livre tornam disponíveis publicamente alguns artefatos produzidos durante o processo de desenvolvimento de software. Esses artefatos incluem código-fonte e documentação mantidos em sistemas de controle de versão, bem como relatórios de *bugs* e mensagens de e-mail trocadas entre desenvolvedores.

A grande disponibilidade de dados relacionados ao desenvolvimento favorece a realização de estudos em larga escala. Nesta pesquisa serão utilizados predominantemente métodos quantitativos de análise de dados para viabilizar o estudo de uma amostra significativa de projetos de software livre.

A seleção da amostra de projetos será feita com base em informações como número de desenvolvedores e tamanho do software. Serão extraídos dados de repositórios de controle de versão, sistemas de acompanhamento de *bugs* e listas de discussão.

Características como idade do projeto e número de downloads poderão ser obtidas de repositórios de projetos de software livre, tais como o SourceForge.net. Segundo dados oficiais¹, em fevereiro deste ano o repositório contava com cerca de 230 mil projetos registrados.

A identificação das práticas adotadas por um projeto será feita a partir do uso de técnicas de mineração de processos [5] sobre o histórico dos dados extraídos do projeto. Tais técnicas têm como finalidade extrair modelos de processos a partir de dados temporais.

¹<http://sourceforge.net/about>

A organização social dos desenvolvedores será estudada através de técnicas de análise de redes sociais [?]. Essas técnicas permitirão investigar aspectos como a transferência e a concentração de trabalho entre desenvolvedores.

Para responder às questões de pesquisa, serão usados métodos de análise exploratória de dados e inferência estatística, relacionando características de projetos e de desenvolvedores, o histórico de desenvolvimento e a aplicação de boas práticas. A partir dos resultados da análise poderão ser realizadas investigações mais aprofundadas sobre projetos selecionados e comparações com resultados obtidos em estudos de caso.

4 Proposta de Cronograma

As seguintes atividades serão realizadas durante este projeto, de acordo com os prazos indicados na Tabela 1:

1. **revisão bibliográfica;**
2. **execução de experimentos preliminares;**
3. **execução de experimentos.**
4. **divulgação científica;**
5. **redação e defesa da tese.**

Atividade	2010.1	2010.2	2011.1	2011.2	2012.1	2012.2	2013.1	2013.2
1	•	•	•					
2			•	•				
3					•	•		
4				•		•		
5						•	•	•

Tabela 1: Proposta de Cronograma de Atividades

Referências

- [1] Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.
- [2] Robert L. Glass. The relationship between theory and practice in software engineering. *Commun. ACM*, 39(11):11–13, 1996.
- [3] Eric S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly and Associates, Inc., Sebastopol, CA, USA, 2001. Foreword By-Young, Bob.

- [4] Gregorio Robles and Jesus Gonzalez-Barahona. Contributor Turnover in Libre Software Projects. *Open Source Systems*, pages 273–286, 2006.
- [5] Vladimir Rubin, Christian Günther, Wil van der Aalst, Ekkart Kindler, Boudewijn van Dongen, and Wilhelm Schäfer. Process Mining Framework for Software Processes. pages 169–181. 2007.
- [6] Gabriel Szulanski. Exploring Internal Stickiness: Impediments to the Transfer of Best Practice Within the Firm. *Strategic Management Journal*, 17:27–43, 1996.
- [7] David A. Wheeler. Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)?: Look at the Numbers! 2007.