

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 comprovadamente contribuem para o sucesso de um projeto de software. Existem centenas de boas práticas documentadas, muitas delas associadas a processos de desenvolvimento, tais como RUP e XP [2]. Exemplos de práticas incluem integração contínua de mudanças e o uso de testes de unidade automatizados.

A aplicação de boas práticas dentro de uma equipe torna-a mais competitiva ao equilibrar as diferenças de desempenho entre seus membros. Não obstante, a transferência de práticas entre desenvolvedores pode ser prejudicada por diversos fatores, como falta de motivação ou de experiência [9].

Projetos de software livre são tipicamente caracterizados pela participação de voluntários com diferentes níveis de envolvimento, o que é um fator que potencialmente dificulta a transferência de práticas na equipe. Outro fator é a alta rotatividade de desenvolvedores [7]. Além disso, comparado a projetos desenvolvidos por empresas, projetos de software livre geralmente possuem uma estrutura de controle menos centralizada [6], o que reforça a importância de interações entre desenvolvedores na propagação de boas práticas.

A estrutura social de uma equipe é composta de indivíduos, suas ações e suas interações [1]. Para entender a disseminação de práticas, é preciso entender aspectos como o tamanho e a tendência de crescimento dos número de desenvolvedores; a distribuição de trabalho e o grau de especialização dos membros do projeto; a comunicação e a transferência de trabalho entre desenvolvedores.

A maioria dos estudos sobre o desenvolvimento de software livre foca a produção de código-fonte, e apenas alguns investigam a interação entre desenvolvedores [1]. Dentre estes últimos, a maior parte consiste de estudos de caso.

Mockus et al. [3], por exemplo, estudaram os projetos Apache e Mozilla e caracterizaram a distribuição do esforço de codificação e de relato de *bugs* entre participantes dos projetos. Uma das conclusões do estudo é que o número de pessoas que reportam *bugs* é uma ordem de magnitude maior do que o número de pessoas que reparam *bugs*. Mas o quão generalizável é essa observação?

Segundo Crowston e Howison [1], a falta de estudos em larga escala, envolvendo múltiplos projetos, reflete “o estágio inicial desse tipo de pesquisa, a complexidade do fenômeno e a dificuldade de se obter dados comparáveis de projetos diferentes”.

2 Objetivo

O principal objetivo desta pesquisa é entender como a adoção de boas práticas em um projeto de software é afetada pela evolução da estrutura social e de outras características 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.

Serão investigadas, entre outras, os seguintes aspectos de projetos de software livre:

- idade do projeto;
- tamanho do software;
- número de desenvolvedores;
- número de downloads do software;
- atividade em listas de discussão;
- atividade em sistemas de acompanhamento de *bugs*;
- distribuição do esforço de codificação entre 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? Existem evidências de que certas práticas preconizadas por métodos ágeis são difíceis de se aplicar em equipes grandes [4].
- A adoção de práticas é mais uniforme em projetos mais centralizados? Uma resposta positiva a esta questão pode sugerir que é importante buscar formas de centralizar o desenvolvimento de projetos de software.
- 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? De que forma? Que estratégias para perpetuar boas práticas face à saída de desenvolvedores têm se mostrado bem sucedidas?

3 Metodologia

Em primeiro lugar será feita uma revisão sistemática de tópicos como engenharia de software experimental, processos e práticas de desenvolvimento de software, desenvolvimento de software livre, mineração de processos e estatística, entre outros. Os objetivos são a identificação de trabalhos correlatos e abordagens complementares, bem como a familiarização com técnicas que poderão ser usadas na pesquisa.

Pretende-se estudar projetos de software livre hospedados no repositório SourceForge.net, que abriga cerca de 230 mil projetos¹. O SourceForge.net disponibiliza publicamente, para cada projeto, informações sobre o número de *downloads* e idade do projeto, além de sistema de controle de versão, sistema de acompanhamento de *bugs* e listas de discussão.

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. Espera-se assim obter conclusões generalizáveis sobre o uso de boas práticas em projetos de software livre.

A princípio serão realizados estudos de caso com um pequeno conjunto de sistemas, em caráter exploratório, a fim de se adquirir familiaridade com as informações disponíveis e com ferramentas de extração e análise de dados. As questões de pesquisa serão refinadas de acordo com observações realizadas nessa etapa.

A seguir será realizado um estudo em larga escala, considerando uma amostra de projetos estratificada de acordo com dados 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.

¹<http://sourceforge.net/about>

A identificação das práticas adotadas por um projeto será feita a partir do uso de técnicas de mineração de processos [10] sobre o histórico dos dados extraídos do projeto. Tais técnicas têm como finalidade extrair modelos de processos a partir de dados temporais e já foram aplicadas em estudos sobre as atividades realizadas durante o desenvolvimento de sistemas de software [8].

A organização social dos desenvolvedores será estudada através de técnicas de análise de redes sociais [5]. Essas técnicas já foram aplicadas com sucesso em estudos anteriores [1] e permitem 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.

Pretende-se divulgar a pesquisa através da publicação de artigos em congressos ou periódicos. Estão previstos pelo menos dois artigos: o primeiro com resultados do estudo de caso e o segundo com resultados do estudo em larga escala.

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. **realização de estudos de caso;**
3. **realização de um estudos em larga escala;**
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] Ivar Jacobson. Enough of Processes- Let's Do Practices. *Requirements Engineering, IEEE International Conference on*, 0, 2007.
- [3] Audris Mockus, Roy T. Fielding, and James D. Herbsleb. Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Methodol.*, 11(3):309–346, 2002.
- [4] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Commun. ACM*, 48(5):72–78, 2005.
- [5] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.
- [6] 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.
- [7] Gregorio Robles and Jesus Gonzalez-Barahona. Contributor Turnover in Libre Software Projects. *Open Source Systems*, pages 273–286, 2006.
- [8] 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.
- [9] Gabriel Szulanski. Exploring Internal Stickiness: Impediments to the Transfer of Best Practice Within the Firm. *Strategic Management Journal*, 17:27–43, 1996.
- [10] Wil van der Aalst, Ton Weijters, and Laura Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Trans. on Knowl. and Data Eng.*, 16(9):1128–1142, 2004.