

Anteprojeto de Tese de Doutorado

# Processos de Desenvolvimento em Software: Discurso e Prática

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

Dezembro de 2009

## 1 Objetivo

O principal objetivo deste trabalho é traçar a evolução dos processos de desenvolvimento e da estrutura social de projetos de software livre. Serão investigadas as causas de mudanças nos processos ao longo do ciclo de vida do software. Além disso, serão evidenciadas diferenças e semelhanças entre os processos praticados em projetos de diversos portes, propósitos e níveis de maturidade.

A análise será focada nas atividades de implementação, documentação e controle de qualidade. Serão respondidas, para cada projeto, questões como as que se seguem:

- São usados testes automáticos?
- Os testes são planejados e escritos antes da implementação das funcionalidades correspondentes?
- Os defeitos são corrigidos antes da implementação de novas funcionalidades?
- A pessoa que escreve os testes para uma funcionalidade é a mesma que implementa a funcionalidade?
- Modificações no código-fonte são integradas frequentemente ao repositório do projeto?
- Há um registro sistemático da descoberta e da resolução de defeitos no software?
- São seguidas convenções de codificação?
- Em que momento a documentação do sistema é escrita?

- Os papéis de programador, testador e documentador são exercidos por pessoas diferentes ou os papéis se misturam?
- Existe posse de código ou o código-fonte é revisado por diversos programadores?
- De que forma se dá a transferência de trabalho entre desenvolvedores?

Ao investigar a resposta dessas questões em diversos projetos ao longo do tempo, se pretende chegar a respostas para as seguintes questões de pesquisa:

- Em que condições a entrada e a saída de desenvolvedores de um projeto afeta o seu processo de desenvolvimento?
- Existem diferenças nos processos aplicados por desenvolvedores com diversos níveis de envolvimento com o projeto?
- Qual é o papel dos desenvolvedores mais envolvidos com um projeto na manutenção de seu processo de desenvolvimento?
- Os subprojetos de um projeto compartilham um mesmo processo?
- Quais práticas são mais difundidas em projetos de software livre? Quais são menos difundidas?
- Como se diferenciam os processos de projetos bem sucedidos e os processos de projetos que foram descontinuados?
- Dentre os projetos mais bem sucedidos, de que forma os processos se diferenciam?
- Como os processos se alteram nos diversos estágios do desenvolvimento de uma versão do software (alfa, beta, candidata a lançamento, em manutenção)?
- Existem práticas que são comumente aplicadas em conjunto?
- Os projetos que dizem seguir um modelo de processo estão de fato aplicando as práticas associadas a ele?

## 2 Motivação

Há muito se discutem formas de tornar o processo de desenvolvimento de software mais previsível, aumentando a produtividade da equipe e a qualidade do software. Dessa discussão surgiram modelos de processos e cartilhas de boas práticas de desenvolvimento.

Existe, no entanto, uma lacuna entre a teoria e a prática do desenvolvimento de software [3]. A aplicação de processos e práticas é dificultada por uma série de questões na vida real, desde a escassez de programas que os suportem até os altos custos de implantação.

Projetos de software livre tornam disponíveis publicamente alguns artefatos produzidos durante o processo de desenvolvimento, como código-fonte e relatórios de *bugs*, o que os torna bons candidatos para o estudo empírico da aplicação de processos e práticas de desenvolvimento.

Pesquisa sobre software livre tem se concentrado em estudos de caso [5, 1], com algumas exceções. Crowston e Howison [2] estudaram padrões de comunicação em sistemas de relatório de bugs de 120 projetos de software livre e encontraram grande variedade entre os projetos. Krishnamurty [4] estudou 100 projetos populares e percebeu que boa parte só possuía um desenvolvedor.

Tal descrição permitiria a identificação de dificuldades e limitações na aplicação de processos e práticas, contribuindo para a proposta de processos e práticas mais realistas.

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. Técnicas de mineração de processos e estatística descritiva viabilizam a realização desta pesquisa com grandes amostras de projetos de software.

### 3 Metodologia

Esta pesquisa consistirá de um estudo abrangente, envolvendo um grande número de projetos de software livre, e de um estudo mais aprofundado sobre um número reduzido de projetos. Os dois estudos cumprem papéis complementares no entendimento da aplicação de processos em projetos reais.

Para o estudo abrangente será selecionada uma amostra significativa de projetos de software livre hospedados em um repositório de código-fonte a ser escolhido, como o SourceForge. A amostra será estratificada de acordo com aspectos como idade do projeto, número de desenvolvedores e número de linhas de código-fonte.

Serão analisados relatórios de bugs registrados em um sistema de gerenciamento de bugs, mensagens de e-mail arquivadas em listas de discussão e arquivos mantidos sob controle de versão, como documentação e código-fonte. Ferramentas para a extração desses dados já foram usadas em pesquisas anteriores [], com bons resultados.

A análise dos dados no estudo mais abrangente será feita de forma automática através de técnicas de mineração de processos. Essa abordagem foi escolhida por tornar viável a análise de grandes volumes de dados. A mineração de processos tem como finalidade de extrair modelos de processos a partir de registros de eventos. Eventos que poderão ser extraídos dos dados coletados incluem a modificação de um arquivo fonte, o registro da resolução de um bug e o envio de uma mensagem à lista de discussão do projeto.

Inicialmente será feita uma análise exploratória sobre uma amostra reduzida de projetos de software com os métodos e ferramentas escolhidos. Essa

análise terá por objetivo investigar como os dados devem ser pré-processados e identificar limitações da abordagem.

A amostra reduzida de projetos de software também será alvo de um estudo mais detalhado. Informações adicionais sobre o processo desses projetos serão coletadas através de questionários e entrevistas com desenvolvedores. Quando for possível, as informações fornecidas pelos desenvolvedores serão confrontadas com as informações extraídas automaticamente.

## 4 Proposta de Cronograma

Atividade	2010.1	2010.2	2011.1	2011.2	2012.1	2012.2	2013.1	2013.2
??								
??								
??								
??								
??								
??						•	•	•
??								•

Tabela 1: Proposta de Cronograma de Atividades

## Referências

- [1] Andrea Capiluppi and Martin Michlmayr. From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects. In Joseph Feller, Brian Fitzgerald, Walt Scacchi, and Alberto Silitti, editors, *Open Source Development, Adoption and Innovation*, pages 31–44. Springer, 2007.
- [2] Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.
- [3] Robert L. Glass. The relationship between theory and practice in software engineering. *Commun. ACM*, 39(11):11–13, 1996.
- [4] Sandeep Krishnamurthy. Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects. *First Monday*, 7(6), 2002.
- [5] 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.