

# Texto Livre em documentação no desenvolvimento de softwares em comunidades

Cássio Elias dos Santos Jr

31 de outubro de 2011

## Resumo

Este trabalho contém um breve estudo sobre a importância de documentar o desenvolvimento de softwares livres em uma comunidade. Podemos mostrar ainda que é possível estruturar o desenvolvimento de forma que informações sobre o software e código fonte estejam disponíveis em um registro de alterações no código fonte.

## 1 Introdução

A importância de documentações sobre funcionamento e estrutura de um software se torna cada vez mais relevante a medida em que a complexidade e as funcionalidades do software aumentam. No contexto de um software livre, esta documentação se torna ainda mais relevante devido ao potencial número de desenvolvedores. De fato, um software pode se tornar descontinuado quando não recebe manutenção de seus desenvolvedores e, sem uma completa documentação, torna-se mais fácil desenvolver um novo software do que fornecer manutenção a um antigo. O objetivo deste documento é mostrar a importância de utilizar mecanismos para facilitar que o conhecimento sobre um software seja disponibilizado de forma prática.

## 2 Desenvolvimento

Um ponto de vista relevante para o desenvolvimento de uma metodologia para gerar documentações é o de uma pessoa que procura determinada informação. Podemos imaginar que, inicialmente, esta pessoa possa filtrar as informações por *bugs* conhecidos, ou suas correções, ou mesmo implementação de novas funcionalidades, histórico de atualizações. Desta forma cada informação deve ser inicialmente anotada. Estas anotações podem ser inferidas por meio da fonte da informação: se uma informação é proveniente de um fórum de dúvidas ou de uma plataforma *wiki*, então é muito provável que esta seja um tutorial; informações sobre *bugs* são normalmente provenientes de softwares do tipo *bug trackers* como o *mantis*, *bug-reporter*, entre outros. Sobre este assunto, técnicas de mineração de dados permitem inferências semânticas não supervisionadas e técnicas de casamento de padrão permitem uma busca mais eficiente da informação.

### 3 Exemplos

É comum encontramos comunidades em torno de softwares livres mais populares como *linux*, *firefox*, entre outros. É importante notar que estas comunidades utilizam de ferramentas como estas para fornecerem, de forma eficiente e prática, informações sobre o software.

Vejam os *firefox* por exemplo, no site da comunidade[ref] podemos encontrar informações sobre o histórico de alterações (proveniente do software de controle de versões), tutoriais (alguns dos mais úteis escritos pelos próprios usuários no fórum da comunidade).

Uma das versões do *linux*, o *Ubuntu*[ref], fornece ainda comunidades em vários países e sua manutenção vem de diferentes comunidades como o *synaptics* e *kde*. Da mesma forma que na comunidade do *firefox*, podemos encontrar informações úteis facilmente devido a grande contribuição de usuários.

É importante resaltar que estas informações se tornam disponíveis por meio de softwares que permitem que estes usuários contribuam para uma documentação assim como os desenvolvedores possam interagir diretamente com dúvidas de usuários.

Uma desvantagem deste método é que a quantidade de informação é proporcional à popularidade do software.

### 4 Conclusão

Neste trabalho discutimos a importância de uma boa documentação na manutenção de um software mantido por uma comunidade livre. Embora não existam muitas pessoas que possam trabalhar diretamente na formulação de uma documentação, mostramos que é possível organizar pequenas informações provenientes de vários usuários afim de construir uma documentação e histórico de um software de forma prática e sem custos adicionais.

### 5 Referências

**Mantis:** <http://www.mantisbt.org/>

**XWiki:** <http://www.xwiki.org/xwiki/bin/view/Main/WebHome>

**Ubuntu:** <http://www.ubuntu.com/>

**Firefox:** <http://www.mozilla.org/community/>