

Engenharia de Software e Software Livre

Tudo a Ver!

Antonio Terceiro

`terceiro@dcc.ufba.br`

Abril de 2008 – fisl9.0

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre
- 3 Desafios do Software Livre
- 4 Bons exemplos do Software Livre
- 5 Considerações Finais

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre
- 3 Desafios do Software Livre
- 4 Bons exemplos do Software Livre
- 5 Considerações Finais

Da Wikipedia (“Software Engineering”):

Estudo de abordagens sistemáticas, disciplinadas e quantificáveis para desenvolvimento, operação e manutenção de software.

Perguntas que não querem calar

- Como **prever** custo e tempo do desenvolvimento de software?
- Como **melhorar a qualidade** o produto do desenvolvimento de software?
- Como **facilitar futuras modificações** em software?
- Como **reduzir custo** do desenvolvimento de software?

Temas comuns da Engenharia de Software

- Qualidade
- Evolução
- Arquitetura

Qualidade de Software

Como disciplina:

Conjunto planejado e sistemático de atividades para garantir que o software seja construído com qualidade.

Como atributo:

- ① *o grau ao qual um software atinge os requisitos especificados*
- ② *o grau ao qual um software atinge as expectativas dos seus usuários.*

Fonte: *IEEE 610.12: IEEE Standard Glossary of Software Engineering Terminology* (tradução livre)

Evolução de Software

- Processo contínuo de transformação no software.
- As 8 leis de Lehman:
 - I) Mudança contínua
 - II) Complexidade crescente
 - VI) Crescimento contínuo
 - VII) Qualidade decrescente

M.M Lehman, J.F. Ramil, P.D. Wernick, and D.E. Perry. **Metrics and laws of software evolution-the nineties view.** In *Proceedings of the 4th International Symposium on Software Metrics*, 1997.

Arquitetura de Software

A arquitetura de software de um programa ou sistema computacional é a estrutura das estruturas do sistemas, o que reúne elementos de software, as propriedades desses elementos que são externamente visíveis, e a relação entre eles.

Fonte: Software Architecture in Practice (2nd edition), (Bass, Clements, Kazman; Addison-Wesley, 2003).

Arquitetura de Software (2)

- A **arquitetura** de um software **se manifesta** na construção e na execução do software.
- Descrição de arquitetura: **documentação** da estrutura de alto nível de um software. Torna a arquitetura **explícita**.
- Usos:
 - **Planejamento** do processo de desenvolvimento.
 - Análise de impacto de possíveis alterações (**rastreabilidade**).
 - Análise de **desempenho** da arquitetura.
 - Planejamento de **implantação**.

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre**
- 3 Desafios do Software Livre
- 4 Bons exemplos do Software Livre
- 5 Considerações Finais

O que é Software Livre

Um software é livre se seus usuários têm 4 liberdades:

- 0 executar
- 1 estudar
- 2 redistribuir
- 3 aperfeiçoar e distribuir

Copyleft representa a manutenção dessas liberdades.

Como se dá o desenvolvimento de software livre?

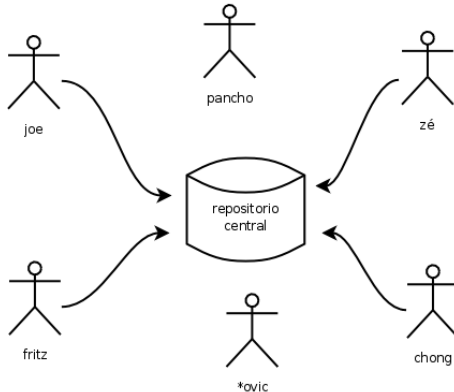


Figura: Desenvolvimento distribuído de software livre

Como se dá o desenvolvimento de software livre? (2)

- Controle de versão (“Gerência de configuração”):
CVS, svn, bzd, darcs, hg, git ...
- Gerenciamento de projeto:
Bugzilla, Trac, debbugs, ...
- Comunicação on-line:
listas, IRC, blogs, planets, fóruns web ...
- Lançamentos:
versões para teste e versões estáveis.

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre
- 3 Desafios do Software Livre**
- 4 Bons exemplos do Software Livre
- 5 Considerações Finais

Qualidade

Quando os desenvolvedores de um software são voluntários ...

- ... como garantir que todo trabalho que precisa ser feito para a próxima versão será feito?
- ... como prever o tempo de desenvolvimento?
- ... como garantir cumprimento de prazos?

Voluntário: um colaborador do projeto sobre cujas atividades não se tem controle.

Evolução

- Como manter a qualidade do código?
- Como manter a arquitetura do software?
- Como facilitar a entrada de novos colaboradores?
- Como manter a portabilidade?

O caso do acoplamento no Linux

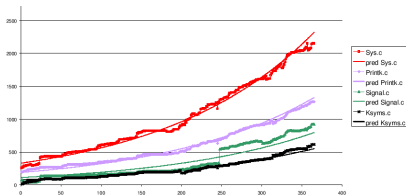


Figura: “Common coupling” no Linux

Fonte: S.R. Schach, B. Jin, D.R. Wright, G.Z. Heller, and A.J. Offutt, **Maintainability of the Linux Kernel**, IEE Proceedings – Software, 149(1), 18-23, February 2002.

<http://citeseer.ist.psu.edu/schach02maintainability.html>

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre
- 3 Desafios do Software Livre
- 4 Bons exemplos do Software Livre**
- 5 Considerações Finais

Qualidade: Portabilidade

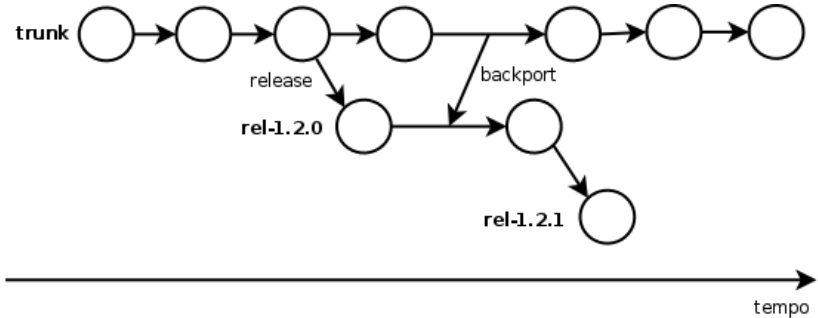
- *build systems* bastantes completos
- Disponibilidade de arquiteturas diferentes para teste
- Debian:
 - 11 arquiteturas Linux com suporte oficial: i386, m68k, sparc, alpha, powerpc, arm, mips, hppa, ia64, s390, amd64
 - 4 arquiteturas Linux sem suporte oficial: ppc64, sh, arceb, m32r
 - 4 Arquiteturas não-Linux, sem suporte oficial: hurd-i386, netbsd-i386, netbsd-alpha, kfreebsd-gnu

Qualidade: Gerenciamento de Lançamentos baseado em tempo

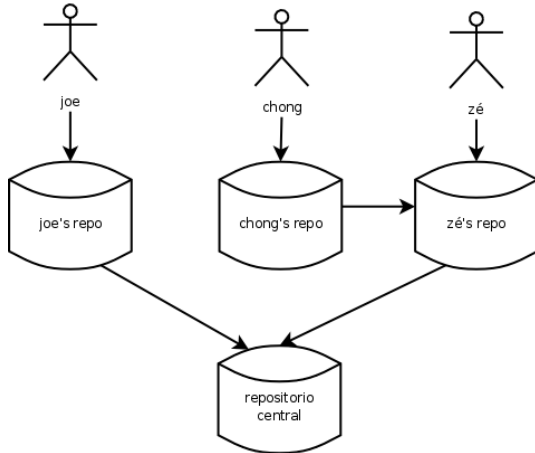
- Projetos grandes.
- Trabalho suficiente é realizado durante o ciclo de desenvolvimento.
- Quando chega a época do lançamento, o que está pronto é lançado e o que não está fica pra próxima.
- Exemplo perfeito: GNOME

Fonte: Michlmayr, M. (2007) Quality Improvement in Volunteer Free and Open Source Software Projects: Exploring the impact of Release Management. PhD Dissertation, University of Cambridge.

Evolução: uso corriqueiro de branches



Evolução: controle de versão distribuído



Arquiteturas de “plugins”

- Core estável
- Extensões desenvolvidas independentemente
- Exemplos:
 - Mozilla *
 - TWiki
 - GNOME: Gedit, EOG, ...
 - outros

Agenda

- 1 O que é Engenharia de Software
- 2 O que é Software Livre
- 3 Desafios do Software Livre
- 4 Bons exemplos do Software Livre
- 5 Considerações Finais

Alguns pontos

- A Engenharia de Software procurar melhorar o desenvolvimento de software. Historicamente, a ES teve seu foco no contexto de “organizações desenvolvedoras de software” .
- EXISTE SIM o uso de diversas técnicas de Engenharia de Software no Software Livre. Mas quase sempre não existe “organização desenvolvedora” .
- Então é necessário repensar a Engenharia de Software ...

Repensando a Engenharia de Software

Além de se perguntar como **aumentar a qualidade** e **reduzir o custo**, é preciso se perguntar também:

Num contexto de desenvolvedores voluntários e espalhados geograficamente que se comunicam quase exclusivamente através da rede ...

- ... como facilitar a entrada de novos colaboradores em projetos complexos?
- ... como comunicar decisões de projeto?
- ... como especificar o Jeito CertoTM de se fazer as coisas?
- ... como garantir a integridade do código, mesmo que não exista alguém designado (ou especialmente motivado) para isso?

Perguntas?

`terceiro@dcc.ufba.br`