

# The FreeBSD project: A replication case study of open source development

MAC5779 - Engenharia de Software Experimental

Higor Amario de Souza<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística  
Universidade de São Paulo  
Programa de Pós-Graduação em Ciência da Computação

22 de Abril de 2013

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.



# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# FreeBSD Project

- Projeto OpenSource de longa data;
- Processo de desenvolvimento razoavelmente bem-definido;
  - Papéis para desenvolvedores;
  - Lidar com melhorias e defeitos;
  - Gerenciar lançamentos;
- Poucos desenvolvedores do núcleo do sistema;
- Muitos desenvolvedores para 80% do código;
- Processo de testes definido;
- Similar - Apache Software Foundation.

# Open Source: Vantagens

- Voluntários;
- Suporte de diversas empresas;
- Sem atribuições ou prazos;
- Baixo custo e alta qualidade;
- Diversos estudos afirmam suas vantagens.

# Open Source: Vantagens

- Voluntários;
- Suporte de diversas empresas;
- Sem atribuições ou prazos;
- Baixo custo e alta qualidade;
- Diversos estudos afirmam suas vantagens.

# Open Source: Vantagens

- Voluntários;
- Suporte de diversas empresas;
- Sem atribuições ou prazos;
- Baixo custo e alta qualidade;
- Diversos estudos afirmam suas vantagens.

# Open Source: Vantagens

- Voluntários;
- Suporte de diversas empresas;
- Sem atribuições ou prazos;
- Baixo custo e alta qualidade;
- Diversos estudos afirmam suas vantagens.

# Open Source: Vantagens

- Voluntários;
- Suporte de diversas empresas;
- Sem atribuições ou prazos;
- Baixo custo e alta qualidade;
- Diversos estudos afirmam suas vantagens.

# Open Source: Desvantagens

- Projeto e arquitetura pobres;
- Ferramentas de desenvolvimento inferiores;
- Crescimento de LOCs maior que projetos comerciais.



# Open Source: Desvantagens

- Projeto e arquitetura pobres;
- Ferramentas de desenvolvimento inferiores;
- Crescimento de LOCs maior que projetos comerciais.

# Open Source: Desvantagens

- Projeto e arquitetura pobres;
- Ferramentas de desenvolvimento inferiores;
- Crescimento de LOCs maior que projetos comerciais.

## Outros estudos (Mockus et al.)

- Apache;
- Mozilla;
- Hipóteses para o sucesso de projetos OS.

## Outros estudos (Mockus et al.)

- Apache;
- Mozilla;
- Hipóteses para o sucesso de projetos OS.

## Outros estudos (Mockus et al.)

- Apache;
- Mozilla;
- Hipóteses para o sucesso de projetos OS.

# Objetivo

- Avaliar essas hipóteses no projeto FreeBSD;
- Replicação do estudo de Mockus et al.;
- Medição da densidade de defeitos do produto.

# Objetivo

- Avaliar essas hipóteses no projeto FreeBSD;
- Replicação do estudo de Mockus et al.;
- Medição da densidade de defeitos do produto.

# Objetivo

- Avaliar essas hipóteses no projeto FreeBSD;
- Replicação do estudo de Mockus et al.;
- Medição da densidade de defeitos do produto.



# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.



# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Desenho do estudo de caso

- Unidade:
  - Commits;
  - Relatórios de defeitos;
  - Mensagens;
- Contexto:
  - FreeBSD;
  - Processo de desenvolvimento OS;
  - Papel dos desenvolvedores;
- Triangulação;
- Entrevista;
- EC Explanatório.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.



# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Hipóteses (Mockus et al.)

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
- 2. Grupos maiores exigem algum tipo de coordenação;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
- 4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;
- 5. Densidade de defeitos menor que em aplicações comerciais;
- 6. Desenvolvedores são também usuários;
- 7. Soluções rápidas para problemas dos clientes.

# Questões de pesquisa

- Adaptadas para o projeto FreeBSD (questionário);
- Relacionadas ao processo de desenvolvimento;
- Processos usados, numero de desenvolvedores, tempo de solução de problemas...;
- Enviadas aos 9 membros do núcleo.

# Questões de pesquisa

- Adaptadas para o projeto FreeBSD (questionário);
- Relacionadas ao processo de desenvolvimento;
- Processos usados, numero de desenvolvedores, tempo de solução de problemas...;
- Enviadas aos 9 membros do núcleo.

# Questões de pesquisa

- Adaptadas para o projeto FreeBSD (questionário);
- Relacionadas ao processo de desenvolvimento;
- Processos usados, numero de desenvolvedores, tempo de solução de problemas...;
- Enviadas aos 9 membros do núcleo.

# Questões de pesquisa

- Adaptadas para o projeto FreeBSD (questionário);
- Relacionadas ao processo de desenvolvimento;
- Processos usados, numero de desenvolvedores, tempo de solução de problemas...;
- Enviadas aos 9 membros do núcleo.

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?



# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Questionário

- 1. Quais os papéis envolvidos na codificação?
- 2. Como fazer para se tornar um desenvolvedor?
- 3. Como fazer para se tornar um desenvolvedor do núcleo?  
Como se escolhe novos desenvolvedores?
- 4. Como uma pessoa comum contribui com código?
- 5. Como a equipe de lançamento checa o código?
- 6. Como um desenvolvedor decide por colocar o código na versão atual ou na estável?
- 7. Qual a diferença entre um desenvolvedor do núcleo e outro desenvolvedor?
- 8. Como se decide o próximo passo ao corrigir um defeito ou gerar uma nova funcionalidade?

# Fontes de dados

- Questionário;
- Consulta ao repositório;
- Base de dados de relatos de defeitos;
- Arquivos de emails.

# Fontes de dados

- Questionário;
- Consulta ao repositório;
- Base de dados de relatos de defeitos;
- Arquivos de emails.



# Fontes de dados

- Questionário;
- Consulta ao repositório;
- Base de dados de relatos de defeitos;
- Arquivos de emails.

# Fontes de dados

- Questionário;
- Consulta ao repositório;
- Base de dados de relatos de defeitos;
- Arquivos de emails.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.



# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- 1. Grupos entre 10 e 15 pessoas desenvolvem o núcleo do código (80%);
  - Grupo menor: 7 a 9;
  - 47% do código;
  - 48 pessoas em média contribuem com 80% do código;
- 2. Grupos maiores do que 15 desenvolvedores exigem algum tipo de coordenação;
  - Existem mecanismos de controle;
  - Grupo para release, procedimentos para testes;
  - Processo continua informal;
- 3. Um grupo maior corrige defeitos e um grupo ainda maior reporta problemas;
  - Similar no FreeBSD.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.



# Testando as hipóteses

- 4. ~~Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- 7. ~~Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- 4. ~~Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- 7. ~~Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Testando as hipóteses

- ~~4. Projetos com poucos desenvolvedores fora do núcleo falham em encontrar e corrigir defeitos;~~
  - Não se aplica;
- 5. Densidade de defeitos menor que em aplicações comerciais;
  - Similar no FreeBSD;
- 6. Desenvolvedores são também usuários;
  - Similar no FreeBSD;
- ~~7. Soluções rápidas para problemas dos clientes;~~
  - Dados insuficientes.

# Referência

- The FreeBSD project: A replication case study of open source development  
Trungh T. Dinh-Trong & James M. Bieman  
Transactions on Software Engineering, Vol. 31 No. 6  
2005.