

# *Incorporating Formal Methods in the Open Source Software Development Process*

*Antonio Cerone and Siraj Shaikh*

*United Nations University*

*International Institute for Software Technology*

*Macau SAR China*

`www@iist.unu.edu/~antonio`

# *OpenCert Community*

From OpenCert 2007

# *OpenCert Community*

From OpenCert 2007

- **Where** do we go from here?

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?
- Collect **more data**?  
(e.g. with interviews and questionnaires)

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?
- Collect **more data**?  
(e.g. with interviews and questionnaires)
- Develop together a **common case study**?

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?
- Collect **more data**?  
(e.g. with interviews and questionnaires)
- Develop together a **common case study**?
  - agree on **modelling and design methodologies and tools**

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?
- Collect **more data**?  
(e.g. with interviews and questionnaires)
- Develop together a **common case study**?
  - agree on **modelling and design methodologies** and **tools**
  - produce/identify a **model** and a **small kernel of code** and make it **available on the Internet**

# *OpenCert Community*

## From OpenCert 2007

- **Where** do we go from here?
- Analyse a **common case study**?
- Collect **more data**?  
(e.g. with interviews and questionnaires)
- Develop together a **common case study**?
  - agree on **modelling and design methodologies and tools**
  - produce/identify a **model** and a **small kernel of code** and make it **available on the Internet**
  - incrementally introduce methods and tools

# *Success of OSS*

- **Delivered** high-quality system and application software

# *Success of OSS*

- **Delivered** high-quality system and application software
- **Why** high-quality?

# Success of OSS

- **Delivered** high-quality system and application software
- **Why** high-quality?

The *high level* of quality of free software is partly due to the high degree of peer review and user involvement

**[Raymond 1999]**

# Success of OSS

- **Delivered** high-quality system and application software
- **Why** high-quality?  
The *high level* of quality of free software is partly due to the high degree of peer review and user involvement  
[Raymond 1999]
- **Has** reliability and efficiency  
[Neumann 1998] [Coverity 2008]

# Success of OSS

- **Delivered** high-quality system and application software
- **Why** high-quality?  
The *high level* of quality of free software is partly due to the high degree of peer review and user involvement  
[Raymond 1999]
- **Has** reliability and efficiency  
[Neumann 1998] [Coverity 2008]
- **Falls short of** security and safety  
[Schneider 2000]

# Success of OSS

- **Delivered** high-quality system and application software
- **Why** high-quality?  
The *high level* of quality of free software is partly due to the high degree of peer review and user involvement  
[Raymond 1999]
- **Has** reliability and efficiency  
[Neumann 1998] [Coverity 2008]
- **Falls short of** security and safety  
[Schneider 2000]
- **Need for** certification process for OSS

# *Problems of OSS*

Fix **four major problems** [McConnell 1999]:

- create central **clearinghouse** for the open-source methodology
- overcome its **addiction to code and fix**
- eliminate **upstream defects earlier**
- collect and publish data to support **effectiveness** of the open-source methodology

# *What has been Done*

Fix **four major problems** [McConnell 1999]:

- create central **clearinghouse** for the open-source methodology
  - ⇒ **proposals** [Halloran and Scherlis 2002] [Yilmaz et al. 2006]
- overcome its **addiction to code and fix**
- eliminate **upstream defects earlier**
- collect and publish data to support **effectiveness** of the open-source methodology
  - ⇒ **surveys** [Zhao and Elbaum 2000 + 2003]
  - ⇒ **interviews** [Michlmayr 2005] [Michlmayr et al. 2005]
  - ⇒ **case studies** [Mockus et al. 2002]

# What's still Missing

Fix **four major problems** [McConnell 1999]:

- create central clearinghouse for the open-source methodology
- overcome its **addiction to code and fix**  
⇒ **negative impact on security**  
[Schneider 2000] [Coverity 2008]
- eliminate **upstream defects earlier**  
⇒ “interesting opportunities for formal models” ⇒  
“augment the robustness” + “evaluation of components and the effect of their composition” [Neumann 1998]
- collect and publish data to support effectiveness of the open-source methodology

# *Towards OSS Certification ['07]*

Our OpenCert **2007** proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]

# *Towards OSS Certification ['07]*

Our OpenCert 2007 proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]
  - quality by access, design and development

# *Towards OSS Certification ['07]*

## Our OpenCert 2007 proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]
  - quality by access, design and development
  - access : specific for OSS

# *Towards OSS Certification ['07]*

Our OpenCert **2007** proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]
  - quality by access, design and development
  - access : specific for OSS
  - design : use of formal methods

# *Towards OSS Certification ['07]*

Our OpenCert 2007 proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]
  - quality by access, design and development
  - access : specific for OSS
  - design : use of formal methods
  - development : effective leadership

# *Towards OSS Certification ['07]*

## Our OpenCert 2007 proposal

- Define a quality model for OSS  
[Shaikh and Cerone 2007]
  - quality by access, design and development
  - access : specific for OSS
  - design : use of formal methods
  - development : effective leadership
- leads to ...

# *Towards OSS Certification ['08]*

## Our OpenCert 2008 proposal

- Incorporating Formal Methods in the OSS Development [Cerone and Shaikh 2008]
  - exploit/address quality notions
  - access  $\implies$  appropriate contributors
  - design  $\longleftarrow$  use of formal models and tools
  - development  $\longleftarrow$  propose choices and offer tools but preserve freedom

# *Towards OSS Certification ['08]*

## Our OpenCert 2008 proposal

- Incorporating Formal Methods in the OSS Development [Cerone and Shaikh 2008]
  - exploit/address quality notions
  - access  $\implies$  appropriate contributors
  - design  $\longleftarrow$  use of formal models and tools
  - development  $\longleftarrow$  propose choices and offer tools but preserve freedom
- 'pilot' project open to volunteers.

# *Motivate Potential Contributors*

[exploiting quality by access]

- address motivating factors

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations
- **Implementation**
  - appropriate choice of software product

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations
- **Implementation**
  - appropriate choice of software product
  - academic projects and dissertations

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations
- **Implementation**
  - appropriate choice of software product
  - academic projects and dissertations
  - problems and challenges posed to the existing open source community

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations
- **Implementation**
  - appropriate choice of software product
  - academic projects and dissertations
  - problems and challenges posed to the existing open source community
  - learning opportunities for novices in FM

# *Motivate Potential Contributors*

[exploiting **quality by access**]

- **address motivating factors**
  - intrinsic motivations
  - extrinsic motivations
- **Implementation**
  - appropriate choice of software product
  - academic projects and dissertations
  - problems and challenges posed to the existing open source community
  - learning opportunities for novices in FM
  - a raised public profile of the members

# *New Classes of Activities*

[addressing **quality by design**]

# *New Classes of Activities*

[addressing **quality by design**]

- leader team should **propose**
  - one (or more) simple **formal modelling framework(s)**
  - the basic (extensible) **model of the system**

# *New Classes of Activities*

[addressing **quality by design**]

- leader team should **propose**
  - one (or more) simple **formal modelling framework(s)**
  - the basic (extensible) **model of the system**
- new categories of actors could
  - **refine** and **extend** the proposed formal model

# *New Classes of Activities*

[addressing **quality by design**]

- leader team should **propose**
  - one (or more) simple **formal modelling framework(s)**
  - the basic (extensible) **model of the system**
- new categories of actors could
  - **refine** and **extend** the proposed formal model
  - **reverse engineering** code

# *Effective Leadership*

[addressing **quality by development**]

- **cannot explicitly enforce** use of a specific formal modelling framework

# *Effective Leadership*

[addressing **quality by development**]

- **cannot explicitly enforce** use of a specific formal modelling framework
- formal methods presented only as a **possible but not mandatory** option

# *Effective Leadership*

[addressing **quality by development**]

- **cannot explicitly enforce** use of a specific formal modelling framework
- formal methods presented only as a **possible but not mandatory** option
- basic (extensible) model of the system
  - **informally presented** (requirements) and
  - **formally modelled** (formal framework)

# *Effective Leadership*

[addressing **quality by development**]

- **cannot explicitly enforce** use of a specific formal modelling framework
- formal methods presented only as a **possible but not mandatory** option
- basic (extensible) model of the system
  - **informally presented** (requirements) and
  - **formally modelled** (formal framework)
- additional effort by the leader team in **integrating non-FM contributions**

# *Effective Leadership*

[addressing **quality by development**]

- **cannot explicitly enforce** use of a specific formal modelling framework
- formal methods presented only as a **possible but not mandatory** option
- basic (extensible) model of the system
  - **informally presented** (requirements) and
  - **formally modelled** (formal framework)
- additional effort by the leader team in **integrating non-FM contributions**  
⇒ attract new potential project actors who are keen to reverse engineer code

# *Implementation: Product?*

- motivate potential contributors

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an existing popular tool or utility is redesigned and rebuilt

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an existing popular tool or utility is redesigned and rebuilt
    - ⇒ familiarity and approach demonstrability

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an existing popular tool or utility is redesigned and rebuilt
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an existing popular tool or utility is redesigned and rebuilt
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation
  - conceive an idea from scratch

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an existing popular tool or utility is redesigned and rebuilt
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation
  - conceive an idea from scratch
    - ⇒ best address intrinsic motivations

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an **existing popular tool or utility is redesigned and rebuilt**
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation
  - conceive an **idea from scratch**
    - ⇒ best address intrinsic motivations
    - ⇒ higher effort in setting up and demonstrating the approach

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an **existing popular tool or utility is redesigned and rebuilt**
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation
  - conceive an **idea from scratch**
    - ⇒ best address intrinsic motivations
    - ⇒ higher effort in setting up and demonstrating the approach

implicitly ⇒ Choice of Project Members

# *Implementation: Product?*

- motivate potential contributors
- appropriate choice of software product
  - an **existing popular tool or utility is redesigned and rebuilt**
    - ⇒ familiarity and approach demonstrability
    - ⇒ undermine intrinsic motivation
  - conceive an **idea from scratch**
    - ⇒ best address intrinsic motivations
    - ⇒ higher effort in setting up and demonstrating the approach

implicitly ⇒ Choice of Project Members  
**Discussion:** Choice of Software Product

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- having managed small project teams

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- having managed small project teams
- having been involved in design and development

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- having managed small project teams
- having been involved in design and development
- having familiarity with FM

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- **having managed small project teams**
- **having been involved in design and development**
- **having familiarity with FM**
- **working in academia, industry and open source development**

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- **having managed small project teams**
- **having been involved in design and development**
- **having familiarity with FM**
- **working in academia, industry and open source development**
- **public dissemination and reporting**

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- having managed small project teams
- having been involved in design and development
- having familiarity with FM
- working in academia, industry and open source development
- public dissemination and reporting

*Discussion:* Choice of Leadership

# *Implementation: Leadership?*

Leader team should demonstrate **experience of**

- having managed small project teams
- having been involved in design and development
- having familiarity with FM
- working in academia, industry and open source development
- public dissemination and reporting

*Discussion:* Choice of Leadership

- Members of the OpenCert and FLOSS-FM communities?

# *Items for Discussion*

- Is the proposal feasible?  
Anybody interested in participating?

# *Items for Discussion*

- Is the proposal feasible?  
Anybody interested in participating?
- Choice of Software Product
  - an existing popular tool or utility is redesigned and rebuilt **Which one?**
  - conceive an idea from scratch: **Domain?**  
**Proposals?**
  - **Alternative Proposals?**

# *Items for Discussion*

- Is the proposal feasible?  
Anybody interested in participating?
- Choice of Software Product
  - an existing popular tool or utility is redesigned and rebuilt **Which one?**
  - conceive an idea from scratch: **Domain?**  
**Proposals?**
  - **Alternative Proposals?**
- Choice of Leadership
  - Members of the OpenCert and FLOSS-FM communities: **Any Candidates?**

# *Discussion: Sponsorship*

# *Discussion: Sponsorship*

- **Academia:**
  - hosting project servers
  - providing tools
  - providing technical support
  - providing training material

# *Discussion: Sponsorship*

- **Academia:**
  - hosting project servers
  - providing tools
  - providing technical support
  - providing training material

**Any Offer?**

# *Discussion: Sponsorship*

- **Academia:**

- hosting project servers
- providing tools
- providing technical support
- providing training material

## **Any Offer?**

- **Industry:**

- domain associated sponsor

# *Discussion: Sponsorship*

- **Academia:**

- hosting project servers
- providing tools
- providing technical support
- providing training material

## **Any Offer?**

- **Industry:**

- domain associated sponsor

## **Any Contact?**

# References

## General References

- [O'Reilly 1999]  
IEEE STD 610.12–1990  
Lessons From Open-Source Development  
Prentice Hall, 1999
- [Raymond 1999]  
E. S. Raymond  
The Cathedral and the Bazaar  
O'Reilly and Associates, 1999
- [McConnell 1999]  
S. McConnell  
Open Source Methodology: Ready for prime time?  
*IEEE Software*, 16(4):6–8, 1999, IEEE Press, 1999

## *References: Security*

- [Schneider 2000]  
Fred B. Schneider  
Open Source in Security: Visiting the Bizarre  
In *Security and Privacy 2000*, IEEE Press, 2000
- [Coverity 2008]  
Coverity  
Open Source Report 2008

## References: Formal Methods

- [Neumann 1998]  
Peter G. Neumann  
Robust Open Source Software *Communications of the ACM*, 41(2):128, 1998
- [Shaikh and Cerone 2007]  
Siraj A. Shaikh and Antonio Cerone  
Towards a quality model for Open Source Software (OSS) presented at *OpenCert 2007*
- [Breuer and Pickin 2008]  
Peter T. Breuer, Simon Pickin  
Approximate verification in an open source world  
*Innovations in System and Software Engineering*, 4(1):87–105, 2008

## *References: Reports on Surveys*

- [Zhao and Elbaum 2000]

L. Zhao and S. Elbaum

*A survey on quality related activities in open source*

*ACM SIGSOFT Engineering Notes*, 25(3):53–57, 2000,  
ACM Press

- [Zhao and Elbaum 2003]

L. Zhao and S. Elbaum

*Quality assurance under the open source development model*

*Journal of System and Software*, 66(1):65–75, 2003,  
Elsevier Science

## *References: Proposals (from Empirical Studies)*

- [Halloran and Scherlis 2002]

T. J. Halloran and W. L. Scherlis

High quality and open source software practices

*Proc. of 2nd Workshop on Open Source Software Engineering, 2002*

- [Yilmaz et al. 2006]

C. Yilmaz, A. M. Memon, A. Porter, A. S. Krisna, D. C. Schmidt, A. Gokhale

Techniques and processes for improving the quality and performance of open-source software

2006

## *References: Interview-based Analyses*

- [Michlmayr 2005]

M. Michlmayr

Quality improvement in volunteer free software projects: Exploring the impact of release management  
*Proc. of 1st Int. Conf. on Open Source Systems*, pages 309-310, Genova, Italy, 2005

- [Michlmayr et al. 2005]

M. Michlmayr, F. Hunt, D. Probert

Quality practises and problems in free software projects  
*Journal of System and Software*, 66(1):65–75, 2003,  
Elsevier Science  
*Proc. of 1st Int. Conf. on Open Source Systems*, pages 24-28, Genova, Italy, 2005

## *References: Case Studies*

- [Mockus et al.]

A. Mockus, R. T. Fielding, J. D. Herbsleb

Two case studies of open source development: Apache and Mozilla

*ACM Transactions on Software Engineering and Methodology*, 11(3):309–346, 2002, ACM Press