Aspects of Software Quality Assurance in Open Source Software Projects: Two Case Studies from Apache Project

Dindin Wahyudin, Alexander Schatten, Dietmar Winkler, Stefan Biffl

Institute of Software Technology and Interactive Systems
Vienna University of Technology

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at

Motivation

- Open source software (OSS) solutions provide mission-critical services to industry and government organizations.
- However, empirical studies on OSS development practices raise concerns on risky practices such
  - as unclear requirement elicitation,
  - ad hoc development process,
  - little attention to quality assurance (QA) and documentation,
  - and poor project management.

- → Need for a QA framework with respect to OSS projects.
- → Which performance indicators can address QA activities?

OSS Product Classification

- Sourceforge[1] investigated OSS projects according to application domains, maturity levels and project size:
  - Top 5 of 18 application domains are: Internet application (15.4%), Software development (15.1%), System (12.4%), Communication (10%), and Game/Entertainment (9.3%).
  - More than 70% of the projects are still in early stages or already at the end of their lifecycle.
  - Only a small subset of the projects have reached their maturity and produce stable releases.
  - Project size: 86.2% of the projects employ less than 6 developers, and less than 1% of the projects have more than 16 developers.
- Pure OSS Projects consist of volunteers with rare formal processes.
- Currently a number of important OSS projects are hybrid project which are supported by companies; some participants are not volunteers (e.g., JBoss, Apache JackRabbit, Myfaces, Sourcefire or OpenOffice projects).

[1] http://www.sourceforge.net

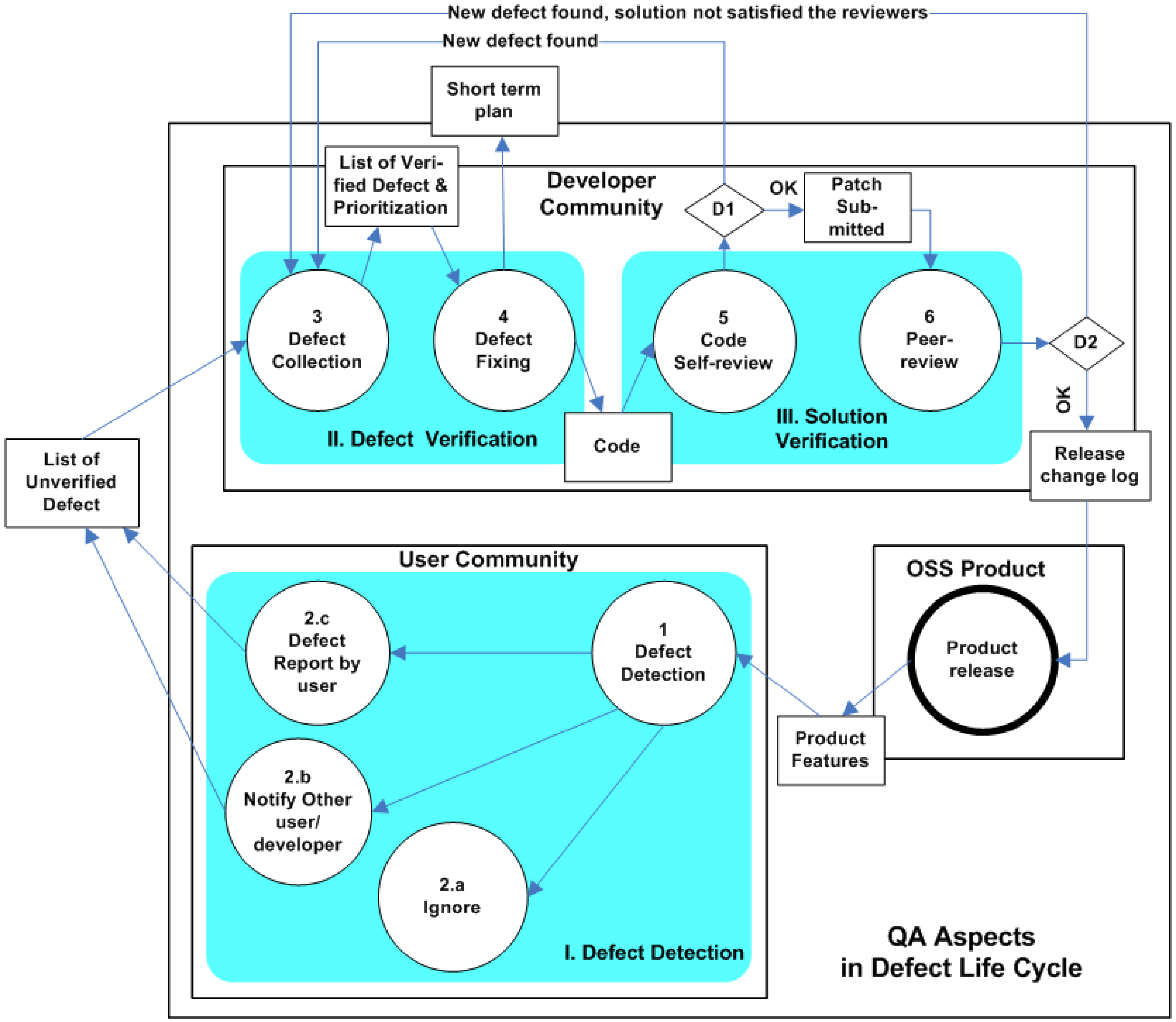
Quality Assurance in OSS Projects

- A growing number of OSS applications show levels of quality comparable to closed source software development products (Abadour 2007).
- High-quality OSS products can be achieved by applying a high degree of peer reviews and user involvement in bug/defect detection (Raymond, 2003.
- Win Conditions regarding quality issues based on OSS key stakeholders:
  - User: software must be easy to use, faster response from the developer community, faster defect closure and stable releases, etc.
  - Developer: access to current development repositories, collaboration tools, less invalid defect reports, etc.
  - Committer: adequate defect reports, higher number of verified solutions (defects were solved according to their specification).
  - Project Manager: easy monitoring of project performance measures to ensure appropriate quality assurance activities.
- An QA framework for OSS projects might support the construction of high-quality products.

QA Framework for OSS Projects

Proposed QA process groups:

- Defect Detection: provide information of a candidate defect.
- Defect Verification includes defect verification, collection and defect correction.
- Solution Verification: code self-reviews by the developers, and peer review by a „third party".



Variables and Study Objects

Variables

- Defect Detection Frequency describes the average number of reported defects (bug tracker) per time interval.
- Defect Collection Effectiveness describes ratio of real reported defects (verified by the developer community) and candidate defects.
- Defect Closure Time
  time interval from defect report opening until closure time (including peer-review and release).
- Ratio of Verified Solutions defines the verification rate (peer reviews) after a defect correction (e.g. patches).

Case Study Objects

- Pure OSS projects: Apache Tomcat release 5 and 6 (pure voluntary projects)
- Hybrid projects: Apache MyFace and the sub-projects Trinidad, Tobago, Tomahawk and Core (OSS projects with industry support)

Case Study Description

Research Methodology

- Intensive literature research.
- OSS expert interviews.
- Project Logfile Analysis (2 pure OSS projects and 5 hybrid projects)
- Duration of observation: 10/2006 – 02/2007

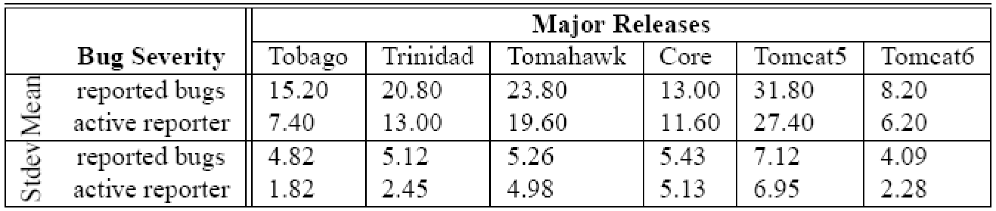
Hypothesis:

- Defect detection frequency is higher in pure OSS projects because of a heterogeneous user community.
- Defect collection Effectiveness is higher for hybrid projects because of better project documentation and better user knowledge on the project.
- Defect Closure Time is longer for hybrid projects because to strict documentation and QA process requirements.
- Ratio of verified solutions is lower for pure OSS projects because the release do not require systematic peer-reviews.

Results: Defect Detection

Defect detection frequency:

- Tomcat 5 has the highest average number of reported defects and reporter per month.
- This indicates that this project has a more active and heterogeneous user community.

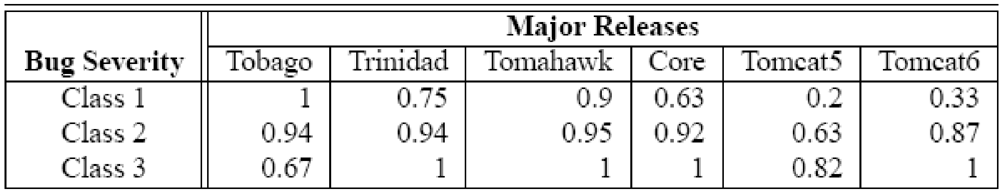


		Major Releases					
	Bug Severity	Tobago	Trinidad	Tomahawk	Core	Tomcat5	Tomcat6
Mean	reported bugs	15.20	20.80	23.80	13.00	31.80	8.20
	active reporter	7.40	13.00	19.60	11.60	27.40	6.20
Stdev	reported bugs	4.82	5.12	5.26	5.43	7.12	4.09
	active reporter	1.82	2.45	4.98	5.13	6.95	2.28

Average defect detection frequency per month

Defect detection effectiveness:

- hybrid projects tend to have a high defect detection effectiveness.
- As expected, in more formal / structured hybrid projects, the community has more knowledge on the software releases; this leads to better bug reports.



	Major Releases					
Bug Severity	Tobago	Trinidad	Tomahawk	Core	Tomcat5	Tomcat6
Class 1	1	0.75	0.9	0.63	0.2	0.33
Class 2	0.94	0.94	0.95	0.92	0.63	0.87
Class 3	0.67	1	1	1	0.82	1

Defect detection Effectiveness

Bug severity classes:

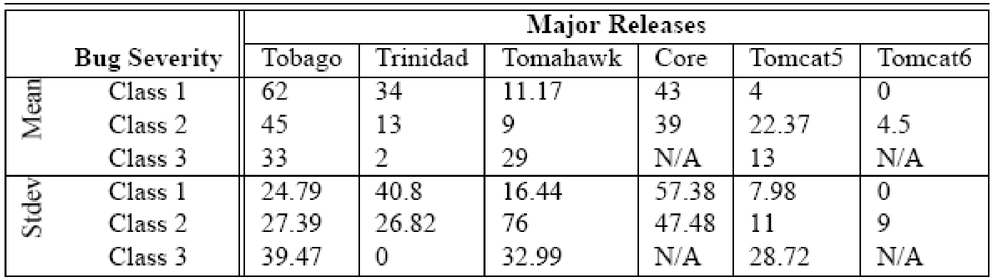
- 1: highest priority, e.g., security issues.
- 2: related to individual features
- 3: trivial and minor defects

Note: severity classes were assigned by developers.

Results: Defect Closure Duration

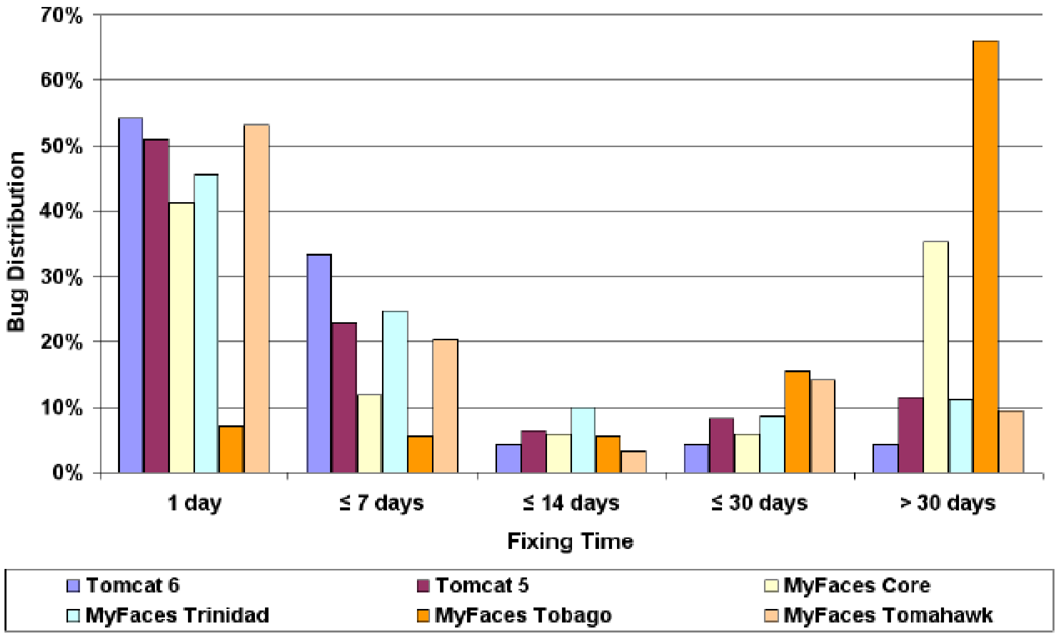
Defect closure duration:

- Defect closure Duration refers to the duration form opening a bug report until it is marked as “closed”.
- The results show that hybrid projects tend to need significantly longer for defect closure.
- This might indicate a defined process for bug handling in hybrid projects.
- Another possible reason for the delay of “Tobago” projects is that the corrections must be verified before they can be declared as “closed”.



		Major Releases					
	Bug Severity	Tobago	Trinidad	Tomahawk	Core	Tomcat5	Tomcat6
Mean	Class 1	62	34	11.17	43	4	0
	Class 2	45	13	9	39	22.37	4.5
	Class 3	33	2	29	N/A	13	N/A
Stdev	Class 1	24.79	40.8	16.44	57.38	7.98	0
	Class 2	27.39	26.82	76	47.48	11	9
	Class 3	39.47	0	32.99	N/A	28.72	N/A

Bug closure time in days.



Bug closure time distribution in reviews projects.

Results: Ratio of Verified Solutions

Bug fixed per bug report:

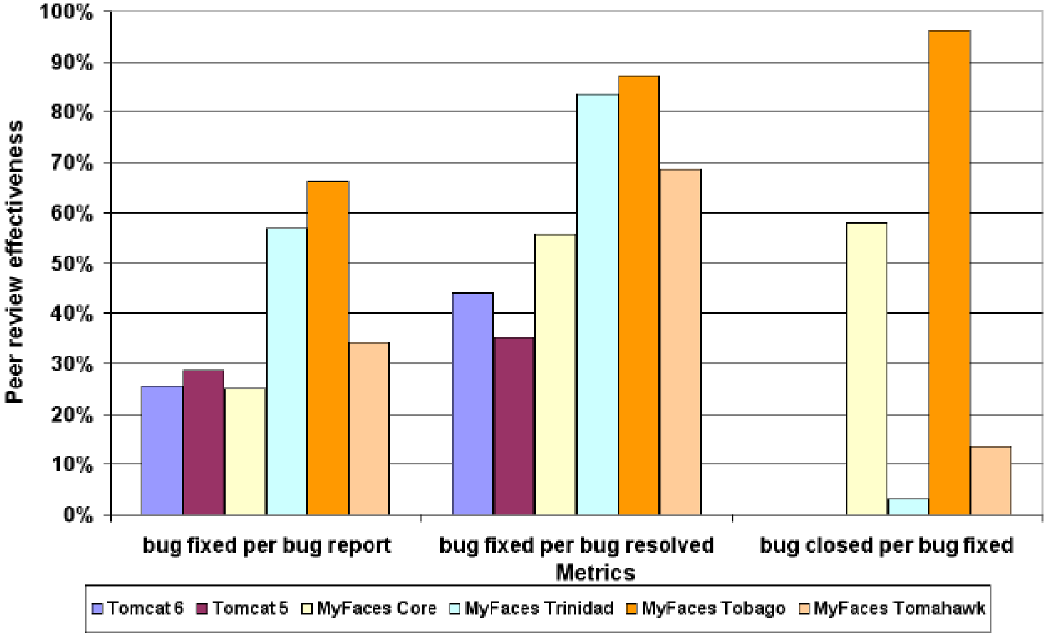
- Bugs were fixed according to the bug report (without considering additional QA activities).

Bug fixed per bug resolved:

- Self-review of bug fixes (resolved)

Bug closed per bug fixed:

- Fixed bugs were peer-reviewed (intensive QA activities)
- Intensive peer-reviews in hybrid projects (especially Core and Tobago project)
- No peer-reviews found in pure OSS projects (e.g. Tomcat)



Verified solution ratio for resolved bugs in the reviews projects.

Summary

- The analysis of current OSS projects based on the Sourceforge Database projects showed that most of the projects are in an early stage of development or at the end of the life-cycle (about 70%).
- Only a small subset of projects have reached maturity and produce stable releases.
- Pure OSS projects (e.g., Apache Tomcat projects) consists of volunteers without any formal processes and only little quality assurance activities.
- Important OSS hybrid projects (e.g., MyFaces) are supported by industry and require more formal processes and QA activities.
- Important stakeholders of OSS projects are user, developers, committers, and project managers.
- The purposed QA framework provides a process for project and product improvement based on bug fix handling in OSS projects.

Conclusion and Further Work

Conclusion

- Based on expert interviews we identified 4 major metrics (Defect detection frequency, defect detection effectiveness, defect closure time, and ratio of verified solutions) which indicate the level of QA integration and evaluated them on 2 large Apache projects Tomcat and MyFaces.
- Main results were that hybrid projects include intensive peer-reviews for fixed bugs, which results in a longer bug closure time.
- Additionally, the number of real bugs is notable higher in hybrid projects

Future Work is

- Include additional OSS projects within this empirical investigation of QA integration for generalization purposes.
- Improve the notification of the status of OSS projects regarding different stakeholder.