City Research Online

City St George's, University of London

Citation: Gacek, C. (2004). An interdisciplinary perspective of dependability in Open Source Software. BUILDING THE INFORMATION SOCIETY, 156, pp. 685-691. ISSN 1571-5736

This is the unspecified version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/250/

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

AN INTERDISCIPLINARY PERSPECTIVE OF DEPENDABILITY IN OPEN SOURCE SOFTWARE

Cristina Gacek
School of Computing Science, University of Newcastle upon Tyne, Newcastle upon Tyne, UK, NE1 7RU

Abstract: Open Source Software (OSS) development needs to be considered during software development as to whether to develop an OSS packages, and/or whether to develop with OSS. In this paper we briefly summarize the various characteristics that can be used to describe an OSS project and then explore the possible relationships between OSS products/projects and dependability.

Key words: open source software; dependability, project's characteristics; interdisciplinary.

1. INTRODUCTION

In the last 6 years, there has been a considerable amount of interest and discussions surrounding Open Source Software (OSS) development as a viable option to be considered during software development. The focus is two-fold. On one hand there is developing OSS packages, and on the other there is developing with OSS. When embarking on a new software development task, or continuing the development of an existing software package, individuals and corporations have to make an explicit decision on whether to make this software open source or not. On the other hand, existing OSS, similarly to Commercial off the Shelf (COTS) packages, can be considered to be used within various projects as a stand-alone package, as a reusable component, or as a platform. Unlike COTS, OSS can also be used as a starting point for changes and evolution.

Considerations of developing OSS or with OSS have not been limited to systems without very stringent dependability requirements, making it imperative to evaluate the links between OSS and dependability. This has triggered discussions with respect to using OSS in the context of high dependability requirements, as well as evaluating whether there are lessons to be learnt from OSS development that can help improve dependability in software development at large.

1.1 Context

The observations and insights discussed here are a result of work in the DIRC (Interdisciplinary Research Collaboration in Dependability) project (UK EPSRC Grant GR/N13999) and further research in the topic, including several students' dissertations at the University of Newcastle upon Tyne.

A range of OSS projects were studied taking into account considerations from Computing Science, Social Sciences, Psychology, Economics, and Law. The number of projects investigated was just below 100, varying from very large and well known ones to fairly small and new ones.

Within DIRC, OSS was investigated by a multidisciplinary team to establish whether this was a silver bullet that could significantly improve dependability. It quickly became clear that the term Open Source was used differently by various people: some focused on the licensing model, many meant only that the source code was visible, far fewer identified the motivation and redundancy (of reviewing) aspects which looked as though they could actually change what was developed (DIRC 2004). Upon further investigation, it also became clear that the way in which various projects implement the "open source model" varies greatly. Most of these findings are discussed in section 2.

Later studies in OSS provided greater insight into various characteristics of OSS projects and how these are interrelated (see section 3).

2. WHAT IS OSS?

Open Source is not a precise term. Some characteristics are common to all OSS projects, but there are much more of these that vary considerably from project to project. Some of these characteristics do exist in traditional (closed source) projects too, although they are likely to be more visible in OSS projects. Previous work has looked into these in more depth (Gacek and Arief, 2004).

The main commonalities among OSS projects are:

- adherence to the Open Source Definition (OSD, 2004);

- and the fact that developers are always users.

The main variable characteristics are:

- project starting points;
- motivation;
- community:
  - balance of centralization and decentralization,
  - how their meritocratic culture gets implemented,
  - and whether the contributors are co-located or geographically distributed, and if so to what degree;
- software development support
  - modularity of the code,
  - visibility of the software architecture,
  - documentation and testing,
  - process of accepting submissions:
    - choice of work area
    - decision making process
    - submission information dissemination process,
  - and tool and operational support;
- licensing;
- and size of community and code base involved.

As highlighted above, OSS projects' characteristics can vary greatly, but that is true of all software projects. Hence, this taxonomy of characteristics can be equally applied to OSS and traditional software development projects, as depicted in table 1, adapted from Gacek and Arief (2004).

A key conclusion from the findings above is that it would not be meaningful for example to compare OSS development with traditional (closed source) software development. Any further studies and conclusions have to be based on more specific terms. The projects under study must always be further qualified by a description of their relevant details.

3. OSS AND DEPENDABILITY CONSIDERATIONS

When discussing OSS products and dependability, there are two facets to be considered. OSS products might need to be evaluated for use in a high dependability context. Additionally, OSS products provide a fertile ground for investigating how various project characteristics can impact dependability. Both of these aspects are discussed below.

Table 1. Software Projects' Characteristics (adapted from Gacek and Arief (2004))

Characteristic	**Open Source**		**Traditional**	
	Common	**Variable**	**Common**	**Variable**
Adherence to the OSD	✔		N/A	N/A
Developers are users	✔			✔
Starting points		✔		✔
Motivation		✔	✔	
Community				
Balance of centralization and decentralization		✔		✔
Meritocratic culture		✔		✔
Geographical distribution		✔		✔
Software development support				
Modularity		✔		✔
Visibility of software architecture		✔		✔
Documentation and testing		✔		✔
Accepting submissions				
Choice of work area		✔		✔
Decision making process		✔		✔
Submission information dissemination process		✔	N/A	N/A
Tool and operational support		✔		✔
Licensing		✔		✔
Size		✔		✔

3.1 Evaluating the dependability of OSS products

The term OSS usually tends to be associated with a few notorious products, such as Apache and Linux, which are far from being the average OSS product. Just like with traditional software, no assumptions can be made about the dependability of an OSS package without further knowledge about the product itself, and the process and methods used while deriving it.[1]

Typical stereotypes about OSS products/projects that relate to dependability are:

1. OSS products contain fewer faults because they have been reviewed by many people.
2. OSS products are more secure because they have been reviewed by many people.
3. OSS products have little to no design documentation available.
4. Having little design documentation available does not impact an OSS project as negatively as it would a traditional one. The reason being that

[1] We have run studies to look into OSS versus traditional software by OSS fans and by OSS sceptics. All of which reached this same conclusion.

OSS developers contribute towards development for their joy and pleasure, and consequently are less likely to leave the project than an employee to change jobs.

5. OSS products are developed by hackers in their free time, who only submit code for consideration once a high standard of quality has been achieved.

An assumption about OSS projects behind both items 1 and 2 above is that their code is extensively reviewed. The reality is that the vast majority of OSS projects consist of a single developer (as can be seen by visiting the very popular hosting sites SourceForge (2004) and Freshmeat (2004)), thus this assumption is invalid.

Item 3 above is an over generalization. It is true that most OSS products have very little documentation available, but there are successful OSS projects that have even created a sub-project dedicated to documentation. It is noteworthy that it is more common to find user and installation manuals than design documentation.

Stereotypes 4 and 5 above rely on the assumption that OSS contributors work for their pleasure in their free time. This assumption is fairly outdated. Many large corporations have employees contributing towards OSS products as part of their paid job. An immediate consequence is that these contributors are as likely to leave the OSS project as any employee in a traditional environment (making item 4 invalid). Another consequence is that these individuals are under higher pressure to deliver the product sooner and might be unable to strive for excellence (as is the assumption behind item 5).

Consequently, evaluating the dependability of an OSS product needs to be done on a case by case basis. Just like with traditional software, even different versions and releases of the same product must be considered individually.

An interesting related issue is that of who would be responsible for pursuing certification of OSS. The best model for this is yet to be identified, yet the same way that companies have been contributing towards the development of OSS products for their advantage; they could also work towards providing needed certification. How to provide the right incentives such that this certification gives companies the proper motivation is still an open question.

3.2 Deriving dependability insights from OSS products

Studies of individual OSS products can be used to derive insights into dependability in general. These are easier to conduct than studies of traditional (closed source) software projects. Much of the information related

to OSS projects is available electronically to the public, although at times requiring a considerable amount of time and effort to locate. Sources include the projects' web sites with descriptions of the projects, various versions of the software, and logs of various mailing lists documenting public email exchanges. It should also be noted that in most cases, the key players within OSS projects are quite receptive to queries by students and researchers and very willing to help.

Recent work with students at the University of Newcastle upon Tyne has started looking for correlations among some OSS projects' characteristics. To date, one strong correlation has been found to hold. It shows that the quality of the documentation available and the readability of the source code are directly related. Other possible correlations are being pursued.

Beneficial findings from studies in OSS projects can be validated in other settings. Depending on the character of those findings, plans can also be drawn towards replicating positive results within a traditional development environment.

4. CONCLUSIONS

An analysis of different modes of working among OSS projects makes it clear that there are many technical, psychological and sociological facets involved and furthermore that most of these could also be deployed in traditional software development projects.

A key conclusion from the findings discussed in this paper is that it is not meaningful to bundle all OSS projects and products together. For example, comparing OSS development with traditional (closed source) software development yields no significant result. Studies, discussions, and conclusions have to be based on more specific terms than just open source or traditional software development. These must always be further qualified by a description of the relevant details being considered.

Evaluating the dependability of an OSS product needs to be done on a case by case basis. No generalization can be made. Still OSS projects provide a good ground for investigating what different (combinations of) characteristics of software development models yield more dependable results. Findings from such investigations can then be implemented in other OSS and/or in traditional software development projects.

REFERENCE:

DIRC, 2004, Interdisciplinary Research Collaboration in Dependability (last accessed 13 May 2004); http://www.dirc.org.uk/index.php.

freshmeat, 2004, freshmeat.net: Welcome to freshmeat.net (last accessed 13 May 2004); http://freshmeat.net.

Gacek, C., and Arief, B., 2004, The many meanings of open source, IEEE Software. 21(1): 34-40.

OSD, 2004, OSI's Open Source Definition (last accessed 10 April 2004); http://www.opensource.org/docs/definition.html.

SourceForge, 2004, SourceForge.net: Welcome (last accessed 13 May 2004); http://sourceforge.net.