Open Source and Agile Methods: Two worlds closer than it seems

Hugo Corbucci[1] and Alfredo Goldman[1]

Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP) - Brazil
corbucci@ime.usp.br and gold@ime.usp.br

Abstract. Agile methods and open source software communities have different approaches to produce high quality and successful software. However agile methodologies are not very difused in open source communities nor the members of those communities follow many agile practices.

Key words: agile software development, open source software, distributed agile,

1 Introduction

Typical Open Source (OS) projects (the scope of of OS project will be narrowed according to Section 2.2) usually receive the collaboration of many geographically distant people [1]. At first glance, this argument could indicate that such projects are not candidates for the use of agile methods since some basic values seem to be missing. In this case, the distance and diversity separating developers deteriorates communication, a very important value within agile methods. However, it is common to identify some principles presented by the agile manifesto [2] on many open source software projects [5]. Being ready for changes, working with continuous feedback, delivering real features, respecting collaborators and users and facing challenges are expected attitudes from agile developers naturally found in the Free and Open Source Software (FOSS) communities.

During a workshop [3] about "No Silver Bullets" [4] held at OOPSLA 2007, agile methods and OS software development were mentioned as two failed silver bullets having both brought great benefit to the software community. During the same workshop the question was raised whether the use of several failed silver bullets simultaneously could not raise production levels by an order of magnitude. This is an attempt to suggest one of those merges to partially tackle software development problems.

2 Scopes

In order to start talking about OS and agile methods, it is necessary to first define which part of each will be analysed in this work. Agile methods in the

scope of ths work are described in Section 2.1 while the OS projects studied in this work are described in Section 2.2.

2.1 Agile methods scope

Thoughout this work, any software engineering method that follows the principles of the agile manifesto [2] will be considered and treated as an agile method. Focus will be given on the most known methods, such as eXtreme Programming (XP) [6], Scrum [7] and the Crystal family [8]. Closely related ideas will also be mentioned from the wider Lean philosophy [9] and its application to software development [10].

2.2 Open source scope

The terms "Open source software" and "Free software" will be considered the same in this work although they have some differences in their specific contexts [12, Ch. 1, Free Versus Open source]. Projects will be called to be open source (or free) if their source code is available and modifiable by anyone with the required technical knowledge, without prior consent from the original author and without any charge.

OS projects essentially controlled by a single company will not be addressed in this work. The reason for such reduction of scope is that projects controlled by companies, whether they have a public source code and accept external collaboration or not, can be run with any software engineering method established in the company since it can be enforced to the employees of this company. Some methods will work better to attract external contributions but the company still controls its own team and can maintain the software without external collaboration.

Considering this scope, it is important to characterize the people involved in such kind of projects. In 2002, the FLOSS Project [13] published a report about a survey they conducted regarding FOSS contributors. Their collected data [14] shows that 78.77% of the contributors are employed or self-employed (question 42) and that only 50.82% of the OS community are software developers while 24.76% do not earn their main income with software development (question 10). In addition to those results, the survey presents the fact that 78.78% of the collaborators consider their OS tasks more joyful (question 22.2) than their regular activities and 42.3% also consider them better organized (question 22.4). As an outcome of those results, we could say that OS contributors perceive their activities both pleasurable and effective.

Another survey [15] points out that 74% of open source projects have teams with up to 5 people and 62% of the contributors work with each other over the Internet and never met physically. It is therefore critical for those projects to have an adequate software process that fits those characteristics and is not a burden on the volunteer work.

3 How closely related are Open source and Agile?

In Martin Fowler's first version of "The New Methodology" [11], he included open source software development as part of the new methodology of software development along with now well known Agile methods. He decided to remove it from the final article because Eric Raymond's description of the development process from "The Cathedral and the Bazaar" [16] is not very well define and is closer to an experience report than to a description of a process. However, Raymond's text presents several actions that could be related to the agile manifesto [2].

OS communities are created are composed of individuals interacting between each other

4 Surveys

4.1 To the FLOSS community

4.2 To the Agile community

5 Survey results

5.1 Individual results from the FLOSS community

5.2 Individual results from the Agile community

5.3 Crossed results

6 Conclusion

Acknowledgments. This work was supported by the QualiPSo project [28].

References

1. Bert J Dempsey and Debra Weiss and Paul Jones and Jane Greenberg: A quantitative profile of a community of open source Linux developers (1999)
2. Kent Beck and Alistair Cockburn and Ward Cunningham and Martin Fowler and Ken Schwaber and al.: Manifesto for Agile Software Development, http://agilemanifesto.org (2001)
3. Dennis Mancl and Steven Fraser and William Opdyke: No silver bullet: a retrospective on the essence and accidents of software engineering (2007)
4. Frederick P. Brooks, Jr.: No Silver Bullet: Essence and Accidents of Software (1987)
5. Ron Goldman and Richard P. Gabriel: Innovation Happens Elsewhere: Open Source as Business Strategy (2005)
6. Kent Beck and Cynthia Andres: Extreme Programming Explained: Embrace Change, 2nd Edition (2004)

7. Ken Schwaber: Agile Project Management with Scrum (2004)
8. Alistair Cockburn: Agile Software Development (2002)
9. Taiichi Ohno: Toyota Production System: Beyond Large-Scale Production (1998)
10. Mary Poppendieck and Tom Poppendieck: Introduction to Lean Software Development (2005)
11. Martin Fowler: The New Methodology, http://martinfowler.com/articles/newMethodologyOriginal.html
12. Karl Fogel: Producing Open Source Software (2005)
13. International Institute of Infonomics - University of Maastricht: Free/Libre/Open Source Software: Survey and Study - Report, http://www.flossproject.org/report/
14. International Institute of Infonomics - University of Maastricht: Free/Libre/Open Source Software: Survey and Study - Report, http://www.flossproject.org/floss1/stats.html
15. Christian Robottom Reis: Caracterização de um Processo de Software para Projetos de Software Livre (2003)
16. Eric S. Raymond: The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary (1999)
17. Andy Oram: Why Do People Write Free Documentation? Results of a Survey (2007)
18. Dirk Riehle: The Economic Motivation of Open Source Software: Stakeholder Perspectives (2007)
19. Jeff Sutherland and Anton Viktorov and Jack Blount and Nikolai Puntikov: Distributed Scrum: Agile Project Management with Outsourced Development Teams (2007)
20. Frank Maurer: Supporting Distributed Extreme Programming (2002)
21. Kent Beck: Tools for Agility, http://www.microsoft.com/downloads/details.aspx?FamilyID=ae7e07e8-0872-47c4-b1e7-2c1de7facf96 (2008)
22. Nachiappan Nagappan and Prashant Baheti and Laurie Williams and Edward Gehringer and David Stotts: Virtual Collaboration through Distributed Pair Programming (2003)
23. Dan North: Behaviour Driven Development, http://dannorth.net/introducing-bdd
24. Danilo Sato and Alfredo Goldman and Fabio Kon: Tracking the Evolution of Object-Oriented Quality Metrics on Agile Projects (2007)
25. J. Surowiecki: The Wisdom of Crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations (2004)
26. Don Tapscott and Anthony D. Williams: Wikinomics: How Mass Collaboration Changes Everything (2006)
27. Yochai Benkler: The Wealth of Networks: How Social Production Transforms Markets and Freedom (2006)
28. Qualipso — Trust and Quality in Open Source systems, http://www.qualipso.org/