

A Systematic Mapping Study on the Open Source Software Development Process

Silvia T. Acuña, John W. Castro
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente 11
28049 Madrid, Spain

silvia.acunna@uam.es, john.castro@estudiante.uam.es

Oscar Dieste, Natalia Juristo
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte, Madrid, Spain
odieste@fi.upm.es, natalia@fi.upm.es

Abstract—Background: There is no globally accepted open source software development process to define how open source software is developed in practice. A process description is important for coordinating all the software development activities involving both people and technology. **Aim:** The research question that this study sets out to answer is: What activities do open source software process models contain? The activity groups on which it focuses are Concept Exploration, Software Requirements, Design, Maintenance and Evaluation. **Method:** We conduct a systematic mapping study (SMS). A SMS is a form of systematic literature review that aims to identify and classify available research papers concerning a particular issue. **Results:** We located a total of 29 primary studies, which we categorized by the open source software project that they examine and by activity types (Concept Exploration, Software Requirements, Design, Maintenance and Evaluation). The activities present in most of the open source software development processes were Execute Tests and Conduct Reviews, which belong to the Evaluation activities group. Maintenance is the only group that has primary studies addressing all the activities that it contains. **Conclusions:** The primary studies located by the SMS are the starting point for analyzing the open source software development process and proposing a process model for this community. The papers in our paper pool that describe a specific open source software project provide more regarding our research question than the papers that talk about open source software development without referring to a specific open source software project.

I. INTRODUCTION

Open Source Software (OSS) has become a key informatics component in recent years. It plays a major role in both business and educational information technologies [1], and powers a service industry worth thousands of millions of dollars to companies like Red Hat, Novell and IBM [2]. The growing importance of OSS in recent years has led researchers to study how open source processes differ from conventional software engineering processes. This research covers many aspects of open source code development, such as stakeholder motivation [3], source code repositories [Mock02], management style [4] and requirements [5]. There is also research work describing the development process enacted in

some popular OSS projects. For example, [Mock00] [Mock02] study the process enacted in Apache. According to Scacchi [Scac04], the starting point for understanding the open source community development process is to investigate how different communities develop OSS. Yet there is no internationally accepted OSS development process model defining how OSS is developed in practice [Scac04]. A process description is important for coordinating all the software development activities, including both people and technology. Coordination is enabled by giving process engineers the chance to collectively discuss and administer the dependencies between people, processes and technologies [6]. For this reason, we have conducted a systematic mapping study to investigate which activities OSS development models contain in conformance with a procedure proposed by Kitchenham et al. [7]. Systematic mapping is a methodology for investigating the literature on a field of particular interest for the purpose of determining the nature, scope and number of published primary studies [6]. Mapping studies categorize primary studies to give an overview of the field of research under consideration. There is now only one article [8] documenting a systematic mapping study related to agile, including open source, practices. But there are no papers using mapping studies addressing research on activities that are part of the OSS development process.

This paper describes the mapping study that we conducted, focusing on the key elements of the research protocol used and the description of the results. The research protocol specifies the methods that will be used to undertake a specific SMS. The remainder of the paper is organized as follows. Section II describes the key elements of the research protocol and describes how the mapping study was conducted. Section III reports the results of the research. Section IV synthesizes the mapping study. Finally, Section V outlines the conclusions.

II. STUDY RESEARCH SELECTION

The aim of the mapping study is to answer the following research question:

RQ: What activities do OSS process models contain?

We started our mapping study by identifying keywords and some search strings that we inferred from the research question. We then ran a trial search. This search returned some

papers that were okayed and rounded out by two expert software engineering researchers. The search strings finally used were: Open Source AND Software Process Model, Open Source AND Software Development Process, Open Source AND Development Process, Free Source AND Software Process Model, Free Source AND Software Development Process, Free Source AND Development Process.

The electronic databases (DB) used in the mapping study were: IEEE Xplore, ACM Digital Library, SpringerLink, Science Direct and Scopus. We applied each of the six defined search strings to each of the selected DBs. The search period began on 1 January 1990. We set 31 March 2010 as the publication deadline for the search. The inclusion and exclusion criteria that we used to determine which primary studies had a bearing on our research question were:

Inclusion criteria: The paper title must contain the words ‘open source’ or ‘free source’; **OR** The keywords refer to the open source software development process; **OR** The abstract alludes to open source software development process issues; **AND** The paper describes the open source software development process; **OR** The paper lists open source software development process activities; **OR** The paper lists free source software development process activities; **OR** The paper discusses the development process enacted in a particular open source software project; **OR** The paper presents an open source software development process proposal.

Exclusion criteria: The paper does not discuss the open source software development process; **OR** The paper does not present open source software development process activities; **OR** The paper does not present free source software development process activities.

Table I shows the number of articles located by applying the six search strings, as well as the number of selected candidate papers. The candidate papers are all studies that comply with the inclusion criteria applied to the title and

keywords only. Applying this strategy, we were able to quickly screen the search results by reducing the number of papers for thorough examination from 12,269 to 621 (just 5.1% of the total). This set of candidate papers does not contain duplicates.

TABLE I. TOTAL NO. ARTICLES LOCATED IN EACH DB

Search Term	Retrieved	Candidates	Primary Studies
IEEE Xplore	387	89	8
ACM Digital Library	6.120	284	16
SpringerLink	2.459	147	3
Science Direct	199	21	1
Scopus	3104	80	1
TOTAL	12.269	621	29

Finally, 29 primary studies were selected after scrupulously applying the inclusion and exclusion criteria. Worthy of mention at this point is that two of the primary studies were obtained by reading the references of candidate articles. These two new primary studies were not stored in any of the five DBs used. The mapping study reported here includes only English papers, as all the search terms are defined in that language. This poses a threat to its validity. Contrary to Kitchenham’s recommendations [7], the primary studies were selected by just one of the authors. In this case, however, there is no risk of primary studies being biased or rejected as the results of the selection process were checked against an independent selection process conducted by another of the authors as part of a comparative case study on the performance of novice and expert researchers.

III. RESULTS

The selected primary studies are listed in Appendix A, and an overview is given in Fig. 1.

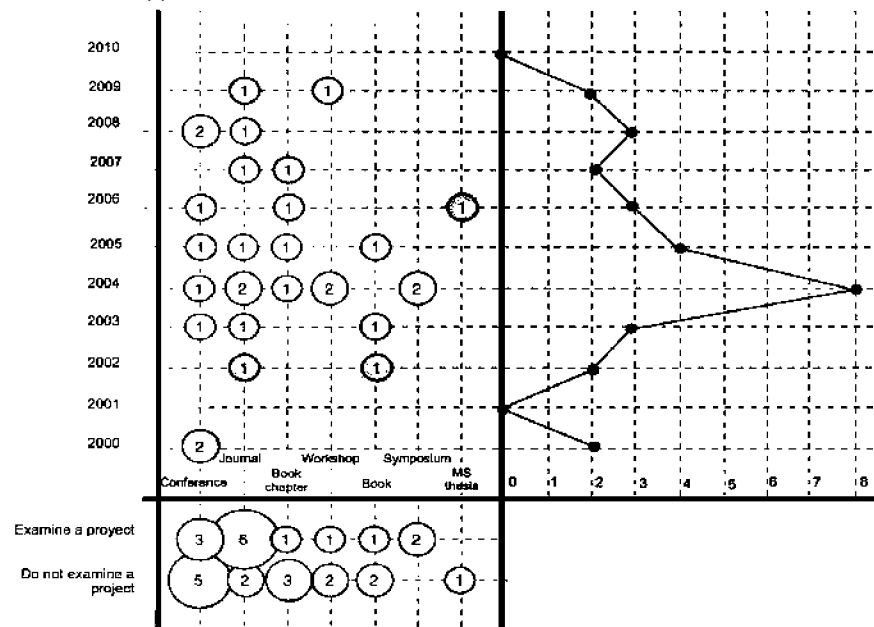


Figure 1. Mapping showing the primary study distribution depending on whether or not they examine a particular project, including publication type and year

Fig. 1 is a bubble chart basically composed of two XY scatter plots with bubbles at the category intersections (left). The categories are determined by primary study publication year, type (journal, conference proceedings, etc.) and whether or not the paper analyses a particular OSS project. The size of each bubble is determined by the number of primary studies that have been classified as belonging to the pair of categories corresponding to the bubble coordinates. Fig. 1 (right) shows the number of primary studies by publication year.

As Fig. 1 shows, there was a constant growth in research papers from 2002 to 2004, which is a sign of there being sizeable interest in understanding the development process enacted by the open source community. Then the number of published papers dropped until 2006, as of when it stabilized. As regards publication type, the selected primary studies were mainly published in journals and conference proceedings (60%), although they also included book chapters, workshops, books, symposiums and master theses, especially in the first half of the examined period (2002-2006). The fact that studies have been published in specialized peer-reviewed media in recent years suggests that the OSS field is gradually maturing.

Finally, note that half of the papers describe the OSS development process enacted in a particular OSS Project. This is an important point because the activities described by these papers indicate how OSS is really developed in practice. Appendix A lists the particular projects studied in each paper. The most popular OSS projects are FreeBSD and Apache Server, which were examined in 6 out of 14 papers (43%).

IV. SYNTHESIS

We have categorized the primary studies by activity types. To do this, we consider just five activity groups based on IEEE International Standard 1074:2006 [9]: *Concept Exploration*, *Software Requirements*, *Design*, *Maintenance and Evaluation*. We have used SWEBOK [10] to categorize software requirements. We can use this categorization to set OSS development against the traditional development process. We can also determine which traditional development process activities are of use to the OSS community. Usefulness can be inferred from the number of studies that include a particular activity: To categorize the papers as described above, we read the primary studies paying special attention to the sections describing the development process enacted by the OSS community. We then extracted the name and description of each activity from the description of the OSS development process. We used this description to match each activity with its counterpart in IEEE International Standard 1074:2006 [9], taking into account the purpose of the activity but not the applied methods and techniques.

A. Concept Exploration

Fig. 2 illustrates the primary studies referencing activities that have been matched with the *Concept Exploration* activity group in IEEE International Standard 1074:2006 [9]. The primary studies are cited using the paper ID that they are allocated in Appendix A. This group contains the activities: *Identify Ideas or Needs*; *Formulate Potential Approaches*; *Conduct Feasibility Studies* and *Refine and Finalize the Idea or*

Need. The grey boxes with rounded corners denote *Concept Exploration* group activities that are examined in primary studies. The only activity not mentioned in primary studies is *Conduct Feasibility Studies*.

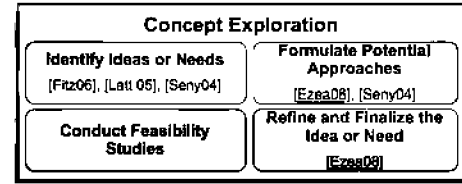


Figure 2. Primary studies examining activities matching Concept Exploration

The underlined primary studies are papers that study a particular OSS project. As Fig. 2 shows, Ezeala et al. [Ezea08] reported the only OSS project that enacted activities matching with Concept Exploration group activities (namely, *Formulate Potential Approaches* and *Refine and Finalize the Idea or Need*).

B. Software Requirements

Fig. 3 illustrates the primary studies referencing activities that have been matched with the *Software Requirements* activity group. The activities that have most primary studies are clearly *Requirements Elicitation* and *Requirements Specification*. Only one primary study was matched with each of the *Requirements Negotiation*, *Requirements Validation* and *Prioritize and Integrate Software Requirements* activities, suggesting that these are low-profile activities in the development process enacted by the OSS community.

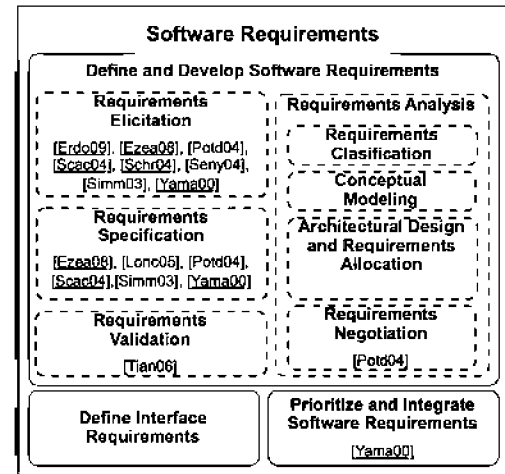


Figure 3. Primary studies examining activities matching Soft. Requirements

The underlined primary studies are papers that study a particular OSS project. As illustrated in Fig. 3, most of the primary studies that match the *Software Requirements* group belong to primary studies that examine a particular OSS project.

C. Design

Fig. 4 illustrates the primary studies referencing activities that have been matched with the *Design* activity group. This

group contains the activities: *Perform Architectural Design*, *Design Database*, *Design Interfaces* and *Perform Detailed Design*. As shown in Fig. 4, the only activity examined by primary studies is the *Perform Architectural Design*, signifying that there are no activities in the OSS development process akin to the *Design Database*, *Design Interfaces*, and *Perform Detailed Design* activities enacted in the traditional software development process.

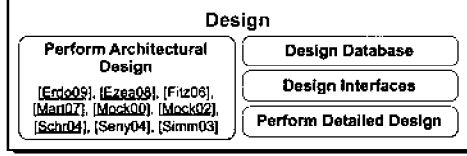


Figure 4. Primary studies examining activities matching Design

The primary studies underlined in Fig. 4 are papers that study a particular OSS project. They are a majority in this group, which is a sign of the importance of the *Perform Architectural Design* activity in particular OSS projects.

D. Maintenance

Fig. 5 illustrates the primary studies referencing activities that have been matched with the *Maintenance* activity group. This group contains the activities: *Identify Software Improvement Needs*, *Implement Problem Reporting Method* and *Reapply SPLCP*. The activities referenced in most primary studies are *Identify Software Improvement Need* and *Implement Problem Reporting Method*. As Fig. 5 shows, most primary studies study a particular OSS project, which is a sign of the importance of the *Maintenance* activity group to the OSS community.

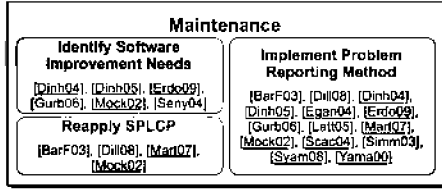


Figure 5. Primary studies examining activities matching Maintenance

E. Evaluation

Fig. 6 illustrates the primary studies referencing activities that have been matched with the *Evaluation* activity group. This group contains the activities: *Conduct Reviews*, *Create Test Data*, *Report Evaluation Results*, *Confirm Security Accreditation*, *Create Traceability Matrix*, *Conduct Audits*, *Develop Test Procedures* and *Execute Tests*. Only three of these activities have been matched with primary studies referencing activities akin to *Conduct Reviews*, *Execute Tests* and *Report Evaluation Results*. The activities examined by most primary studies are *Conduct Reviews* and *Execute Tests*, whereas there are only three primary studies referencing *Report Evaluation Results*. As Fig. 6 shows, the traditional development process activities belonging to the *Evaluation* activity group that are of most use to the OSS community for development process enactment are *Conduct Reviews* and *Execute Tests*. The primary studies underlined in Fig. 6 are

papers that study a particular OSS project. As Fig. 6 shows, most of the primary studies that matched with the *Evaluation* activity group are examined in primary studies that study a particular OSS project.

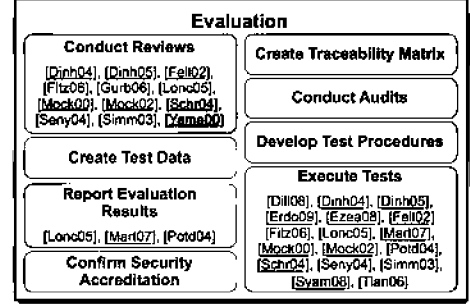


Figure 6. Primary studies examining activities matching Evaluation

V. CONCLUSIONS

This paper describes the systematic mapping study conducted to answer the research question, What activities do OSS process models contain? We located a total of 29 primary studies, published mainly in journals and conference proceedings. Half of the primary studies study a particular OSS project. The most popular OSS projects studied are FreeBSD and Apache Server. Primary studies addressing OSS development processes containing activities that match with the greater part of traditional process activities are precisely papers that study OSS projects. This signifies the importance of studying particular OSS projects.

Traditional software development activities most often present in the development process enacted by the OSS community are: *Identify Ideas or Needs*, *Requirements Elicitation*, *Perform Architectural Design*, *Implement Problem Reporting Method*, *Conduct Reviews* and *Execute Tests*. None of the primary studies describes an OSS development process with an activity that we have not been able to map to the traditional process activity groups covered in this paper. *Maintenance* is the only traditional process activity group that has primary studies addressing all the activities that it contains. The *Evaluation* activity group contains the two activities most often cited in primary studies: *Conduct Reviews* and *Execute Tests*. This is evidence of the key role that OSS users play as bug reporters and beta testers. These two activities account for about 70% of all primary studies. This is a sign of their importance in the development process enacted by the OSS community. Primary studies do not reference any activities related to: *Conduct Feasibility Studies*, *Define Interface Requirements*, *Design Database*, *Design Interfaces* and *Perform Detailed Design*. The OSS community efforts focus primarily on software maintenance and evaluation (*Conduct Reviews* and *Execute Tests*).

REFERENCES

- [1] W. Scacchi, "Free/open source software development: recent research results and emerging opportunities," in *ESEC-FSE'07*, ACM, 2007, pp. 459-468.
 - [2] P. Rooney, "IBM builds dedicated sales channel for red hat, novell linux," 2005. [Online] Available: <http://www.crn.com/software/175002626> [Accessed in february 2012].
 - [3] A. Hars, and S. Ou, "Working for free? motivations for participating in open-source software projects," *International Journal of Electronic Commerce*, 6(3), 2002, pp. 25-39.
 - [4] S. O'Mahony, "Guarding the commons: how community managed software projects protect their work," *Research Policy*, 32(7), 2003, pp. 1179-1198.
 - [5] W. Scacchi, "Understanding requirements for open source software," K. Lyytinen et al. (Eds.), *Design Requirements Workshop*, Springer-Verlag Berlin Heidelberg, LNBP 14, 2009, pp. 467-494.
 - [6] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08)*, 2008, pp. 71-80.
 - [7] B. A. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering version 2.3," Technical Report S.o.C.S.a.M. Software Engineering Group, Keele University and Department of Computer Science University of Durham, 2007.
 - [8] S. Jalali, and C. Wohlin, "Agile practices in global software engineering - a systematic map," in *5th IEEE International Conference on Global Software Engineering (ICGSE'10)*. Princeton, New Jersey, USA, 2010, pp. 45-54.
 - [9] IEEE Std 1074:2006, "IEEE Standard for Developing Software Life Cycle Processes," IEEE Computer Society, 2006.
 - [10] IEEE Computer Society Professional Practices Committee, "Guide to the Software Engineering Body of Knowledge - SWEBOK, Version 2004," IEEE Computer Society. Los Alamitos, CA, 2004.
- APPENDIX A: ARTICLES INCLUDED IN REVIEW
- This appendix lists the references of the primary studies located during the mapping study described in this paper.
- [BarF03]: Bar, M., and Fogel, K. (2003). *Open Source Development with CVS - 3rd Edition*. Libro Online. http://cvsbook.red-bean.com/OSDevWithCVS_3E.pdf.
 - [Capi07]: Capiluppi, A., and Michlmayr, M. (2007). From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects. In *IFIP International Federation for Information Processing*, 234, Open Source Development, Adoption and Innovation, eds. Feller, J., Fitzgerald, B., Scacchi, W., Sillitti, A., (Boston; Springer), pp. 31-44.
 - [Dill08]: Dillon, T.S., and Simmons, G. (2008). Semantic Web Support for Open-Source Software Development. In *Proceedings of the IEEE International Conference on Signal Image Technology and Internet Based Systems (SITIS'08)*, pp. 606-613.
 - [Dinh04]: Dinh-Trong, T., and Bieman, J.M. (2004). Open Source Software Development: A Case Study of FreeBSD. In *10th International Symposium on Software Metrics (METRICS'04)*, pp. 96-105.
 - [Dinh05]: Dinh-Trong, T., and Bieman, J.M. (2005). The FreeBSD Project: A Replication Case Study of Open Source Development. *IEEE Transactions on Software Engineering*, 31, pp. 481-494.
 - [Egan04]: Egan, S. (2004). The Open Source Development Process. In *Open Source Messaging Application Development: Building and Extending Gaim*. Chapter 2, Apress, pp. 23-36.
 - [Erdo09]: Erdogmus, H. (2009). A Process That Is Not. *IEEE Software*, 26(6), pp. 4-7.
 - [Ezea08]: Ezeala, A., Kim, H., and Moore, L.A. (2008). Open Source Software Development: Expectations and Experience from a Small Development Project. In *46th Annual Southeast Regional Conference on ACM-SE'08*, Auburn, AL, USA, pp. 243-246.
 - [Ezea08]: Ezeala, A., Kim, H., and Moore, L.A. (2008). Open Source Software Development: Expectations and Experience from a Small Development Project. In *46th Annual Southeast Regional Conference on ACM-SE'08*, Auburn, AL, USA, pp. 243-246.
 - [Fell02]: Feller, J., and Fitzgerald, B. (2002). *Understanding Open Source Software Development*. In Publisher Addison-Wesley Professional, London.
 - [Fitz06]: Fitzgerald, B. (2006). The Transformation of Open Source Software. *Forthcoming in MIS Quarterly*, 30(3), pp. 1-26. Including subseries Lecture Notes in Artificial Intelligence - 3840 LNCS.
 - [Foge05]: Fogel, K. (2005). *Producing Open Source Software: How to Run a Successful Free Software Project*. Publisher O'Reilly Media, Inc.
 - [Fugg03]: Fuggetta, A. (2003). Open Source Software: An Evaluation. *Journal of Systems and Software*, vol. 66, pp. 77-90.
 - [Fugg04]: Fuggetta, A. (2004). Open Source and Free Software: A New Model for The Software Development Process? In *European Journal for the Informatics Professional, UPGRADE*, vol. V(5), pp. 22-26.
 - [Gurb06]: Gurbani, V.K., Garvert, A., and Herbsleb, J.D. (2006). A Case Study of a Corporate Open Source Development Model. In *28th ACM International Conference on Software Engineering (ICSE'06)*, pp. 472-481.
 - [Latt05]: Latterman, C., and Stieglitz, S. (2005). Framework for Governance in Open Source Communities. In *38th Annual Hawaii International Conference on System Sciences*, pp. 1-11.
 - [Lell09]: Lelli, F., and Jazayeri, M. (2009). Community Support for Software Development in Small Groups: The Initial Steps. In *2nd International Workshop on Social Software Engineering and Applications (SoSEA'09)*, pp. 15-22.
 - [Lonc05]: Lonchamp, J. (2005). *Open Source Software Development Process Modeling*. In *Software Process Modeling* edited by Silvia T. Acuña, Natalia Juristo, Chapter 2, Springer, pp. 29-64.
 - [Mart07]: Martin, K., and Hoffman, B. (2007). An Open Source Approach to Developing Software in a Small Organization. *IEEE Software*, 24, pp. 46-53.
 - [Mock00]: Mockus, A., Fielding, R.T., and Herbsleb, J. (2000). A Case Study of Open Source Software Development: The Apache Server. In *21st International Conference on Software Engineering (ICSE'00)*, Los Angeles, CA, pp. 263-272.
 - [Mock02]: Mockus, A., Fielding, R.T., and Herbsleb, J. (2002). Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3), pp. 309-346.
 - [Mong04]: Monga, M. (2004). From Bazaar to Kibbutz: How Freedom Deals with Coherence in the Debian Project. In *4th Workshop on Open Source Software Engineering Engineering - 26th International Conference on Software Engineering (ICSE'04)*, pp. 71-75.
 - [Potd04]: Potdar, V., and Chang, E. (2004). Open Source and Closed Source Software Development Methodologies. In *4th Workshop on Open Source Software Engineering Engineering - 26th International Conference on Software Engineering (ICSE'06)*, pp. 105-109.
 - [Scac04]: Scacchi, W. (2004). Free and Open Source Development Practices in the Game Community. *IEEE Software*, 21(1), pp. 59-66.
 - [Schr04]: Schroeder, W.J., Ibáñez, L., and Martin, K.M. (2004). Software Process: The Key to Developing Robust, Reusable and Maintainable Open-Source Software. In *2nd IEEE International Symposium on Biomedical Imaging (ISBI'04)*, vol.1, pp. 648-651.
 - [Seny04]: Senyard, A., and Michlmayr, M. (2004). How to Have a Successful Free Software Project. In *11th Asia-Pacific Software Engineering Conference (APSEC'04)*. IEEE Computer Society. Busan, Corea del Sur, pp. 84-91.
 - [Simm03]: Simmons, G.L., and Dillon, T. (2003). Open Source Development and Agile Methods. In *7th IASTED International Conference Software Engineering and Applications*. Marina del Rey, CA, USA, pp. 523-527.
 - [Syam08]: Syamlal, M., O'Brien, T.J., Benyahia, S., Gel, A., and Pannala, S. (2008). Open-Source Software in Computational Research: A Case Study. *Modelling and Simulation in Engineering*, vol 2008, pp. 1-10.
 - [Tian06]: Tian, Y. (2006). *Developing an Open Source Software Development Process Model Using Grounded Theory*. Universidad of Nebraska at Lincoln, NB, USA.
 - [Yama00]: Yamauchi, Y., Yokozawa, M., Shinohara, T., and Ishida, T. (2000). Collaboration with Lean Media: How Open-Source Software Succeeds. In *ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, pp. 329-338.