

Evaluation of Open Source Software with QualiPSO OMM: a case for Bungeni and AT4AM for All

Kennedy Ndenga Malanga¹, Jean Mehat², Ivaylo Ganchev³, John Wandeto⁴ and Casper Shivachi Shikali⁵

Abstract - In this study, we tested the reliability of QualiPSO OMM by using it to evaluate the maturity and quality of Bungeni and AT4AM for All Parliamentary Legislative Systems. We found out that Qualipso OMM assessment has limitations; it relies heavily on product documentation, it is tedious to implement and it does not test code quality effectively. Evaluation and selection of Open Source Software are main challenges affecting adoption of OSS projects. Various methods of evaluating quality of Open Source Software exist, for example Qualipso Open Maturity Model. Unfortunately very few of these methods have been used widely to test OSS projects due to various reasons. We concluded that Qualipso OMM's partial evaluation cannot be relied upon to make decisions on adopting or refraining from an OSS project. Finally, we realized that community metrics can be used to measure quality of OSS projects effectively since such projects are community driven.

Index terms - Evaluation, maturity, metrics, Open Source Software, quality.

XIV. INTRODUCTION

MANY organizations using Free/Open Source Software (FOSS) are dealing with a major problem of selecting the most appropriate software products corresponding to their needs. Open Source Software products targeting the same kind of applications are now very common. This makes choosing an OSS product a tricky task. Quality is one factor that should be considered when choosing among similar open source solutions. The selection process becomes more complicated considering that reliable tools for assessing Open source software quality are rare. It is more challenging to evaluate an OSS project as compared to a proprietary project since the former is usually developed in a dynamic environment. Most organizations select FOSS projects using ad-hoc techniques [1]. Research has shown that practitioners rarely use formal selection procedures. Instead they select OSS products based on familiarity or recommendations by colleagues [4]. Stol and Babar [4] believe that one of the main challenges of using OSS is its evaluation and selection. Different indicators can provide hints on the quality of a FOSS project, for example: the

number of users, the longevity of the project, documentation available on-line, number of product downloads and number of bugs reported, etc. All these indicators can have different interpretations. Therefore, it is important to have a structured set of criteria to be used in assessing the quality of OSS projects [2].

A. Methods of Evaluating OSS

Various methods of evaluating quality of Open Source Software have been developed over time which include; Open Source Maturity Model (OSMM) from Cap Gemini, Open Source Maturity Model (OSMM) from Navica, Methodology of Qualification and Selection of Open Source Software (QSOS), Open Business Readiness Rating (OpenBRR), Open Business Quality Rating (Open BQR) and QualiPSO Open Source Maturity Model (OMM). Stol and Babar [4] argue that it is difficult for practitioners to choose a suitable evaluation method. They continue to assert that the key contributing factor to this situation is lack of clarity of the evaluation methods. Petrinja et al. [2] did a comprehensive comparison amongst OpenBRR, QSOS and QualiPSO OMM assessment models. From their analysis they concluded that in as much as QualiPSO OMM performed as good as OpenBRR and QSOS, it was perceived as being better than the other two. Based on this conclusion, we chose to use QualiPSO OMM to evaluate the maturity and quality of Bungeni and AT4AM for All Parliamentary and Legislative Systems (PLS).

B. Objectives

The first objective of this study was to assess the reliability of QualiPSO Open Maturity Model of evaluating Open Source Software. Second objective was to establish limitations of this tool when evaluating OSS projects based on partial approach. We were also interested in determining maturity levels of Bungeni and AT4AM for All Parliamentary and Legislative Systems using partial QualiPSO OMM evaluation at basic level. Lastly we proposed a better way of evaluating Open Source Software projects.

This study contributes to knowledge in quality of Open Source software since it highlights limitations of using partial QualiPSO OMM assessment tool. It also suggests areas that could be focused on when modeling an OSS quality evaluation tool.

¹University of Paris 8, Laboratoire d'informatique avancée de Saint-Denis (LIASD), 2, rue de la Liberté 93526 Saint-Denis Cedex 02, France (phone: +33 1 49 40 64 15 e-mail: liasd@univ-paris8.fr)

²Dedan Kimathi University of Technology, School of Computer Science & IT, P.O. Box 657-10100 Nyeri, Kenya (phone: +254 061 2050000 or +254713 835 965 e-mail: info@dkut.ac.ke)

XV. OVERVIEW OF QUALIPSO OMM, BUNGENI AND AT4AM FOR ALL

A. Qualipso Open Maturity Model (OMM)

Qualipso Open Maturity Model (OMM) is a process model for OSS development by developers and integration of OSS components by integrators [6]. It is organized as an evolutionary model, inspired by the Capability Maturity Model Integration (CMMI) [6]. Qualipso OMM is organized in three levels i.e. Basic, Intermediate and Advanced levels. Each level builds on the lower levels and also on the Trustworthy elements (TWEs) at the lower level [6]. A trustworthy element (TWE) is a specific factor or aspect of a software development process, or of product results that indirectly influence the perception of the trustworthiness of a FOSS development process. Trust correlates with the quality of a software product and it is influenced by the perception of the quality of trustworthiness of a software product [6]. OMM is based on twelve trustworthy elements as enumerated below [6];

- i. Product Documentation (PDOC)
- ii. Popularity of the Software Product (REP)
- iii. Use of Established and Widespread Standards (STD)
- iv. Availability and Use of a (product) Roadmap (RDMP)
- v. Quality of Test Plan (QTP)
- vi. Relationship between Stakeholders (Users, Developers etc) (STK)
- vii. Licenses (LCS)
- viii. Technical Environment (Tools, Operating System, Programming Language, Development Environment.) (ENV)
- ix. Number of Commits and Bug Reports (DFCT)
- x. Maintainability and Stability (MST)
- xi. Contribution to FLOSS Product from SW Companies (CONT)
- xii. Results of Assessment of the Product by 3rd Party Companies (RASM)

Figure 1 summarizes the TrustWorthy elements in three levels i.e. Basic level, Intermediate level and Advanced level.

To establish the maturity level of an OSS project using Qualipso OMM, it requires an aggregation of the assessment from the practice level up to the maturity level. The following rating algorithms were used to determine the rating of an OSS project at various levels [3];

Rating algorithm of a Practice – R(P);

$$R(P_i) = \frac{\sum_{All\ metrics} Metric_i}{number\ of\ Metrics} \tag{1}$$

Rating algorithm of a Goal - R(G);

$$R(G_i) = \frac{\sum_{All\ Practices} Practice_i}{number\ of\ Practices} \tag{2}$$

Rating algorithm of a Trustworthy Element - R(TWE);

$$R(TWE_i) = \frac{\sum_{All\ Goals} Goal_i}{number\ of\ Goals} \tag{3}$$

Rating algorithm of a Maturity Level - R(ML);

$$R(ML) = \frac{\sum_{All\ Practices} P_i}{max\ \sum_{All\ Practices} P_i} \tag{4}$$

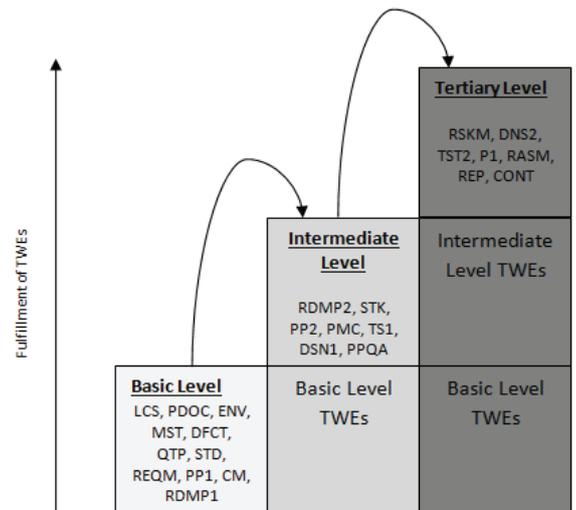


Fig.1: This figure shows three OMM maturity levels. An OSS project qualifies for an Intermediate Level evaluation once all TWEs at Basic Level have been fulfilled. Similarly, Basic Level and Intermediate Level TWEs must be fulfilled for a Tertiary Level evaluation to be carried out.

For an OMM level to be fulfilled, the Maturity Level of the OSS project being assessed should be greater than or equal to 90 percent for that level.

B. Bungeni Parliamentary and Legislative Information System

Bungeni is a system that aims at making parliaments more open and accessible to citizens. This system allows citizens to virtually get inside Parliament" or "Bungeni" the Kiswahili word for "inside Parliament"[8]. Bungeni is an Open Source Software system based on Akoma Ntoso open standards and open source applications. Akoma Ntoso is a framework that defines a set of simple technology neutral electronic representations of parliamentary, legislative and judiciary documents for e-services in a worldwide context and provides an enabling framework for the effective exchange of "machine readable" parliamentary, legislative and judiciary documents such as legislation, debate record, minutes, judgments, etc [7]. Bungeni is an initiative of Africa i-Parliament Action Plan - a programme of United Nations Department of Economic and Social Affairs (UNDESA) [8].

Like many other Open Source software, Bungeni has not been embraced by many parliaments or parliament like organizations. As of today, the stakeholders involved in

Bungeni project include: the national parliaments of Kenya, Mauritius, Mozambique, South Africa, Uganda, Zambia, Nigerian Institute for Legislative Studies, SADC Parliamentary Forum and ECOWAS Parliament [7]. According to Bungeni's project evaluation report of 29th November 2012, this Parliamentary Information System was reported as having proven to be more challenging than had been anticipated. However the report affirms that as of then, the system had matured and that it could be used not just by parliaments but also by other parliament-like organizations such as regional parliaments, municipalities and parliamentary monitoring organizations [7].

C. AT4AM for All Parliamentary and Legislative Information System

AT4AM for All is a free or an open source release of AT4AM. AT4AM is a web-based amendment authoring tool used by the European Parliament to create and table amendments on proposals and present reports of parliamentary committees of the European Commission and the Council of the European Union [12]. The European Parliament decided to use the knowledge and experience gained on the AT4AM project to develop AT4AM for All primarily to help national and regional parliaments to implement their own XML-based amendment authoring systems. Like Bungeni, AT4AM for All uses Akoma Ntoso XML format as input format for source texts and as output format for amendments [12].

Remark: Both Bungeni and AT4AM for All are Parliamentary and Legislative Systems. The reason why these projects were chosen is that they belong to the same family of software. Secondly, most research on OSS focuses on large commonly used projects. Most of the small projects with a high potential like these two projects are mostly overlooked. Despite Bungeni being described as having matured, there is no information that quantifies or describes this maturity. As Deprez and Alexandre point out, organizations want assurance regarding the quality of FOSS projects before integrating them in their solutions [1]. This could explain why adoption of Bungeni is still low. Potential adopters would like to be sure about the level of stability, the degree of support, possibility of adding new features and general continuity of Open Source software before adopting it [5]. We found it necessary to evaluate Bungeni and AT4AM for All quantitatively so as to generate information that potential adopters of these systems can use to gauge the level of maturity of the system before they decide to assimilate them into their work processes.

XVI. METHODOLOGY

A. Approach that was used

As mentioned earlier, Qualipso OMM evaluation is organized in three levels. This study focused on the Basic level of evaluation. To implement this study, a partial OMM assessment was carried out at basic level on Bungeni and AT4AM for All. Qualipso OMM provides for two approaches of assessment. The first approach is a complete assessment of the OSS project, also called internal assessment where the assessment team has got free access to all development documents, process documents and

development teams for interview. The second approach is a partial assessment, also known as external OMM assessment, where the assessment team has limited access to the development documentation, process documents and the development team [6]. The first approach is very difficult to achieve in a true Open Source Software development environment since such an environment is usually very dynamic. One reason is that OSS development is usually driven by a community. The developers who contribute to coding are dispersed geographically. They get in and out of the project at different times. Thus it is difficult to assemble the team for purposes of interviews. Secondly, access to documentation is a challenge in a typical OSS development environment. Thirdly, the first approach is feasible in a situation where the project is still under development. However projects at this stage may not be safe to use at industrial level. Projects that have matured most likely have a small support team while the core team will have moved to other projects. As a result, the second approach was chosen for this study since it applies to a majority of OSS projects and that it is the most probable approach small scale potential OSS adopters may choose to quickly evaluate a project.

Remark: Petrinja et al. [2] realized that some aspects of OMM have to be improved so as to enhance its usability. These areas include; identification of misleading questions in QTP (Quality of Test Plan) and ENV (Environment) TrustWorthy Elements, simplifying complexity of questions, simplification of questions' threshold values, automation of scoring mechanism and creation of an easy to understand description of OMM model. We took care of these issues apart from the last two during the development of data collection tool. We found the LookFor for selection of integrated management and communication tool under the Environment (ENV) TrustWorthy Element to be misleading. We realized that a project could have only one environment. As such, all the environments were treated as one group. In addition we did not assess the LookFor on testability testing since we considered testing of testability to be misleading. Any project at any level is testable.

B. Questionnaire

Using the Trustworthy Elements (TWEs) described by Qualipso.org (2014), we designed an assessment questionnaire for the Basic level of the model. We used LimeSurvey to develop an on-line questionnaire for this purpose [11]. Assessment items were based on LookFors and practices. The questionnaire was pilot tested to ensure its reliability and validity. We rated question items using Likert scale of five points where value of 4 corresponded to greater than 75 percent implementation of a practice, value of 3 - a 50 percent to 75 percent implementation of a practice, value of 2 - a 25 percent to 50 percent implementation, value of 1 - less than 25 percent implementation while 0 - for not applicable. To reduce level of bias, the assessment was carried out by participants who were geographically separated. For Bungeni we gathered information from Bungeni portal, Bungeni site and Akoma Ntoso site [10, 8, 7], while for AT4AM for All it we gathered from AT4AM for All site and Atlassian Bitbucket website [12, 9].

C. Data

Table 1 shows data that was collected from this study;

Table 1: TWE scores for Bungeni and AT4AM for All

Trust Worthy Elements (TWE)	Bungeni	AT4AM for All
LCS - Licenses management	2.73	3.12
PDOC - Product Documentation	2.44	1.75
ENV – Environment	3.06	1.31
MST - Maintainability and Stability	2.01	1.02
DFCT - Number of commits and bug reports	1.62	0
QTP - Quality of Test Plan	1.96	1.12
STD - Use of established and widespread Standards	3.19	2.04
REQM - Requirements management	0	0
PP1 - products and Project planning	0	0
CM - Configuration Management	0	0
RDMP – Roadmap	0	0

A total of seventy three practices were assessed for both projects. However out of these practices, data was available for only forty seven practices. The results indicate that both projects scored zero for Requirements management (REQM), Project Planning-1(PP1), Configuration Management (CM) and Roadmap-1 (RDMP) Trustworthy elements. However, AT4AM for All also scored zero for Number of Commits and Bug reports (DFCT) TWE.

XVII. RESULTS AND ANALYSIS

Results show that Bungeni scored better in most of the TWEs as compared to AT4AM for All. Figure 2 summarizes this comparison;

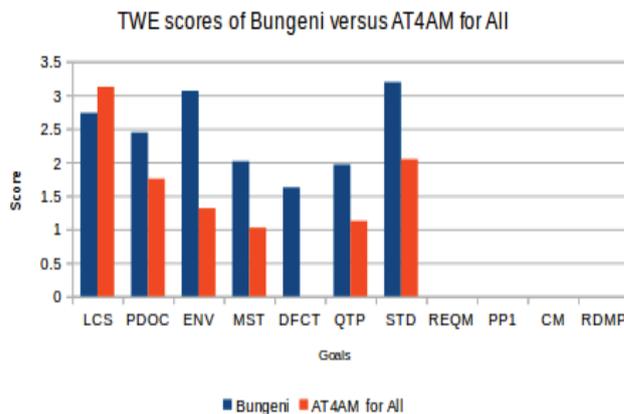


Fig. 2: A comparison graph of TWE scores of Bungeni and AT4AM for All

An analysis of fulfilled Trustworthy elements for each project was done as depicted by the graphs in figure 3 and figure 4.

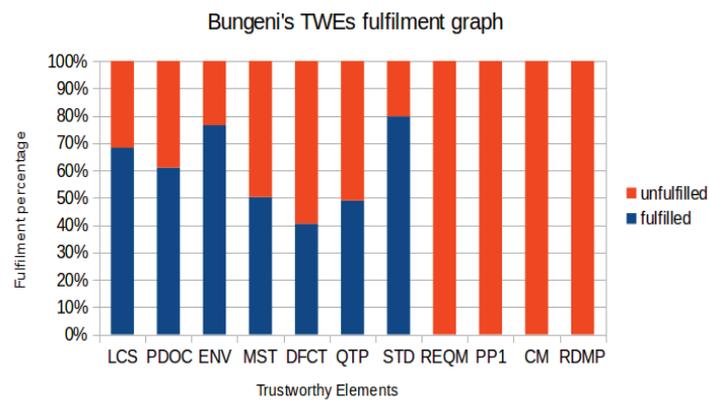


Fig. 3: Bungeni's Trustworthy Elements (TWEs) fulfillment graph

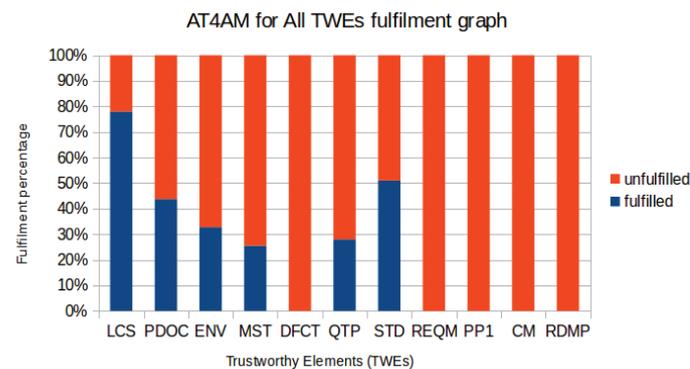


Fig. 4: AT4AM for All's Trustworthy Elements (TWEs) fulfillment graph

Using equation 4, we determined the Maturity level for the two projects. For all seventy three practices under Basic level of OMM, Bungeni scored a Maturity Level of 40.08 percent while AT4AM for All scored a Maturity Level of 29.28 percent. Qualipso OMM gives a leeway for a partial assessment not to evaluate all practices. The assessors are given freedom to select the practices to be considered for assessment [6]. Using this rationale we selected forty seven practices whose information was accessible over the internet. Subsequently we determined a Maturity Level for each project. For this scenario Bungeni scored 62.25 percent while AT4AM for All scored 45.47 percent.

A. Discussion

Partial Assessment: For partial assessment of OSS projects, the assessor does not get full insight into the project but only gets an outside view [6]. This statement raises many questions on the understanding of what an assessment is. An assessment includes all the various methods used to determine the extent of an achievement of a standard. It is very difficult to measure outside view of an item since viewing something is very subjective. Qualipso has set a standard of greater than or equal to 90 percent for an OSS project to qualify to be considered as trustworthy and of high quality. However this standard applies only for full assessment. For every kind of assessment, a standard must be involved so as to act as a reference point when making decisions. However, Qualipso OMM does not define the minimum standard for partial assessment. Since partial assessment should be able to assist a potential adopter to

make a decision for either adopting or rejecting an OSS project, there is need for some level of a minimum standard. The task of setting a standard is left to the assessors. This situation introduces high degree of subjectivity that ends up influencing the validity and reliability of the results. In a nutshell, construct validity is negatively affected since the scores achieved from partial assessment may not necessarily reflect the items being tested. For example considering the scores of Bungeni and AT4AM for All as shown in the previous section, these projects are depicted as being not mature. It may appear as if the projects have not reached a usable maturity level. Actually this is not the case. Both projects are already operational. The fact is that partial Qualipso OMM assessment does not provide valid information that potential adopters can rely on to make decisions to either adopt or refrain from a specific OSS project. However the model can be enhanced by introducing a standard of some minimum requirements that can be measured reliably for partial assessment.

The role of documentation: Evaluation criterion for Qualipso OMM is highly dependent on documentation. This model will only guarantee reliable results in situations where documentation is available. Developers working on small scale non-funded OSS projects tend to focus more on the end product rather than documentation. If such projects are assessed by Qualipso OMM, definitely they will score poorly and yet the end product could be of good quality.

The role of the development team: At some point during evaluation process using Qualipso OMM, it becomes very necessary to be in contact with a person from the development team. This is true since sixty seven percent of LookFors at the Basic level of this model may require clarification from the developer. The availability of the developer may not be guaranteed considering the dynamic nature of OSS development. For example due to code contribution, there could be no distinct owner of the project, or the developer may have stopped supporting the project. In this situation then Qualipso OMM will deliver a biased evaluation as it depends on the project's scanty documentation found on the Internet.

Satisfaction level of users: The practices on use of Established Standards (STDs) and Number of Commits and bugs (DFCT) within Qualipso OMM focus on assessing whether mechanisms for measuring satisfaction level of users have been implemented by the project. We envisaged that, user satisfaction is a very significant element in evaluating quality of software. Reza et al. maintain that user satisfaction has always been a major factor in the success of software, regardless of whether it is closed proprietary or OSS [19]. Therefore it would have been prudent for Qualipso OMM to assess the satisfaction level of users directly from users of the project being assessed instead of inferring from the developer. As mentioned earlier many OSS projects may not have the luxury to carry out elaborate surveys to assess the usability of their products.

Process versus product assessment: Qualipso OMM emphasizes the assessment of the development process of an OSS project. Zieliski and Szmuc argue that successful projects employ more mature processes than unsuccessful projects [13]. However this may not necessarily always be

the case. In as much as the process of development may be an indicator of quality, the ultimate quality is defined by the final product. Very little assessment of quality of the code for the product has been carried out by this method. It is very necessary to measure the quality of code so as to enable potential integrators understand how easy or difficult it is to integrate or reuse the code of an OSS project.

FLOSS Community metrics: Finally, Open Source Software community metrics can be used as an indicator of quality of an OSS project. Falkner [14] has defined second order FLOSS community metrics which include; crowd sourced quality, time between contribution, location of community members, evolution of contributor age, evolution of active or inactive user, time of bug report, time of bug being fixed, number of ignored contributions and distribution of commits across functional areas. If such FLOSS metrics are well understood, they can be used as a pointer towards quality of an Open Source project. Many tools are upcoming that will enable mining of data from on-line repositories for the FLOSS Community Metrics. Such tools include; MetricsGrimoire, Riscoss Analytics, OSSMETER and Eclipse Community Metrics [15 - 18]. Using the output from these tools, potential OSS adopters will be able to easily establish indicators of quality of an OSS project with less effort.

XVIII. CONCLUSION

Qualipso OMM is a good tool for evaluating Open Source Software. However it can be rendered unreliable if project documentation is not available. Qualipso OMMs partial assessment cannot be relied on when making decisions of either adopting or refraining from a particular Open Source Software project. The ultimate quality of a software project is defined in terms of the code quality and its usability. Qualipso OMM has not explored these two areas in details and yet they are very important. Generally, Qualipso OMM is bound to be effective in an environment where Open Source Software project is developed following a proprietary-like work-flow.

We should therefore build a better Open Source Software assessment tool that can automate the process of evaluation. For example an on-line evaluation tool that can automatically mine for data from repositories and generate maturity information of OSS projects. The tool should also be able to compare two or more similar OSS projects statistically in terms of quality. Finally such a tool should be reliable with an ability to predict the life expectancy of an OSS project in relation to support from the community.

Future work: Our future work entails developing a model of evaluating quality of Open Source Software projects based on Open Source Software community metrics. Such a model should be able to determine the quality of past, current, and future versions of an open source software product.

ACKNOWLEDGMENT

We sincerely thank the Advanced Computing Laboratory of Saint-Denis (LIASD) for supporting this study.

REFERENCES

- [1]. Deprez, J.-C. & Alexandre, S. (2008). Comparing assessment methodologies for free/open source software: OpenBRR and QSOS. *Product-Focused Software Process Improvement* (pp. 189203). Springer.
- [2]. Petrinja, E., Sillitti, A. and Succi, G. (2010). Comparing OpenBRR, QSOS, and OMM assessment models. *Open Source Software: New Horizons* (pp. 224238). Springer.
- [3]. Petrinja, E. and Succi, G. (2012). Assessing the open source development processes using OMM. *Advances in Software Engineering*, 2012, 8. Hindawi Publishing Corp.
- [4]. Stol, K.-J. and Babar, M. A. (2010). A comparison framework for open source software evaluation methods. *Open Source Software: New Horizons* (pp. 389394). Springer.
- [5]. QSOS core team et al. (2013). Collaborative technological watch: Qualification and Selection of Opensource Software. Retrieved Sept 13th, 2014, from QSOS: <http://www.qsos.org/>
- [6]. Qualipso.org. (2014). Retrieved August 2nd August 2014, 2014, from Qualipso: Quality platform for Open Source Software: <http://qualipso.icmc.usp.br/OMM/>
- [7]. UNDESA - Akoma Ntoso. (2014). Project Evaluation. Retrieved August 8th August 2014, 2014, from Akoma Ntoso: www.akomantoso.org
- [8]. UNDESA. (2014). Bungeni. Retrieved August 1, 2014, from Bungeni: <http://www.bungeni.org/>
- [9]. Atlassian. (2014). AT4AM code. Retrieved from Atlassian Bucket: <http://code.at4am.org/>
- [10]. UNDESA Bungeni-Portal. (2014). Project Home. Retrieved from Bungeni-Portal: <https://code.google.com/p/bungeni-portal>
- [11]. LimeSurvey. (2014, October 2014 27th). Home. Retrieved from LimeSurvey: <https://www.limesurvey.org/en>
- [12]. European Parliament. (2014, October 27th). About. Retrieved from AT4AM for All: <http://www.at4am.org/overview/>
- [13]. Zieliski, K. and Szmuc, T. (2005). Software process maturity and the success of free software projects. *Frontiers in Artificial Intelligence and Applications*, 3.
- [14]. Falkner, J. (2014, November 2). Metrics Are Fun, but which ones really matter? Retrieved from Speaker Deck: <https://speakerdeck.com/ossmetrics/metrics-are-fun-but-which-ones-really-matter>
- [15]. GitHub . (2014, November 2nd). MetricsGrimoire. Retrieved from MetricsGrimoire - Tools for software and development community analytics: <http://metricsgrimoire.github.io/>
- [16]. Milinkovitch, M. (2014, November 2nd). Eclipse Community Metrics. Retrieved from Speaker Deck: <https://speakerdeck.com/ossmetrics/eclipse-community-metrics>
- [17]. OSSMETER Project. (2014, November 2nd). Main. Retrieved from OSS-METER - Automated Measurement and Analysis of Open Source Software: <http://www.ossmeter.org>
- [18]. OW2. (2014, November 2). Main. Retrieved from riscoss - managing risk and cost in open source software: <http://www.riscoss.eu/bin/view/Main/>
- [19]. Reza, A., Capretz, L. F. & Ahmed, F. (2012). An open source usability maturity model (OSUMM). *Computers in Human Behavior*, 28(4), 1109–1121.