Open Source Software Resilience Framework

Apostolos Kritikos, Ioannis Stamelos

Open Source Software Resilience Framework

Apostolos Kritikos and Ioannis Stamelos

Aristotle University of Thessaloniki, University Campus, 54124 Thessaloniki, Greece
akritiko@csd.auth.gr, stamelos@csd.auth.gr,
WWW home page: http://www.csd.auth.gr

Abstract. An Open Source Software (OSS) project can be utilized either as is, to serve specific needs on an application level, or on the source code level, as a part of another software system serving as a component, a library, or even an autonomous third party dependency. There are several OSS quality models that provide metrics to measure specific aspects of the project, like its structural quality. Although other dimensions, like community health and activity, software governance principles or license permissiveness, are taken into account, there is no universally accepted OSS assessment model. In this work we are proposing an evaluation approach based on the adaptation of the City Resilience Framework to OSS with the aim of providing a strong theoretical basis for evaluating OSS projects.

Keywords: open source software, software resilience, software engineering, software quality, software metrics

1 Background

Open Source Software (OSS) has been continuously growing and evolving for over two decades now. It started as a revolutionary software engineering approach, producing software that was freely and openly available to be used, edited, copied, even commercially utilized. Since then, it has been fueling the development process of companies with source code, testing and bug fixing. Moreover it has been one of the main drivers for the birth of healthy, successful companies [1], [2].

In the beginning, the focus about OSS revolved mainly around its development style. Eric Raymond in [3] compares closed source development to a Cathedral because of its structured and concrete definition of work and roles. In contrast, open source development resembles to a bazaar, a place with no strict rules, where people can openly and freely contribute. As Raymond specifically states *"No quiet, reverent cathedral-building here rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who'd take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles"* meaning that in a time where closed (or proprietary) software development was the norm, an open approach of collectively developed software seemed odd and likely to fail.

The fact that OSS succeeded is a result of several factors. A primary and extensively studied factor is structural quality which led to the proposal of several quality models like ISO25010 [4] for evaluating software in general or Open Source specific ones like OpenBRR [5]. In [6] Midha and Palvia identify several other factors outside the scope of source code and software structure. The level of permissiveness of the license under which the OSS is published, community aspects such as number of active developers and end users, language translations are some indicative examples. In [7] the authors study Open Source Governance models, another important factor for OSS success.

In [8] Miguel presents the evolution of quality models between 1977 and 2013. The models are further categorized to Basic Models (1977-2001) with the aim of evaluating the software product as a whole and Tailored Quality Models (2001-2013) which extend to component evaluation. The latter category includes the OSS specific quality models as well. The study compares and contrasts the models based on their characteristics and concludes that generic quality evaluation models cannot easily be applied in specific cases. On the other hand tailored quality models usually cover the needs of very specific domains limiting their applicability.

In [9] Wasserman describes the evolution of Business Readiness Rating (BRR) model to OSSpal. As Wasserman very aptly states, software quality, maturity, stability, documentation community and so forth vary in different OSS projects and continuously change as those projects evolve. Therefore it is important for OSS evaluation models to include, apart from numerical scores and metrics, qualitative criteria as well.

In [10] the authors are combining quality assurance methodologies with social network analysis techniques to study competition and collaboration in large OSS ecosystems. Community wise, in [11] the authors are proposing a reference model for evaluating the maturity of an OSS project's community. They focus on the socio-technical practices of OSS software development and try to combine characteristics of existing software quality models to this end.

The aforementioned diverse set of factors which are responsible for the success of an OSS dictates the need to approach it as an evolving system in order to be able to study it in a holistic way. Moreover, in order for an OSS to be able to succeed and achieve longevity, it is crucial to be resilient to survive potential stresses and crises that might occur. Such stresses or crises could be related with forks of the project that might drive the attention of the original project's community away, migration of lead developers or even part of the development community to other forks or projects, an unsuccessful major release that might hurt the reputation of the project, changes to the license, migration to another forge and so forth. In [12] Gamalielsson and Lundell study the case of Libre Office, an OSS project that started as a fork of Open Office, but managed to retain the development community and evolve, as of the time of writing, to a successful OSS project.

In this publication we are trying to adapt the concept of City Resilience, a term found in the scientific field of Urban Planning and Architecture, to OSS

evolution. The term resilience is defined in [13] as the ability [of a system] to cope with change. In [14] the author defines a resilient system as one that can take a hit to a critical component and recover and come back for more in a known, bounded and generally acceptable period of times. In [15] the authors of the City Resilience Framework, which inspired this paper, define city resilience as follows: *"city resilience describes the capacity of cities to function, so that the people living and working in cities particularly the poor and vulnerable survive and thrive no matter what stresses or shocks they encounter"*.

From the aforementioned literature it becomes clear that an OSS project can be approached from several perspectives. The structure of its source code, the community that was built around it (developers, end users, testers), business related aspects (license, business models). In order to be able to holistically evaluate an OSS project we need to study it as a continuously evolving system. In this work we are proposing a framework with the aim of studying the resilience of an OSS project trying to combine several of the aforementioned perspectives.

The rest of the paper is organized as follows. In Section 2 we analyze the City Resilience Framework, a framework designed to measure Urban Resilience, which inspired our work. In Section 3 we present the adaptation of the City Resilience Framework to OSS. In Section 4 we demonstrate the application of the Open Source Resilience Framework to two (2) Open Source projects. In Section 5 we discuss possible threats to validity and, finally, in Section 6 we summarize our conclusions and we discuss ideas for future work.

2 City Resilience Framework

The City Resilience Framework (CRF), as presented in [15], is the result of research undertaken with the aim of establishing an accessible, evidence-based definition of Urban Resilience by Arup and the Rockefeller Foundation. The CRF is, as of the time of writing, used by the 100 Resilient Cities [16] non profit organization to primarily evaluate the Urban Resilience of more than 90 cities around the world and, additionally, to assist the cities on crises with tailored made resilience strategies.

Trying to tackle the fact that every city is unique the City Resilient Index (CRI) was proposed. It is a set of indicators, variables and metrics that allow cities to understand, baseline and subsequently measure local resilience over time. As the authors of [17] state *"The CRI will measure relative performance over time rather than comparison between cities. It will not deliver an overall single score for comparing performance between cities, neither will it provide a world ranking of the most resilient cities."*

2.1 City Resilience Index

City Resilience Index (CRI) suggests that resilience of a city is related to four (4) key dimensions that are further decomposed to twelve (12) goals that can help a city achieve resilience. These dimensions and goals are the following:

1. **Health & well-being:** Related to **people**, working and living in the city. Goals: (1) Minimal human vulnerability, (2) Diverse livelihoods & employment (3) Effective safeguards to human health & life.
2. **Economy & society:** Related to the **organization** of cities on a social and economical level. Goals: (1) Sustainable economy, (2) Comprehensive security & rule of law, (3) Collective identity & community support.
3. **Infrastructure & environment:** Related to **place**, the quality of infrastructure and ecosystems. Goals: (1) Reliable mobility & communications, (2) Effective provision of critical services, Reduced exposure & fragility.
4. **Leadership & strategy:** Related to **knowledge** of the past and adapting appropriately for the future. Goals: (1) Effective leadership & management, (2) Empowered stakeholders, (3) Integrated development planning

Finally, the aforementioned goals are, on a third level, analyzed in indicators in order to identify the critical factors that contribute towards the resilience of urban systems.

3 Adaptation of City Resilience Index to Open Source Software

We argue that Open Source Software projects share a conceptual similarity with cities. They are dynamic and continuously evolving systems with their own structural properties, they attract people that form communities around them which, on a second level, may utilize a governance model. Some OSS projects have commercial activity. As it is happening with cities, OSS projects can face stresses and crises (i.e. developers abandoning the project to work on a fork or users massively migrate to a competitive project).

In this section we are presenting our attempt to adapt the City Resilience Index to Open Source Software projects. We aim in utilizing the models and metrics of the extensive literature on OSS quality, metrics and evaluation and provide a framework that will measure the relative performance of an OSS project over time rather than a comparison between projects.

3.1 Open Source Software Resilience Framework (OSSRF)

Following the City Resilience Framework paradigm, Open Source Software Resiliency Framework (OSSRF) is also being primarily structured to four (4) dimensions that are then analyzed in twelve (12) goals and, on a third level, on a set of indicators for each goal.

3.1.1 Dimensions

As in the CRF we propose four key dimensions related to Open Source Software as follows:

1. **Source Code:** The first dimension of CRF is Health & Well-being and it is related with people. In Open Source Software we consider source code (i.e. classes) to be the structural unit of the project. In this dimension we will take under consideration aspects like the activity and growth rate of an OSS project along with some other related aspects.
2. **Business & Legal:** The second dimension of CRF is Economy & Society and is related with organization. In Open Source Software the norm is voluntary work but, more mature projects are utilizing Open Source Business Models to offer commercial services (be it pro features or support). For those types of projects licensing plays a key role when it comes to commercialization.
3. **Integration & Reuse:** The third dimension of CRF is related to place. Open Source Software projects usually reuse components of other OSS projects or are being reused themselves. In this spirit, in the third dimension of the Open Source Software Resilience Framework we will be dealing with the aspects of integration and reuse.
4. **Social (Community):** Finally the last dimension of CRF is about Leadership & Strategy and is related with utilizing knowledge from the past to become better and more resilient in the future. In Open Source Software both leadership and strategy related processes are usually connected with the community. Moreover most of the knowledge related to an Open Source Software usually comes from its community activity (i.e. feature proposal, bug reports, translations, documentation, testing and so forth).

3.1.2 Goals

The aforementioned dimensions are further decomposed to the following twelve (12) goals:

Source Code

1. Continuous Growth
2. Holistic Documentation
3. Systematic Testing & Violation Minimization

Business & Legal

1. Economic Sustainability
2. Flexible Licensing
3. External Organization Support

Integration & Reuse

1. Low Dependability
2. Low Complexity
3. Ease of Integration

Social (Community)

1. Well defined Project Standards
2. Well Defined Governance
3. Developer Base Activity

3.1.3 Indicators

Finally, the twelve (12) goals are further analyzed to indicators in order to provide a more specific description of the goals. For the purposes of this paper we will analyze the indicators related to the goals of the Business & Legal dimension. Due to space limitations, we provide the full analysis of the indicators to the following url: http://users.auth.gr/akritiko/ossrf_indicators.html for the intended audience.

1. **Economic Sustainability**
   1.1. Donations: 0 (no) or 1 (yes) based on whether the OSS project accepts donations. 0 is considered a non resilient value.
   1.2. Commercial features: 0 (no) or 1 (yes) based on whether the OSS project offers commercial features or a pro (paid) version. The indicator was based to the work of [18]. 0 is considered a non resilient value.
   1.3. Paid support: 0 (no) or 1 (yes) based on whether the OSS project offers a paid plan for support. [18]. 0 is considered a non resilient value.
2. **Flexible Licensing**
   2.1. Level or permissiveness: 0 (all restrictive - i.e. commercial), 1 (persistent i.e. GPL), 2 (all permissive - i.e. BSD). We base the indicator to the of [19]. The indicator is considered non resilient when it is less than 1.
   2.2. Level of persistence: 0 (no) or 1 (yes) based on whether there are parts of the project's code or dependencies published under persistent licenses (i.e. GPL). We base the indicator to the of [19]. 1 is considered a non resilient value.
3. **External Organization Support**: 0 (no) or 1 (yes) based on whether the OSS project is supported by an external organization (non profit, governmental or corporate). 0 is considered a non resilient value.

3.1.4 Resilience determination mechanism

Since the evaluation of a project regarding its resilience is based on indicators we need a mechanism to determine whether the OSS project under review is resilient and, on a second level, how its resiliency changes as it evolves. Starting to the indicators level we will consider an OSS project successful towards a resilience goal when it is considered resilient at least to 50% of the goals ingredients.

Moving to the dimensions level, an OSS project will be considered successful towards a resilience dimension when it is considered resilient at least to 50% of the goals of the specific dimension. Likewise, on a project level, the OSS project is considered resilient when at least two (2) out of four (4) dimension (50%) are considered resilient.

4 Open Source Software Resilience Framework Application

In this section we will be demonstrating the application of the OSSRF to two OSS projects, the OKapi and WooCommerce, using the indicators analyzed in

the previous sections. Due to space limitations, for our demonstration purposes will be using only the fully analyzed dimension "Business & Legal".

Since the OSSRF takes under consideration the evolution of OSS projects we would normally have to apply the framework to all the major releases of the two projects. Due to limitation spaces we chose to use the last major release of each year from the beginning of each project until either the year the project became inactive (i.e. for the case of OKapi) or the last year that ended (i.e. 2017) for the case of WooCommerce. We do not consider major releases, alpha, beta release or release candidates so if some early years of the project have been excluded is because there were no major release back then.

4.1 OKapi - A non resilient project

The first project, OKapi is a small framework for building web applications. It's built on PHP and is hosted in Github [20]. It started during 2008 and hasn't been updated since July 2011. We selected this case in order to test our framework to a project that intuitively seems non resilient.

The versions of the releases of OKapi to which we will be applying the OSSRF, with their corresponding dating (year / month) are shown to the following table: By applying the Business & Legal dimension's indicators to OKapi we are

Table 1. OKapi releases

Version	1.1.5	1.2.1	1.2.3
Date	2008/12	2009/12	2010/12

getting the results as shown in Table 1. In order to make the demonstration of

Table 2. OKapi - Business & Legal dimension. Goals & Indicators

Indicator	**v1.1.5**	**v1.2.1**	**v1.2.3**
		Economic Sustainability	
Donations	0 (F)	0 (F)	0 (F)
Commercial features	0 (F)	0 (F)	0 (F)
Paid support	0 (F)	0 (F)	0 (F)
		Flexible Licensing	
Level or permissiveness	0 (F)	0 (F)	0 (F)
Level of persistence	0 (S)	0 (S)	0 (S)
		External Organization Support	
External Organization Support	0 (F)	0 (F)	0 (F)

the Resilience Determination Mechanism we proposed at Section 3.1.4. easier, we have marked the values of the indicators to Table 2. with (F), if an indicator is considered non resilient, and (S) if it is considered resilient.

OKapi is officially not maintained but even when it was there is no indication that supported donations, commercial features or support. There is no official website or reference in any related document, to our best knowledge, about revenue streams for the project. Therefore we consider the project non resilient regarding the indicators of the first goal. As far as the licensing part is concerned, there is no license defined to the project and OKapi does not seem to reuse other open source projects. Therefore the project is considered non resilient as far as the "Level of permissiveness" is concerned but it is considered resilient towards the "Level of persistence". Therefore regarding the goal "Flexible Licensing" OKapi is considered resilient. Finally, there is no indication that OKapi was ever supported by an external organization therefore we consider the project non resilient towards this goal. The aforementioned results apply to all the versions of the project.

Based on the results two out of three goals are considered non resilient for the OKapi project. Therefore based on OSSRF, the project is considered non resilient.

4.2 WooCommerce - A resilient project

The second project is WooCommerce, an open source eCommerce plug-in for WordPress content management system, written also in PHP and hosted in Github [21]. It started as an OSS project in 2011 and as of the time of writing is active. We selected this case in order to test our framework to a project that intuitively seems resilient.

The versions of the releases of WooCommerce to which we will be applying the OSSRF, with their corresponding dating (year / month) are shown to Table 3. By applying the Business & Legal dimension's indicators to WooCommerce

Table 3. WooCommerce releases

Version	1.3.2	1.6.6	2.0.20	2.0.10	2.0.12	2.6.11	3.2.6
Date	2011/12	2012/12	2013/12	2014/12	2015/12	2016/12	2017/12

we are getting the results as shown in the Table 4.

WooCommerce is a free and OSS WordPress plug-in but there are commercial plug-ins created for WooCommerce. There is also paid support offered. To our best knowledge it does not support donations. Additionally, WooCommerce is published under GPL license and some of its dependencies are published under persistent license (i.e. wc-e2e-page-objects). Finally WooCommerce is backed from both the WordPress Foundation and the Automattic Company, both well-known organizations. Towards "Economic Sustainability" goal the project is considered resilient since 2 out of 3 indicators are considered resilient. Regarding "Flexible Licensing" it is also considered resilient since 1 of 2 indicators is considered resilient. Finally, "External Organization Support" is also considered resilient. The aforementioned findings apply to all versions.

Table 4. WooCommerce - Business & Legal dimension. Goals & Indicators

Indicator	**v1.3.2**	**v1.6.6**	**v2.0.20**	**v2.0.10**	**v2.0.12**	**v2.6.11**	**v3.2.6**
				Economic Sustainability			
Donations	0 (F)	0 (F)	0 (F)	0 (F)	0 (F)	0 (F)	0 (F)
Commercial features	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)
Paid support	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)
				Flexible Licensing			
Level or permissiveness	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)
Level of persistence	1 (F)	1 (F)	1 (F)	1 (F)	1 (F)	1 (F)	1 (F)
				External Organization Support			
External Organization Support	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)	1 (S)

Therefore based on OSSRF, the project is considered resilient.

5 Threats to validity

We should note that OSSRF should be applied to project of a relative maturity in terms of community and age (we would intuitively suggest at least 10 contributors and a maturity of more than a year). Applying it in solo maintained OSS projects, or projects that not yet have reached the proposed maturity may lead to misleading results.

OSSRF is an adaptation of the City Resilience framework to Open Source Software. Although the structure of the original framework was retained, despite the conceptual similarities that we have already mentioned earlier, the mapping of dimensions, goals and indicators is a product of the subjective lens of the authors.

In the application of the OSSRF to the two projects we have applied the indicators of one of the four available dimensions of the OSSRF, "Business & Legal" dimension.

Regarding the goals and indicators, some of them are based on metrics available for object oriented source code. Additionally, as far as control version systems are concerned, for the needs of this publication we selected projects that are hosted in Github.

As far as the scales and their aggregation is concerned, in this preliminary approach we considered each criterion equally important and the threshold for defining a project as resilient or non resilient is 50%.

Finally, both of the projects analyzed in this paper, were developed in PHP and their domains are close (OKapi was a framework for web applications and WooCommerce is a plugin for a WordPress which is also considered by some a kind of web framework).

6 Conclusions & Future work

This is a preliminary work proposing an evaluation approach with the aim of providing a strong theoretical basis for evaluating OSS projects adapting the concept of Urban Resilience to OSS. By doing so, we are trying to benefit form the advantage of resiliency to follow the evolution of a dynamic system (such as a city or an OSS) while at the same time we are providing a framework that can take under consideration concepts such as crises and or stresses that can theoretically impact the survival of an OSS. We applied a single dimension of the OSSRF to an intuitively non resilient OSS project and an intuitively resilient one and the results seem to concur with the initial intuitions.

For future work we intend to thoroughly fine-tune the rest of the indicators by testing it to a variety of OSS projects. This will also allow us to investigate how the OSSRF responds to projects of different age, community size or source code size and complexity. We also intend to investigate whether the software domain of an OSS project affects the results of the application of the OSSRF.

Regarding the framework itself we will experiment with other approached regarding the "Resilience determination mechanism" (i.e. weighted goals).

In addition we will be extending the OSSRF to be able to work with a variety of control version systems (not only git-like but also Mercurial, SVN, CVS). In a similar spirit will would like to experiment with projects of different programming languages (i.e. Java).

Another challenging idea for future work would be to apply OSSRF to OSS projects that are known to have faced specific stresses or crises in order to identify how those crises relate with the resiliency levels of an OSS project.

Finally we intend to attempt and request feedback, in the form of a survey, from key players of the Open Source Software international community (lead developers, stakeholders, academics and so forth) about OSSRF.

References

1. Steve Weber. *The success of open source*. Harvard University Press, 2004.
2. Dharmesh Thakker Max Schireson. The Money In Open-Source Software. https://techcrunch.com/2016/02/09/the-money-in-open-source-software/, 2016. [Online].
3. Eric Raymond. The cathedral and the bazaar. *Philosophy & Technology*, 12(3):23, 1999.
4. Organización Internacional de Normalización. *ISO-IEC 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models*. ISO, 2011.
5. Anthony Wasserman, Murugan Pal, and Christopher Chan. The business readiness rating model: an evaluation framework for open source. In *Proceedings of the EFOSS Workshop, Como, Italy*, 2006.
6. Vishal Midha and Prashant Palvia. Factors affecting the success of open source software. *Journal of Systems and Software*, 85(4):895–905, 2012.
7. Vision Mobile. Open governance index-measuring the true openness of open source projects from android to webkit. 2011.

8. José P Miguel, David Mauricio, and Glen Rodríguez. A review of software quality models for the evaluation of software products. *arXiv preprint arXiv:1412.2977*, 2014.
9. Anthony I Wasserman, Xianzheng Guo, Blake McMillian, Kai Qian, Ming-Yu Wei, and Qian Xu. Osspal: Finding and evaluating open source software. In *IFIP International Conference on Open Source Systems*, pages 193–203. Springer, 2017.
10. Jose Teixeira, Gregorio Robles, and Jesús M González-Barahona. Lessons learned from applying social network analysis on an industrial free/libre/open source software ecosystem. *Journal of Internet Services and Applications*, 6(1):14, 2015.
11. Sandro Andrade and Filipe Saraiva. Principled evaluation of strengths and weaknesses in floss communities: A systematic mixed methods maturity model approach. In *IFIP International Conference on Open Source Systems*, pages 34–46. Springer, 2017.
12. Jonas Gamalielsson and Bjrn Lundell. Sustainability of open source software communities beyond a fork: How and why has the libreoffice project evolved? *Journal of Systems and Software*, 89:128 – 145, 2014.
13. Andreas Wieland and Carl Marcus Wallenburg. The influence of relational competencies on supply chain resilience: a relational view. *International Journal of Physical Distribution & Logistics Management*, 43(4):300–320, 2013.
14. C Warren Axelrod. Investing in software resiliency. 2009.
15. J Da Silva and B Morera. City resilience framework. *Arup & Rockefeller Foundation. Online: http://publications. arup. com/Publications/C/City_Resilience_Framework. aspx [12/15/2015]*, 2014.
16. 100 Resilient Cities. http://www.100resilientcities.org/, 2013. [Online].
17. City Resilience Index. City resilience framework. *The Rockefeller Foundation and ARUP*, 2014.
18. Neeshal Munga, Thomas Fogwill, and Quentin Williams. The adoption of open source software in business models: a red hat and ibm case study. In *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 112–121. ACM, 2009.
19. M Välimäki and Ville Oksanen. Evaluation of open source licensing models for a company developing mass market software. *Law and Technology*, 2002.
20. OKapi Github Repository. https://github.com/liip/Okapi. [Online].
21. WooCommerce Github Repository. https://github.com/liip/Okapi. [Online].