

Evaluation of OpenProject, OrangeScrum and ProjeQtor using QSOS Methodology

Any Keila Pereira¹ and Jorge Bernardino^{1,2} 

¹Polytechnic of Coimbra – ISEC, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal

²CISUC - Centre of Informatics and Systems of University of Coimbra, DEI, 3030-290 Coimbra, Portugal

Keywords: Open Source Project Management Tools, QSOS Methodology, OpenProject, OrangeScrum, ProjeQtor.

Abstract: The market of project management tools is fruitful in Open Source solutions that are able to provide similar functionalities or even superior to those offered by commercial solutions. Many organizations using these tools deal with the problem of selecting the appropriate one, that corresponds to their requirements. Qualification and Selection of Open Source Software (QSOS) is a methodology designed to qualify, select and compare Open Source solutions in an objective and argued way, based in the functionalities and maturity of the software. This paper provides an evaluation of three Open Source Project Management tools: OpenProject, OrangeScrum, and ProjeQtor using QSOS methodology.

1 INTRODUCTION

According to Project Management Body of Knowledge (PMBOK) (Project Management Institute, 2017), a project is a temporary endeavor undertaken to create a unique product, service, or result.

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. It enables organizations to execute projects effectively and efficiently (Project Management Institute, 2017).

Managing a project is not an easy task, and to facilitate it, several tools have been developed. These tools offer a variety of features. Currently, a large number of project management tools are available, and the Open Source alternatives are increasingly demonstrating better functionalities, similar or even superior to those offered by commercial solutions (Abramova et al., 2016).


With the increase of the quality of Open Source Project Management tools, choose one over the other implies a prior assessment. Many companies are currently selecting the tool to managing their projects using *ad-hoc* techniques (Deprez and Alexandre, 2008), instead of using a certified methodology that allows to do the evaluation taking into account various characteristics of the software.

The QSOS (Qualification and Selection of Open Source software), is a methodology that allows to qualify, compare and select Open Source software in an objective and argued way, based, not only in the functionalities of the software, but also on a set of maturity criteria proposed by the methodology (Semeteys, 2013).

In this paper, we use the QSOS methodology, because we wanted an evaluation based not only in the functionalities but also in the maturity of the software. We evaluated OpenProject, OrangeScrum and ProjeQtor, once they appear in the list of the Top Open Source Project Management Software.

We also present the main functionalities, advantages, and limitations for each of the open source management tool analyzed.

The present paper is organized as follows. Section 2 describes the related work existent about this theme, Section 3 describes the three Project Management tools that will be evaluated. Section 4 presents a description of the QSOS methodology and in Section 5 is presented the results of the evaluation of the tools with the application of QSOS methodology. Finally, Section 7 presents the conclusions and future work.

 <https://orcid.org/0000-0001-9660-2011>

2 RELATED WORK

There are some published works that compare Project Management tools.

In Pereira et al., (2013), the authors compare the most popular Free/Open Source web-based Project Management tools with respect to their compliance to PMBOK and Capability Maturity Model Integration (CMMI). In general, they observed that exist a very large number of so-called Project Management Tools, however, a review of those tools indicated that in fact only a few tools actually provide support for Project Management as understood by PMBOK/CMMI. Many of these tools are rather for managing to-do lists and/or management of issue tracking, such as, Bugzilla or Redmine.

In Cicibas et al., (2010), the authors do a Comparison of Project Management Software tools. Half of the tools they analysed are Open Source or do not require licenses. In this study, they developed criteria to determine which tool would be subject to the analysis. Then, they developed criteria to compare and evaluate these tools. For each of the tools, the authors investigated whether it supports the functionality or features selected as the criteria developed for the study. The analysis indicates that none of the tools provides all the functionality or features and some of the features exist in most all of the tools.

There are also, some published works that use the QSOS methodology to evaluate and select software.

In Ferreira et al., (2012), the authors use the QSOS methodology for qualification and selection of Open Source Software (Koha, Evergreen and PhpMyBibli) and applies it in the context of integrated library management software. Based on the analysis performed, the software that showed more ability to meet the needs of the national librarians was the Koha. After the analysis, they present an evaluation and selection of Big Data Analytics using and adapting the QSOS methodology for qualification and selection of Open Source software. The comparison demonstrated that the HPC Systems Platform is more efficient and reliable than the Hortonworks Data Platform. There are also some works comparing other types of software such as Big Data platforms (Almeida and Bernardino, 2015).

However, to the best of our knowledge, this is one of the first papers that applies the QSOS Methodology to evaluate Open Source Project Management Tools.

3 OPEN SOURCE PROJECT MANAGEMENT TOOLS

Most of the failed projects have in common the fact that they are not adequately managed. Doing effective and efficient management of projects still remains a big challenge to organizations. In order to achieve the best way to manage projects, "best practice" models, such as, the CMMI or the PMBOK, (Project Management Institute, 2017) are being developed to assist organizations in improving project management.

The dimensional growth and increasing difficulty in Project Management promoted the development of different tools that serve to facilitate it. Although not required, software tools can help implement the Project Management process in practice and becomes a key in supporting the effort to complete a project successfully (Margea and Margea, 2011).

There are many Project Management tools that provide a large number of features and the market is rich in Open Source solutions that are able to provide similar functionalities or even superior to those offered by commercial solutions (Abramova et al., 2016).

We selected the tools: OrangeScrum, OpenProject, and ProjeQtor to evaluate, since the 3 tools are in the list of best Project Management Tools (Conrad et al., 2018). The metrics for the choice were, number of features, number of users and friendly interface.

In the next sections, we describe the main characteristics of the Project Management Open Source tools: OpenProject, OrangeScrum, and ProjeQtor.

3.1 OpenProject

OpenProject (OpenProject, 2019) is a powerful Open Source Project Management solution written in Ruby on Rails and compatible with Linux operating systems.

The OpenProject Foundation was established by OpenProject's developers and users in October 2012. Is continuously developed by an active Open Source community.

The association provides an organizational framework for technical decisions and the propagation, acceleration and perpetuation of development by the worldwide community and by a full-time development team, funded by the members of the OpenProject Foundation.

Its modules support: Project planning and scheduling, it is possible to easily define the project

objectives and specify the work to be done related to this scope, analyse the required activities and create a detailed plan that shows how and when the project will provide the deliverables defined in the project scope; Roadmap and release planning, it is easy to plan, visualize, and communicate the product roadmap, share product roadmap with stakeholders, get feedback about the ideas and break it down into a detailed release plan; Time tracking, cost reporting and budgeting, OpenProject makes time tracking easy, it is possible to create custom reports for accurate, current insight into project performance and allocated resources, plan the cost for each project phase, and see how much of the allocated budget has been spent; Bug tracking, OpenProject excels at bug tracking, offering Quality Assurance managers and testers a platform to capture, classify and prioritize bugs; Kanban, Agile and Scrum, its agile features, including creating stories, prioritizing sprints, and tracking tasks, it is the perfect tool for agile teams that want to use agile methodologies such as Scrum; Project Wiki, OpenProject allows users to create a knowledge base for their projects and share it with their team and other stakeholders. Paid plans offer additional capabilities, including customization, security, and support (OpenProject, 2019). The price for the upgrade is 4,95€ per user for the Cloud Edition and 9,95€ per user for the Enterprise Edition.

OpenProject is licensed under GPLv3. Its latest version, 7.3.2. is available for download and its source code is available on GitHub (Conrad *et al.*, 2018).

The main advantages of OpenProject are:

- The free version includes the entirety of Project Management capabilities;
- Easy to use and user friendly.

The main limitations of OpenProject are:

- Windows Operation System is not supported;
- OpenProject Community offers minimal support outside of user guides.

Figure 1 shows the interface of OpenProject.

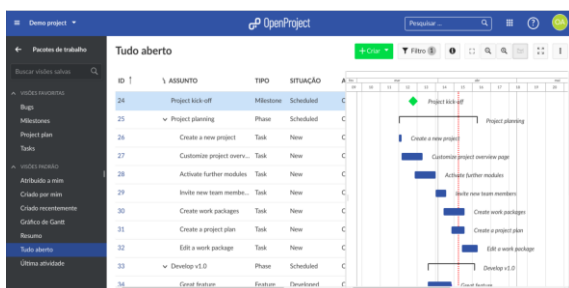


Figure 1: Interface of OpenProject.

3.2 OrangeScrum

OrangeScrum is a product of AndolaSoft Inc., a privately held web and mobile app development company based in San Jose, USA, founded in 2009 by Jay Das (Andolasoft, 2019a).

It is a cloud-based Project Management software and Task Management to help organize, projects, teams, and tasks in one place. It gives users full visibility and control over their projects, enabling them to deliver projects on time and within budget. It is a valuable tool that brings people and projects together, providing clear structures (Andolasoft, 2019b).

As an agile Project Management software, it offers capabilities such as epics and stories, sprints, scrum boards, reports and velocity charts.

The Premium features include: time tracking, that allows time tracking spent on tasks, with daily and weekly timesheets for evaluating team performance; Recurring tasks, that allows auto-creation of tasks with the defined frequency of time; Gantt charts, that allows to drag and drop tasks, draw linkages to define task dependencies, arrange the tasks in a sequence and many from a single view; Project templates, that enables to replicate and reuse the structure and content of existing projects, including milestones and task with start date, end date, estimated hours to speed up new project creation and standardize workflow processes; Client management, that allow clients to easily contribute to the project with their account access, mark a user as a client to involve clients in project progress, maintain log of all client communication to prevent unwanted interference from clients and customers, create a private task which a client user can't see; User role management, that allows to define clear roles and responsibilities for the smooth functioning of the teams as well as the projects, it also ensures how everyone can contribute to a project (Andolasoft, 2019a).

Training and onboarding support are available for an additional fee (Project Management Zone, 2019). As we can see, most of the Project Management capabilities are Premium features, which has an associated cost.

OrangeScrum offers several versions. The Startup plan is priced at 7€ per month, it includes for 10 users with a storage limit of 5GB. The Basic plan is priced at 23€ per month, includes for 20 users and 15GB storage limit. The Standard plan is priced at 39€ per month, includes for 35 users, the Professional plan is priced at 71€ per month, with up to 60 users, and there is also an Unlimited Users plan, priced at 88€ per

month. All plans include unlimited tasks, projects, and task groups or sprints.

OrangeScrum is licensed under GPLv3 and is based on the CakePHP framework. It requires Apache, PHP 5.3 or higher, and MySQL 4.1 or higher, and works on Windows, Linux, and MacOS. Its latest release, 1.6.1. is available for download, and its source code can be found on GitHub.

The main advantages of OrangeScrum are:

- Easy to use and user friendly;
- It allows self-hosting;
- Mobile application for iOS, Android;
- Ideal for task management, small organizations and small and midsize businesses.

The main limitations of OrangeScrum are:

- Limited number of free features;
- Confusing in distinguishing the free and paid versions, once they have same name;
- Although it is easy to use, the process of installation is explained in a confusing way in the tool website.

Figure 2 shows the interface of OrangeScrum.

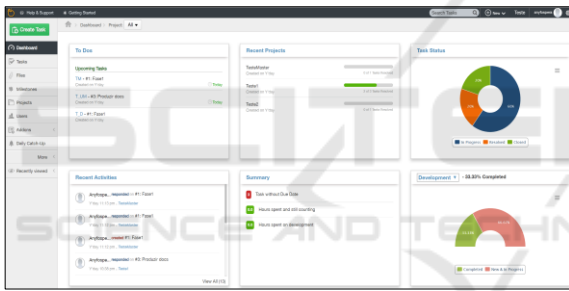


Figure 2: Interface of OrangeScrum.

3.3 ProjeQtOr

ProjeQtOr (ProjeQtOr, 2019) is a solid, Open Source Project Management tool originally released in 2009 by Pascal Bernard.

Its name means Quality based Project Organizer, as it claims to be Quality oriented. This software has changed its name in November 2013. It was formerly called Project'Or RIA (Project Organizer Rich Internet Application).

Over the years many contributors have put significant work into the project, expanding it into a deep Project Management system with a dizzying number of features, including portfolio management, bug tracking, risk management, and budget management. ProjeQtOr main features are: Planning management, it provides all the elements needed to build a planning from workload, constraints between tasks and resources availability; Resource

management, it manages the availability of resources that can be affected to multiple projects, it calculates a reliable, optimized and realistic planning; Incident management, it includes a Bug Tracker to monitor incidents on your projects, with possibility to include work on planned tasks of your projects; Costs management, all elements related to delays can be followed as costs (from resources work) and managing other expenses, all costs of the project are monitored and can generate invoices; Quality management, it is Quality Oriented, it integrates the best practices that help to meet the quality requirements on the projects; Risks management, it includes a comprehensive risks and opportunities management, including the action plan necessary to mitigate or treat them and monitoring occurring problems; Perimeter management, it allows you to monitor and record all events on your projects and helps in managing of deviations, to control the perimeter of projects; Commitments management, it allows to follow the requirements on the projects and measure at any time coverage progress, making it easy to reach your commitments; Documents management, ProjeQtOr offers integrated Document Management, documents can be versioning and an approver process can be defined, it is efficient to manage project and product documents; Tools, ProjeQtOr includes some tools to generate alerts, automatically send emails on chosen events, import or export data in various formats (ProjeQtOr, 2017).

The system is regularly updated, with new patches coming out several times per month and a new major update to add features and address issues roughly every other month. The community forum is also very active (ProjeQtOr, 2019).

The main advantages of ProjeQtOr are:

- It is a completely free and fully operational;
- It includes all of Project Management capabilities and a few more things;
- It is customizable, due to its high configurability;
- It has an active international community, participating to its constant improvement and rapid enrichment.

The main limitations of ProjeQtOr are:

- The number of menu icons available after installing the application is impressive, and it may be complex for some users;
- The tool seems not to be "agile methods" oriented;
- Users complain that after some time of use in business environment, the software starts to become slow.

Figure 3 shows the interface of ProjeQtor.

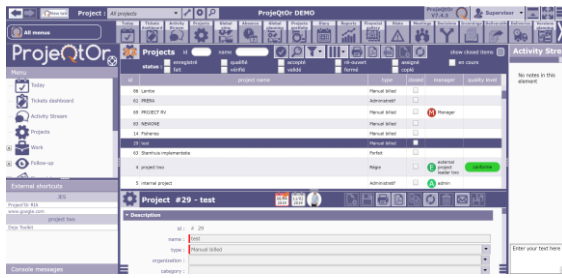


Figure 3: Interface of ProjeQtor.

4 QSOS METHODOLOGY

The QSOS (Qualification and Selection of Open Source software), is a methodology designed by Atos to qualify, select and compare Free and Open Source software in an objective, traceable, and arguable way. The evaluation of the software using this method is based, not only in the functionalities, but also in the maturity of the software (Semeteys, 2013).

It consists of four stages, namely, Definition, Evaluation, Qualification, and Selection. The model is supported by a tool called Open Source Selection Software (Adewumi et al., 2019).

The general approach of the QSOS methodology is composed of four interdependent steps.

4.1 Step 1: Define

The purpose of this step is to define different elements of typology that will be used during the next three steps of process.

The different typological references are:

- Type of software: the hierarchical classification of types of software and the description of functional coverage;
- Type of license: classification of types of free and open source licenses in use;
- Type of community: classification of types of community organizations around the software to ensure the life cycle.

4.1.1 Type of Software

This reference is composed of hierarchical criteria, grouped by axes:

- Maturity analysis of the project in charge of the software development;
- Functional coverage analysis of the software.

The QSOS Method defines and imposes the maturity criteria of a project. The QSOS Manifesto (Semeteys, 2013) defines how the maturity criteria are scored.

Figure 4 shows the maturity criteria proposed by QSOS methodology.

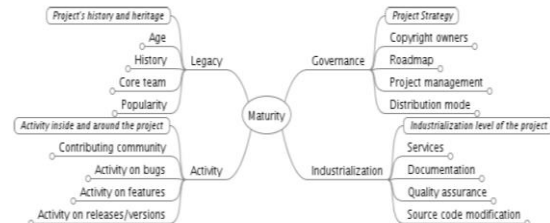


Figure 4: Maturity criteria of a project (Semeteys, 2013).

4.1.2 Type of License

The purpose of this reference is to identify and categorize the software license according to the following axes:

- Copyleft: Can derivative works become proprietary or have to stay under the same conditions?
- Virality: Does the use of the software from a module implies that this module has to be under the same license?
- Inheritance: Does the derivative work inherit from the license or is it possible to add restrictions?

4.1.3 Type of Community

The types of identified communities are:

- Sole developer: The software is developed by a sole person;
- Group of developers: Several persons working together without formal processes;
- Developers organization: A group of developers using a software lifecycle management system formalized and respected, based on roles and meritocracy;
- Legal entity: A legal entity, often a non-for-profit, manages the community and the sponsorship and holds the copyrights;
- Commercial entity: A commercial entity employs the core developers of the project and gets revenue from the sale of services or commercial version of the software.

4.2 Step 2: Evaluate

The purpose of this step is to evaluate the software. The evaluation criteria of the maturity of the tool are imposed by the method and described further.

The criteria are assigned a discrete score from 0 to 2 and the evaluation templates contain the meaning of the three scores 0, 1 and 2 for every criterion.

Regarding the functional coverage, the scoring rule is: 0 for Functionality not covered; 1 for Functionality partially covered and 2 for Functionality fully covered.

These scores will be used in the selection step to compare and filter the software depending on the weighting specified during the qualification step.

It is possible to apply the general approach in an iterative way. At the evaluation level, it means to have the possibility to score the criteria several times.

4.3 Step 3: Qualify

The purpose of this step is to define a set of elements translating the needs and constraints lined to the selection approach of a Open Source software. The context in which the software will be used has to be set, in order to get a filter used in the Selection step.

In the Maturity filter, the degree of relevance of every maturity criterion is defined as: Not relevant criterion; Relevant criterion and Critical criterion. In the Functional coverage filter, every functionality is assigned a level of requirement: Required functionality; Optional functionality and Not required functionality.

The degree of relevance of the criteria and the level of requirement of the functionalities will be translated into a weighting value in the next step of the process, depending on the chosen selection mode.

4.4 Step 4: Select

The purpose of this step is to select the software matching the user's needs, or to compare the software of the same type.

Two modes of selection are available: Strict selection and Loose selection.

The Strict selection is made by a process of elimination as soon as a piece of software does not comply with the demands:

- Elimination of the software that don't go through the identity filter;
- Elimination of the software that don't provide the required functionalities;
- Elimination of the software whose maturity criteria don't match with the degree of relevance defined by the user:
 - the score of a relevant criterion must be greater than or equal to 1;
 - the score of a critical criterion must be equal to 2.

The Loose selection mode is less strict than the previous one. Instead of eliminating software that are non-eligible, it sorts them while measuring the difference compared to the filters previously defined.

It is based on the weighting values whose rules are detailed in the following paragraphs:

A. Weighting of Functionalities.

The weighting factor is based on the level of requirements of every functionality of the functionality coverage.

The weight of the functionality according to the level of requirement is: 3 for Required functionality; 1 for Optional functionality and 0 for Not required functionality.

B. Weighting of Maturity.

The weighting factor is based on the degree of relevance of every maturity criterion.

The weight of the criteria according to degree of relevance is:

- 3 for Critical criterion;
- 1 for Relevant criterion;
- 0 for Not relevant criterion.

5 EVALUATION

To do the evaluation, it is necessary to build the maturity matrix according to the method, the matrix of functionalities considered important in a Project Management Tool and identify the software license and community.

For this evaluation we considered the functionalities: Hierarchical tasks; Milestone tracking; Task dependencies; Gantt charts; PERT charts; Resource management; Time tracking; Cost tracking; Risk management; Scrum support; Kanban support and Project portfolio management, since they are the most common functionalities in a Project Management tool (Karlson, 2018) (Kashyap, 2019).

In terms of licence all of the three tools have the GNU Public License, which means that the license is not an evaluation factor.

In terms of community of developers, we identify that: OrangeScrum is developed by a Commercial Entity; OpenProject is developed by a Developers Organization and the ProjeQtor by a Group of developers. Although the community of developers may be a relevant factor in the evaluation of a software, we think that it does not influence in the choice of one tool in relation to another one, so we will not consider it in the evaluation.

Considering all the criteria on the maturity matrix, after assigning the score of each criterion for each tool, the tool that achieved the highest score, with the sum of the scores assigned to each criterion, was OpenProject with a score of 20, followed by ProjeQtor with 19, and finally the OrangeScrum with 14 in a maximum of 32.

Considering all the functionalities, defined on step 1, after assigning the score of each one, (steps 1 and 2) the tool that achieved the highest score, with the sum of the scores assigned to each functionality, was ProjeQtor with a score of 18, followed by the OpenProject with 12, and finally the OrangeScrum with 8 in a maximum of 24.

The step 3 is to qualify the criteria and functionality considered in the previous step. The Tables 1 and 2 qualifies the maturity criteria and functionality coverage respectively.

Table 1: Degree of relevance of the maturity criteria.

Maturity	Degree of relevance
Age	Not relevant criterion
History	Not relevant criterion
Core team	Not relevant criterion
Popularity	Relevant criterion
Contributing community	Relevant criterion
Activity on bugs	Critical criterion
Activity on features	Critical criterion
Activity on releases/versions	Relevant criterion
Copyright owners	Not relevant criterion
Roadmap	Not relevant criterion
Project management	Relevant criterion
Distribution mode	Not relevant criterion
Services	Critical criterion
Documentation	Critical criterion
Quality assurance	Relevant criterion
Source code modification	Not relevant criterion

Table 2: Level of requirement of the functionalities.

Functionalities	Level of requirement
Hierarchical tasks	Not required functionality
Milestone tracking	Required functionality
Task dependencies	Not required functionality
Gantt charts	Optional functionality
PERT charts	Optional functionality
Resource management	Required functionality
Time tracking	Required functionality
Cost tracking	Required functionality
Risk management	Required functionality
Scrum support	Optional functionality
Kanban support	Required functionality
Project portfolio management	Optional functionality

To qualify we removed of the evaluation the “Not relevant criteria” and “Not required functionality”.

For the next step, Selection, we chose the Loose selection. According to this mode of selection we calculated the final score of each tool by multiplying

the score attributed in the evaluation step by the weight of the criteria/functionality level.

After multiplying the score of each criterion on the maturity matrix by the weight of maturity, according to degree of relevance, the tools OpenProject, and ProjeQtor achieved the same score, 21, and then OrangeScrum with 14 in a maximum of 34. In the functionality matrix, after multiplying the score of each functionality by the weight of functionalities, according to the level of requirement of the functionality, the tool that achieved the high score is ProjeQtor, with a score of 38, followed by OpenProject, with a score of 22 and then OrangeScrum with 20 in a maximum of 44.

As we can see the ProjeQtor is the tool that we select as the best tool applying the QSOS methodology, since it obtained the best score in both matrixes.

6 CONCLUSIONS AND FUTURE WORK

There is a variety of Open Source Project Management tools that provide a large number of features. Considering available Open Source alternatives, it is important to notice that there are a vast number of them that provide similar features to proprietary ones.

In this paper, we analysed three Open Source Project Management tools: OpenProject, OrangeScrum and ProjeQtor. This evaluation is based in technical documentation and on the websites of the respective tools.

The application of the QSOS methodology allowed us to obtain a more precise assessment, based not only in the functionalities of the tools but also on maturity criteria proposed by this methodology.

With the application of the QSOS methodology we conclude that ProjeQtor is the best tool, once this tool achieved the highest score in both matrixes, the functionalities and the maturity matrixes proposed by this method. Although this tool is powerful for project management with plenty of features, the number of features available scares some users, and this characteristic that should be its main advantage ends up being its main disadvantage. So, if the user-friendly interface of a software is an important thing for you, then ProjeQtor is not a good choice, but if you are looking for features it is an excellent choice.

As we can see, ProjeQtor just goes beyond OpenProject in terms of functionality, because in the maturity matrixes they achieved the same score.

As a future work, we intend to evaluate ProjeQtor comparing with other Open Source tools and even commercial ones. We also propose to use the best tool in a real environment.

REFERENCES

- Abramova, V., Pires, F. and Bernardino, J. (2016) 'Open Source vs Proprietary Project Management Tools', *New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing*, 2. doi: https://doi.org/10.1007/978-3-319-31232-3_31.
- Adewumi, A., Misra, S., Omeregbe, N. and Sanz, L. (2019) 'Research Article Fosses : Framework for open-source software evaluation and selection', (January), pp. 780–812. doi: 10.1002/spe.2682.
- Almeida, P. and Bernardino, J. (2015) 'Big Data Open Source Platforms', 2015 IEEE International Congress on Big Data. IEEE, pp. 268–275. doi: 10.1109/BigDataCongress.2015.45.
- Andolasoft (2019a) OrangeScrum. Available at: <https://www.orangescrum.com/>.
- Andolasoft (2019b) OrangeScrum Blog. Available at: <http://blog.orangescrum.com/2019/02/4-proven-strategies-for-increasing-employee-productivity-in-2019.html>.
- Cicibas, H., Unal, O. and Demir, K. (2010) 'A Comparison of Project Management Software Tools (PMST).', *Software Engineering Research*, (January). Available at: http://www.softwaresuccess.org/SERP10_Demir_Comparison_PM_Tools.pdf.
- Conrad, A., O'Loughlin, E., Lauren, M. and Tamang, P. (2018) The Top 29 Free and Open Source Project Management Software for Your Small Business. Available at: <https://wp-stage.capstage.net/free-open-source-project-management-software/>.
- Deprez, J. C. and Alexandre, S. (2008) 'Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5089 LNCS, pp. 189–203. doi: 10.1007/978-3-540-69566-0_17.
- Ferreira, M., Ferros, L. M. and Fernandes, V. (2012) 'Avaliação e seleção de software open-source para Gestão Integrada de Bibliotecas', *Actas do Congresso Nacional de Bibliotecários, Arquivistas e Documentalistas*, p. 6. Available at: <http://repositorium.sdum.uminho.pt/handle/1822/20491>.
- Karlson, K. (2018) 6 PM Software Features That Will Skyrocket Your Project Performance. Available at: <https://www.scoro.com/blog/features-of-project-management-software/>.
- Kashyap, V. (2019) 12 Amazing Features to Look for in a Project Management System. Available at: <https://www.proofhub.com/articles/project-management-system>.
- Margea, R. and Margea, C. (2011) 'Open source approach to project management tools', *Informatica Economică*, 15(1), pp. 196–206.
- OpenProject (2019) Open source project management software. Available at: <https://www.openproject.org/>.
- Pereira, A., Gonçalves, R., Wangenheim, C. and Buglione, L. (2013) 'Comparison of Open Source Tools for Project Management', *International Journal of Software Engineering and Knowledge Engineering*, 23(02), pp. 189–209. doi: 10.1142/s0218194013500046.
- Project Management Institute (2017) A Guide to the Project Management Body of Knowledge, *International Journal of Production Research*. doi: 10.1002/pmj.21345.
- Project Management Zone (2019) Orangescrum System Properties. Available at: <https://project-management.zone/system/orangescrum>.
- ProjeQtOr (2017) 'ProjeQtOr User guide'. Available at: <https://www.projeqtor.org/files/ProjeQtOrUserGuide.pdf>.
- ProjeQtOr (2019) ProjeQtOr free project management software Quality Based Open Source Project Organizer. Available at: <https://www.projeqtor.org/en/>.
- Semeteys, R. (2013) 'Qualification and Selection of Open Source Software (QSOS)', *Open Source Business Resource*, (May 2008), pp. 1–29. Available at: http://backend.qsos.org/download/qsos-2.0_en.pdf%5Chttp://timreview.ca/article/146%5Chttp://www.qsos.org/Method.html.