# Evaluating GitLab, OpenProject, and Redmine using QSOS Methodology

André Vicente[1] and Jorge Bernardino[1,2][a]
[1]*Polytechnic of Coimbra, ISEC, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal*
[2]*CISUC - Centre of Informatics and Systems of University of Coimbra, Pinhal de Marrocos, 3030-290 Coimbra, Portugal*

Keywords: Project Management, QSOS Methodology, Open-source, Filter, Maturity, Redmine, Open Project, GitLab.

Abstract: Measure and planning all aspects and variables of a project is extremely important to have success. Therefore, having the right tool is extremely important. To evaluate project management tools, we can use several methodologies, that allow to choose the best tools according to our criteria. QSOS is one of the methodologies that allows to make a more weighted choose. In this paper, we evaluate popular open source project management tools GitLab, OpenProject, and Redmine, using QSOS methodology.

## 1 INTRODUCTION

Project management is important from the beginning of the project, to ensure that what is being delivered, is right and aligned to what was intended to be done. Accordingly, it can be considered as the discipline of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria.

According to the Project Management Institute (PMI) and PMBOOK (2017), Project management knowledge draws on ten areas: Integration; Scope; Time; Cost; Quality; Procurement; Human resources; Communications; Risk management; and Stakeholder management.

With the landscape of project management changing every day, it is important to know what we have to manage. Puttying things into perspective, project management is the art of making a plan, then execute it to deliver an output(s) that will benefit the organization (Kashyap, 2018).

Having the right tool to analyse this “perspective” is essential to keep the tracks of the project on the right way and always updated.

To evaluate the tools, we also must keep in mind not only opinions, but functionality’s and formal evaluations of what the tool does and their type of community and licence.

The QSOS - Qualification and Selection of Open source project is a free project, created in 2004, to evaluate open source software and there functionalities. This methodology focus not only in functionalities, but in maturity of the software, community and license bases of the Team/Project.

In this paper, we analyze three open-source management tools, using Qualification and Selection of Open Source Software (QSOS) methodology.

The Project Management Zone, (Project-management zone, 2019), ranks project management systems according to their popularity. The ranking is updated monthly. And this classification bases itself on number of mentions of the system on websites, number of jobs offers, in which the system is mentioned, number of profiles in professional networks, in which the system is mentioned, relevance in social networks and importance of the system's website. Therefore, according to this ranking, we choose the following three open-source management tools: GitLab, OpenProject, and Redmine.

The rest of this paper is organized as follows. Section 2 describes the three open-source managing project tools that will be evaluated. Section 3 presents a description of the QSOS methodology and Section 4 presents the evaluation of the tools with the parameters and measures of QSOS methodology. Section 5, presents a personal opinion and evaluation about the experiencing functions and plugins of the

[a] https://orcid.org/0000-0001-9660-2011



Vicente, A. and Bernardino, J.
Evaluating GitLab, OpenProject, and Redmine using QSOS Methodology.
DOI: 10.5220/0008364603800387
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 380-387
ISBN: 978-989-758-386-5

Redmine. Finally, Section 6, presents the conclusions and future work.

# 2 MANAGING PROJECT OPEN SOURCE TOOLS

To correct manage a project, there are several variables that we should have in count. What exactly make a good tool for this goal?

When it comes to the development of a software project, it is indispensable to always have in mind a system that allows us to manage all stages of project development, from documentation, to control deadlines. It is this management that will define a lot of the quality of the software or project that is being developed and is success.

According to PMBOOK (2017) is important to control and keep in mind all the 5 phases of a project, each one will influence the other: Initiating, Planning, Executing, Monitoring and Closing as is illustrated in Figure 1. The tools are all open-source, being this the main criteria for choosing the tools.

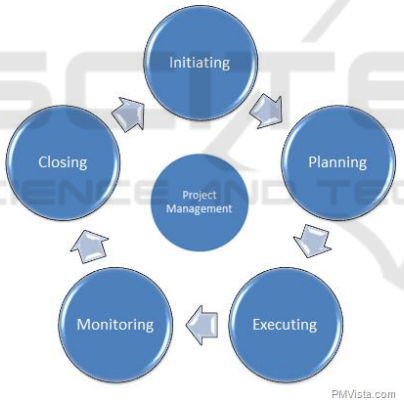


Figure 1: PMBOOK Project Stages relationship (source: http://www.pmvista.com/pmbok-knowledge-areas-and-processes/).

Redmine has been created at 15 years, with several developers and a strong community, as evolved, thanks to the plugins that can be yearly add to the program, gaining more functionalities.

OpenProject as start from the root of Redmine, diffing apart in terms of visualization applying. The integrations of new functionalities in the tool claimed to be easier to install and use.

Finally, the GitLab CE, a project that have grown a lot in the last years, the software main goal is not to manage a project, but the tool as starting to gain functionalities that are for project management.

## 2.1 GitLab

GitLab is a web-based platform more directed the principals of Agile and DevOps lifecycle. Provides a Git-repository with wiki, issue-tracking and CI/CD pipeline features.

As split into two distinct versions: GitLab CE: Community Edition and GitLab EE: Enterprise Edition.

Git tools have grown much in the past years becoming almost a standard in repository management, other management interfaces like GitLab got developed.

Combining the functionality's of managing a repository, integrating whit the life cycle of an Agile project and managing the other variables of the project, the uses for this tool have continuing to grow.

All this in the same platform that differentiates itself by having have a nice, clean and recent design and that is easy to learn and work in, with a rapid learning curve for the user. Manage and track issues, visualize work with issue boards, agile project management are strong features of this tool, but reporting and Gantt graph functionalities are not supported features and only archived by external add-ons.

More information can be found in (GitLab, 2019). GitLab interface is illustrated in the Figure 4.

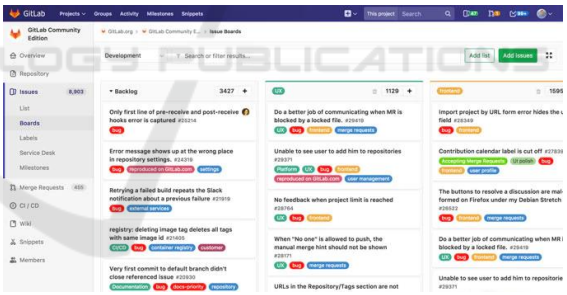

Figure 2: GitLab interface (source: https://www.getapp.com/it-management-software/a/GitLab/#gallery-1).

## 2.2 Open Project

OpenProject is a web-based project management system for location-independent team collaboration, starting from the root of Redmine.

OpenProject has been developed since 2010. The initial motivation for this fork was that the founding members want to have more performance, security and accessibility requirements, which could not be easily reached by plugins for either Redmine or ChiliProject (another fork of Redmine).

One of the main goals is to offer the highest standards in data security and privacy as well as

accessibility of most features. Most big companies have limited possibilities for choosing their project management software because many restrictions need to be fulfilled. This limits the choices and often even eliminates the possibility of using open source software for project management.

OpenProject supports those requirements and have several required functionality's for managing project as product Roadmap, Agile and Scrum, Time Tracking, Cost Reporting, Budgeting, Bug Tracking, and more.

The visual interface also got a huge update, standing simplistic, but looking much more appealing, but adding more functionality´s is not easy and open-source edition as limitations.

Can be found more information in (OpenProject.org, 2016).

It was first released in 2012, written in ruby, rails and angular is currently 8.0.0 version a support over 30 languages.

The OpenProject interface is shown in Figure 3.

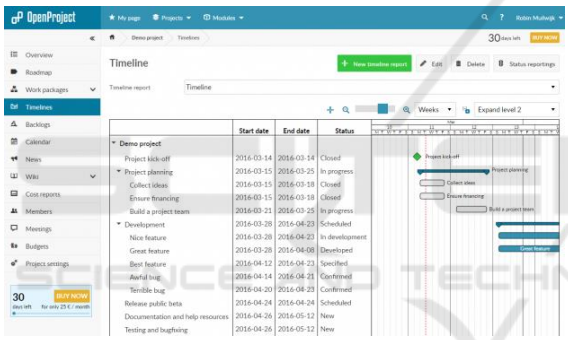


Figure 3: OpenProject interface (source: https://opensource.com/business/16/3/top-project-management-tools-2016).

## 2.3 Redmine

The Redmine is a web-based platform for project managing and an issue tracking tool, (Softwareadvice.com, 2017).

Users can manage multiple projects and associated subprojects. It features per project wikis and forums, time tracking, and flexible role-based access control. It includes a calendar and Gantt charts to aid visual representation of projects and their deadlines. Redmine integrates with various version control systems and includes a repository browser and diff viewer, allowing multiple plugins that add more functionalities to the tool, and some themes that make the interface friendlier.

This addable plugins and themes for this tool it's what set it apart from other and making it highly customisable. With a community of strong developers and contributors, if a feature is required or strongly need, certainly will be develop and available in Redmine free plugins.

Despite this some functionality's and plugins can be rudimentary and insecure. Opinions, main characteristics and limitations of this software can be found in (Quora, 2015). Is simplistic interface helps the users to not be lost.

It has first released in 2006, written in Ruby and Rails is currently in 4.0.0 version and supports over 34 languages. The basic interface is illustrated in the Figure 4.

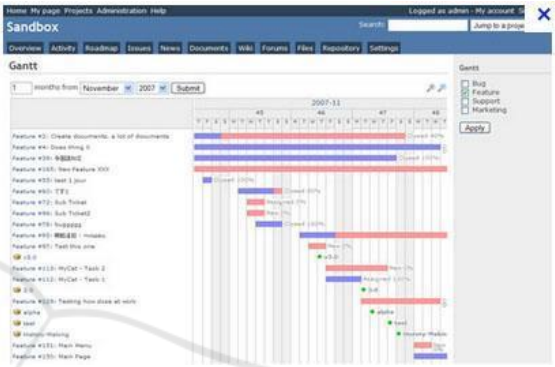


Figure 4: Redmine Interface (source: https://www.researchgate.net/figure/a-Redmine-Interface-b-Easy-Redmine-Interface_fig3_309747233).

# 3 QUALIFICATION AND SELECTION OF OPEN SOURCE PROJECT

It is necessary to have a method of qualification and selection suited to free and open source software while studying its adoption.

Which piece of software matches my current and anticipated technical and functional needs? Before adopting a software, every company should considerate this question´s before deciding:

- What is the continuity of this piece of software? What are the odds of choosing a fork? How to anticipate and manage it?
- What is the level of stability to expect? How to manage dysfunctions?
- What is the required and available level of support of this piece of software?
- Is it possible to influence the software development (adding new features)?

To answer that and make an informed decision, it is required to have a methodology that allows to:

- Qualify of a piece of software by integrating the specificity of free and open source software;
- Compare several pieces of software depending of the needs and weighting criteria to make a correct final decision.

QSOS methodology considers the following criteria/points: Maturity, Community and Licencing of the software, points that are explained in the next sections. QSOS completed information can be found in (Dist.qsos.org, 2013).

## 3.1 Target of the Method

This methodology as an approach for evaluation free and open source software can be used by:

- People inquiring on the method either as professionals or as non-professionals;
- Free and open source communities;
- IT experts that want to know and apply this method in their day-to-day activities of evaluation and selecting components/programs to build software solutions meeting their own or their customers' needs.

## 3.2 General Approach

The QSOS approach is composed by four independent steps/phases:

- Define - definition and update of the reference used for the next steps;
- Evaluate - evaluation of a version of a piece of software (functional coverage and maturity of the project);
- Qualify - weighting the criteria according to the context;
- Select - Comparison and selection of software, based on previous steps data.

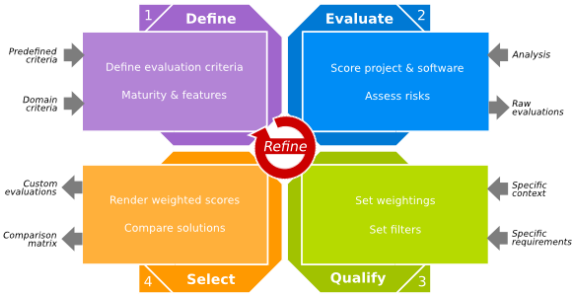


Figure 4: QSOS Phases (source: http://dist.qsos.org/qsos-2.0_en.pdf ).

### 3.2.1 Step 1: Define

Here we define different elements of typology that will be used during the next three steps of process.

- Type of software: the hierarchical classification of types of software and the description of functional coverage in the form of templates;
- 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.

The templates are 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. These criteria must be used in every single QSOS evaluation.

### 3.2.2 Step 2: Evaluate

This step evaluates the free and open source software. Information on the open source community are retrieved to score the software based on the criteria from the previous step. This analysis grid or template is then a tree of criteria (see Figure 5).

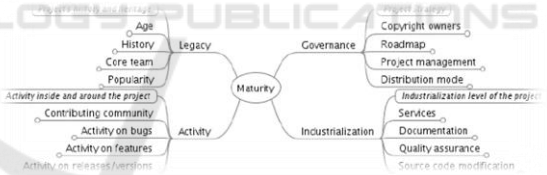


Figure 5: Phases of evaluation QSOS (source: http://dist.qsos.org/qsos-2.0_en.pdf).

Criteria are assigned a discrete score from 0 to 2. 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 usually:

| Score | Description |
|---|---|
| 0 | Functionality not covered. |
| 1 | Functionality partially covered. |
| 2 | Functionality fully covered. |

Figure 6: Criteria scores QSOS (source: http://dist.qsos.org/qsos-2.0_en.pdf).

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

### 3.2.3 Step 3: Qualify

This step is to define a set of elements translating the needs and constraints lined to the selection approach of a piece of 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.

#### Filters.

The first level of filtering can be set on the data relative to the software identity.

It can be, for example, to consider only the software of a certain type, or only of a certain type of license.

#### Maturity Filter.

The degree of relevance of every maturity criterion is set depending on the context:

- Not relevant criterion, not to be included in the filter;
- Relevant criterion;
- Critical criterion.

This degree of relevance will be translated into a weighting value in the next step of the process, depending on the chosen selection mode.

#### Functional Coverage Filter.

Every functionality described in the evaluation template is assigned a level of requirement, in the following list:

- Required functionality;
- Optional functionality;
- Not required functionality.

These requirements will be associated to weighting values during the Select step, depending on the chosen selection mode.

### 3.2.4 Step 4: Select

This step is selected the software that matches the user's needs.

Two modes are available to make the decision:

- Strict selection;
- Loose selection.

#### Strict 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 pass in 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.

Depending on the demands of the user, this strict selection can return no eligible software.

#### Loose Selection.

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

Is based on the weighting values that obey to certain roles.

**Weighting of Functionalities.**

The weighting is based on the level of requirements of every functionality of the functionality coverage and is shown in Figure 7.

| Level of requirement | Weighting |
|---|---|
| Required functionality | 3 |
| Optional functionality | 1 |
| Not required functionality | 0 |

Figure 7: Criteria weight 1(source: http://dist.qsos.org/qsos-2.0_en.pdf**)**

**Weighting of Maturity.**

The weighting is based on the degree of relevance of every maturity criteria as illustrated in Figure 8.

| Degree of relevance | Weighting |
|---|---|
| Critical criterion | 3 |
| Relevant criterion | 1 |
| Not relevant criterion | 0 |

Figure 8: Criteria weight 2 (source: http://dist.qsos.org/qsos-2.0_en.pdf)

# 4 EVALUATING GitLab, OPEN PROJECT AND REDMINE WITH QSOS

For analyse the 3 tools was used the QSOS methodology in the next terms.

## 4.1 Step 1: Define

The 3 parameters considered are:

**Type of Software:**
- Legacy: Project's history and heritage – Group1;
- Activity: Activity inside and around the project – Group 2;
- Governance: Project's strategy – Group 3;
- Industrialization: Industrialization of the project – Group 4.

**Type of License:**
- Redmine: GPL/LGPL
- Open Project: GNU Public License
- GitLab: MIT License

**Type of Community:**
- Redmine: Developers organization
- Open Project: Developers organization / Commercial entity
- GitLab: Developers organization / Commercial entity

As open-source products, Redmine, Open Project and GitLab (CE) there are not much difference between them in terms of the type of license or community. The MIT Licence of GitLab is the more restrictive type of licencing that GPL/LGPL and GNU Public License.

## 4.2 Step 2: Evaluate

The matrix to evaluate the maturity is the following illustrated in Table 1.

## 4.3 Step 3: Qualify

In this phase, there are no Identity filters to be set. As maturity filter and functional coverage filter the group 3 and 4, are critical and required feature the most relevant, for choosing the most completed open-source tool. Other groups can be considered as relevant and optional.

Table 1: Groups of maturity and score´s.

| Measures | Score |
|---|---|
| ***Group 1*** | |
| Age | From 0 to 2 |
| History | From 0 to 2 |
| Core team | From 0 to 2 |
| Popularity | From 0 to 2 |
| ***Group 2*** | |
| Contributing community | From 0 to 2 |
| Activity on bugs | From 0 to 2 |
| Activity on features | From 0 to 2 |
| Activity on releases/versions | From 0 to 2 |
| ***Group 3*** | |
| Copyright owners | From 0 to 2 |
| Roadmap | From 0 to 2 |
| Project management | From 0 to 2 |
| Distribution mode | From 0 to 2 |
| ***Group 4*** | |
| Existing service (support, training, audit) | From 0 to 2 |
| Documentation | From 0 to 2 |
| Quality assurance: QA process | From 0 to 2 |
| Source code modification | From 0 to 2 |

Table 2: Measure and normalized weight for evaluation.

| **Weighting of functionalities and Maturity** | **Max: 8** | 100 % |
|---|---|---|
| Required functionality and Critical Criterion | 6 | 75% |
| Optional functionality and Relevant Criterion | 2 | 25% |
| Not Required functionality | 0 | 0% |
| No relevant criteria | 0 | 0% |

## 4.4 Step 4: Select

The final classification made join the criteria scores of loose selections for criteria weight 1 and criteria weight 2. The final score is the addition of the classification normalizing the weight´s in total of 100%.

Comparing the final results, the scores for the tree tools are presented in Table 3.

Joining all the points, the final classification of the tools is shown in Table 4.

According to the analysed QSOS methodology and the applied filters to these 3 open-source management tools, the best tool is Redmine.

Table 3: Evalution of Redmine, Open Project and GitLab.

| **Required and Critical - Group 3 and 4** | | | |
|---|---|---|---|
| **GitLab (CE)** | **Open Project** | **Redmine** | **Value** |
| 6 | 5 | 7 | |
| 5 | 5 | 6 | + |
| 11 | 10 | 13 | = |
| 8,25 | 7,5 | 9,75 | **75%** |
| **Optional and Relevant Group 1 and 2** | | | |
| GitLab (CE) | Open Project | Redmine | **Value** |
| 7 | 7 | 7 | |
| 8 | 7 | 6 | + |
| 15 | 14 | 13 | = |
| 3,75 | 3,5 | 3,25 | **25%** |

Table 4: Normalized score of the Redmine, Open Project.

| **Weight** | **GitLab (CE)** | **Open Project** | **Redmine** |
|---|---|---|---|
| **25% - Group 1, 2** | 3,75 | 3,5 | 3,25 |
| **75% - Group 3, 4** | 8,25 | 7,5 | 9,75 |
| **Final Score** | 12 | 11 | 13 |

# 5 EXPERIENCE WITH REDMINE

The Redmine software can be installed for several platforms as Windows, Linux, and others. One of the easiest way to correct install an initiate Redmine, is to use the official image existing in Docker Image Hub, running a container. Using Docker and containers was become almost a standard and one of the fastest way to run service or application, providing a based and clean environment that is configurable based on the image that is used.

## 5.1 Using Redmine

After installing or running Redmine we can access their interface in a browser using the account User: admin, Password: admin.

The basic interface and login menu are illustrated in the Figure 9.


Figure 9: Redmine Login interface.

The projects are configured in the tab Projects, and the base functionalities are Overview, Activity, Tasks, Time spent, Gantt, Calendar, News, Documents, Wiki, Files/Documents and the Configurable Settings by the project manager. Project interface and sub menus are shown in the Figure 10.

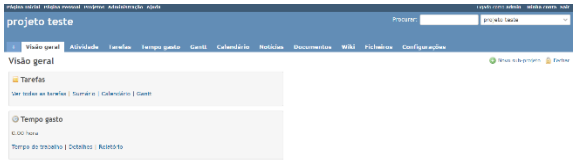


Figure 10: Redmine Project interface.

Options and functionalities available for the projects are configurable in administration global tab. The available options are Projects, Users, Groups, Functions and Permissions, Types, Task States, Workflow, Custom fields, Enumerations, Settings, LDAP Authentication, Extensions and Information.

Extensions and their information are visible and activated and configured in Extensions tab.

Themes and visual interface are configured in Configuration tab, as illustrated in Figure 11.

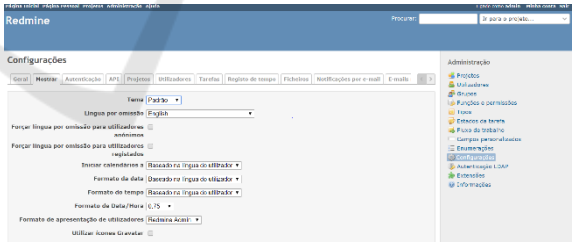


Figure 11: Redmine Configuration Themes interface.

## 5.2 Evaluating Redmine Functionalities

Redmine default functionalities help managing a project with functionalities like Gant charts, Time tracking, Wiki, Tasks and Calendar. Experiencing Redmine was shown that the tool, although is also an issue tracker, also can manage and have functionalities for project manager that help controlling variables over the time.

Is not difficult to find a pretended functionality, and downloaded as extension for the Redmine. The

community interactions, provides a repository off on progress functionalities being developed, that allows the users to install and uninstall extensions, with proper instructions easily. The existing functionalities are in general for project managing.

The themes work in same way as the extensions, can be added and deleted, improving the visual interface.

Redmine is also easy to use, the simple and functional interface helps users to work more quickly and efficient.

Therefore, Redmine can be intuitive and easy tool to manage a project and also an issue tracking, it is functionalities growing by the community each day. The functionalities are not directly related with Project Management Institute (PMI) and PMBOOK (2017), and cannot be the most appropriated tool in these aspects. It will depend on the aspects that will be considered relevant for managing a project.

# 6 CONCLUSIONS AND FUTURE WORK

QSOS methodology gives a weighting about the maturity and type of software and community of open-source software. Because it is specialized in open-source software and can give a good view of the software maturity and type, although it does not directly evaluate features, not providing a good view and evaluation in this aspect.

According to this selection and looking to governance project's strategy and Industrialization of the project the best tool is Redmine.

Created in 2004, have been evolving and growing in relevant features for project management. Could not be the best-looking software, but with is one of the more stable and functional.

As future work, we intend to consider some functionality and particular aspects of project management tools, as Kanban board, Pert, Gantt and burndown charts, time tracking and others, as relevant aspects of these tools.

We also intend to use some frameworks of evaluation, based on QSOS, that had not been used in this work.

The Redmine software and its plugins did not focus in any particularly aspect of PMBOOK. The community will continue to develop helpful plugins, and maintaining the project upgradable, runnable, costumized, wich makes it one of the more completed. Therefore, new versions will be evaluated.

# REFERENCES


Dist.qsos.org. (2013). [online] Available at: http://dist.qsos.org/qsos-2.0_en.pdf [Accessed 23 Apr. 2019].

Kashyap, S. (2018). Why is Project Management Important: Benefits, Importance, and Tools. [online] ProofHub. Available at: https://www.proofhub.com/articles/why-is-project-management-important [Accessed 8 Jun. 2019].

Jean-Christophe Deprez, Simon Alexandre, 2008, https://www.researchgate.net/publication/225142074_Comparing_Assessment_Methodologies_for_FreeOpen_Source_Software_OpenBRR_and_QSOS

Pmi.org. (2017). PMBOK 6ªth edition - Guide and Standards. [online] Available at: https://www.pmi.org/pmbok-guide-standards [Accessed 4 Apr. 2019].

Semeteys, R. (2008). Method for Qualification and Selection of Open Source Software. [online] Available at: https://timreview.ca/article/146 [Accessed 12 Apr. 2019].

Project-management.zone. (2019). Popularity Ranking of 35 Open Source, Project Planning Project Management Systems. [online] Available at: https://project-management.zone/ranking/open-source,planning [Accessed 22 Apr. 2019].

GitLab. (2019). Product. [online] Available at: https://about.GitLab.com/product/project-management/ [Accessed 23 Apr. 2019].

OpenProject.org. (2016). OpenProject – the user friendly alternative to Redmine » OpenProject.org. [online] Available at: https://www.openproject.org/openproject-an-alternative-to-redmine/ [Accessed 23 Apr. 2019].

Quora. (2015). Quora. [online] Available at: https://www.quora.com/What-is-your-experience-using-Redmine-for-project-management-and-collaborating [Accessed 23 Apr. 2019].

Opensource.com. (2016). Top 11 project management tools for 2016. [online] Available at: https://opensource.com/business/16/3/top-project-management-tools-2016 [Accessed 8 Jun. 2019].

Anon, (2016). [online] Available at: https://www.researchgate.net/figure/a-Redmine-Interface-b-Easy-Redmine-Interface_fig3_309747233 [Accessed 8 Jun. 2019].

Pmvista.com. (2017). PMBOK 5 Knowledge Areas and Processes | Project Management Guide. [online] Available at: http://www.pmvista.com/pmbok-knowledge-areas-and-processes [Access 10 Jun. 2019].

Reviewer, A., Reviewer, A., Maclennan, C. and Vaghela, N. (2018). GitLab Pricing, Features, Reviews & Comparison of Alternatives. [online] GetApp. Available at: https://www.getapp.com/it-management-software/a/GitLab/#gallery-1 [Accessed 8 Jun. 2019].

Softwareadvice.com. (2017). Redmine Software - 2019 Reviews, Pricing & Demo. [online] Available at: https://www.softwareadvice.com/project-management/redmine-profile/ [Accessed 10 Jun. 2019].