

Project Management Tools Assessment with OSSPal

Samuel Cruz¹ 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, Open-source Project Management Tools, Microsoft Planner, Gitlab,]Project-Open[.

Abstract: In this paper it is highlighted the importance of using a methodology to evaluate project management tools and chose the one that will make the project management tasks easier. Three project management tools, GitLab, Microsoft Planner and]Project-Open[, will be analysed by an open-source assessment methodology OSSPal that focus on important features of this kind of tools. This is one of the most correct and efficient way to choose which tool should be used in a project.

1 INTRODUCTION

With the increasing complexity of technologies, the human being invented new tools that will help him doing certain tasks in order to save time and effort, in this particular case, the focus will be project management tools that make every single task related to the project management clearly easier.

The bigger and longer the task is the hard is to estimate and control each and every single aspect related to it. Most people will have a quite accurate answer when asked how long will they take to walk ten meters because it is an easy task to do however when asked to walk a kilometre the accuracy of the answers will greatly decrease. Longer distance, exhaustion, and weather are a few of the characteristics that will most likely change along the walk and eventually affect the time needed. This hypothetical example intends to highlight the fact that the bigger and longer the task is, the harder will be to control it because the number of aspects that can influence its outcome tends to be bigger too.

Project management has an important part in the success of the project although it does not always have the right amount of effort dedicated to it. Fortunately, this is something that has been decreasing and as time passes by, people start to pay more attention and dedicate more time to tasks related to project management.

To encourage the use of project management tools and the importance of a good project management

this article will focus mostly on open-source free project management tools that everyone can use upon acquiring a certain level of knowledge about this field.

The three project management tools presented in this article will be GitLab, Microsoft Planner, and]Project-Open[. In this paper, all three project management tools will be analysed and evaluated by the OSSPal methodology.

The OSSPal methodology is well known on the scientific community and it is used to compare a vast type of tools, software, and many other products. There are many articles available with information about how to apply it (Ferreira, Pedrosa and Bernardino, 2018) and examples of it (Pereira, Sousa and Santos, 2018). This paper gives another example of the application of this methodology to a totally different subject. Here the methodology will be adapted to project management tools.

The present paper is organized in the following way. Section 2 will describe all three previously mentioned tools. Section 3 presents a description of the OSSPal methodology and Section 4 presents the evaluation of the tools with the application of OSSPal methodology. Finally, Section 5 presents the conclusions and future work.

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

2 PROJECT MANAGEMENT TOOLS

Today there are many open-source management tools available for free online. By doing a quick search on any browser it is possible to find many of them and even make a simple comparison between them.

Based on research (Top 7 open source project management tools for agile teams | Opensource.com n.d.) and on previous experience with the tools the three tools chosen were GitLab, Microsoft Planner and]Project-Open[. This section is exclusively focus on introducing these three tools by giving short aspects on their origin, advantages and disadvantages.

2.1 GitLab (about.Gitlab.Com)

GitLab is a widely known open-source tool that can be used both for repository management and project management. It is the only tool in this list can has both of these functions. It is important to remind that although it has some interesting features related to repository management that will not be the focus of this article.

GitLab project began in 2011 and since then there have been multiple releases, being the latest the GitLab 11.9 (Friday, March 22, 2019) (History of GitLab | GitLab n.d.) . It is written in several programming languages like Ruby on Rails, Go and Vue.js (Why we use Ruby on Rails to build GitLab | GitLab n.d.).

GitLab has been growing a lot and many important tech companies like NASA, Siemens, Avast and many others decided to adapt this tool to their processes (Case studies from GitLab customers | GitLab n.d.).

The main strengths of GitLab on project management are:

- It can be accessed through a browser;
- It can be installed in different ways and there is documentation supporting each one of them.
- It has an excellent GUI – simple and very intuitive;
- It has a good amount of documentation on the official site that can be accessed by everyone;
- It has a solid community that has been increasing along the years where people can expose their problems;
- The free version, the Community Edition, has many features and the costs of upgrade are lower compared to other project management tools (GitLab Pricing | GitLab n.d.).

Problems that this tool has:

- When accessing from the browser it can be slow
- To have access to burndown charts it is needed a upgrade which means it has costs.

Figure 1 shows a standard interface of GitLab.

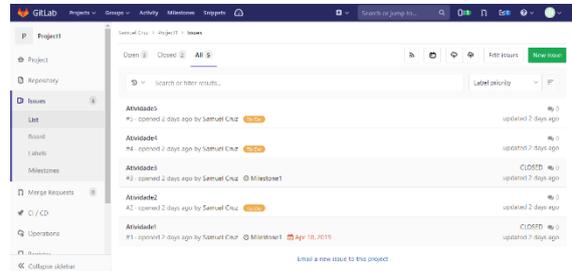


Figure 1: Example of Gitlab interface.

2.2 Microsoft Planner (Tasks.Office.Com)

Microsoft Planner is a tool used mainly for project management associated with Microsoft Office 365. It is available in premium version for business and for students who have a license which is the case. It was launched in 6th July 2016.

It is the only tool in this list that is not free to everyone and also it is not open-source meaning that probably should not even be here but there are some good aspects about it that took an important weight in the decision of keeping this tool in the list.

It is important to mention that some of the following characteristics were written based on personal experience working with this tool.

The main strengths of Microsoft Planner on project management are:

- It can be accessed through the browser. No need for installation;
- Very simple and intuitive GUI. It is extremely easy to use;
- It is free for students with a Microsoft Office 365 license;
- It is a Microsoft product which means it has multiple languages.

Problems that this tool has:

- It is not open-source nor free for everyone;
- It lacks many features like for example the Gant chart or the burndown chart previously mentioned (Microsoft Planner Features & Capabilities | GetApp@ n.d.);
- It is not easy to export reports since plenty of times they end up deformed.

Figure 2 shows the interface of Microsoft Planner.

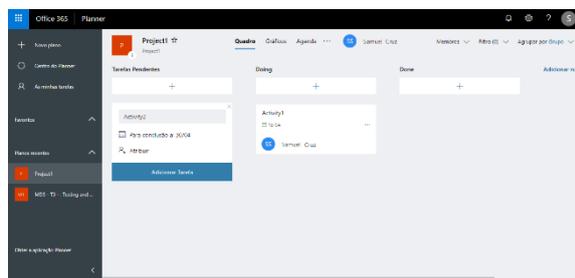


Figure 2: Example of Microsoft Planner interface.

2.3]Project-Open[(Www.Project-Open.Com)

The final tool in this list is the]Project-Open[. This is a pure project management tool that respects every aspect demanded on this search: it is open-source and it is free.

]Project-Open[is the oldest of these three tools. The project started in June of 2002 although it was not]Project-Open[yet. Past that there have been a few new releases and updates and the last version is V 5.0 (Roadmap n.d.).

The main strengths of]Project-Open[on project management are:

- It has more features available than the other two tools previously mentioned since it is a mainly focus project management tool (]project-open[Enterprise Project Management - Community Edition - Professional Edition - Enterprise Edition n.d.);
 - It has a good amount of support documentation on the official site;
 - It has budget and human resources management.
- Problems that this tool has:
- Unlike the previous presented tools, it requires an installation process;
 - There are few tutorials about this tool available. It is not a well known tool or company;
 - The GUI is not as simple and intuitive as the other two tools previously mentioned.

3 OSSPal METHODOLOGY

OSSPal is a well-known methodology created after the Business Readiness Rating (BRR) whose main objective is to evaluate open-source software from a precise perspective following criteria that assures that

the result reached is logic and consistent from an expert point of view (About OSSPal | OSSPAL n.d.).



Figure 3: Example of]Project-Open[interface (Demo n.d.).

OSSPal evaluation criteria is based on seven categories according to Wasserman et al., (2017):

- **Functionality:** How well will the software meet the average user’s requirements?
- **Operational Software Characteristics:** How secure is the software? How well does the software perform? How well does the software scale to a large environment? How good is the UI? How easy to use is the software for end-users? How easy is the software to install, configure, deploy and maintain?
- **Support and Services:** How well is the software component supported? Is there commercial and/or community support? Are there people and organizations that can provide training and consulting services?
- **Documentation:** Is there adequate tutorials and reference documentations for the software?
- **Software Technology Attributes:** How well is the software architected? How modular, portable, flexible, extensible, open, and easy to integrate is it? Is the design, the code, and the tests of high quality? How complete and error free are they?
- **Community and Adoption:** How well is the component adopted by community, market, and industry? How active and lively is the community for the software?
- **Development Process:** What is the level of the professionalism of the development process and of the project organization as a whole?

The implementation of this methodology can be divided in four phases which will be shortly described in this article although if any doubt appears this information is clearly justified in articles who are referenced in the end (Ferreira, Pedrosa and Bernardino, 2018). The four phases are:

1. Identification of the features and components that will be the scope of this evaluation.

2. Assigned a weight to each category totalling 100%. If a category needs to be analysed in-depth it may be a good idea to define subcomponents or subtasks that will help evaluate with more accuracy. Each of this subcomponents or subtasks will also receive a weight and once again the sum of every subcategory weight should be 100%.
3. Once the evaluation methodology is defined (steps 1 and 2) it is time to start collecting the data that will help to give a classification value to each category or subcategory. This value is contained within the scale presented on the Table 1.

Table 1: Evaluation scale.

1	Unacceptable
2	Poor
3	Acceptable
4	Very Good
5	Excellent

4. Finally, after every category and subcategory as a classification value it is time to calculate the final value for each software that is being evaluated. Starting from the subcategories each classification value is multiplied by its weight and then sum every result to get the final result for the category. The process is repeated to every complex category who needs it and once its over the process is repeated one last time to the general and final categories. The results are compared with each other to take the final conclusions.

4 EVALUATION

The first task to do in this section is to give the proper weight to each one of the seven categories previously described. This part can be somehow subjective because not everyone may have agreed with the values given to every category. Each individual person has their own approach to what they think is more important. In this work the weights were give as described in the Table 2.

Functionality and operational software characteristics are the most valuable categories since both of them focus on important aspect of the software like security, user's requirements, installation and learning process.

Since the focus is open-source software the weight given to software technology attributes should

also have a high weight because it evaluates the architecture, code, flexibility and tests for example.

Table 2: Weights given to each of the seven categories that will be the focus of this evaluation.

Category	Weight
Functionality	25%
Operational Software Characteristics	25%
Software Technology Attributes	14%
Documentation	10%
Community and Adoption	12%
Support and Service	8%
Development Process	6%
Total	100%

Community and adoption are the next category with the highest weight followed by documentation. Both of these weight values were given thinking about how the users rely on them to improve their working performance with the tool. It is a fact that most users rely on community and documentation to solve problems and not so much on the support and service that the official team provides simply because it is usually faster and easier. This is the reason why support and service have such a low weight.

Finally, the development process has lowest weight on the scale because it studies aspects that are not heavily related to user satisfaction which is the focus of this article.

Functionality is a vast category to evaluate so it will be subdivided in twelve subcategories. Each subcategory will represent a necessary feature that is a requirement to the user. Once again, the twelve features chosen may change from individual to individual since there are many ways to approach this problem.

The twelve chosen subcategories were:

- Task management features like the possibility to plan and assign a task, keep tracking that task and emit warnings once it is reaching the deadline.
- The following features are related to the project management in general like the construction of milestones and a baseline that can be used to maintain the project on the scope on project tracking and a risk management possibility.
- Some additional features like tagging, the Kanban board which makes the management of issues easier, priority support and the possibility to have scrum and agile approach.

The weights defined to each subcategory can be found on Table 3.

Table 3: Weights for the subcategories of the functionality.

Features	Weight
Assignment Management	2
Assignment Tracking	2
Deadline Tracking	3
Tagging	1
Task Planning	1
Milestone Tracking	2
Baseline Management	2
Project Tracking	3
Risk Management	3
Scrum & Agile	2
Issue Board / Kanban Board	2
Priority Support	2
Total	25

The next step is to evaluate the strength of each subcategories in the previously mentioned tools and assign a value between 1 and 3:

- 1 – Does not have the feature;
- 2 – It has the feature but is underdeveloped;
- 3 – It has the feature optimized.

The evaluation given to each category can be found on Table 4.

Table 4: Value given to each tool according to the functionality subcategories.

Category	GitLab	MS Planner]Project-Open[
Assignment Management	3	3	3
Assignment Tracking	2	2	2
Deadline Tracking	3	3	3
Tagging	3	3	2
Task Planning	2	2	2
Milestone Tracking	3	1	3
Baseline Management	1	1	2
Project Tracking	2	2	3
Risk Management	1	1	3
Scrum & Agile	3	2	2
Issue Board / Kanban Board	3	3	2
Priority Support	3	1	2

Once the table is filled sum the multiplication of each assigned value of each feature by its previously defined weight and then divide the result by the total of weight, which is 25 in this case. For space reasons it will only be presented the procedure made to GitLab as a demonstration, this applies on every example until the end of the article.

$$GitLab = (3*2 + 2*2 + 3*3 + 3*1 + 2*1 + 3*2 + 1*2 + 2*3 + 1*3 + 3*2 + 3*2 + 3*2) / 25 = 2.36$$

The Table 5 presents the value each tool got according to the pre-defined scale from 1 to 3.

The next step is to convert that result to a 1 to 5 scale that will be used in the final table. Divide the tool total by the total of the scale and the result should be a percentage that once multiplied by 5 will give the end result to the functionality category. The results can be seen on Table 6.

$$GitLab = (2.36 / 3) * 5 = 3.933333$$

Table 5: Total that each tool got in a scale from 1 to 3.

	GitLab	MS Planner]Project-Open[
Total	2.36	1.96	2.52

Table 6: Total that each tool got in a scale from 1 to 5.

	GitLab	MS Planner]Project-Open[
Total	2.36	1.96	2.52
Percentage	79%	65%	84%
Final	3.933333	3.266666667	4.2

Now that the hardest category to evaluate is finished the next step is simply to evaluate the general conditions of the tool and assign a number. Bellow there will be presented the reasons of why that was the assigned value.

GitLab

- Operational Software Characteristics: it does not even need installation, can be accessed via browser; the UI is really accessible and simple; it scales easily and it is secure.
- Support and Services: the software is well supported.
- Documentation: the official site has a good amount of documentation.
- Software Technology Attributes: GitLab has a well constructed architecture and they are continuously releasing new versions to fix bugs (in recent times it has been one release per month).
- Community and Adoption: GitLab has a big community where people are actively present.

- **Development Process:** although this category’s weight does not affect the result that much, GitLab is a well-known company and it surely has a good organization and professionalism behind its work otherwise it would never get this big.

In the Table 7 it can be seen the value given to each tool in every category in a scale from 1 to 5.

Table 7: Value given to each tool in every category in a scale from 1 to 5.

	GitLab	MS Planner]Project-Open[
Functionality	4	3.2	4.2
Operational Software Characteristics	4.5	4.2	4
Software Technology Attributes	4.5	3.7	4
Documentation	4.6	4.2	3.2
Community and Adoption	4.6	3.5	4
Support and Service	3.2	3	3.2
Development Process	4.5	4.2	3.6

For an easy comparison of the results of every aspect each one was indicated in a radar chart that can be seen on Figure 4.

The last step of this methodology is to multiply each category value by its weight and then sum all the results to get the total of the tool.

$$\text{GitLab} = 0.25*4 + 0.25*4.5 + 0.14*4.5 + 0.10*4.6 + 0.12*4.6 + 0.08*3.2 + 0.06*4.5 = 4.293$$

Table 8: Final result of each tool in a scale from 1 to 5.

	GitLab	MS Planner]Project-Open[
Total	4.293	3.724	3.882

It is possible to assume that GitLab is the best tool to use since it was the one that got the highest rate of all three tools that were the scope of this analysis as it can be confirmed on Table 8.

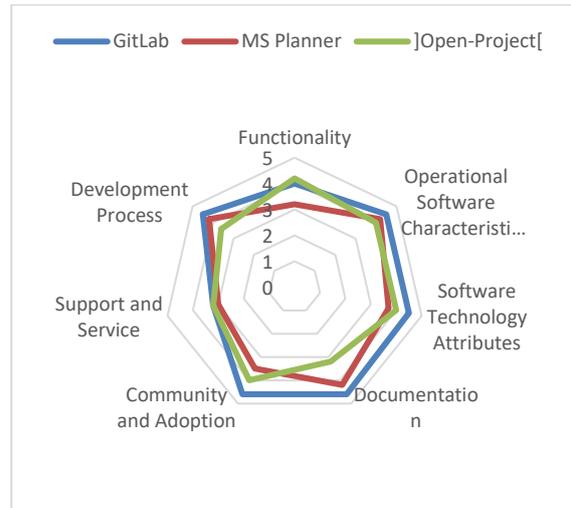


Figure 4: Radar chart.

5 CRITICAL ANALYSIS

Each tool has its own vantages and advantages so it might not be totally fair to choose one over the other but still this section is reserved for that only.

Both Microsoft Planner and GitLab were tools that we have already had the pleasure to work with on the past and that was one of the main reasons they were chosen to be part of this list. They are both very simple to use and they require no installation process. A downside to Microsoft Planner is that the tool is not free for a regular user neither open-source. On the opposite side, GitLab is both free and open-source and has many more features than Microsoft Planner. GitLab access by browser can be slow sometimes but it is rare for this to happen. Still, if these problems are occurring very often it can be installed in your personal or work devices and, as long as they meet the right requirements, this problem is easily solved.

Finally, the]Project-Open[is a good tool use specifically for project management and has some excellent features but it is harder to use since it requires a installation process that is not very clear and the acces to the tool is one of the main aspects of the evaluation done previously.

Overall GitLab is probably the best tool presented on this list according to all aspects previously mentioned although opinions may differ from one person to another. If your project also needs a repository management then there is no better tool you can use than GitLab.

6 CONCLUSIONS AND FUTURE WORK

In this paper the OSSPal methodology was used to compare three project management tools, GitLab, Microsoft Planer and]Project-Open[and the results reached were exactly what was expected. GitLab is a widely known tool which can only mean that it is well built and has good features for the task it was tested, project management.]Project-Open[is also a good tool but is not as nearly known as GitLab. There are plenty other project management tools that are better than this one such as Open Project, Orange Scrum. And finally, Microsoft Planner is a easy to use tool but it is not free neither open source and has a very limited amount of features. According to this, GitLab should have the best score followed by]Project-Open[followed by Microsoft Planner. This is exactly the result this methodology gave.

This methodology is very flexible, it can adapt to the complexity of the categories under analysis and the process involves easy calculations.

The train of thought can be used to help decision taking on many areas, not only with project management tools. If anyone needs to have a solid vision over which tool, product or method should implement this methodology can really help although might need to have some changes on the scopes of evaluation.

As future work, we intend to test the same tools with other open-source evaluation methodology and compare the results with the ones presented in this work.

REFERENCES

- “]Project-Open[Enterprise Project Management - Community Edition - Professional Edition - Enterprise Edition.” <http://www.project-open.com/en/products/editions.html> (April 18, 2019).
- “About OSSPal | OSSPAL.” <http://osspal.org/content/about-osspal> (April 19, 2019).
- “Case Studies from GitLab Customers | GitLab.” <https://about.gitlab.com/customers/> (April 18, 2019).
- “Demo.” <http://demo.project-open.net/intranet/> (April 19, 2019).
- “GitLab Pricing | GitLab.” <https://about.gitlab.com/pricing/> (April 18, 2019).
- “History of GitLab | GitLab.” <https://about.gitlab.com/company/history/> (April 18, 2019).
- “Microsoft Planner Features & Capabilities | GetApp®.” <https://www.getapp.com/collaboration-software/a/microsoft-planner/features/> (April 18, 2019).
- “Roadmap.” <http://www.project-open.com/en/project-open-roadmap> (April 18, 2019).
- “Top 7 Open Source Project Management Tools for Agile Teams | Opensource.Com.” <https://opensource.com/article/18/2/agile-project-management-tools> (June 6, 2019).
- “Why We Use Ruby on Rails to Build GitLab | GitLab.” <https://about.gitlab.com/2018/10/29/why-we-use-rails-to-build-gitlab/> (April 18, 2019).
- Baker, J. and Muilwijk, R. (2018). Top 7 open source project management tools for agile teams. [online] Opensource.com. Available at: <https://opensource.com/article/18/2/agile-project-management-tools> [Accessed 25 May 2019].
- En.wikipedia.org. (2019). GitLab. [online] Available at: <https://en.wikipedia.org/wiki/GitLab> [Accessed 18 Apr. 2019].
- Ferreira, T., Pedrosa, I. and Bernardino, J. (2018) ‘Evaluating Open Source E-commerce Tools using OSSpal Methodology’. 20th International Conference on Enterprise Information Systems. doi: 10.5220/0006790902130220.
- GitLab. (2019). *CHANGELOG.md · master · GitLab.org / gitlab-runner*. [online] Available at: <https://gitlab.com/gitlab-org/gitlab-runner/blob/master/CHANGELOG.md> [Accessed 18 Apr. 2019].
- Pereira, A. K., Sousa, A. P., Santos, J. R., and Bernardino, J. (2018). Open Source Data Mining Tools Evaluation using OSSpal Methodology. In *ICSOF* (pp. 706-712).
- Wasserman, A. I. et al. (2017) ‘Open Source Systems: Towards Robust Practices’, 496, pp. 193–203. doi: 10.1007/978-3-319-57735-7.