

Agile Project Management with Jira

Fredrik Björkman

`fbjorkma@kth.se`

Adam Henriksson

`adahen@kth.se`

April 2021

1 Introduction

In this essay the agile project management tool Jira will be discussed, as well as advantages and disadvantages of using Jira in the software development process.

We will not go into detail about how to set up and use specific features of Jira, but instead focus on the features that Jira provides, the problems Jira is trying to solve and the benefits of agile project management in workplaces that have embraced agile development practices.

2 Background

Traditional project management has viewed development as a linear or incremental process, where you follow a well-defined sequence of events when developing software [2]. This assumes that the developing team has complete and perfect information about the goals of the project and expected solutions. These traditional linear models were often referred to as the “Waterfall Model” [5], illustrated in Figure 1.

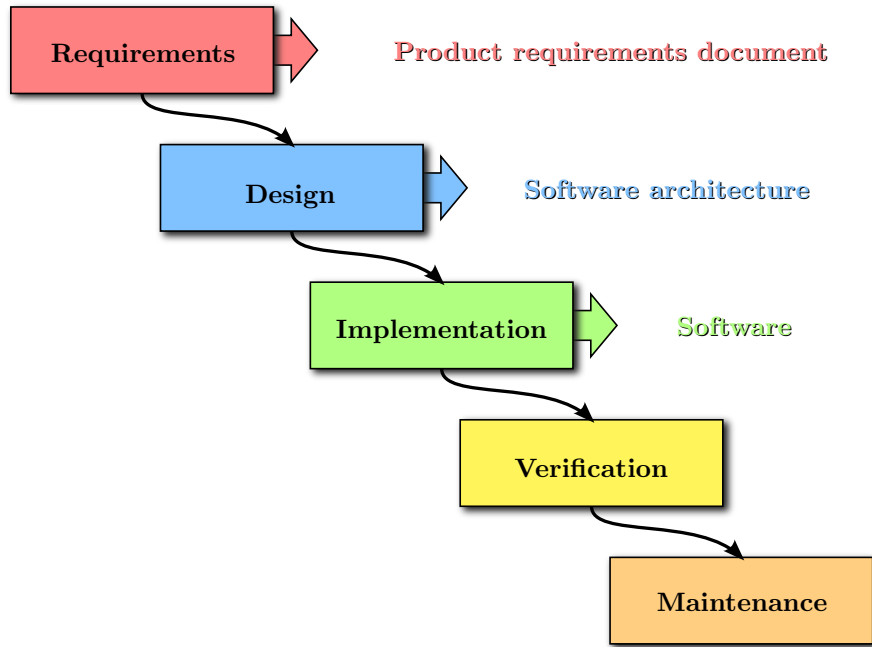


Figure 1: Waterfall Model. Source: Wikipedia [9]

In modern software development, radical changes in the customer requirements may occur frequently. New innovative ideas may arise from competitors or other unexpected events may happen that need to be addressed, there is a need for flexibility and the ability to react quickly is essential for the development process. One way to deal with this increased complexity in the project development is with agility and working with agile project management [4].

With the need for a more agile development process, agile methodologies such as Scrum and Kanban became more popular due to the more iterative approach of the software development [5]. Advantages that comes with a more iterative development strategy is that one can have continuous releases, which allows the customers to be more included in the development process and will be able to give feedback based on the current deployment [3]. Figure 2 illustrates

the iterative agile development process.

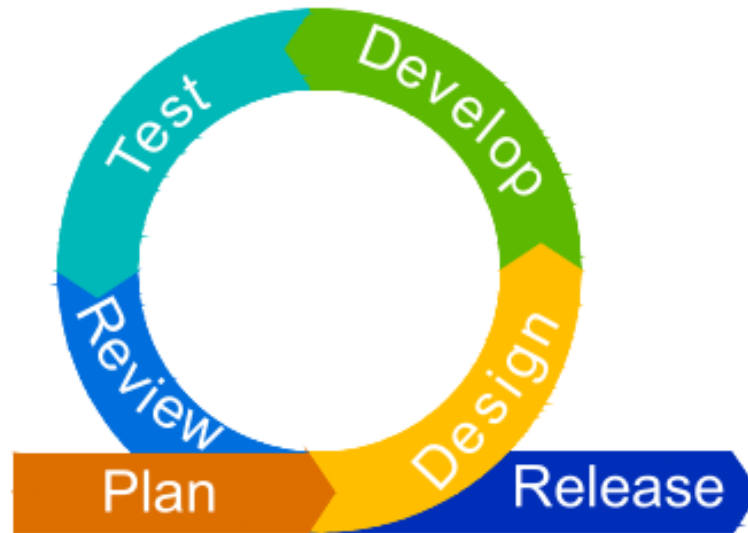


Figure 2: The agile development process.

With short iterations and where all the team members in a team working towards the same goal, coordinating and managing what needs to be done becomes an essential problem. For these problems tools like Jira can be of great use to assist in all the daily activities [8].

3 The project management tool Jira

In this chapter the project management tool Jira will be introduced and discussed and the advantages of using a project management tool for agile project management.

3.1 What is Jira?

Jira is a project management tool developed by Atlassian and was originally developed as a bug and issue tracker, but has today grown to be a powerful management tool [11].

First the features that help individual developers and small teams to organize their work and optimize their time will be shown. After that, the features that help managers plan and organize important milestones, get actionable data, achieve long-term goals, beat the competition and deliver value to customers will be discussed.

First things first, what are the tools that individual teams can use to organize their work in the simplest and most clear way possible?

3.2 Organize tasks, introducing Kanban

Kanban is a tactic introduced in lean manufacturing that was later adopted by the software development industry and integrated into agile development. If you cannot see what problems exist, it's hard to fix it. By visualizing problems, backlog, and future improvements, individuals can be more productive by working on these problems individually. According to Atlassian themselves:

Kanban is the Japanese word for visual signal. If you work in services or technology, your work is often times invisible and intangible. A kanban board helps make your work visible so you can show it to others and keep everyone on the same page. [10]

Jira improves on this very simple concept by adding advanced filtering and visualization (see Figure 3), so anyone can find answers to important questions:

- How much of the work that's assigned to me is left, and how many hours of work does that represent?
- How many tasks are more than a month old?
- How many "high-priority" bugs are there?
- How many to-do issues are blocked by a specific issue?

Visualizing tasks that need to be finished for the entire team in a clear and concise way, and show the progress of those tasks is central to Agile. It helps

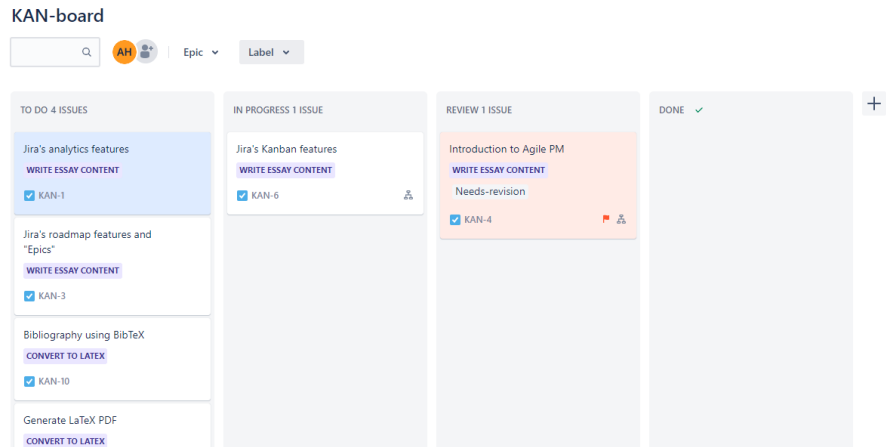


Figure 3: Jira Board using “Kanban template” configuration

the team plan, work towards the common goal and collaborate more effectively. Although, this system breaks down if you have too many tasks that need to be completed, or even worse, too many tasks that never get completed.

3.2.1 Limiting Work-In-Progress

By creating constraints on the number of tasks being worked on at a time, you will reduce distractions that pull individual team members in every direction and increase the throughput of the entire team. Though limiting work-in-progress and clutter on issue tracker is important for short-term productivity, managers still need some way to plan features and releases in the long-term.

3.2.2 Iterative development and sprints

By using a Scrum board template for your project instead of Kanban, you can group issues in “sprints”. A sprint represents a cycle in the “Design”, “Develop”, “Test”, “Review” cycle. Issues included in a sprint will be a part of that sprint’s release candidate when finished, and the goal of the team is to complete all issues in a sprint. Issues that are not completed will be put into a backlog of issues that will be prioritized for the next sprint.

By structuring development into sprints, teams can release new product versions more reliably. There are plenty of tools that managers can use to plan sprints and track the progress of the teams. A burndown chart similar to the one in the one in Figure 4 is a typical way for managers to track the progress of a sprint.

Early adopters of Agile principles and tactics like Kanban and sprint planning were small teams and startups. More and more today, large companies

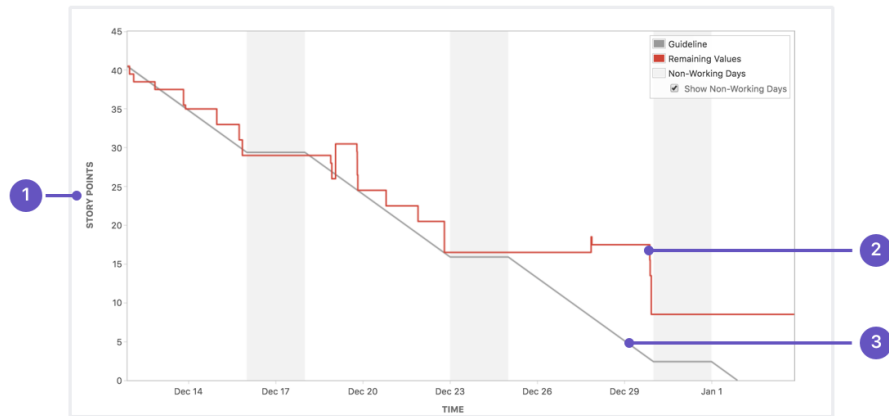


Figure 4: Burndown chart. Source: Atlassian [1]

are throwing away waterfall development practices in favor of agile principles to achieve the same velocity.

3.3 Roadmaps

Large companies have their own challenges unique to large enterprises.

- How can you make long-term time estimates when you have to respond to changes and new demands from the stakeholders?
- What tools can you use to make sure that your teams are neither under-utilizing or exhausting your teams?

Jira's Roadmap features have been designed to solve these types of problems that company executives and project managers face. Roadmaps bring a set of features that allow managers to formulate long-term plans that show what the team is working towards and explain why it is important. According to Atlassian:

A product roadmap is essential to communicating how short-term efforts match long-term business goals. Understanding the role of a roadmap—and how to create a great one—is key for keeping everyone on your team headed in the same direction. [6]

You can have different roadmaps for different stakeholders:

- Executives need a roadmap that places emphasis on how development goals support high-level company goals.
- Developers need a roadmap to organize their work in sprints and keep the team on track for success.

- Sales personnel need a roadmap to support sales conversations about new potential features and to assist in customer conversion.
- PR needs it to market new features to the public and to generate hype.

Roadmaps are not just a specification of features though. Knowing the overall goals of an epic, and the amount of sprints you have to achieve that goal helps teams know both if they can finish their planned work on time, and how many times they can afford to make new design decisions before they reach the end of their runway.

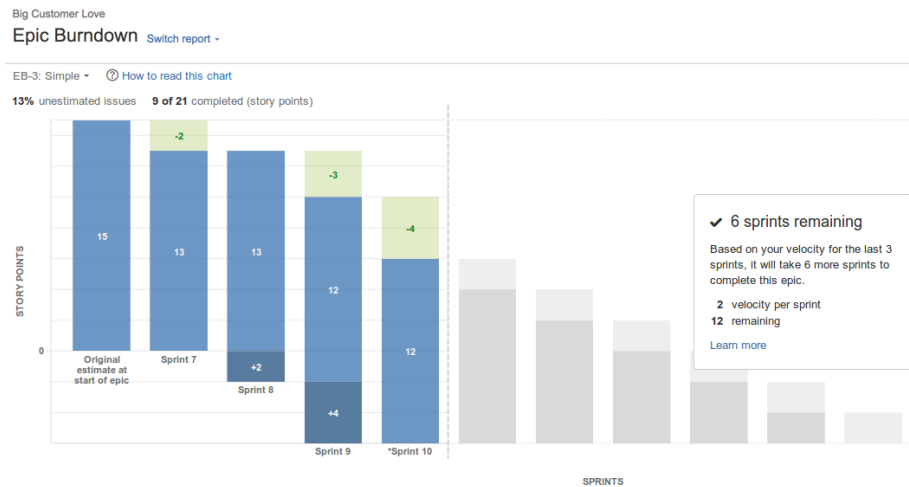


Figure 5: Roadmap burndown chart. Source: Atlassian [7]

Having enough runway to make changes to the product is not optional, it's essential to making sure that the team can adapt to new knowledge. It's important to manage the teams collective knowledge responsibly. That is why the Alassian ecosystem has developed tools to manage knowledge in an organization, and integrated those tools into the Jira platform.

3.4 Knowledge management with Confluence

Confluence is a standalone service created as a complement to the Agile project management features in Jira. Confluence is designed to be the teams central knowledge-hub, and is designed to host knowledge documents like:

- Product requirement documentation - What are the formal requirements of the product and the regulations that needs to be followed.
- Retrospectives and analysis - What did the team do well and what did the team do poorly during a sprint.

- Decision documentation - What decision was approved, and what data was that decision based on.

Confluence is designed to store knowledge from all parts of the organization and restrict access to critical business documents within your organization wherever necessary. The collective knowledge base is one of the most valuable resources a technology company has, and organizing knowledge well improves the effectiveness, longevity and overall value of an entire organization.

4 Conclusion

This suite of products deliver a powerful toolset for managers striving to implement the agile workflow, so teams can quickly learn, change and grow over time, while still meeting its obligations and long-term goals.

- Boards illustrate what work needs to be completed in a simple, concise and intuitive way to increase productivity and collaboration, and reduce downtime between tasks.
- Scrum and Sprints help developers group these tasks into larger units of work, manage backlog of work, and prioritize what is important right now.
- Roadmaps helps executives make long-term goals for the product, and helps the team understand how their work contributes towards that shared goal. It helps teams stay on track to deliver on major initiatives and helps teams prioritize what issues are important for achieving this initiative, and what is not.
- Knowledge management in Confluence helps teams manage knowledge that is learnt during this entire process. Agile development is all about feedback and learning, and maximizing shared knowledge is a good step towards maximizing success.

All these features are making agile development more manageable to work with. This makes project management tools like Jira very useful when it comes to managing project of a larger scale and we believe that it is worth spending the time to set up a tool like Jira to help manage your project and make the project work more efficiently.

References

- [1] *Burndown chart*. <https://www.atlassian.com/agile/tutorials/burndown-charts>. Accessed: 2021-04-25.
- [2] Tore Dybå, Torgeir Dingsøy, and Nils Brede Moe. “Agile Project Management”. In: *Software Project Management in a Changing World*. Ed. by Günther Ruhe and Claes Wohlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 277–300. ISBN: 978-3-642-55035-5. DOI: 10.1007/978-3-642-55035-5_11. URL: https://doi.org/10.1007/978-3-642-55035-5_11.
- [3] Daniel J. Fernandez and John D. Fernandez. “Agile Project Management—Agilism versus Traditional Approaches”. In: *Journal of Computer Information Systems* 49.2 (2008), pp. 10–17. DOI: 10.1080/08874417.2009.11646044. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/08874417.2009.11646044>. URL: <https://www.tandfonline.com/doi/abs/10.1080/08874417.2009.11646044>.
- [4] Raphael Kaim, Ralf-Christian Härting, and Christopher Reichstein. “Benefits of Agile Project Management in an Environment of Increasing Complexity—A Transaction Cost Analysis”. In: *Intelligent Decision Technologies 2019*. Ed. by Ireneusz Czarnowski, Robert J. Howlett, and Lakhmi C. Jain. Singapore: Springer Singapore, 2019, pp. 195–204. ISBN: 978-981-13-8303-8.
- [5] Howard Lei et al. “A statistical analysis of the effects of Scrum and Kanban on software development projects”. In: *Robotics and Computer-Integrated Manufacturing* 43 (2017). Special Issue: Extended Papers Selected from FAIM 2014, pp. 59–67. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2015.12.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0736584515301599>.
- [6] *Product Roadmaps*. <https://www.atlassian.com/agile/product-management/product-roadmaps>. Accessed: 2021-04-25.
- [7] *Roadmap burndown chart*. <https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-epic-burndown-report>. Accessed: 2021-04-25.
- [8] Hasliza Md Sarkan, Tengku Puteri Suhilah Ahmad, and Azuraini Abu Bakar. “Using JIRA and Redmine in requirement development for agile methodology”. In: *2011 Malaysian Conference in Software Engineering*. 2011, pp. 408–413. DOI: 10.1109/MySEC.2011.6140707.
- [9] *Waterfall model*. https://en.wikipedia.org/wiki/Waterfall_model. Accessed: 2021-04-25.
- [10] *What is a kanban board?* <https://www.atlassian.com/agile/kanban/boards>. Accessed: 2021-04-25.
- [11] *What is Jira used for?* <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for>. Accessed: 2021-04-25.