Research Report

Agile Project Development

1 INDEX

2 SETUP

2.1 GOAL

An important term that gets thrown around often in the software engineering space is agile.

An agile software project consists of smaller iterative sprints, which, compared to the waterfall methodology, drastically improves a project’s flexibility.

It is undeniable that currently it is crucial to know how to manage a project in an agile fashion, so this research is focused on learning the ins-and-outs of agile software development.

2.2 RESEARCH QUESTIONS

2.2.1 Main question

How, from commissioning to delivery, is a software project managed in an agile manner?

2.2.2 Sub questions

- What defines working agile?
- What are the significant frameworks for agile project management?
- What are the significant differences between agile and traditional project management?
- What are significant common pitfalls when applying agile project management?

2.3 APPROACH

2.3.1 Literature study

Because of the short timeframe and the rich source of information available online, only a literature study will be applied for this research.

3 RESULTS

Agile is characterized by quickness, lightness, and ease of movement; nimble.

3.1 THE AGILE MANIFESTO

Agile software development was popularized by the agile manifesto, which consists of four main values and twelve principles.

These definitions are derived from the agile manifesto website:

3.1.1 Values

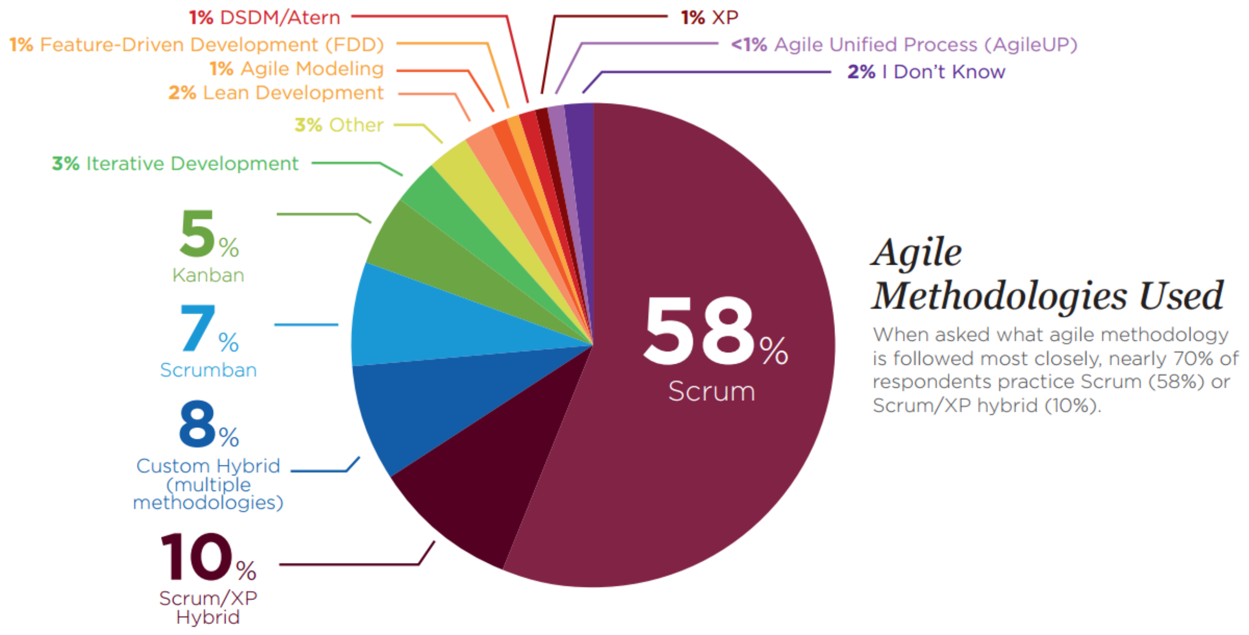
1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

3.1.2 Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Businesspeople and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity –the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly.

3.2 IMPLEMENTATIONS

While agile is not a framework itself, there are a lot of different framework which implement working agile. Here the two most popular frameworks, scrum and kanban, are defined.



3.2.1 Scrum

Scrum is, by far, the most popular agile framework. Scrum chains multiple short development iterations, called sprints, which end with a delivery.

Roles

The framework consists of three roles:

Product owner

The primary contact point of behalf of the customer.

- Defines user stories.
- Creates the product backlog.

Scrum master

Enables the Scrum team to improve its agile practices.

Scrum Team

Manages and develops the project.

Ceremonies

Daily standups

A daily meeting for the entire scrum team at the start of the day, maximum of 15 minutes, to discuss:

1. What was done since the last meeting?
2. Any problems that occurred.
3. What is going to be done for next meeting?

Sprint planning

A meeting at the start of the sprint where the sprint backlog will be defined by the team and the product owner. Main questions to answer in the meeting is what the backlog consists of this sprint and how the chosen work will get done.

Sprint review

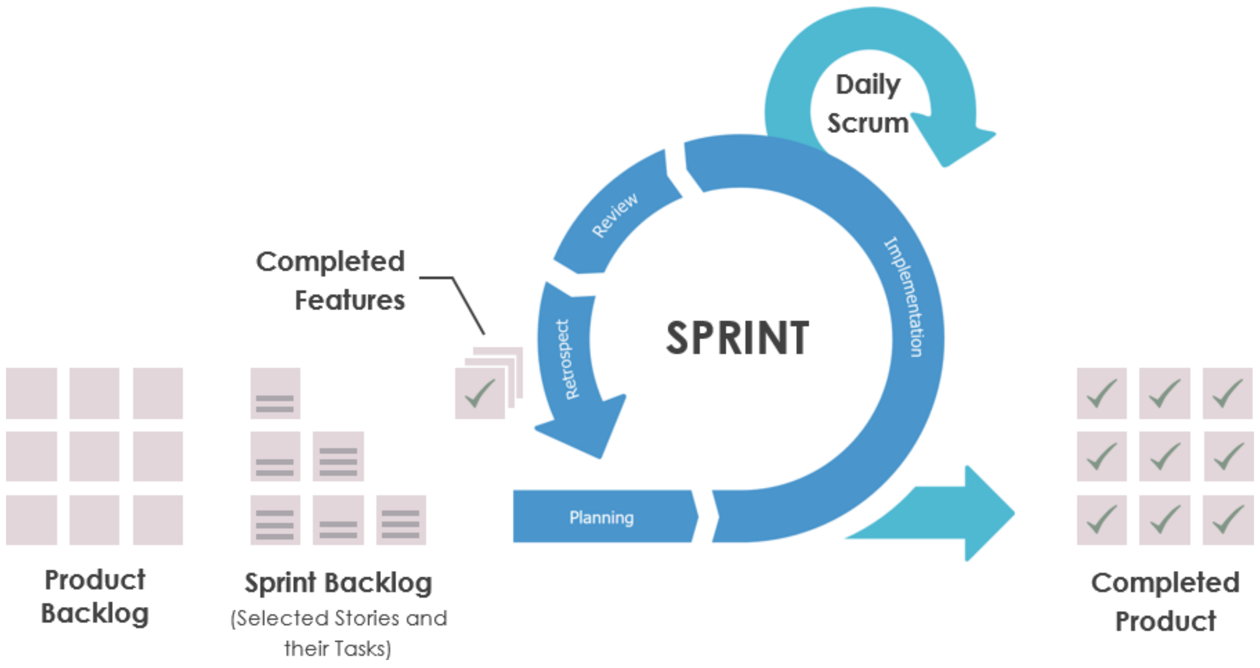
Meeting at the end of the sprint where the scrum team presents the result of the sprint to the stakeholders and reflects on the sprint and what to do next.

1. Explain what has been done and not done.
2. Discuss what went well, what problems occurred, and how these problems were resolved.
3. Demonstrate the work that was done and answer any questions.
4. Discuss the state of the product backlog and project likely target and delivery dates.
5. Review any potential necessary changes to the product.
6. Review the timeline, budget, and capabilities for the next release.

Retrospective

Meeting at the end of the print with the scrum team to review the previous sprint. Here you can discuss:

1. What went well in the sprint?
2. What could be improved?
3. What will be done to improve the next sprint?



Artifacts

Burndown charts

A chart where the project's progress is tracked over the entire sprint. This can be done by using user stories or points.

Points can be added to user stories using planning poker, which determines the time a user story will take to complete. Adding points to user stories makes the planning more precise.

Product backlog
The product backlog is the entire list of user stories that comprise the project. This backlog can be changed at any time.

Sprint backlog
The sprint backlog is a smaller backlog of userstories that will be completed during a sprint. This backlog can only be changed at the beginning of the sprint.

3.2.2 Kanban

Board
When using kanban it is important that everything is visualized on the project board.

Backlog	**Selected (2)**	**Designing (2)**		**Implementing (2)**		**Testing (2)**		**Approving**	**Done**
		Doing	Done	Doing	Done	Doing	Done		

During the entire project, the board is never cleaned. Unlike Scrum, Kanban is a continuous process.

WIP-limits
Work in Progress limits, shortened to WIP-limits, are limits that are set on tables of the board. In the example, there cannot be any more than two tasks at the time in the selected, designing, implementing, and testing tables.

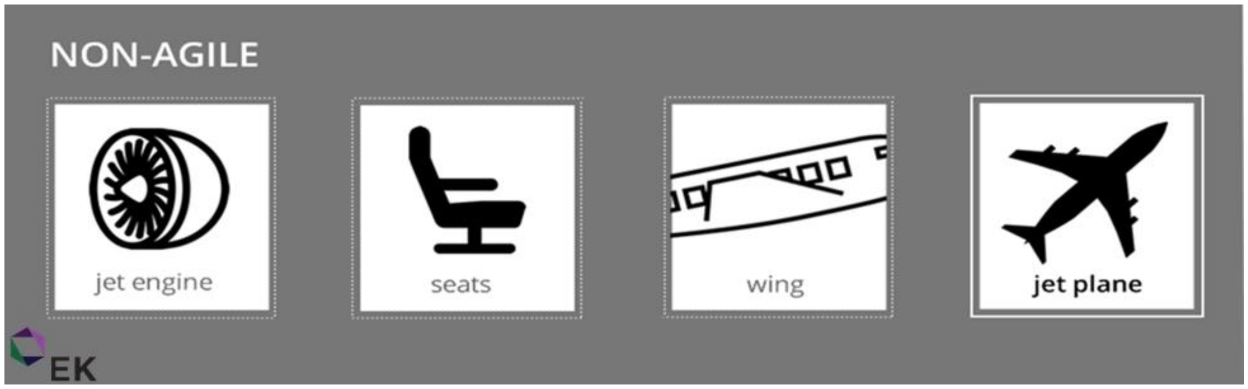
Using limits ensures that any bottlenecks that occur during the project are quickly resolved, since new tasks cannot move forward on the board before the previous are solved.

Roles
In kanban there are no predefined roles. Kanban is an iterative process, so there do not have to be any set deliverables. With Kanban, you can meet with the product owner every day to update the backlog, making it extremely flexible.

3.3 TRADITIONAL PROJECT MANAGEMENT

3.3.1 Waterfall
With traditional, also known as waterfall, project management all requirements and designs are defined at the start of the project. For big project this can mean months of planning before the development

begins. Then, at the end of the entire project, the customer is presented with the product for the first time.

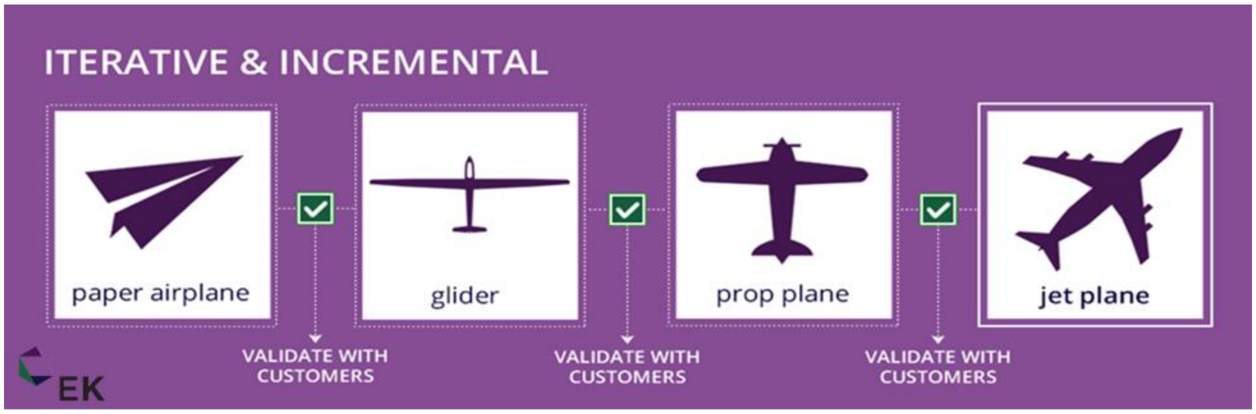


Because of this, it is not possible to change anything about the project during development. The application gets created part by part that do not have any value to the customer by itself.

Another problem with this is that, since projects can take years, at the end of the project the customer's needs or existing technologies may have changed.

3.3.2 Agile

With agile project management the customer is continuously delivered with valuable software. This also allows for constant feedback and input into the project.



3.4 Pitfalls

While agile project management is extremely popular and has many benefits, there are significant pitfalls that must be avoided.

3.4.1 Agile waterfall

Today, agile is almost as much of a buzzword as blockchain. It gets thrown around a lot, but often the switch from waterfall is not correctly made.

Working agile does not mean waterfall project management with iterative deliveries. Instead of continually delivering small components that have no value on their own, you deliver smaller working products.

3.4.2 Points

To be able to properly plan out the spring backlog, it is important to assign each user story points.

When using points, you get a good estimate how long the task will take, and thus an accurate estimate of what can be achieved in a sprint.

Additionally, knowing beforehand how long a task is going to take makes it easier to detect problems that are occurring during the daily standups, and motivates the team members to finish their tasks on time.

3.4.3 Definition of done

Since the project is often delivered and continuously changing, documentation quickly falls by the wayside. Still, it is important to have a clear picture of what a task consists of and exactly when it is done. This makes sure that all components fit together properly, and the product satisfies the product owner's requirements.

3.4.4 Contact

Contact with the product owner is crucial. Whenever changes are made, something goes wrong, or maybe there is even time left in the sprint, it is crucial to keep the product owner up to date.

4 Conclusion

A few big problems with tradition, waterfall, project management are solved using agile project management.

Agile project management makes sure valuable products are continuously delivered, problems are detected and solved early, and needed changes can be implemented quickly.

Agile is not a methodology, but it is implemented by frameworks like Scrum and Kanban.

Often these frameworks can be combined.

5 References

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved from agile manifesto: https://agilemanifesto.org/

GSA. (n.d.). *Agile FAQs*. Retrieved from Tech at GSA: https://tech.gsa.gov/guides/Agile_FAQs/

scrum.org. (n.d.). *Burndown Charts Defined*. Retrieved from scrum.org: https://www.scrum.org/resources/burndown-charts-defined

scrum.org. (n.d.). *What is a Daily Scrum?* Retrieved from scrum.org: https://www.scrum.org/resources/what-is-a-daily-scrum

scrum.org. (n.d.). *What is a Scrum Master?* Retrieved from scrum.org: https://www.scrum.org/resources/what-is-a-scrum-master

scrum.org. (n.d.). *What is a Sprint Retrospective?* Retrieved from scrum.org: https://www.scrum.org/resources/what-is-a-sprint-retrospective

scrum.org. (n.d.). *What is a Sprint Review?* Retrieved from scrum.org: https://www.scrum.org/resources/what-is-a-sprint-review

scrum.org. (n.d.). *What is Sprint Planning?* Retrieved from scrum.org: https://www.scrum.org/resources/what-is-sprint-planning

Singh, M. (2018, March 2). *Kanban = Continuous Delivery? Not Necessarlily*. Retrieved from digite: https://www.digite.com/blog/kanban-continuous-delivery/

van der Wardt, R. (n.d.). *Wat is Kanban? Een volledige uitleg van de Kanban Methode + template*. Retrieved from Agile Scrum Group: https://agilescrumgroup.nl/wat-is-kanban-methode/