

Agile vs Waterfall:

which project management methodology is better for software development?



The software development process starts long before the first line of code is written. One of the tasks to solve before coding is selecting the project management methodology that would effectively work for this particular project.

To make the right choice, a lot of things have to be considered: from the project size and technical requirements to the industry's specifications. In this whitepaper, we compare the two most famous and contrasting management methodologies – Waterfall and Agile – to help you understand which one will better suit your product development.

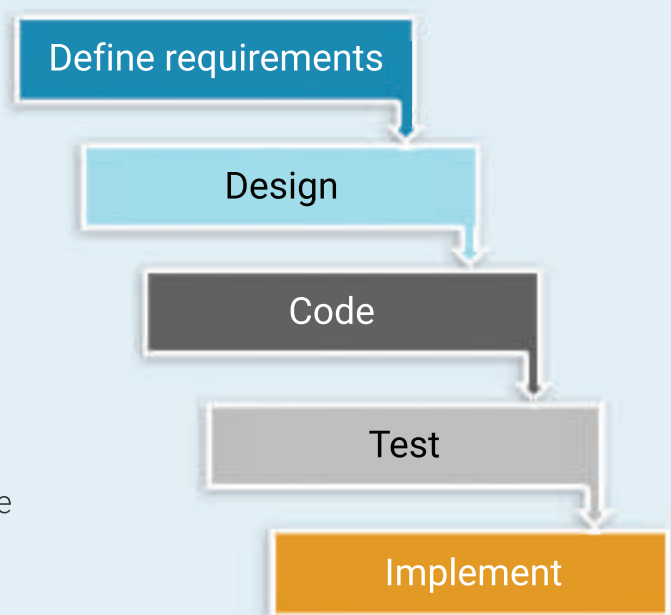
It seems there are no people left nowadays who haven't heard about Agile and Waterfall methodologies. And more and more discussions arise daily on these project management methodologies, their pros and cons and cases where it is better to use one or the other approach. And, of course, each methodology has its loyal followers who insist on using their project management scheme. However not everything is so obvious as it might seem and not for every project each of these methods can be applied. The implementation of any methodology depends on many factors such as project type, its size and number of people involved, etc.

So the choice between these two most prominent methodologies may not be easy. In this article, we'll give you a simple and unbiased overview of Agile and Waterfall, and help you to decide which one suits you best. But let's find out what is Agile and Waterfall first.

Waterfall: Brief Description

Waterfall methodology is also called traditional, because of its origin (it was borrowed by software development companies from other industries). Obviously, its name indicates its workflow: each phase of SDLC (software development life cycle) takes place in sequence: conception, initiation, analysis, design, construction, testing, implementation, and maintenance. Basically there's a review of requirements at the end of each stage, but the main idea is in-must-complete one phase before going to another, it means you can only move downstream and you can't go back to the previous phase without completing a full development cycle.

Thus, your team needs to be very careful and precise on planning and implementation stages and should have a number of processes for managing workflow, exceptions, reporting, risks and issues. There's no place and time for many changes or bug fixes.



Key PROs of the Waterfall methodology

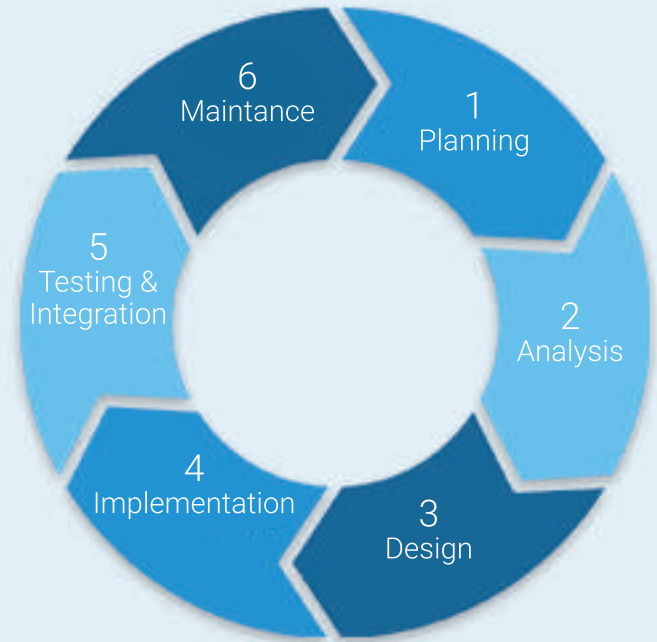
1. It's a good choice for small and mid size projects, as waterfall allows to turn the development process into a conveyor.
2. Waterfall helps to deliver the project faster when the planning is precise and detailed, and the project size, cost and deadlines are clearly stated.
3. If a client has a clear understanding of how the final product should look like and has an SRS (Software Requirements Specification), the released project will completely meet the customer's expectations.
4. In a waterfall approach all processes and results are strongly documented. It helps to avoid some negative impacts in case of employee replacement or absence. Also, the detailed documentation allows improving the development process in the future or making support and maintenance more efficient.
5. Generally a waterfall project is easier to manage as it has clear predefined requirements and milestones. It's simple and predictable.

Key CONs of the Waterfall methodology

1. Waterfall methodology is not a good choice for large size projects as it is almost impossible to foresee all possible challenges requested by the customer during the development process.
2. A waterfall project presupposes significant customer's involvement. If there are no clear, detailed and comprehensive requirements, it would significantly affect the final result that might not meet client's expectations.
3. In sequential development model, the testing phase begins only after full implementation cycle is over. Thus, it might be difficult to identify when the errors were made and as a result it will be more expensive to fix them. Additionally, your team might be tempted to sacrifice thorough tests and reviews in favor of being in compliance with schedule at the early stages.
4. Even a small error at the beginning of the project implementation may terribly affect the whole result and how other developers write their code. Waterfall methodology does not allow making a small step back and immediately fix detected bugs.
5. Waterfall method does not presuppose customer's participation in the development process, it is fully outsourced. So the end product can be less customer oriented, neglecting evolving market needs.

Agile: Brief Description

Agile methodology appeared as a response to the waterfall methodology drawbacks. Unlike Waterfall sequential processes, Agile method provides iterative ongoing approach to software development. Instead of downstreaming phases Agile divides implementation process into 'sprints'. Each sprint starts with planning, team members create a list of tasks, define their roles, and sprint duration time, and finally tests and review. If not all planned work is complete in time, team tasks are re-prioritized and then used for future sprint planning.



Agile is much more team-based and gives developers more freedom. With the Agile method there is no predefined course of action or complete development process plan.

Also, such an approach allows customers to introduce changes at any time of the development process. Through end-of-sprint demos customer can participate in the testing process, evaluate results, and provide feedback to the project team. Agile methodology significantly relies on a high level of customer involvement into the project development process.

Key PROs of the Agile methodology

1. Projects on Agile methodology don't require strict SRS, and basically are more friendly to non-technical customers. A brief description of the final product is enough for the development team to start working.

2. If you are running an Agile project, it means that its requirements are certainly expected to change and evolve, that makes the final product more up-to-date with market trends, users' needs and latest technologies. It is extremely useful in the highly competitive market: as new features can be easily introduced into the product that will favorably allocate it against competitors.

3. With Agile method fewer unforeseen errors arise at the deployment stage in comparison to traditional approaches. Continuous (repeatable) iterative QA-tests at the end of each sprint assure that all bugs are caught, resolved and won't appear again. Agile actually reports fewer risks as compared to traditional approaches due to shorter sprint cycles, ability to quickly implement the ad hoc requirements and quickly adapt to the customer needs.

4. In some cases, you may launch your product earlier than expected. The shorter the sprints, the faster the product reaches the end user.

5. Agile method can significantly improve product's quality and provide better results. In addition to continuous integration, testing, peer reviews and automation, direct communication with the customer usually makes Agile development teams better motivated and self-organized.

6. Agile approach is highly transparent and flexible both for developers and customers. Collaborative work and tracking each other's progress helps everyone to stay focused. And with the use of project management and various communication tools, it becomes simpler day by day.

Key CONs of the Agile methodology

1. Agile projects are not so easy to manage as they may seem. Without experienced project manager the whole project can become amorphous set of code sprints. The project may be released later than expected and may go over budget.

2. While carrying out all client's whims, development team can exceed the time-frames and provide the results that will not satisfy the customer needs. As the initial project doesn't have strict requirements, the final product can become completely different and may not correspond to the primary objectives.

3. The very high degree of customer involvement, which is basically good for the project, may also pose a problem if the customer does not have time or interest for such kind of participation in the development process. His role is extremely important in Agile.

4. One person - one role, such policy leads to higher employee turnover risks. By putting emphasis on the working process, there can be an illusion that documentation can sometimes be neglected. In case of employee disease, dismissal or any other reason of absence, it's more difficult to transfer tasks and responsibilities to another person.

5. The complex nature of Agile methodology sometimes focuses more on the process rather than the final goals.

When to use Waterfall and Agile?

Answering what is the difference between Agile and Waterfall Model? – we should say that these two software development methodologies are completely different and each one is good in its respective way. However, we can highlight the following differences that will help you to decide when to use Waterfall and Agile:

1. You have clearly defined requirements and there are no changes expected? In this case Waterfall model is ideal for your project. In respect of the remaining, if there is no clear picture of how the final product should look like, Agile will help you to overcome frequent requirement changes.

2. The Waterfall as sequential and rigid method is easy to manage, it doesn't require additional managing skills and easy to apply. Better used if you don't have an ability to hire a skilled project manager.

3. When you want to be involved in the development process and is ready to give feedback regularly – transparent and flexible Agile methods will satisfy your needs. If you prefer to entrust that work to the development team and stay away until the final work is done use Waterfall.

4. If your team consists of skilled and responsible developers who are able to adapt and think independently, follow Agile model and leave Waterfall for projects with clear requirements and instructions.

5. You develop solution for industry with rapidly changing standards? Agile will allow you to introduce the latest technologies even at the final phase of the development process. When definition, not speed, is the key to the success of your project use Waterfall.

ALIGNING PROJECT TRAITS with DEVELOPMENT METHODOLOGIES			
Project Trait/Factor	Agile	Plan-Driven (Waterfall)	Comments
Customer Availability	Prefers customer available throughout project	Requires customer involvement only at milestones	Customer involvement reduces risk in either model
Scope/Features	Welcomes changes, but changes come at the expensive of Cost, Schedule, or other Features. Works well when scope is not known in advance	Works well when scope is known in advance, or when contract terms limit changes	Change is a reality so we should prefer adaptability where possible. Contract terms sometimes restrict it
Feature Prioritization	Prioritization by value ensures the most valuable features are implemented first, thus reducing risk of having an unusable product once funding runs out. Funding efficiency is maximized. Decreases risk of complete failure by allowing "partial" success	"Do everything we agreed on" approach ensures the customer gets everything they asked for; "all or nothing" approach increases risk of failure	Contract terms may not permit partial success and may require "do everything"
Team	Prefers smaller, dedicated teams with a high degree of coordination and synchronization	Team coordination/synchronization is limited to handoff points	Teams that work together work better, but when contracts are issued to different vendors for different aspects of the project, high degrees of synchronization may not work
Funding	Works extremely well with Time & Materials or other non-fixed funding, may increase stress in fixed-price scenarios	Reduces risk in Firm Fixed Price contracts by getting agreement up-front	Fixed price is tough when scope is not known in advance, but many government contracts require it
Summary	Agile is better, where it is feasible	Plan-Driven may reduce risk in the face of certain constraints in a contract between a vendor and external customer such as the government	Through educating our customers about the strengths and weaknesses of each model, we hope to steer them towards a more Agile approach. This may require changes to how our customers, particularly the government, approach software development projects

Conclusion

The above-mentioned factors are a how to guidelines, however the final choice on the use of Agile and Waterfall project management methodologies depends on individual project requirements.

About SCAND

SCAND Ltd. is an Eastern European company, located in Minsk, Belarus. The company is focused on custom software development outsourcing across multiple technologies including .NET, Java, PHP, JavaScript, and C++. With more than 19+ years in IT and 600 successful projects for dozens of customers globally, and over 250 highly experienced engineers on board, we are rated as one of the top software outsourcing companies in Eastern Europe.

Have questions? Feel free to contact us: info@scand.com
