Agile Methodology

Deliver products faster,
in sync with market trends
and user requirements.

serengeti
software tech

Abstract

Agile project management is a methodology that has been widely accepted by many organizations around the world. What makes it so successful and dominant, what are its benefits, and are there any concerns you should be aware of? How does it work, is it hard to implement it in your company and how does it differ from traditional methods?

Those are just some of the questions that we'll cover in this white paper.

Agility

If you google the meaning of the word Agile, along with its definition, you'll end up with a picture of a cheetah – the fastest land animal. Of all big cats, cheetahs are hunters with the highest hunting success rate. Most of us would attribute this success rate to its speed, but it turns out that the key to its success lies largely in its ability to make quick adjustments during the hunt. They can adjust their trajectory in real-time based on the target's speed, direction, and obstacles they encounter by making quick movements, stops, and sharp turns.

Now, we may all agree that there aren't many similarities between a cheetah's hunt and project management, but just like the cheetah is focused on its target, the project manager is focused on his – delivering the product scope with given time and resources. And just like a cheetah's hunting success is based on its ability to respond and adapt to changes, the project is successful if the project manager can do the same.

It is no surprise then that there is a project management methodology based on the trait, that makes those who possess it, so successful in the natural world – agility.

Taking a New Approach

As we already said, to be Agile in project management is to be able to adapt to changes. In the context of project management, those changes refer to changes in project scope. Project scope is something that shouldn't change during the project's life span and even if it does, those changes should be minimal. And although this is the case in some projects, it often isn't in others, and if your project falls under that category, it's highly likely you'll need to use the Agile project management approach.

Projects which have requirements that are very well known, a stable product definition, a technology that is well understood, and a rather short lifetime aren't usually candidates for Agile software development. In such cases, you're probably better off sticking to a conventional model known as Waterfall, where you'll base on the detailed specification and with minimum customer intervention, deliver the solution once all features have been implemented.

Taking a New Approach

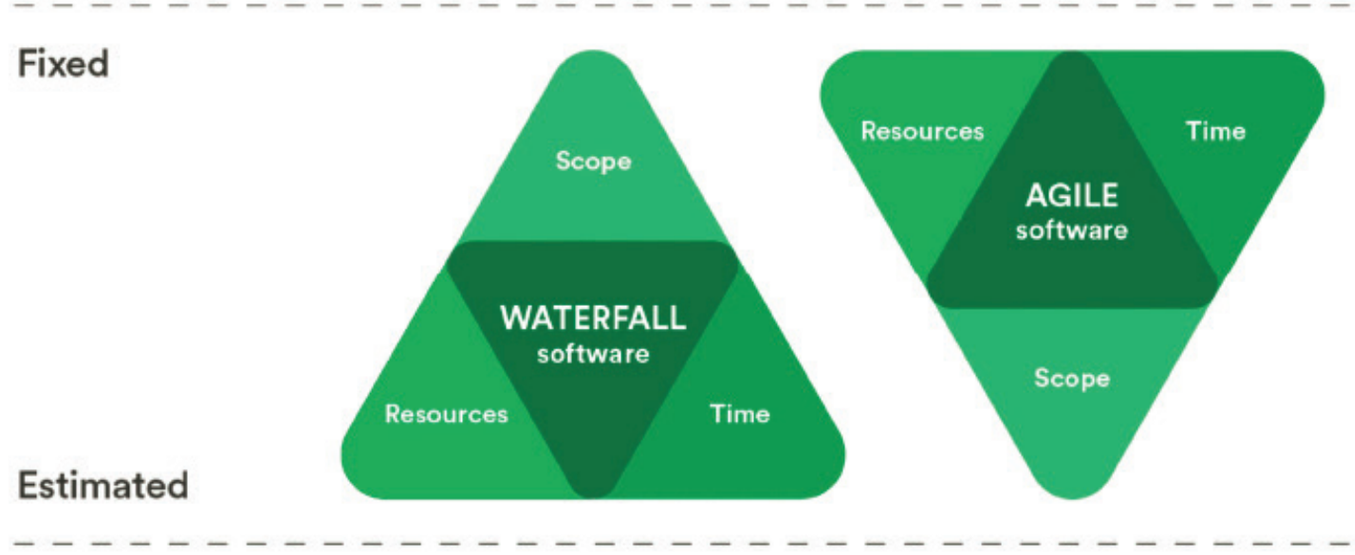
Unlike other scopes, software development project scopes usually experience a lot of changes during their life span. Clients have trouble defining and communicating what they want and often have a hard time visualizing a solution before they see it. In other words, you'll get the feedback once you provide them with the final solution and that is, as you may already have concluded, something you want to avoid.

In the Agile approach, you provide the client with chunks of application functionalities so they can get the picture of what the application will eventually look like. The client can then provide you with feedback, giving you the ability to change the things you did wrong and steer the project in the right direction. It is much easier to make changes while your application is still under development, and when you're getting feedback every few weeks, it's easier to know what clients want.

Taking a New Approach

By doing this, we have turned the project management triangle upside down and changed the project scope from something that has been fixed and predefined at the beginning of the project into something that's being defined during the project life span.

Features are developed in short iterations that usually last from two to four weeks. Each iteration is like a small independent project that goes through all project phases: specification, estimation, development, testing, and deployment.

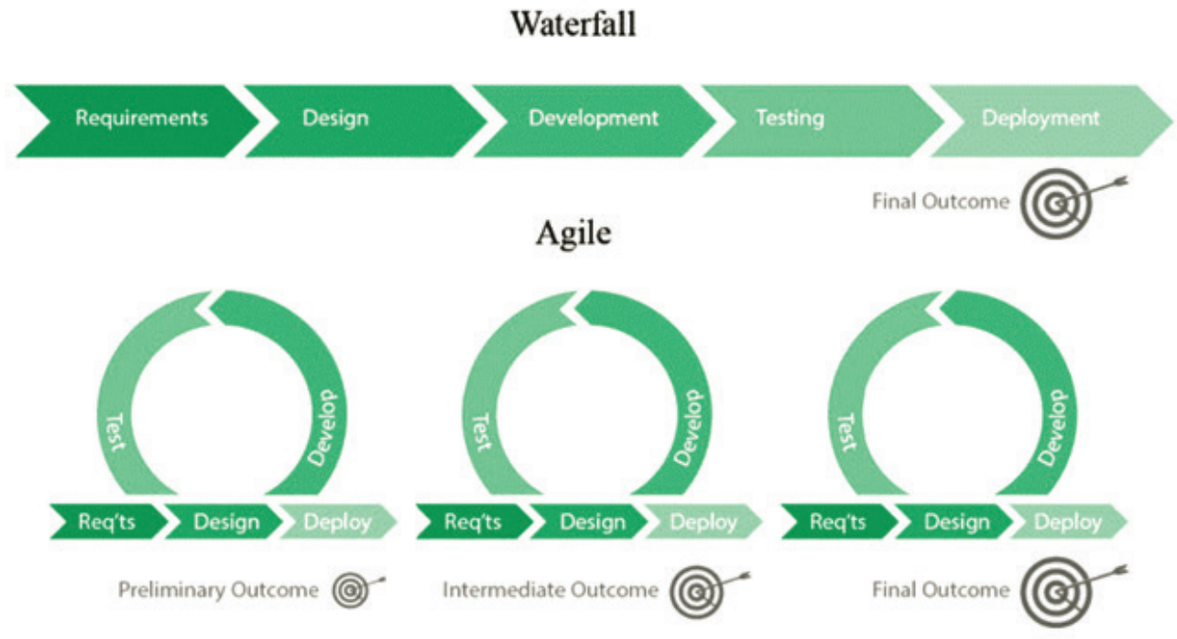


Taking a New Approach

The client, being included in the specification phase of each iteration, is now an active participant in the project and a part of the team. During each iteration, the focus is both on the product and the customer. The feedback the customer provides at the end of each iteration can be used by the team to adjust their work and fix potential deficiencies.

Some of the benefits that arise from the client being included in Agile development are:

- reduced risk due to the client's feedback on delivered iterations
- reduced uncertainty due to focus on both prod uct and customer
- the client can get a look at the product much faster
- fewer errors and better decision making due to better communication



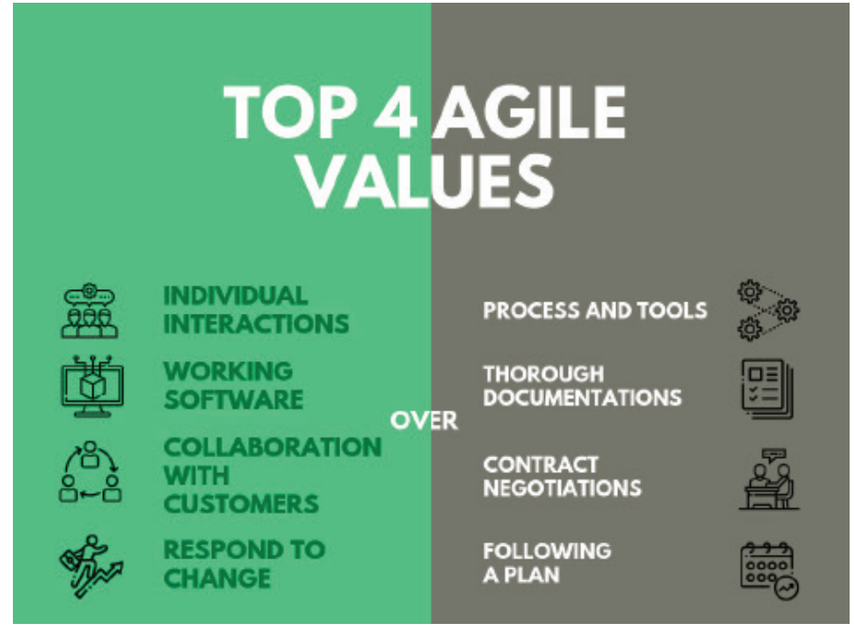
Agile Values and Principles -The Agile Manifesto

Agile as a methodology for software development was significantly popularized after the creation of the Manifesto. The Agile Manifesto was created in order to solve the problem of companies that focused on excessive planning and documentation. One of the negative consequences of that kind of software development is that they often lost sight of what was truly important – their customers.

During the 1990s, a number of lightweight software development methods evolved, such as rapid application development (RAD), dynamic systems development method, extreme programming, feature-driven development, etc. The Agile Manifesto was created as a result of the combined experience of 17 software developers (source: https://agilemanifesto.org/), who met to discuss these lightweight software development methods.

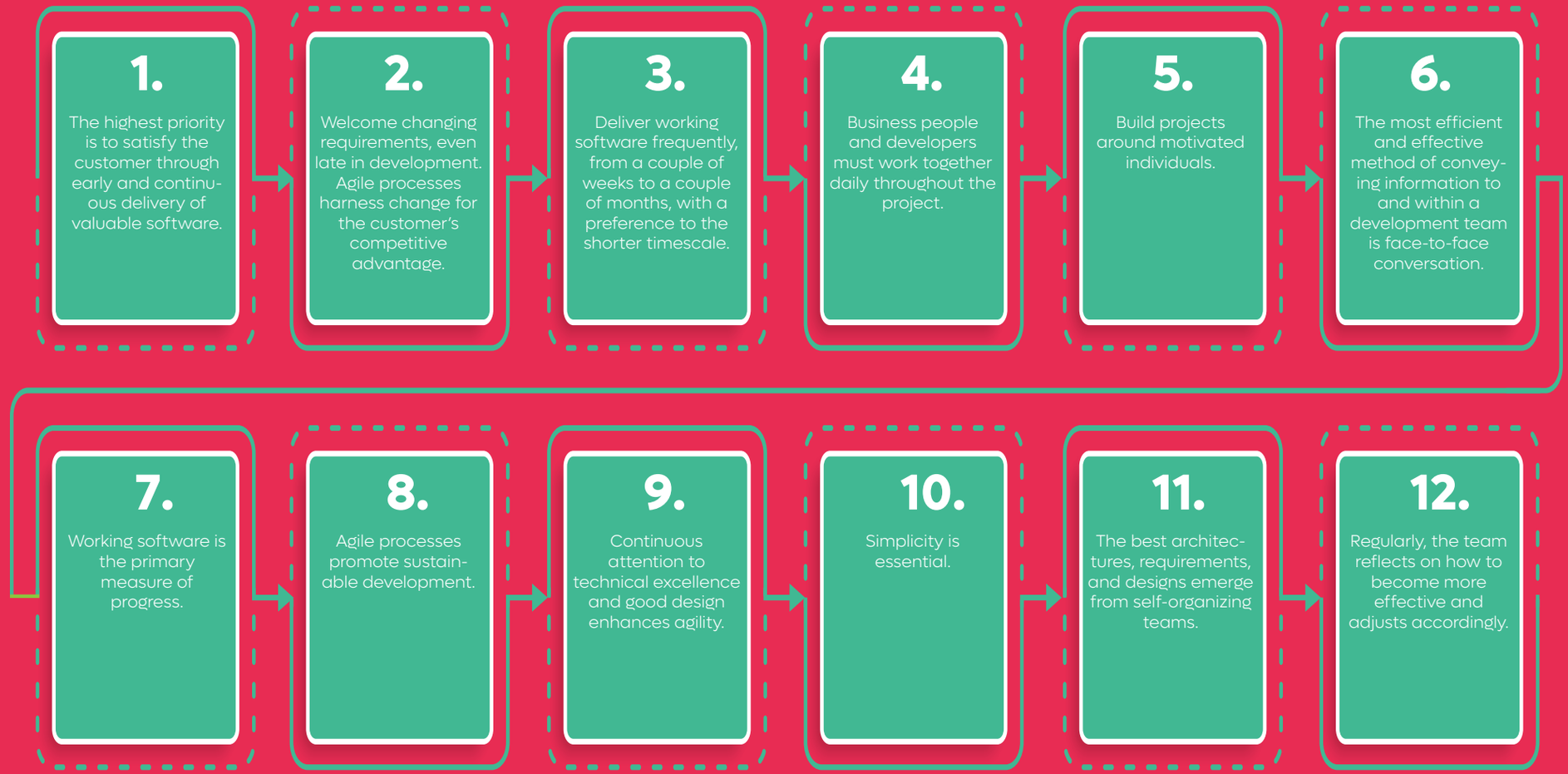
As a result, they created 4 Agile software development values and 12 principles.

In every project, processes and tools are important, but having team members working together is more important. Additionally, documentation is also useful, but the main point should be to create working software, not documents. Having a project plan is important, but the plan must be flexible enough to accommodate additional changes.



Drivers for implementing DevOps

The principles behind the Agile Manifesto are:



Agile Methodology - Deliver products faster, in sync with market trends and user requirements.

serengeti
software tech

Agile Frameworks

The Agile framework can be defined as a specific development approach based on the Agile values and principles previously mentioned. Frameworks are like tools you use to carry out Agile principles. There are many popular Agile frameworks, so we will offer a short overview of some commonly used frameworks.

Scrum

Scrum is probably the most popular framework. Its focus is on team-led projects. It utilizes defined team roles to implement the responsive style of Agile project management.

Kanban

Its focus is on the workflow aspect of a project. This framework usually involves the use of a Kanban board or flowchart.

XP

The focus of Extreme Programming (XP) is on frequent releases. This framework is ideal for projects where continuous delivery is a top priority.

FDD

Feature-Driven Development also focuses on delivering value regularly. This framework is client-centric, paying attention to the client's engagement.

DSDM

Dynamic System Development Method (DSDM) aims for regular value delivery and clear communication. It concentrates on delivering the project on time and within budget.

When it comes to Agile frameworks, it can be hard to choose the best one. The good thing is, you don't have to select just one framework. It is possible to combine several frameworks to achieve the best possible result for your project.

Agile in Practice

The most popular Agile methodology is Scrum, which borrowed its name from rugby. A scrum is a method of restarting play after the game has been interrupted – the players are gathered around and packed closely. This union of players, which emphasizes teamwork, shares similarities with members of a team cooperating in software development.

Scrum is widely used by software development teams and according to some reports, 70% of software teams use Scrum or some hybrid version of Scrum. It makes information transparent so the team can adapt to current conditions, rather than predicting future conditions, and ensure that what is being delivered is what is was expected.

Agile in Practice

Scrum is composed of multiple events, also called ceremonies, that occur inside each Sprint. Below is a summary of how Scrum works along with its main events:

- Each feature that needs to be implemented is described in a story. A story is defined from the user's perspective and it describes features through three main points:
    - value – purpose
    - actor – user that will consume it
    - action – what can be done
- Story complexity is estimated by the develop ment team, which is later used to determine the number of features that can be implemented in one iteration (Sprint).
- User stories get prioritized based on the client's preferences and development preconditions (in case a feature needs to be implemented in order to enable the development of other fea tures, it will be given a higher priority to avoid blocking).

The team works on selected stories during one iteration. Those iterations are called Sprints in the Scrum methodology, and usually they last from 1 to 4 weeks. The process of selecting which stories will be included in one Sprint is called Sprint planning.

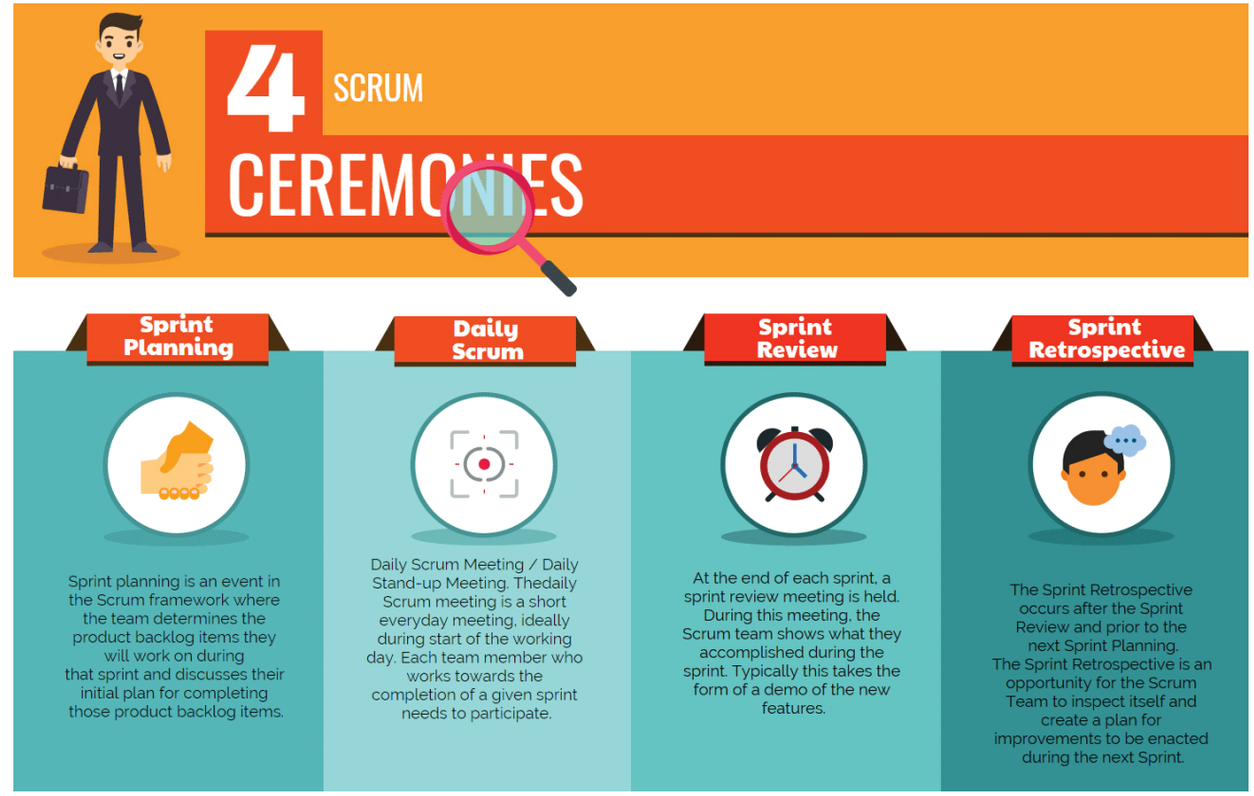
The team meets on a daily basis to share progress and information. Daily meetings (Daily Scrum) are meant to be short, lasting anywhere from 15 to 30 minutes (depending on team size). These meetings benefit both developers and management since it is easy to keep track of project/sprint. Progress is often tracked on burndown charts and Gantt charts.

Agile in Practice

At the end of each Sprint, the features are presented to the stakeholders. The team receives feedback which is later used to make any necessary changes and adjustments. They all get a sense of the rate at which the team is able to perform and plan the next steps in terms of features that need to be implemented.

Depending on the clients/stakeholders, preference release can be made from a single or from multiple Sprints.

The team has a retrospective meeting at the end of each Sprint to talk about the things that went well and those that didn't work. They discuss their efficiency and how well they estimated their capacity in terms of the stories they can implement in one Sprint. They use that information in the next Sprint planning.



Agile Project Managment

Unlike in traditional project management, Agile development doesn't have a defined baseline, there is no need for the detailed specification on the beginning of the project and requirements can be defined during the project life cycle.

Focus is shifted from knowing exactly what needs to be done to understanding what the client wants. This is a process that involves all team members, from business analysts and product owners to developers. They all need to communicate with each other and the more informed they are, the better. It is that very communication that reduces the risk in Agile development because developers are less likely to deliver a functionality that won't match the client's requirements.

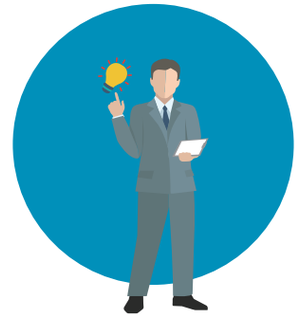
Agile Methodology - Deliver products faster, in sync with market trends and user requirements.

Agile Project Managment

In this kind of project organization, the role of the project manager is somewhat different. The project manager can oversee the project, report the status to higher management, and take on the role of coordinator between other roles on the project. But most of the project manager's traditional work is scattered between other team roles:

CLIENT

Defines what the application should do from the user perspective. Which functionalities are expected? Reviews delivered features and gives feedback if changes need to be done. Accepts or rejects a solution.

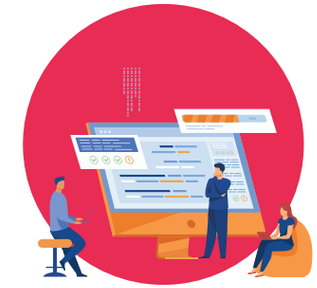


PRODUCT OWNER

Translates the client's requirements to developers. He is the link between the business and technical world. Suggests how to implement certain functionalities while considering both user experience and technical capabilities. Scrum Master – Coordinates Scrum events (ceremonies). Arranges and coordinates all Scrum activities and meetings.

DEVELOPMENT TEAM

Developers in charge of implementing features defined in User stories.



QA/TESTERS

Testers and QA are in charge of testing implemented features.
They also check on what level the delivered features match the client's requirements.

Should I Go Agile?

Although Agile brings many benefits to your project, there some concerns for both the client and the development team. Besides trying to figure out the answers to basic questions regarding team size, project complexity, and compatibility with the Agile methodology, the client if often concerned about the time he'll need to invest in projects and would prefer to be deprived of frequent meetings and continuous feature specification.

On the other hand, developers would also like to be left alone and make their own schedules. They fear that constant meetings and strict Scrum guidelines will take away their precious development time and that they'll lose their flexibility and the ability to organize themselves.

Although their concerns can be partly justified, they soon realize the benefits that the Agile methodology brings. Scrum events keep them focused all the time and neither side can escape their part of the responsibility. The client needs to provide developers with constant feedback and approve or disapprove of the delivered features, which prevents developers from deviating from the desired implementation.

The development team reports its progress and issues on a daily basis, so the client can have insight into project progress. They soon realize that working under this structure is good and that time, they initially feared will be wasted, is well invested.
Of course, there are situations when Agile isn't necessarily the right approach:

- If you need to have all the requirements defined and documented at the beginning of the project
- If the scope and requirements are not likely to change much
- If the customer won't be able to participate in the project as much as necessary
- If the development team is too large

Conclusion

When implemented the right way, Agile development simplifies development, testing, and project management. It improves communication on projects, ensures business value, due to the client's involvement throughout the project's life cycle, and reduces the risk of delivering an inadequate solution.

Higher speed, flexibility, and productivity achieved through such approaches are the key drivers that motivate more and more organizations to switch to Agile.

Fast-paced industries, like software development, rely on flexibility, and Agile methodologies allow them to deliver products in sync with market trends and user requirements.

Who are we?

An international nearshoring and offshoring software development and consulting company. With our unique Team Extension Model, we help companies with each stage of the development lifecycle, not only with development but also with consulting services. Technical consulting in different domains like Architecture Design, Automated Testing, DevOps and SCRUM was recog-nized in 300+ projects across different industries such as finance, logistics, hospitality, industrial manufacturing, healthcare, energy, and retail.

Serengeti d.o.o.
Ulica Pile I. br. 1
HR – 10000 Zagreb, Croatia
sales@serengetitech.com
serengetitech.com

serengeti
software tech