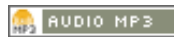


[Player](#) | [Play in Popup](#) | [Download](#)



Career Model For Agile Coaches [5:36m]:

[Hide Player](#) | [Play in Popup](#) | [Download](#)



Did Somebody Invent Agility? [2:43m]: [Hide](#)

[Player](#) | [Play in Popup](#) | [Download](#)

## Agile Methodology

### What Is Agile?

Agile methodology is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.

### What is Scrum?

Scrum is the most popular way of introducing Agility due to its simplicity and flexibility. Because of this popularity, many organizations claim to be “doing Scrum” but aren’t doing anything close to [Scrum’s actual definition](#). Scrum emphasizes empirical feedback, team self management, and striving to build properly tested product increments within short iterations. Doing Scrum as it’s actually defined [usually comes into conflict with existing habits at established non-Agile organizations](#).

Scrum has only three roles: Product Owner, Team, and [Scrum Master](#). These are described in detail by [the Scrum Training Series](#). The responsibilities of the traditional project manager role are split up among these three Scrum roles. Scrum has five meetings: [Backlog Grooming \(aka Backlog Refinement\)](#), [Sprint Planning](#), [Daily Scrum \(aka 15-minute standup\)](#), the [Sprint Review Meeting](#), and the [Sprint Retrospective Meeting](#).

Many books and [classes](#) are available from a variety of competing sources of varying accuracy and quality. One place to start would be the [Scrum Training Series](#), which uses an entertaining approach to cover the most popular way of introducing Agile to teams. You can also download the [6-page illustrated Scrum Reference Card](#).

### Where Did Agile Come From?

In 1970, Dr. Winston Royce presented a paper entitled “Managing the Development of Large Software Systems,” which criticized sequential development. He asserted that software should not be developed like an automobile on an assembly line, in which each piece is added in sequential phases. In such sequential phases, every phase of the project must be completed before the next phase can begin. Dr. Royce recommended against the phase based approach in which developers first gather all of a project’s requirements, then complete all of its architecture and design, then write all of the code, and so on. Royce specifically objected to this approach due to the lack of communication between the specialized groups that complete each phase of work.

It’s easy to see how the “waterfall” methodology is far from optimized compared to agile methodology.

First of all, it assumes that every requirement of the project can be identified before any design or coding occurs. Put another way, do you think you could tell a team of developers everything that needed to be in a piece of software before it was up and running? Or would it be easier to describe your vision to the team if you could react to functional software? Many software developers have learned the answer to that question the hard way: At the end of a project, a team might have built the software it was asked to build, but, in the time it took to create, business realities have changed so dramatically that the product is irrelevant. In that scenario, a company has spent time and money to create software that no one wants. Couldn't it have been possible to ensure the end product would still be relevant before it was actually finished?

## Why Agile?

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there's always time to steer it in another direction.

The results of this “inspect-and-adapt” approach to development greatly reduce both development costs and time to market. Because teams can develop software at the same time they're gathering requirements, the phenomenon known as “analysis paralysis” is less likely to impede a team from making progress. And because a team's work cycle is limited to two weeks, it gives stakeholders recurring opportunities to calibrate releases for success in the real world. Agile development methodology helps companies build the right product. Instead of committing to market a piece of software that hasn't even been written yet, agile empowers teams to continuously replan their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Development using an agile methodology preserves a product's critical market relevance and ensures a team's work doesn't wind up on a shelf, never released.



Introduction to Scrum video

Be Sociable, Share!



+ MORE

Tweet 178

Like 541

G+1 58

Share 514

St

Pin

Posted under