Agile





User req -- > design -- > develop -- >BRD -- > Unit/ Integration / System Testing / UAT -- > Demo/ Release

What Is Agile?

The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability.

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 building properly tested product incremented within short iterations. 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.

<http://agilemethodology.org/>

Agile Methodology

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen tight iterations throughout the development cycle.

Proponents of lightweight agile methods contend that they are returning to development practices that were present early in the history of software development.

Agile values

The Agile Manifesto reads, in its entirety, as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

* Individuals and interactions over Processes and tools
* Working software over Comprehensive documentation
* Customer collaboration over Contract negotiation
* Responding to change over Following a plan

The meanings of the manifesto items within the agile software development context are:

Individuals and interactions – in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

Working software – working software will be more useful and welcome than just presenting comprehensive documents to clients in meetings.

Customer collaboration – requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.

Responding to change – agile development is focused on quick responses to change and continuous development.

**Agile principles**

The Agile Manifesto is based on twelve principles:

Customer satisfaction by rapid delivery of useful software

Welcome changing requirements, even late in development

Working software is delivered frequently (weeks rather than months)

Working software is the principal measure of progress

Sustainable development, able to maintain a constant pace

Close, daily cooperation between business people and developers

Face-to-face conversation is the best form of communication (co-location)

Projects are built around motivated individuals, who should be trusted

Continuous attention to technical excellence and good design

Simplicity—the art of maximizing the amount of work not done—is essential

Self-organizing teams

Regular adaptation to changing circumstances

**Philosophy**

Compared to traditional software engineering, agile development is mainly targeted at complex systems and projects with dynamic, undeterministic and non-linear characteristics, where accurate estimates, stable plans and predictions are often hard to get in early stages, and big up-front designs and arrangements will probably cause a lot of waste, i.e. not economically sound. These basic arguments and precious industry experiences learned from years of successes and failures have helped shape Agile's favor of adaptive, iterative and evolutionary development.

<http://en.wikipedia.org/wiki/Agile_software_development>

Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal". It challenges assumptions of the "traditional, sequential approach" to product development. Scrum enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members and daily face to face communication among all team members and disciplines in the project.

A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly and respond to emerging requirements.

**Role of Product Owner in Defining and Communicating Product Requirements**

Communication is a main function of the product owner. The ability to convey priorities and empathize with team members and stakeholders are vital to steer the project in the right direction. Product owners bridge the communication gap between the team and their stakeholders. As Figure 1 shows, they serve as a proxy stakeholder to the development team and as a project team representative to the overall stakeholder community.[10]

As the face of the team to the stakeholders, the following are some of the communication tasks of the product owner to the team: - demonstrates the solution to key stakeholders who were not present in a normal iteration demo - announced releases - communicates team status - organizes milestone reviews - educates stakeholders in the development process - negotiates priorities, scope, funding, and schedule

Empathy is a key attribute for a product owner to have—the ability to put one’s self in another’s shoes. A product owner will be conversing with different stakeholders in the project— different people, with a variety of backgrounds, job roles, and objectives. A product owner needs to be able to see from these different points of view. To be effective, it would also be wise for a product owner to know the level of detail his audience needs from him. The development team would need thorough feedback and specifications so they build a product up to expectation, while an executive sponsor may just need summaries and cliff notes of progress. Providing more information than necessary may lose their interest and waste time. There is also significant evidence that face-to-face communication around a shared sketching environment is the most effective way to communicate information instead of documentation. A direct means of communication is then most preferred by seasoned agile product owners.[11]

A product owner’s ability to communicate effectively is also enhanced by being skilled in techniques that identify stakeholder needs, negotiate priorities between stakeholder interests, and collaborate with developers to ensure effective implementation of requirements.

**Development Team**

The Development Team is responsible for delivering potentially shippable increments (PSIs) of product at the end of each Sprint (the Sprint Goal). A Team is made up of 3–9 individuals with cross-functional skills who do the actual work (analyse, design, develop, test, technical communication, document, etc.). The Development Team in Scrum is self-organizing, even though there may be some level of interface with project management offices (PMOs).

**Scrum Master**

Scrum is facilitated by a Scrum Master, who is accountable for removing impediments to the ability of the team to deliver the product goals and deliverables. The Scrum Master is not a traditional team lead or project manager, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended. The Scrum Master is the enforcer of the rules of Scrum, often chairs key meetings, and challenges the team to improve. The role has also been referred to as a servant-leader to reinforce these dual perspectives.

The Scrum Master differs from a project manager in that the latter may have people management responsibilities unrelated to the role of Scrum Master. The Scrum Master role excludes any such additional people responsibilities. In fact, there is no role of project manager in Scrum at all, because none is needed. The traditional responsibilities of a project manager have been divided up and reassigned among the three Scrum roles, and mostly to the Development Team and the Product Owner, rather than to the Scrum Master. Practicing Scrum with the addition of a project manager indicates a fundamental misunderstanding of Scrum, and typically results in conflicting responsibilities, unclear authority, and sub-optimal results.[12]

Sprint[edit]

**The Scrum process**

A sprint (or iteration) is the basic unit of development in Scrum. The sprint is a "timeboxed" effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, although two weeks is typical.

Each sprint is started by a planning meeting, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made, and ended by a sprint review-and-retrospective meeting, where the progress is reviewed and lessons for the next sprint are identified.

Scrum emphasizes working product at the end of the Sprint that is really "done"; in the case of software, this means a system that is integrated, fully tested, end-user documented, and potentially shippable.

**Events**

**Meetings**

**Sprint planning meeting**

**At the beginning of the sprint cycle (every 7–30 days), a "Sprint planning meeting" is held:**

Select what work/product is to be done

Prepare the Sprint Backlog that details the specific time it will take to do that work, with the entire team

Identify and communicate how much of the work is likely to be done during the current sprint

Eight-hour time limit

(1st four hours) Entire team: dialog for prioritizing the Product Backlog

(2nd four hours) Development Team: hashing out a plan for the Sprint, resulting in the Sprint Backlog

**Daily Scrum meeting**

Each day during the sprint, a project team communication meeting occurs. This is called a Daily Scrum (meeting) and has specific guidelines:

All members of the development team come prepared with the updates for the meeting.

The meeting starts precisely on time even if some development team members are missing.

The meeting should happen at the same location and same time every day.

The meeting length is set (timeboxed) to 15 minutes.

All are welcome, but normally only the core roles speak.

**During the meeting, each team member answers three questions:**

**What have you done since yesterday?**

**What are you planning to do today?**

Any impediments/stumbling blocks? Any impediment/stumbling block identified in this meeting is documented by the Scrum Master and worked towards resolution outside of this meeting. No detailed discussions shall happen in this meeting.

**End meetings**

At the end of a sprint cycle, two meetings are held: the "Sprint Review Meeting" and the "Sprint Retrospective".

**At the Sprint Review Meeting:**

**Review the work that was completed and the planned work that was not completed**

Present the completed work to the stakeholders (a.k.a. "the demo")

Incomplete work cannot be demonstrated

Four-hour time limit

**At the Sprint Retrospective:**

All team members reflect on the past sprint

**Make continuous process improvements**

**Two main questions are asked in the sprint retrospective:**

**What went well during the sprint?**

**What could be improved in the next sprint?**

Three-hour time limit

This meeting is facilitated by the Scrum Master

**Terminology**

The following terms are often used in a Scrum process.

**Scrum Team(roles)**

Product Owner, Scrum Master and Development Team

**Product Owner**

The person responsible for maintaining the Product Backlog by representing the interests of the stakeholders, and ensuring the value of the work the Development Team does.

**Scrum Master**

The person responsible for the Scrum process, making sure it is used correctly and maximizing its benefits.

**Development Team**

A cross-functional group of people responsible for delivering potentially shippable increments of Product at the end of every Sprint.

**Sprint burn down chart**

Daily progress for a Sprint over the sprint's length.

**Release burn down chart**

Sprint level progress of completed product backlog items in the Product Backlog.

**Product backlog (PBL)**

A prioritized list of high-level requirements.

**Sprint backlog (SBL)**

A prioritized list of tasks to be completed during the sprint.

**Velocity**

The total effort a team is capable of in a sprint. The number is derived by evaluating the work (typically in [user story](http://en.wikipedia.org/wiki/User_story) points) completed from the last sprint's backlog items. The collection of historical velocity data is a guideline for assisting the team in understanding how much work they can do in a future sprint.

<http://en.wikipedia.org/wiki/Scrum_(development)>

<https://www.youtube.com/watch?v=vm5kGy6URjM>

<https://www.youtube.com/watch?v=x6zRZet8IEs>

<https://www.youtube.com/watch?v=t9VaLg-eh0w>

**Test execution strategy**

Agile testing must be iterative

Testers canot rely on having completer specification

Testers must be flexible

They need to be independetn and independently empowered in order to dffective

be generalizing specialists.

Be preapred to wrk closely with developers.

focus on value added activities

be flixible

focus on what an not how to test.

Testers shuld be embedded in agile team.

Flexibile t o contribute in any way then can.

Have wide range of skills with one or more specialities.

Shorter feedback cycles.

Focus on sufficient and straightforward situations.

Focus on exploratory testing.

Specify the meaning of "done"

Define when to continue or stop testing before release the testing product. Specify evaluation criteria such as limited time, user coverage and quality.