

An Empirical Framework For Learning (Not a Methodology)



The Scrum Reference Card

Scrum is part of the Agile movement. Agile is a response to the failure of the dominant software development project management paradigms (including waterfall) and borrows many principles from lean manufacturing. In 2001, 17 pioneers of similar methods met at the Snowbird Ski Resort in Utah and wrote the Agile Manifesto, a declaration of four values and twelve principles. These values and principles stand in stark contrast to the traditional Project Manager's Body Of Knowledge (PMBOK). The Agile Manifesto placed a new emphasis on communication and collaboration, functioning software, team self organization, and the flexibility to adapt to emerging business realities.

How Does Scrum Fit With Agile?

The Agile Manifesto doesn't provide concrete steps. Organizations usually seek more specific methods within the Agile movement. These include Crystal Clear, Extreme Programming, Feature Driven Development, Dynamic Systems Development Method (DSDM), Scrum, and others. While I like all the Agile approaches, for my own team Scrum was the one that enabled our initial breakthroughs. Scrum's simple definitions gave our team the autonomy we needed to do our best work while helping our boss (who became our *Product Owner*) get the business results he wanted. Scrum opened our door to other useful Agile practices such as test-driven development (TDD). Since then we've helped businesses around the world use Scrum to become more agile. A truly agile enterprise would not have a "business side" and a "technical side." It would have teams working directly on delivering business value. We get the best results when we involve the whole business in this, so those are the types of engagements I'm personally the most interested in.

What's The Philosophy Behind Scrum?

Scrum's early advocates were inspired by empirical *inspect and adapt* feedback loops to cope with complexity and risk. Scrum emphasizes decision making from real-world results rather than speculation. Time is divided into short work cadences, known as sprints, typically one week or two weeks long. The product is kept in a potentially shippable (properly integrated and tested) state at all times. At the end of each sprint, stakeholders and team members meet to see a demonstrated potentially shippable product increment and plan its next steps.

Scrum is a simple set of roles, responsibilities, and meetings that never change. By removing unnecessary unpredictability, we're better able to cope with the necessary unpredictability of continuous discovery and learning.

Scrum Roles

Scrum has three roles: Product Owner, Scrum Master, and Team.



The Example Scrum Master's Checklist

- **Product Owner:** The Product Owner should be a person with vision, authority, and availability. The Product Owner is responsible for continuously communicating the vision and priorities to the development team.

It's sometimes hard for Product Owners to strike the right balance of involvement. Because Scrum values self-organization among teams, a Product Owner must fight the urge to micro-manage. At the same time, Product Owners must be available to answer questions from the team.

- **Scrum Master:** The Scrum Master acts as a facilitator for the Product Owner and the team. The Scrum Master does not manage the team. The Scrum Master works to remove any impediments that are obstructing the team from achieving its sprint goals. This helps the team remain creative and productive while making sure its successes are visible to the Product Owner. The Scrum Master also works to advise the Product Owner about how to maximize ROI for the team.
- **Team:** According to Scrum's founder, "the team is utterly self managing." The development team is responsible for self organizing to complete work. A Scrum development team contains about seven fully dedicated members (officially 3-9), ideally in one team room protected from outside distractions. For software projects, a typical team includes a mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers. Each sprint, the team is responsible for determining how it will accomplish the work to be completed. The team has autonomy and responsibility to meet the goals of the sprint.

Scaled Agile

When several teams work on one product, they should generally use a single Product Owner (who can make real business decisions) and a single Product Backlog with customer-centric requirements. Each Scrum team should strive to become a *feature team*, able to build a complete slice of product which could be delivered to a customer. When interdependencies arise, Scrum's feature teams must learn to use team self organization principles to coordinate with other teams. Unfortunately, most teams are not initially accustomed to this level of responsibility, and pre-existing management habits and hierarchies present organizational impediments. Scrum eliminates traditional co-ordination roles such as project manager and PMO, as these interfere with team self organization. Scrum eliminates traditional technical czar roles such as "architect," as technical decisions are made by collaborative teams.

While optimal Agility requires fundamental changes to organizational design, it's tempting to use one of the hybrid approaches that combine a watered-down version of Scrum with traditional hierarchical

management. Large organizations that are more committed to the values and principles of the Agile Manifesto are advised to learn about [Large Scale Scrum \(LeSS\)](#).