

Agile project management

- The principal responsibility of software project managers is to manage the project so that the software is delivered on time and within the planned budget for the project.
- The standard approach to project management is plan-driven. Managers draw up a plan for the project showing what should be delivered, when it should be delivered and who will work on the development of the project deliverables.
- Agile project management requires a different approach, which is adapted to incremental development and the practices used in agile methods.

1

Another popular method...

SCRUM

Scrum - Overview

- An agile process that allows us to focus on **delivering the highest business value** in the **shortest time**
- It allows us to **rapidly** and **repeatedly inspect actual working software** (every two weeks to one month).
- The **business sets the priorities**. Our teams self-manage to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone **can see real working software** and decide to release it as is or continue to enhance for another iteration.

History of Scrum



- **1995:**
 - Design of a new method: Scrum by Jeff Sutherland & Ken Schwaber
 - Enhancement of Scrum by Mike Beedle & combination of Scrum with Extreme Programming
 - **1995:** introduction of Scrum at international conference
 - **2001:** publication "Agile Software Development with Scrum" By Ken Schwaber & Mike Beedle
- Successful appliance of Scrum in many companies
Founders are members of the Agile Alliance



From agilealliance.org

"In the late 1990's several methodologies began to get increasing public attention. Each had a different combination of old ideas, new ideas, and transmuted old ideas. But they all emphasized close collaboration between the programmer team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organizing teams; and ways to craft the code and the team such that the inevitable requirements churn was not a crisis."

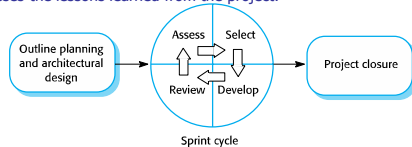
Encompasses everything that the agile alliance think agile is all about.

Characteristics of Scrum

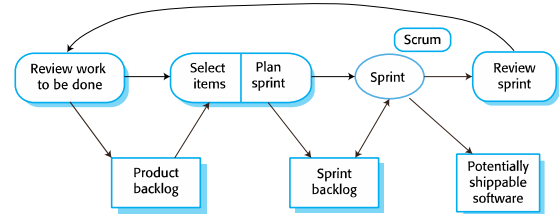
- Self-organizing teams
- Product progresses in a series of month-long "sprints"
- Requirements are captured as items in a list of "product backlog"
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the "agile processes"

Scrum Approach

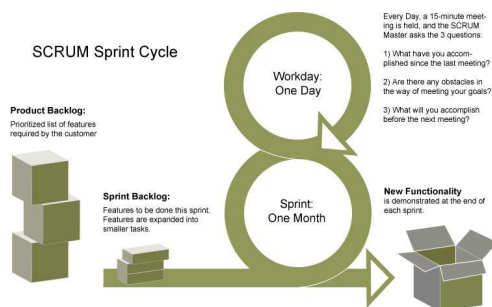
- The Scrum approach is a general agile method but its **focus is on managing iterative development** rather than specific agile practices.
- There are three phases in Scrum.
 - The **initial phase** is an **outline planning phase** where you establish the general objectives for the project and design the software architecture.
 - This is followed by a series of **sprint cycles**, where each cycle develops an increment of the system.
 - The **project closure** phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.



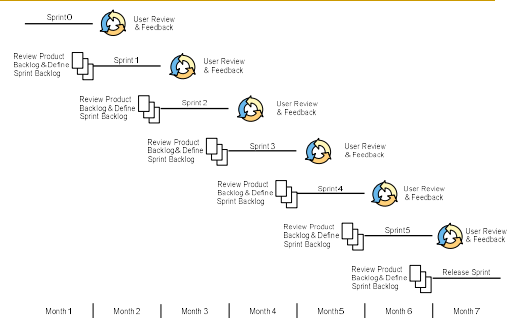
Scrum sprint cycle



Sprint Iteration



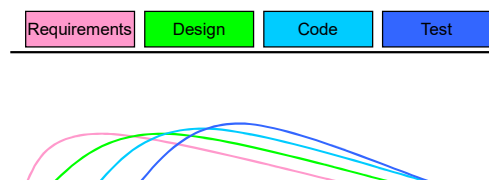
Potentially releasable code delivered every 4 weeks



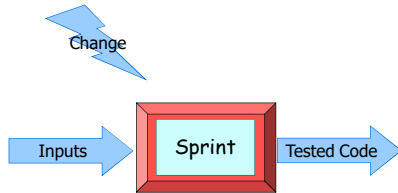
Sprints

- Scrum projects make progress in a series of "sprints"
 - Analogous to XP iterations
- Target duration is one month
 - +/- a week or two
 - But, a constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Sequential vs. Overlapping Dev.



No changes during the sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint
- But is this insistence sacrosanct? There is some debate on that topic.

Scrum Framework

- **Roles** : Product Owner, ScrumMaster, Team
- **Ceremonies** : Sprint Planning, Sprint Review, Sprint Retrospective, & Daily Scrum Meeting
- **Artifacts** : Product Backlog, Sprint Backlog, and Burndown Chart

Product Owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results.

The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

Scrum Team

- Typically 5-10 people
- Cross-functional
 - QA, Programmers, UI Designers, etc.
- Members should be full-time
 - May be exceptions (e.g., System Admin, etc.)
- Teams are self-organizing
 - What to do if a team self-organizes someone off the team??
 - Ideally, no titles but rarely a possibility
- Membership can change only between sprints

Scrum Ceremonies

- Sprint Planning Meeting
 - Held at beginning of a sprint
 - Define scope & goals of next sprint
- Scrum meeting
 - AKA 'Daily Stand Up'
 - Held every day (morning) before team starts work
- Sprint Review Meeting
 - Last day of Sprint
 - Identify missing feature and future direction of product
- Sprint retrospective meeting
 - Team reflect on last sprint "process"
 - Reviews process not product

Parts of Sprint Planning Meeting

- 1st Part:
 - Creating Product Backlog
 - Determining the Sprint Goal.
 - Participants: Product Owner, ScrumMaster, Scrum Team
- 2nd Part:
 - Participants: ScrumMaster, Scrum Team
 - Creating Sprint Backlog

Pre-Project/Kickoff Meeting

- A special form of Sprint Planning Meeting
- Meeting before the beginning of the Project

Sprint

- A month-long iteration, during which product functionality is incremented
- NO outside influence can interfere with the Scrum team during the Sprint
- Each Sprint begins with the Daily Scrum Meeting



Daily Scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
 - Not for problem solving
- Three questions:
 1. **What did you do yesterday**
 2. **What will you do today?**
 3. **What obstacles are in your way?**
- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a ScrumMaster to track the progress of the Team

Scrum FAQs

- Why daily?
 - "How does a project get to be a year late?"
 - "One day at a time."
 - Fred Brooks, The Mythical Man-Month.
- Can Scrum meetings be replaced by emailed status reports?
 - No
 - Entire team sees the whole picture every day
 - Create peer pressure to do what you say you'll do

Scalability of Scrum

- A typical Scrum team is 6-10 people
- Jeff Sutherland - up to over 800 people
- "Scrum of Scrums" or what is sometimes called a "Meta-Scrum"
- Frequency of meetings is based on the degree of coupling between packets

Scaling agile methods

- Agile methods have proved to be successful for small and medium sized projects that can be developed by a small co-located team.
- It is sometimes argued that the success of these methods comes because of improved communications which is possible when everyone is working together.
- Scaling up agile methods involves changing these to cope with larger, longer projects where there are multiple development teams, perhaps working in different locations.

25

Scaling out and scaling up

- **'Scaling up'** is concerned with using agile methods for developing large software systems that cannot be developed by a small team.
- **'Scaling out'** is concerned with how agile methods can be introduced across a large organization with many years of software development experience.
- When scaling agile methods it is important to maintain agile fundamentals:
 - Flexible planning, frequent system releases, continuous integration, test-driven development and good team communications.

26

Scrum of Scrums

- A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10. Each **daily scrum** within a sub-team ends by designating one member as "ambassador" to participate in a daily meeting with ambassadors from other teams, called the Scrum of Scrums.
- Depending on the context, ambassadors may be technical contributors, or each team's ScrumMaster, or even managers of each team.
- The Scrum of Scrums proceeds otherwise as a normal daily meeting, with ambassadors reporting **completions, next steps and impediments** on behalf of the teams they represent.

Scrum Pros/Cons

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ Advantages <ul style="list-style-type: none"> ▪ <i>Completely developed and tested features in short iterations</i> ▪ <i>Simplicity of the process</i> ▪ <i>Clearly defined rules</i> ▪ <i>Increasing productivity</i> ▪ <i>Self-organizing</i> ▪ <i>each team member carries a lot of responsibility</i> ▪ <i>Improved communication</i> ▪ <i>Combination with Extreme Programming</i> | <ul style="list-style-type: none"> ▪ Drawbacks <ul style="list-style-type: none"> ▪ <i>"Undisciplined hacking" (no written documentation)</i> ▪ <i>Violation of responsibility</i> ▪ <i>Current mainly carried by the inventors</i> ▪ <i>Lack of defined process... is this a problem?</i> |
|---|---|

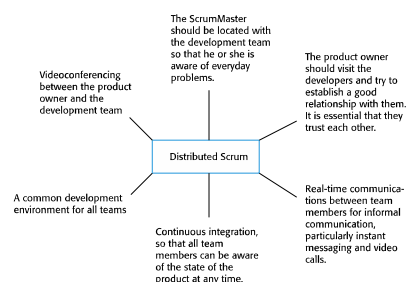
Contractual issues?



- Most software contracts for custom systems are based around a specification, which sets out what has to be implemented by the system developer for the system customer.
- However, this precludes interleaving specification and development as is the norm in agile development.
- A contract that pays for developer time rather than functionality is required.
 - However, this is seen as a high risk by many legal departments because what has to be delivered cannot be guaranteed.

29

Distributed Scrum?



30

Reflections on Scrum

- In the context of agile software development, Scrum is probably the most significant software project management methodology to be introduced over the past 20 years.
- Scrum has some limitations, noted in the earlier slides, especially in terms of scaling to larger teams or larger projects / organisations.
- Scrum is inherently suited to agile project management, and therefore all of the applicability and benefits/limitations of general agile development can be relevant to Scrum itself.
- Scrum is also used outside of software development.