

Agile Methodology

Introduction to Agile Methodology

SOFTWARE PROCESS IMPROVEMENT

NASA Goddard Space Flight Center

DEFINE • EXECUTE • ANALYZE • REVISE



Agile Methodology

- **Introductions**
 - **Alex Durkin, Software Process Improvement Team**
<http://software.gsfc.nasa.gov>

Goals

- **Common understanding of Agile fundamentals**
 - **Origins of Agile**
 - **Key Concepts / Activities / Roles**
 - **Common Misconceptions**
 - **Strengths and Weaknesses**

What is Agile?

- **Definition*:** The Agile Alliance defines Agile Software Development in this way:

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 working cooperatively
- Ways to craft the code and the team such that the inevitable requirements churn was not a crisis

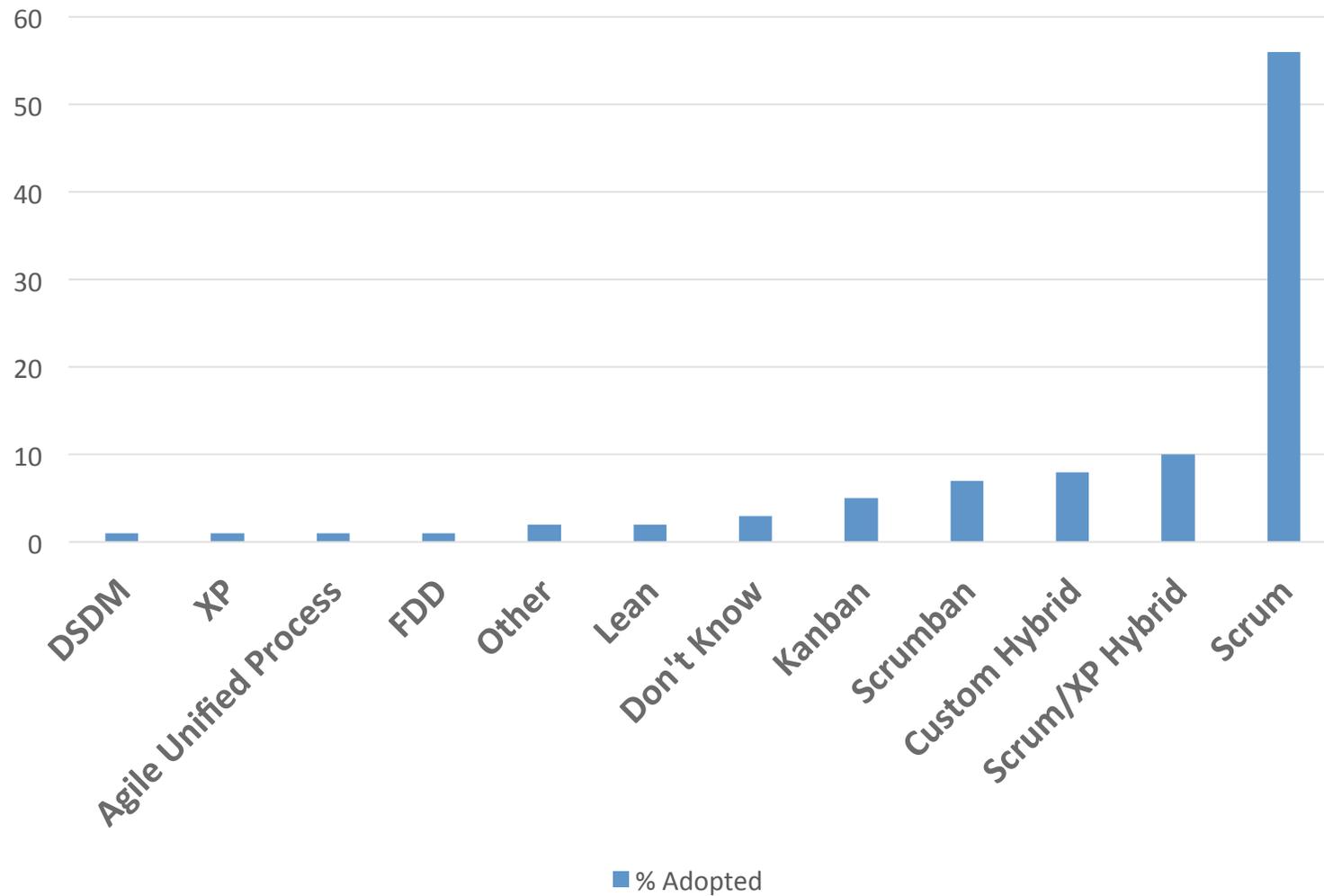
- **The Agile Manifesto was written in February of 2001, at a summit of 17 independent-minded practitioners of several programming methodologies.**

- *Definition of Agile from Agile Alliance, a non-profit organization founded by some authors of the original Agile Manifesto to advance Agile concepts; <http://www.agilealliance.org/the-alliance>

What is Agile?

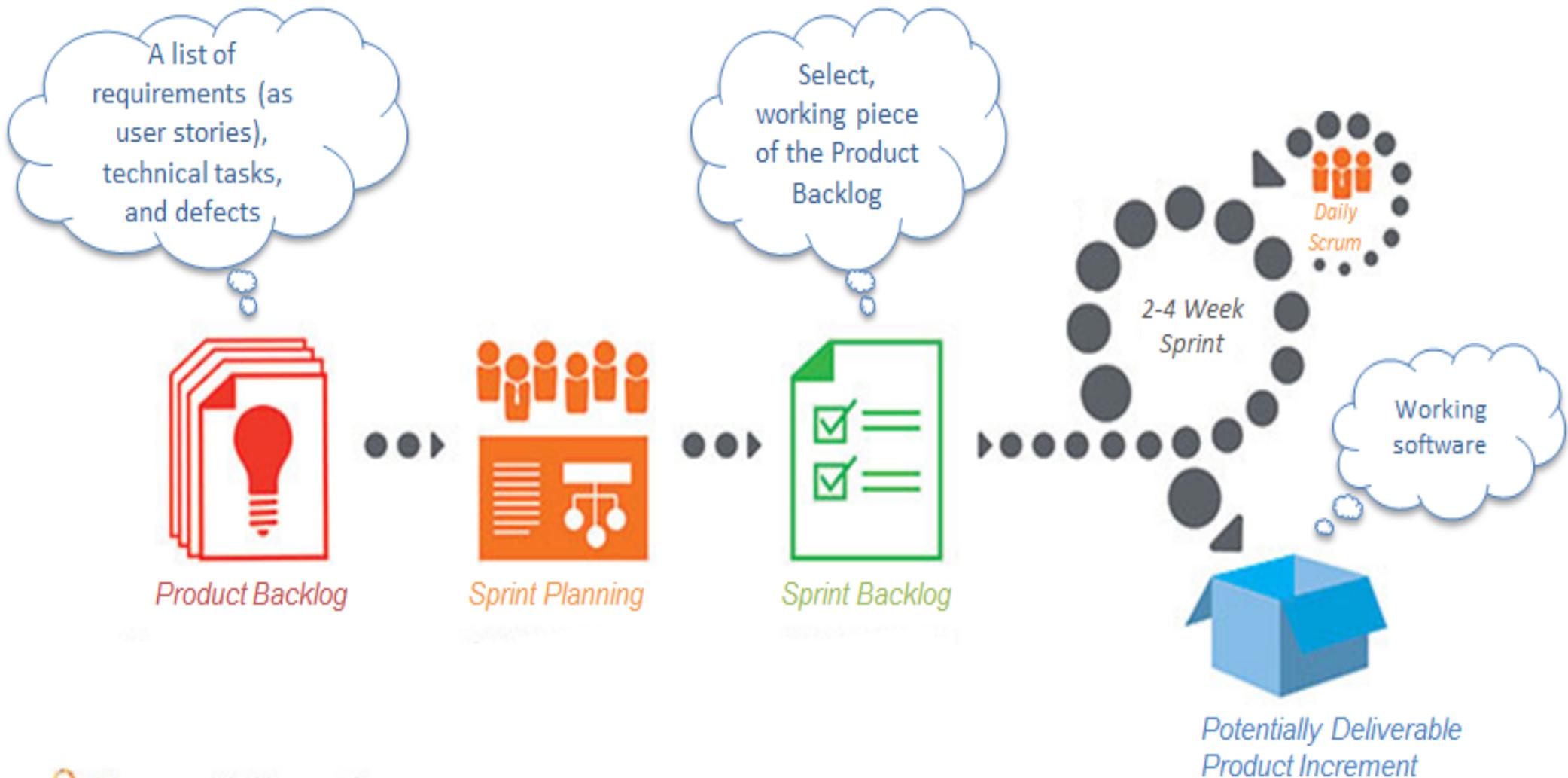
- **Interestingly, Agile concepts are applied to fields outside of software**
 - **Toyota implemented Agile techniques long before software development**
 - **One of which was a task board in 1953, in Agile software development this became known as a Kanban board**
 - **Team Wikispeed is using an Agile method called Scrum to build a fast, affordable, ultra-efficient, safe commuter car that should sell for less than \$20,000**
- **There are many methods implementing the Agile Software Development Methodology. Most notable:**
 - **Lean**
 - **Kanban**
 - **Extreme Programming (XP)**
 - **Scrum -- most popular**

Agile Methods In Use



- Version One [9th Annual State of Agile™ Survey](#)

Scrum Framework



Original diagram from the [Scrum Alliance](https://www.scrumalliance.org/)

Scrum – Key Concepts

- **Product Backlog**

The product backlog is a list of requirements (as user stories) and technical tasks which the team maintains and that, at a given moment, are known to be necessary and sufficient to complete a product or a release

- The product backlog is the primary point of entry for knowledge about requirements, and the single authoritative source defining the work to be done
- Includes features, defects, and other technical work
- Any format: Excel document, a text file, a database
Agile teams commonly use a collection of index cards or Post-It notes on a task board (Kanban)

- **Sprint Backlog**

The sprint backlog is a working subset of the product backlog, small enough to be completed in a single sprint

- All Term definitions from the [Agile Alliance](#), the [Scrum Alliance](#), the [Mountain Goat Software](#) website by Mike Cohn, a founder of the Agile Alliance; and the [Agile Modeling](#) website by Scott Ambler, author of “Disciplined Agile Delivery”.

Scrum – Key Concepts

- **User Story**

A user story is a method to capture desired product features from the perspective of a user or customer.

- Defined and sorted by priority into the Product Backlog by Product Owner
- Concise enough to fit on an index card
- Each user story is expected to yield, once implemented, a contribution to the value of the overall product, irrespective of the order of implementation
- Generally high level at first, they can evolve and be refined into low-level user stories, or requirements
 - It's often best to think of the written part as a pointer to a real requirement
 - Detail can be added by adding “conditions of satisfaction”
- Stories can be assigned points based on size and complexity
- Commonly takes the form of:
As a <type of user>, I want <some goal> so that <some reason>.
 - Example:
“As a power user, I want to specify files or folders for backup based on file size, date created and date modified, so I can prioritize items for backup.”
 - May belong to the higher-level user story (epic): *“As a user, I can backup my entire hard drive.”*

Scrum – Key Concepts

- **Sprint**

The sprint is a set amount of time (either 2 or 4 weeks) determined to be sufficient to develop a working deliverable software based on the product backlog

- It is a time-boxed effort, completing all 4 phases (analysis, design, implementation, testing)
 - Automated testing is widely used in Agile
- The end product of the sprint is working software that is complete and potentially deliverable

- **Daily Scrum, or Daily Stand-up – during the Sprint, less than 15 minutes**

- Purpose: Inspect progress, identify impediments to progress
- Participants: Scrum Master and Team – Product Owner and other stakeholders are invited in listen-only mode

The daily stand-up is 10-15 minute daily briefing while standing

- Each team member addresses:
 - What did I do yesterday that helped the Development Team meet the Sprint Goal?
 - What will I do today?
 - Are there any roadblocks in my way?

Scrum – Key Concepts

- **Sprint Planning** – at most 8 total hours for a one month Sprint

- Purpose: Determine work for next Sprint
- Participants: Owner, Scrum Master, Team

The sprint planning meeting is when the product owner describes the highest priority features to the team, and the team moves user stories from the product backlog into the sprint backlog

- Artifacts resulting are a Sprint goal and the Sprint backlog

- **Sprint Review** – after the Sprint, at most 4 hours

- Purpose: Inspect completed work, adapt priorities for next Sprint
- Participants: Owner, Scrum Master, Team, Stakeholders, other developers

The sprint review is a review meeting held at the conclusion of a sprint where the team presents this software, as a demo or a test drive, to the external stakeholders and discuss how to adapt the product to the overall feedback.

- Artifacts resulting may be new user stories for the product backlog
- Typically held to no more than 2 hours, and attended by Product Owner, Scrum Team, Scrum Master, and external stakeholders

Scrum – Key Concepts

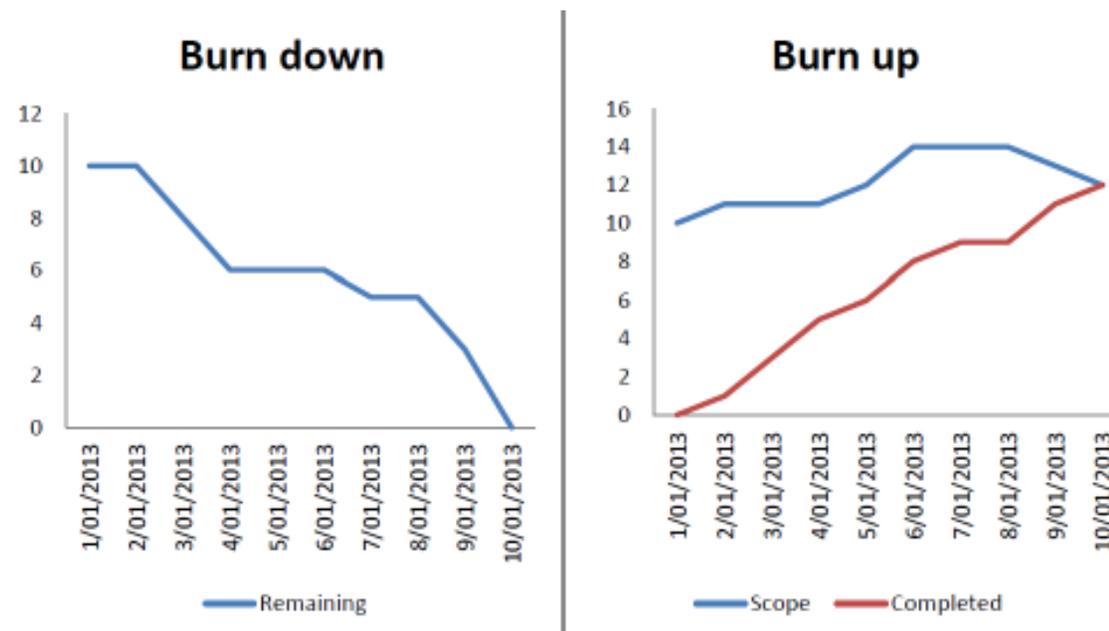
- **Burn-down chart**

A product burn-down chart shows the number of backlog items completed vs yet to be completed overall (updated after each sprint)

A sprint burn-down chart shows the number of remaining sprint backlog items completed vs yet to be completed in the current sprint (updated daily)

- **Burn-up chart**

More visible when the scope of the product changes, but harder to convey when the end of the process will be



<http://www.clariostechology.com/productivity/blog/burnupvsburndownchart>

Scrum – Key Concepts

- **Retrospective** – after the Sprint Review, before the next Sprint Planning, at most 3 hours
 - Purpose: Identify improvements to Team development process
 - Participants: Scrum Master and Team

The retrospective is a dedicated period at the end of each sprint to deliberately reflect on how the team is doing and to find ways to improve.

- Typical artifacts are lessons learned to be reviewed next retrospective
 - Scrum master asks:
 - “What went well during the sprint?”
 - “What could be improved in the next sprint?”
-
- **Velocity**

The velocity is a measure of how much work was completed during the sprint

 - Usually measured in stories completed or story points instead of hours
 - Allows for scoping of future sprints

Scrum – Key Concepts

- **Technical debt**

The technical debt is unfinished work after a sprint is over

- Can build up over several sprints
- Gets prioritized and placed on the Product Backlog

- **Sprint Zero**

Sprint Zero, or the sprint before sprints, is a phrase coined for an initial sprint limited to project setup and initiation

- Used for planning the sprint backlog, performing installs, and other prep activities
- No released software
- Still a 2-4 week period, like any other sprint

Scrum – Additional Concepts

- **Product Backlog Refinement** – less than 10% of total team effort

Backlog refinement, or grooming, is the process of soliciting, refining and ranking requirements by the Product Owner with input from the Team and Customers

- Users stories can be added, removed or split
- Priorities and estimates are developed or changed
- All stakeholders should understand the ranking criteria and resulting priorities

- **Hardening Sprint**

A hardening sprint is an additional sprint that some Scrum Teams perform to ready the product for release

- Controversial concept that isn't universally accepted
- Scrum teams often use it to tie up the loose ends identified during this sprint or earlier sprints

- All Term definitions from the [Agile Alliance](#), the [Scrum Alliance](#), the [Mountain Goat Software](#) website by Mike Cohn, a founder of the Agile Alliance; and the [Agile Modeling](#) website by Scott Ambler, author of "Disciplined Agile Delivery".

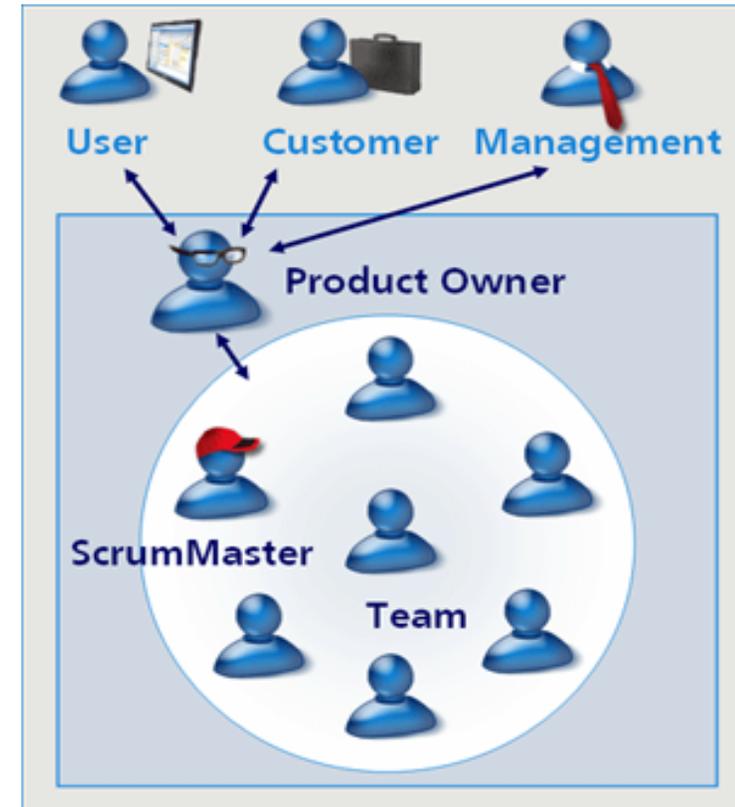
Scrum Roles

External Stakeholders:

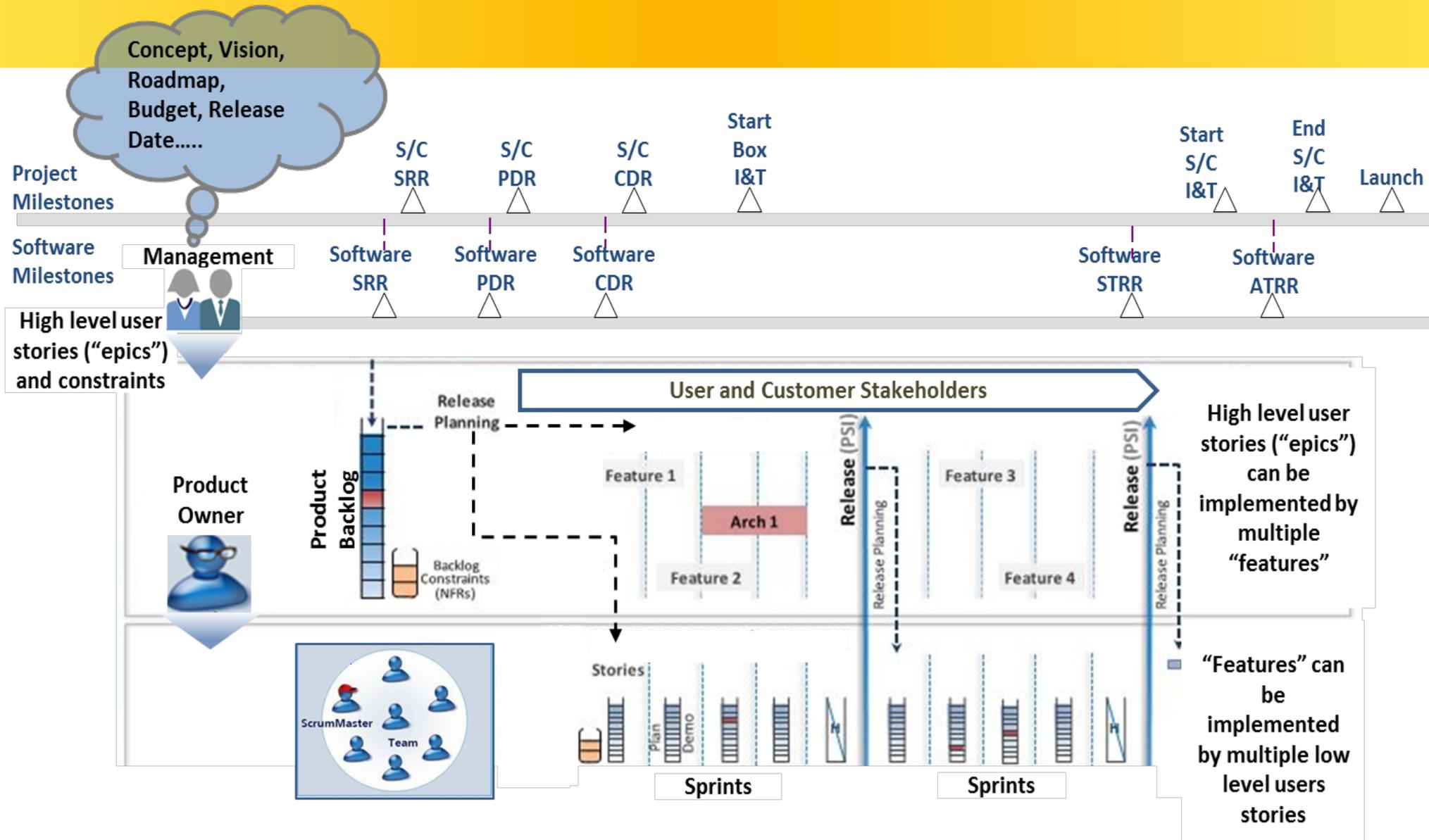
- User – The end user of the product
- Customer – Client often paying for the work
- Management – Use a hands-off approach with the technical team, **coordinate with product owner**, attend milestone reviews as “listen-only”

Internal Stakeholders:

- Product Owner – Represent the interests of the external stakeholders, act as a filter between team and external stakeholders, maintain the product backlog
- Scrum Master – Leading *member of the team* that focuses on the success of the sprint, runs the daily stand-up
- Scrum Team – A cross-functional team made up of the developers, testers, architects, administrators and software assurance
 - Co-located, frequent communication - at the least meets daily
 - Recommended size of 3-9 people
 - Self-organizing but within a set boundary
 - The focus is to get work done, even if you need help
- Scrum Role definitions derived from the [Scrum Methodology](#) and the [Scrum Alliance](#)



- <https://noogony.wordpress.com/2010/09/19/5-tips-voor-een-succesvolle-sharepoint-implementatie-adhv-scrum-deel-1/>



NFRs = Non-functional requirements

PSI = Potentially shippable increment

Adapted from www.scalingsoftwareagility.wordpress.com

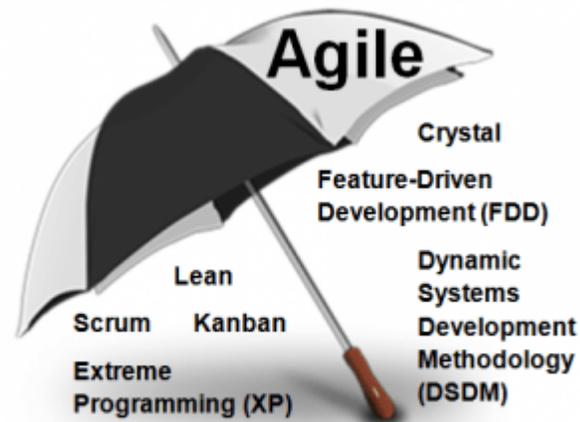
Common Misconceptions

- **Agile is synonymous with Scrum**
- **Agile means no documentation**
- **Agile means no planning – just coding**
- **Agile only works when everyone is co-located**
- **Agile means faster and cheaper**

• The [Agile Manifesto](http://www.agilealliance.org/the-alliance/the-agile-manifesto/): <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>

Common Misconceptions

- **“Agile is synonymous with Scrum”**
 - Scrum is one development method that implements Agile (the most popular one)
 - Scrum is constantly evolving through an active community, leading to many hybrid adaptations
 - Scrum implements all the values and principles of Agile



- [Scrum and Agile, synonyms? \(and other Agile methodologies\)](#)

Common Misconceptions

- **“Agile means no documentation”**
 - A focus in Agile is working software over documentation (but not *instead of* documentation)
 - Agile calls for more frequent updates to documentation
 - Documentation is done real time focusing on the sprint, rather than documenting everything all at once
 - The Agile version of documentation says:
 - There is an overall plan
 - There are requirements descriptions
 - There are cost and schedule estimates
 - There are risk assessments
 - There is training material (as appropriate)
 - There is documentation (as appropriate)
 - There are lessons learned (based on retrospectives)

- [What Does the Agile Manifesto Mean?](#) and [Agile software development, the principles. Principle 9 : Continuous attention to technical excellence and good design enhances agility](#)

Common Misconceptions

- **“Agile means no planning – just coding”**
 - Agile seeks to eliminate “analysis paralysis” through shorter planning cycles, hopefully 2-4 weeks
 - Teams are self-organizing, so Agile is not focused on task assignments
 - In Scrum, sprint planning can evolve each sprint
 - Sprint planning before each sprint
 - Sprint Zero is 2-4 weeks of planning before coding starts
 - Developers can’t start coding in a sprint until the Scrum master has picked out the User Stories for the Sprint Backlog
 - After the end of every sprint, a few hours are spent planning
 - While the development lifecycle is dynamic, product plans and system testing are still necessary

- [What Does the Agile Manifesto Mean?](#) and [Agile software development, the principles. Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams](#)

Common Misconceptions

- **“Agile only works when everyone is co-located”**
 - Agile is heavily focused on face-to-face communication
 - This does not mean teams have to be co-located - video conferencing is one solution
 - Be aware that a more geographically distributed team could then increase cost
 - The Agile methodology stresses that email and phone conversations are no substitute for the daily meetings
 - A mitigation for a user not co-located is to employ a local user representative
 - There are Agile scaling techniques to accommodate multiple teams in separate locations
 - Two different scrums in two different locations operate as normal, and the Scrum Masters from each meet to define Sprint Backlog
- [What Does the Agile Manifesto Mean?](#) and [Agile software development, the principles. Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.](#)

Common Misconceptions

- **“Agile means faster and cheaper”**
 - Agile defines short development and delivery cycles
 - This means that a working product is delivered faster
 - From the customer’s point of view, they receive success faster
 - Leads to a ‘fail fast’ scenario, where they realize that a product solution may be flawed, correcting their course early and potentially saving money
 - However Agile has a steep learning curve
 - Inexperience with Agile can lead to delays, building up technical debt or difficulty planning sprints
 - Sometimes leads to performing traditional methods when you thought you were being Agile
 - It’s common not to see added benefit to Agile until late in the lifecycle
 - Other Agile implementation can have different cost impacts
 - Pair programming and other XP techniques often cost more
- [Agile Alliance](#), the [Scrum Alliance](#), the [Mountain Goat Software](#) website by Mike Cohn, a founder of the Agile Alliance; and the [Agile Modeling](#) website by Scott Ambler, author of “Disciplined Agile Delivery”.

Agile: Strengths and Weaknesses

Strengths

- Users are kept involved in the process daily, increasing likelihood of validation
- Accepts changes in requirements during all stages of development
- High amount of risk analysis done
- Avoid “analysis paralysis” by diving into development early
- Working products are available throughout the process
- Thrives with co-located teams
- Documentation is done real-time, during each iteration

Weaknesses

- Dependent on availability of users/customers or their representatives
- Increases in difficulty for larger development teams
- Requires strict adherence to activities
- Some methods are more expensive (Pair-programming)
- Customer involvement can increase rework or gold-plating
- Added complexity for dispersed teams
- Documentation not done real-time can lack important details

- Strengths and weaknesses assisted by the [Scrum Alliance](#) and [Version One – Agile made easy](#) / as well as [Before you make the leap to Agile – Ten weaknesses of Agile](#)

Is Agile right for me?

A successful Agile methodology depends on embracing several key concepts (by both the development team and by management):

- Having continuous user involvement
- Having frequently deliverable software
- Accept changes in requirements, even late in development
- Daily communication and focus on efficiency through daily face-to-face stand-ups
- Using retrospectives to reflect and improve every iteration
- Trust that all team members and stakeholders will support the approach

Even if you're not fully doing Agile, many of these concepts can still be used to construct strong project practices, like user involvement and demonstrations. And most importantly, tailor to the task. Agile only asks you to inspect and adapt, how you do that is up to you.

- See [Water-Scrum-Fall](#) article from the Scrum Alliance

Questions

Discussion

Closeout

- Contact Alex Durkin with questions
- Final slides on Software website, Training section, <http://software.gsfc.nasa.gov>

References

- Agile Alliance – <http://www.agilealliance.org>
- Scrum Alliance – <https://www.scrumalliance.org/>
- Mountain Goat software - <https://www.mountaingoatsoftware.com/>
- Ken Schwaber's Blog - <https://kenschwaber.wordpress.com/>
- Some Agile Methods - <http://www.devx.com/architect/Article/32761/0/page/3>
- Top 10 Agile myths - <http://www.agileconnection.com/>
- Version one - <http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>
- Kanban - <https://www.atlassian.com/agile/kanban>
- About Agile - <http://www.agilealliance.org/the-alliance/>
- About Scrum - <http://scrummethodology.com>
- Scrum Roles - <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>
- State of Scrum - <https://www.scrumalliance.org/>
- Weaknesses: <http://www.cedarpointconsulting.com/delivery/articles/before-making-the-leap-to-agile>
- Principles and definitions - <https://technology.amis.nlhttps://technology.amis.nl/2008/07/01/>
- Bob Winter (2015). *Agile Performance Improvement Synergy Technology*, retrieved from <http://www.amazon.com/Agile-Performance-Improvement-Synergy-Technology/dp/1484208935>