Scrum

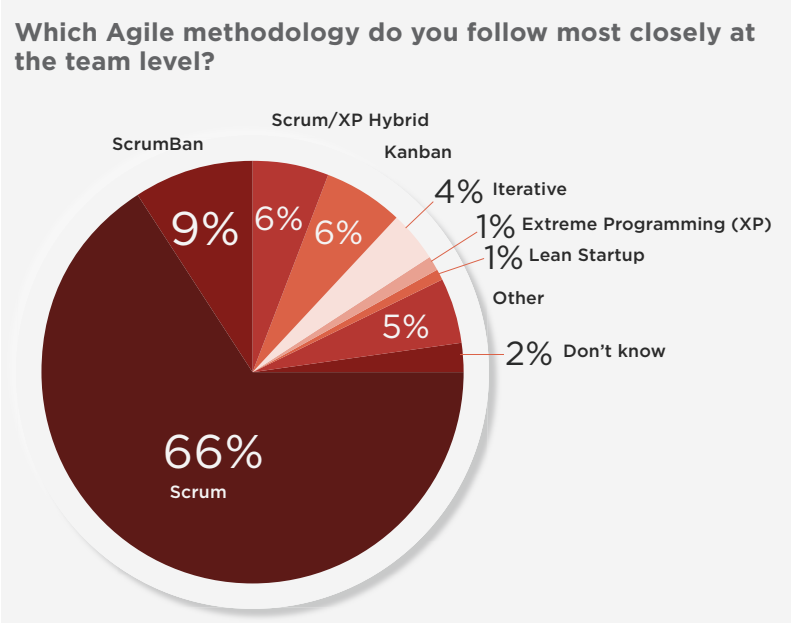
Version 1.3.1

Licensing Note

Definition

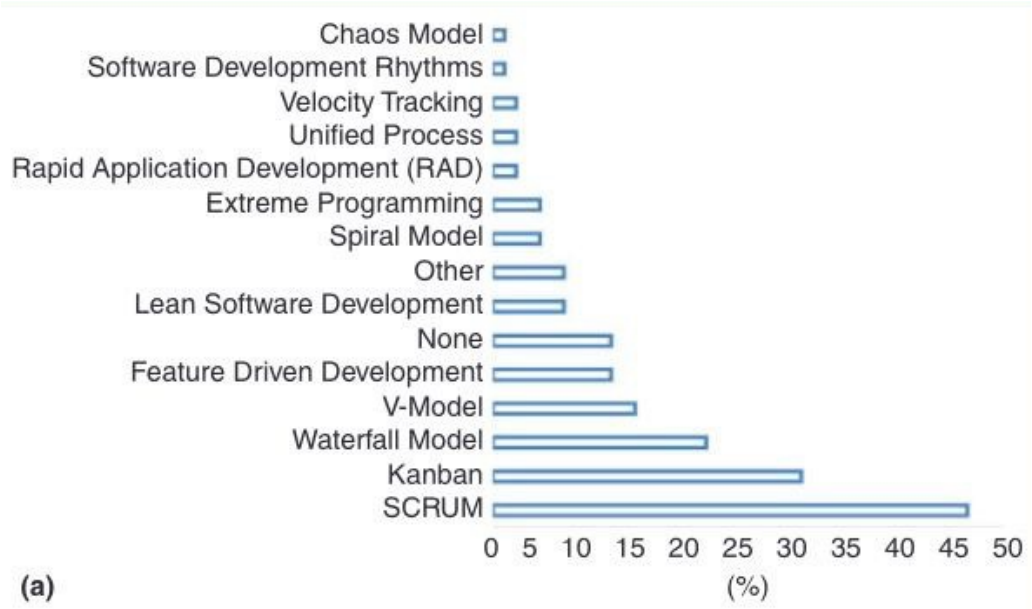
- Scrum:
  - An ordered formation of players, used to restart play in rugby
  - A team-based framework to develop complex systems and products.

Diffusion



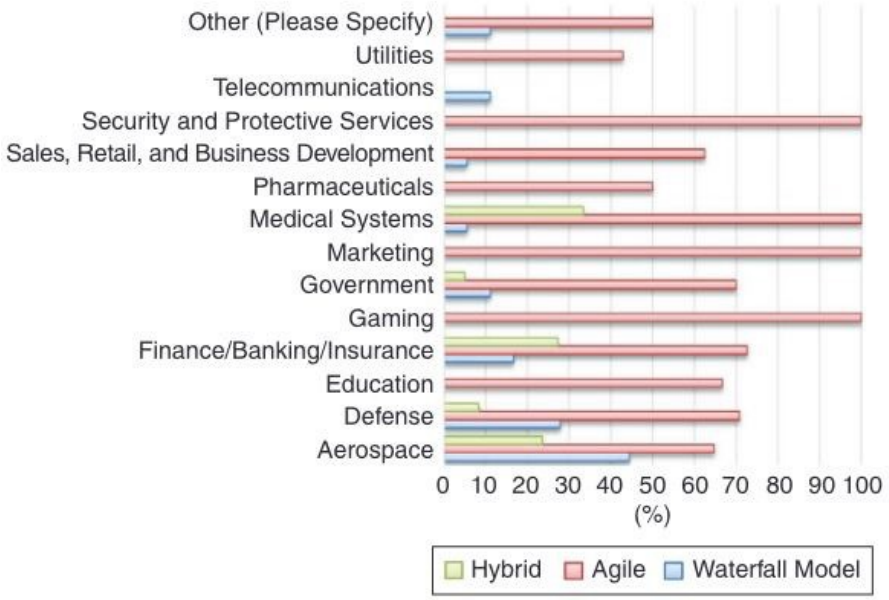
Source: 15th State of Agile Report,
available at: https://explore.digital.ai/state-of-agile/15th-state-of-agile-report

Diffusion (USA data)



M. Kassab and P. Laplante, "The Current and Evolving Landscape of Requirements Engineering in Practice," IEEE Software, vol. 39, no. 5, pp. 76–83, Sep. 2022, doi: 10.1109/MS.2022.3147692.

Diffusion (industrial sector – USA)



M. Kassab and P. Laplante, "The Current and Evolving Landscape of Requirements Engineering in Practice," IEEE Software, vol. 39, no. 5, pp. 76–83, Sep. 2022, doi: 10.1109/MS.2022.3147692.

Outline

- Principles
- Activities
- Roles
- Artifacts

Principles

Agile Development Principles

- Test as you go
- Deliver product early and often
  - Feedback
- Document as you go, only as required
- Build cross-functional teams

Theoretical foundations

- Empirical process control theory
- Iterative and incremental approach
  - to optimize predictability
  - to control risk

Scrum pillars

- Transparency
  - Process visible to whom is responsible
- Inspection
  - On artifacts and goals to detect variances
- Adaptation
  - To meet the goals

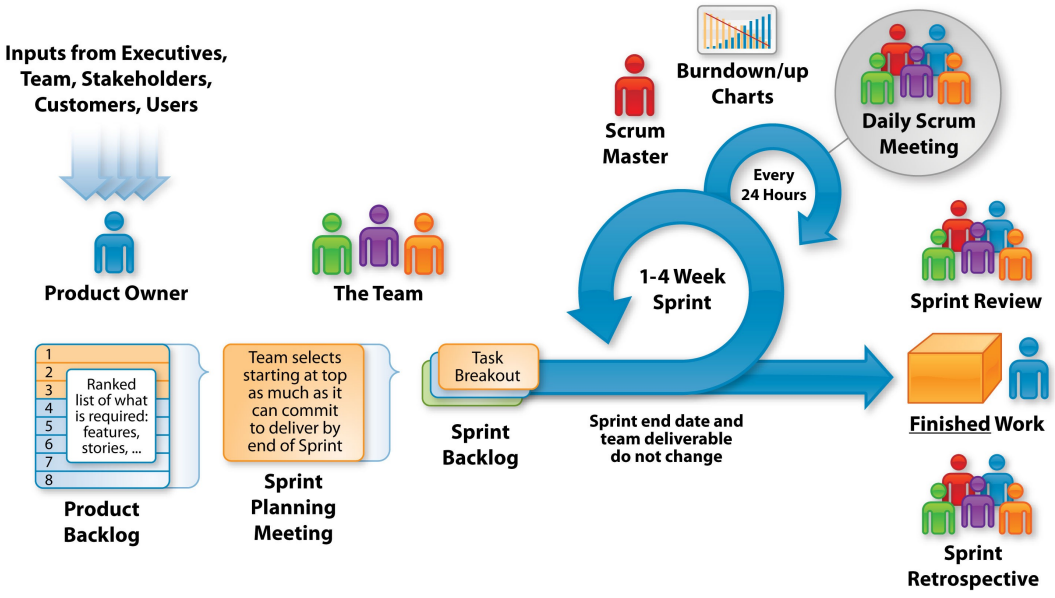
Adherence to pillars

- Four formal events:
  - Sprint planning
  - Dailiy scrum
  - Sprint review
  - Sprint retrospective

Scrum values

- Commitment
- Courage
- Focus
- Openness
- Respect

Scrum process in a nutshell



From: Rupert Dürre, «Analysis of User Stories and Effort Estimations in Agile Software Development» Technische Universität München

Roles

Scrum Roles

- Product Owner
- Scrum Master
- Team member

Product Owner

- Controls the priority order of items in the team's backlog
  - Directs team toward most valuable work
  - Work closely with the stakeholders in order to deliver the maximum business value
    - What needs to be built and when

Product Owner

- Makes sure that the needs of the customers and end-users are understood by the team
  - Requirements
  - Collaboration
    - Available to the team
- Keeper of the product vision
  - Who the product is built for
  - Why they need it
  - How they will use it

Product Owner

- Holds the vision for the product
- Represents the business
- Represents the customers
- Owns the product backlog
- Prioritizes stories
- Creates acceptance criteria for stories
- Is available to answer team members' questions

Scrum Master

- Acts as a coach
  - His goal is to produce a self-organizing team
  - Is a facilitator, NOT a boss
    - Good shepherd
    - Champion
    - Guardian
    - Scrum expert

Scrum Master

- Adjust to the team experience
  - As the team becomes self-managing the master steps back
- Remove impediments for the team
  - External
    - Escalates issues
    - Help solving resource issues (e.g. hw)
  - Internal
    - Helps the team see the problem
    - Encourages the team find a solution

Scrum Master

- Scrum expert and advisor
- Coach
- Impediment *bulldozer*
- Facilitator

Team Member

- The team is self-organizing
  - Tools
  - Techniques
  - Task assignment
- Team member estimate the effort for the features
- Team size: typically 5 to 9
  - Enough skill variety
  - Limited communication overhead

Team Member

- Cooperate to achieve the goal
- Willing to work outside his comfort zone when the team needs it
- The Scrum Team
  - P.Owner + S.Master + Team Members
- The Team
  - Scrum team except P.Owner and S.Master

Events

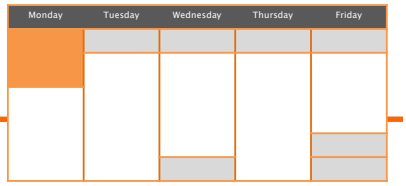
Sprint

- The basic iteration in the Scrum approach
- Produces a piece of working software to be demonstrated and reviewed at the end of the sprint
- Length
  - From 1 to 4 weeks long

Sprint organization (1 week)

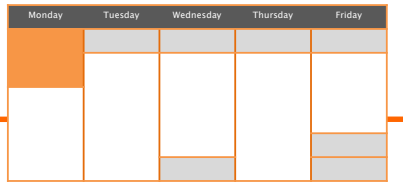
Monday	Tuesday	Wednesday	Thursday	Friday
Sprint Planning (2h)	Scrum (15m)	Scrum (15m)	Scrum (15m)	Scrum (15m)
				Sprint Review (1/2 h)
		Story Time (1h)		Retrospective (1h)

Sprint Planning



- 1. What will we do?
  - Set of committed stories
  - Product owner proposes story
  - Team members decide whether commit
    - Team's velocity / Yesterday's weather
- 2. How will we do it?
  - Decompose stories into tasks
    - May trigger renegotiation of stories
  - Max effort per task: half-day

Sprint Backlog



- List of stories
- Related tasks

- Estimation:
  - Task hours
  - Task points
  - Task count

Effort Estimation

- Essential to define a predictable schedule
- Relative vs. Absolute estimates
  - Former are much easier
  - Stories are sized in term of “points”
    - Using Fibonacci series
  - After a sprint is complete an estimate of stories per sprint i.e. “velocity” is done
    - Velocity is NOT a performance measure

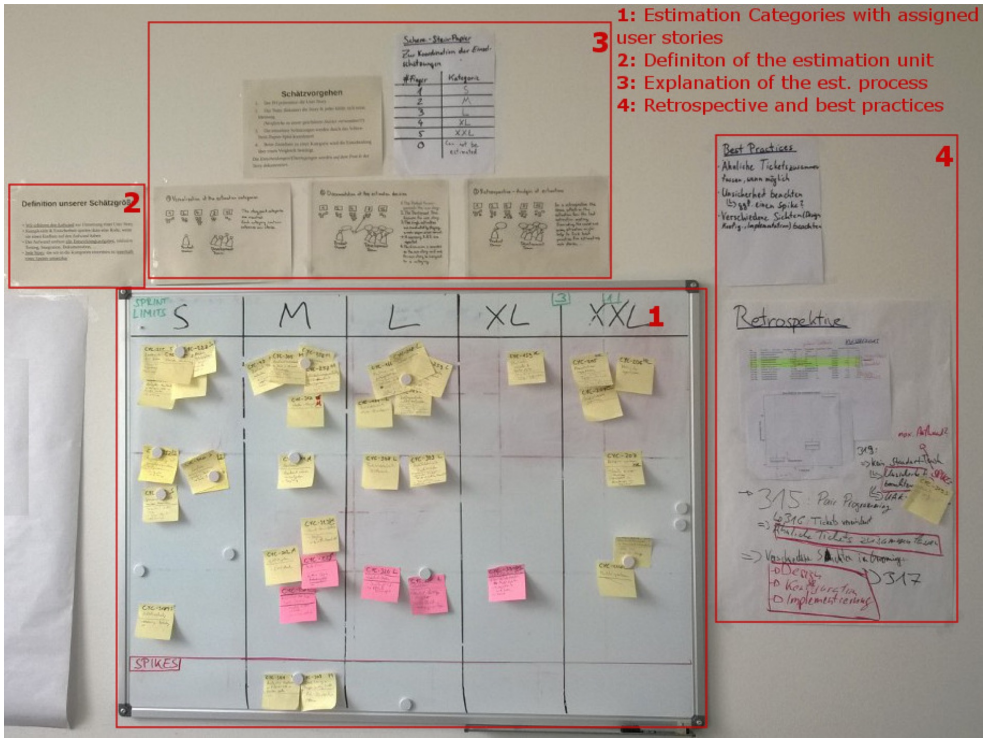
Estimation Poker

- Every team member pick a card and show it at the same time as the others
- High and low discuss the rationale
- Consensus is achieved
  - It is possible to get feedback from the product owner during the process

Estimation Poker

- Every team member pick a card and show it at the same time as the others
- Then each one explain the rationale
- Consensus is achieved with feedback from the product owner

«Estimation wall»



Rupert Dürre, op. cit..

Planning and estimation

What if planning for a piece of software was like (disciplined) shopping?

Adapted from: Kent Beck and Martin Fowler, "Planning Extreme Programming (XP)"

Analogy to shopping

- The items
- The prices
- The budget
- The constraints

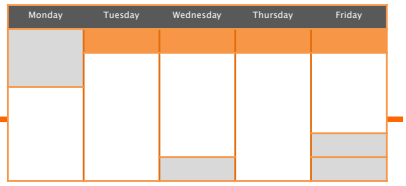
Adapted from: Kent Beck and Martin Fowler, "Planning Extreme Programming (XP)"

Analogy to shopping

- The items
  - stories
- The prices
  - estimates
- The budget
  - how many people can work on the project, and how much they can work on it
- The constraints
  - business and/or technology constraints

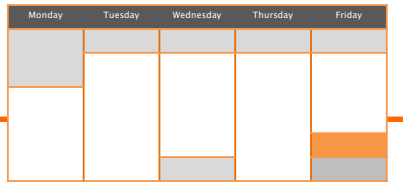
Adapted from: Kent Beck and Martin Fowler, "Planning Extreme Programming (XP)"

Daily Scrum



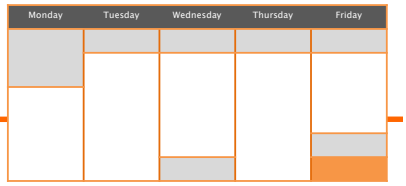
- Daily
  - Any time suitable to the team
- Small
  - Only members of the dev team
- Brief
  - Keep updated, standing, 15 min max
- Pointed
  - What has been done
  - What will be done
  - Current obstacles

Sprint review



- Goal:
  - Show off some piece of working software to stakeholders
  - Report on incomplete stories
  - Transparency
  - Record reactions of stakeholders
    - Basis for Product Owner future decisions
    - No planning at this time

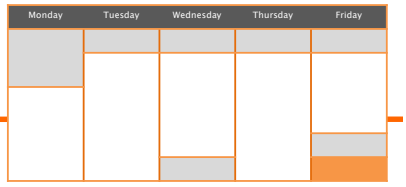
Retrospective



- Set the stage
- Gather data
  - What happened?
  - Time-line and artifacts
- Generate insights
  - Understand causes, no finger pointing
- Decide what to do
  - One team-improvement task for next sprint, under the team's control
- Close
  - Appreciation exercise

Esther Derby and Diana Larsen,
"Agile Retrospectives: Making Good Teams Great"

Retrospective

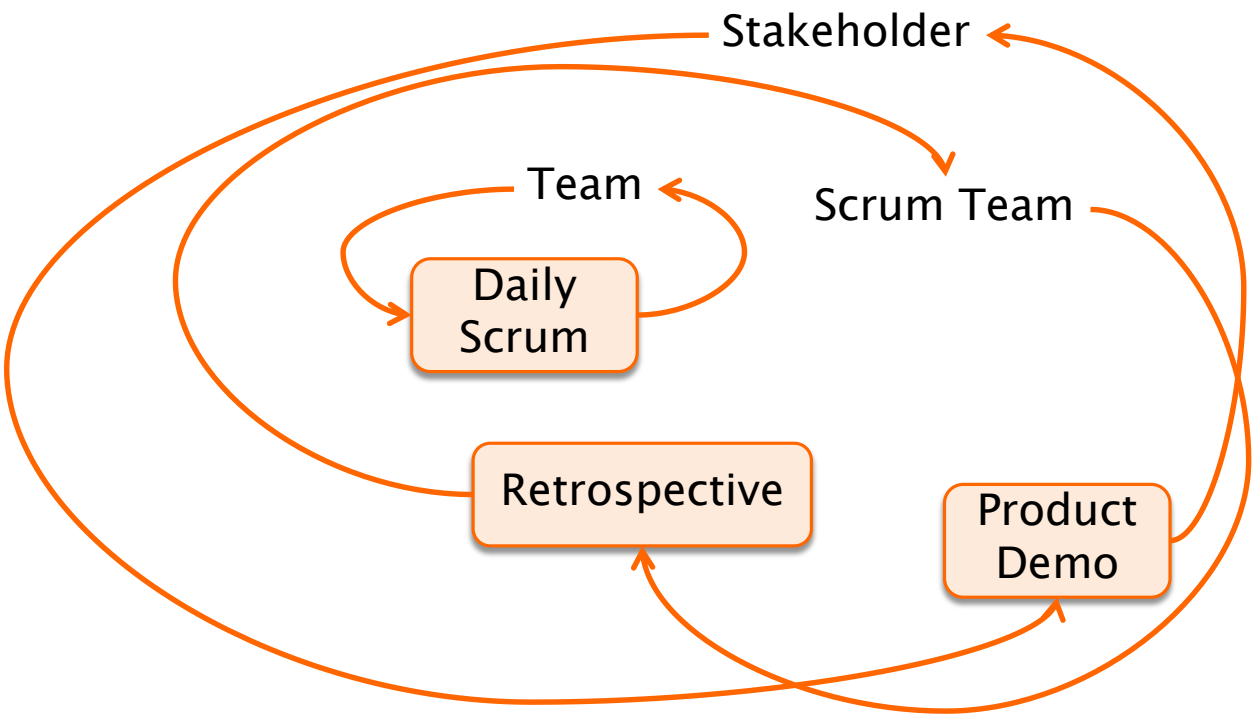


- Focus on lessons learned
  - How they can be applied to next sprint
- Goal:
  - Identify one/two things to improve
  - Define the relative action plan

"At the end of a project everyone knows so much more. Naturally we will discover decisions and actions we wish we could do over. This is wisdom to be celebrated…"

Norm Kerth,
"Project Retrospectives: A Handbook for Team Reviews"

Inspect and Adapt



Release (long term) Planning

- Fixed Scope
- Fixed Date
- Fixed Scope & Fixed Date
  - Scrum is not a 

Adding manpower to a late software project makes it later

Fred Brooks,
"The Mythical Man-Month"

Artifacts

Artifacts

- Product Backlog
  - Includes Product Backlog Items (PBI)
  - We will use User Stories as PBIs
- Sprint Backlog
  - Detailed small tasks

Product Backlog

- Anything that *will* consume team resources
- Ever changing
- Strict ordering of the **stories**
- Property of the Product Owner
- Can be a spreadsheet or a wall

Example

Story	Story points
As an *unregistered user* I want *to create a new account* So that *I can buy items*	3
As a *customer* I want *to use my Card* So that *I can buy the items*	8
As a *user* I want *to add items* *to my wish list*	5
	TOT 16

Forecasted velocity: 12 pt per sprint
Sprint length: 2 weeks

Product Backlog

- Product Backlog Item / Story
  - Independent
  - Negotiable
  - Valuable
  - Estimable
  - Small
  - Testable

"INVEST"

Properties of a good Product Backlog

- "DEEP"
  - Detailed appropriately
  - Estimated
  - Emergent
  - Prioritized

Sprint Backlog

- Committed stories
  - Relative tasks (may change during sprint)
- Additional tasks, for example:
  - Team improvement
  - Research work
  - Performance and security requirements
  - Bug fixing
- Frozen at the end of sprint planning
- Property of the team

Task board – example

To do	In progress	Done
Story #12	Story #4	Story #28
Story #7	Story #2	Story #1
Story #33		Story #3
Story #34		
Story #22		

Definition of Done

- Team shared definition
- Typically: shippable
  - Code review
  - Design review
  - Refactoring
  - Performance testing
  - Unit tests passed

Overview of main artifacts, roles and activities

		Participants				Artifacts & Activities	
		Product owner	Scrum Master	Devel. Team	Stakeholder	Used Artifacts	Activities
Events	Sprint Planning	X	X	X	(opt)	Product backlog Sprint backlog	Definition of a sprint goal Selection of user stories Definition of tasks
	Daily Scrum	(opt)	(opt)	X	(opt)	Sprint backlog Burn-Down/up chart	Update each other
	Story time	(opt)	X	X	(opt)	Product backlog	Backlog refinement Estimation of future stories
	Sprint review	X	X	X	X	Software Sprint backlog	Show piece of working software Collect feedback
	Sprint Retrospective	(opt)	X	X	(opt)	What is deemed useful/necessary	Examination of the last sprint in order to identify possible improvements

References

- K. Schwaber and J. Sutherland,The Scrum Guide, 2017 http://www.scrumguides.org
- K.Schwaber, M.Beedle.
  Agile Software Development with Scrum, Prentice-Hall, 2001
- C.Sims, H.L.Johnson.
  “The Elements of Scrum”
  DYMAXICON, 2011.
- Jeff Sutherland's Scrum Handbook
- H. Takeuchi and I. Nonaka,
  "The New New Product Development Game," Harvard Business Review, 1986.