

Agile Development Methods

HUSZERL Gábor
huszerl@mit.bme.hu



Méréstechnika és
Információs Rendszerek
Tanszék



**Critical Systems
Research Group**

Learning Outcomes

- At the end of the lecture the students are expected to be able to
- (K1) understand the basic principles of agile software development,
- (K1) understand and identify the main elements and functions of the Kanban method and the Scrum, and the roles and steps defined by the Scrum.

Further Topics of the Subject

I. Software development practices

Steps of the development

Version controlling

Requirements management

Planning and architecture

High quality source code

Testing and test development

II. Modelling

Why to model, what to model?

Unified Modeling Language

Modelling languages

III. Processes and projects

Methods

Project management

Measurement and analysis

Reminder: Basic Principles of Agile

- Individuals and communication \succ predefined processes
- Working software \succ comprehensive documentation
- Real cooperation with the customer \succ contract negotiation
- Adapting to changes \succ following a plan
- Short, iterative steps
- Small but “multidisciplinary” teams
- **Methods**
 - Kanban
 - Scrum

Kanban Method

Kanban

- Origin: production technology (Meaning: sign board)
- [Toyota Production System](#)
 - Taiichi Ohno, 1943, shop floor control
- (Original) Motivation:
 - Too many parts and semi-finished products
 - Too many products between the phases
 - Overstocking (raw material, parts, products)
- Nowadays: Just In Time
 - To produce only what is needed, when it is needed
- Many [applications](#) in the production

Kanban General Rules

Never pass on defective products

Take only what is needed

Produce the exact quantity required

Level the production

Fine-tune production, if necessary

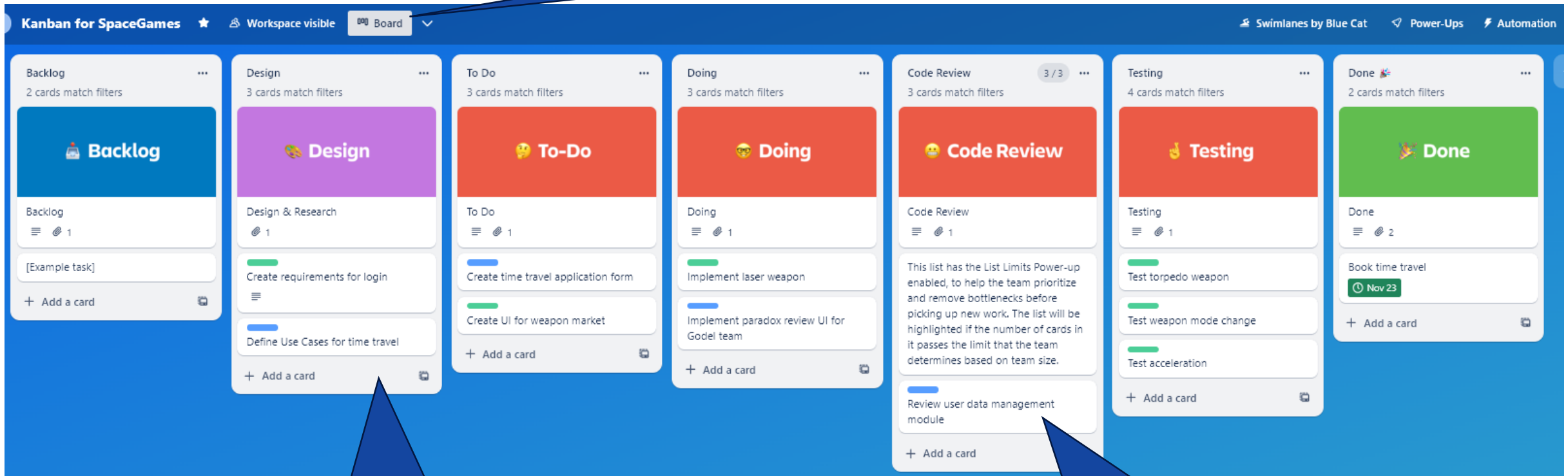
Stabilize and rationalize the process

Use Cases for Kanban

- Developing a new product or service + product support
- Supporting an existing product
- Developing a new product (without active users)

Kanban Board Example

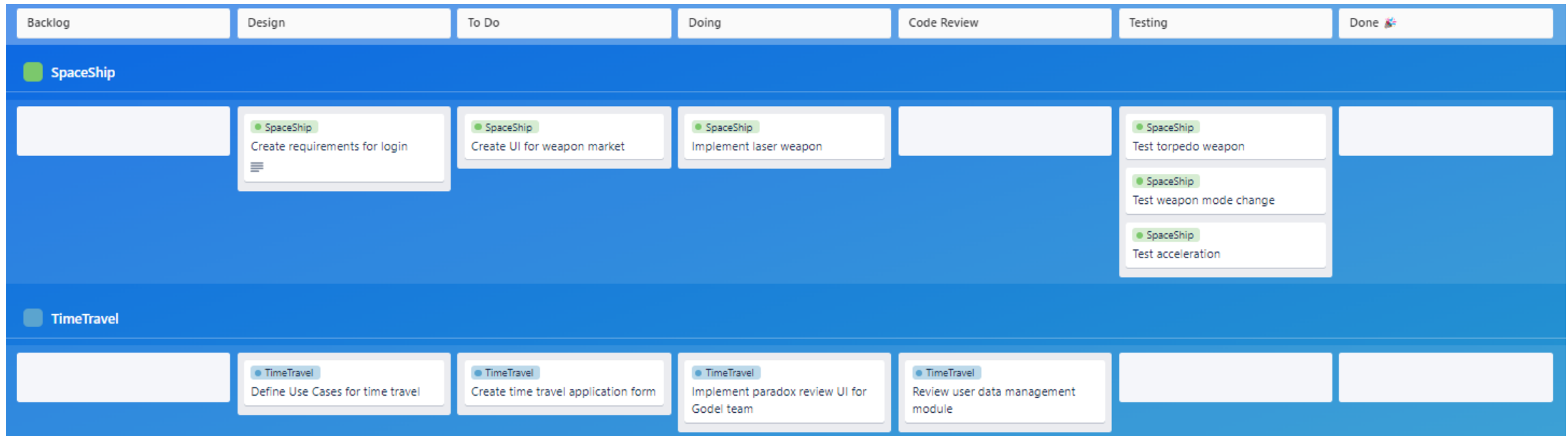
Board: representing one team



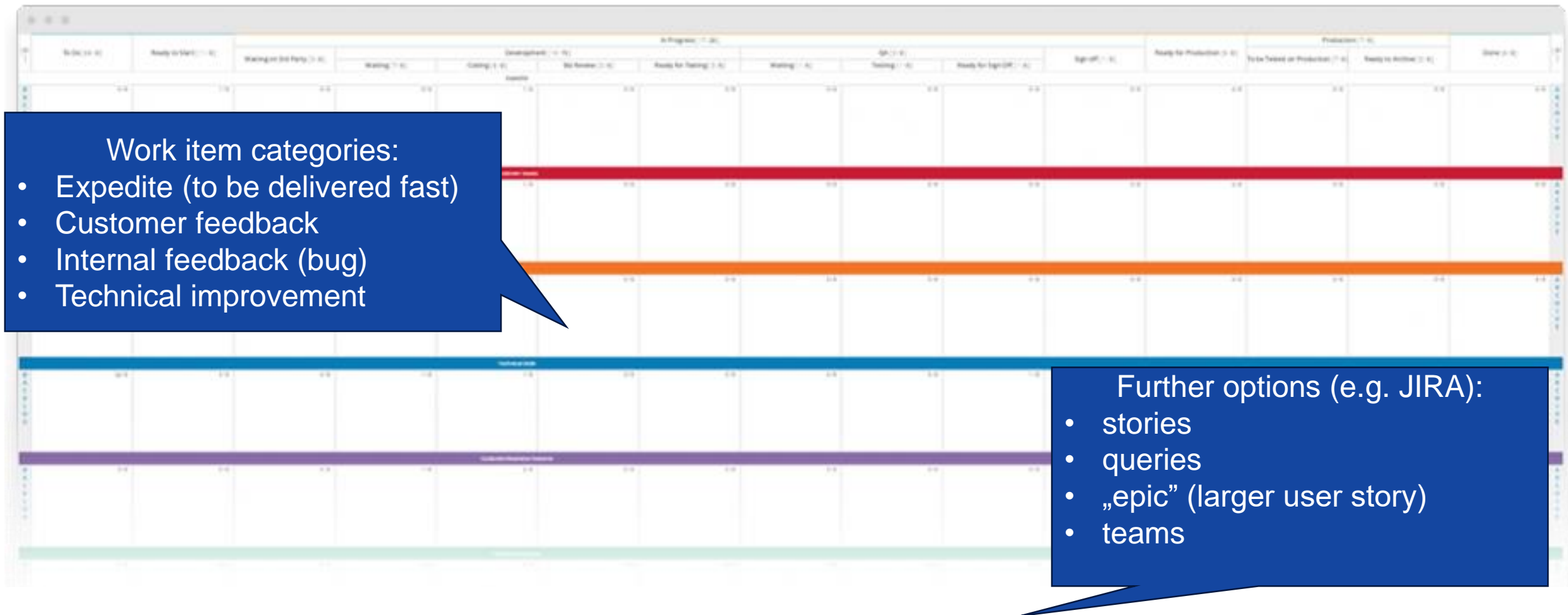
List: main states of the life cycle

Card: a single work item

Kanban: Swimlane = Project



Kanban: Swimlane = Category



Kanban: Automating with GitHub

OctoArcade Invaders

Planning | Sprint Board | Alpha | Roadmap | My work | Features | Priority | By person | Status Board | By status | By Sprint | Done

Filter by keyword or by field

Not Started 19 Estimate: 37

- planning-tracking-demo #810
Beta go-no-go meeting
- planning-tracking-demo #800
Save score across levels
- planning-tracking-demo #784
Interviews with media outlets
epic

Planning 19 Estimate: 109

- planning-tracking-demo #823
Updates and bug fixes to engine from Beta
bug demo
- planning-tracking-demo #824
Beta signup page
need help
- planning-tracking-demo #806
[Tracking] Upsell / Growth experience
backlog feature
- planning-tracking-demo #818
Account subscription design
- planning-tracking-demo #828
Acquire domain for launch
- planning-tracking-demo #832
Final creative shots from game
- planning-tracking-demo #829

Building 8 Estimate: 40

- planning-tracking-demo #1160
Update documentation
- planning-tracking-demo #814
Updates to collision logic
enhancement
- planning-tracking-demo #816
Free and paid levels
need help
- planning-tracking-demo #831
Documentation and Support
need help
- planning-tracking-demo #821
Updates to alien, beam, bomb and cannon sprites
#370
- planning-tracking-demo #802
Updates to velocity of the ship and alien movements

Review 5 Estimate: 17

- planning-tracking-demo #822
Hero site - Development
#12 #1160 in-review task urgent
- planning-tracking-demo #808
General bug fixes from Alpha feedback
#992
- planning-tracking-demo #1151
Design new launch screen
#374 web
- planning-tracking-demo #793
Polished alien, beam, and cannon sprite files
- planning-tracking-demo #1101
[Tracking] Integrate payments system
backlog feature

Work items “move” automatically
Triggered by

- comments
- pull requests
- issue changes

Draft
Prevent the Konami code from bringing down all of GitHub

+ Add item

Further Tools

- Checklist
 - Partition of a work item
- Integration
 - With the output of continuous integration or testing
 - With teamwork software, calendar, messaging (e.g. Slack)
- Limiting the length of the lists (number of WIPs)
- Examples
 - Trello
 - JIRA
 - Monday.com
 - Miro
 - MS Project

SCRUM

Framework for agile development

Sutherland, Schwaber (1996)

Basic Philosophy

Transparency

- Everyone knows all the current information

Adaptation

- Tactics may be changed

Basis of Decisions

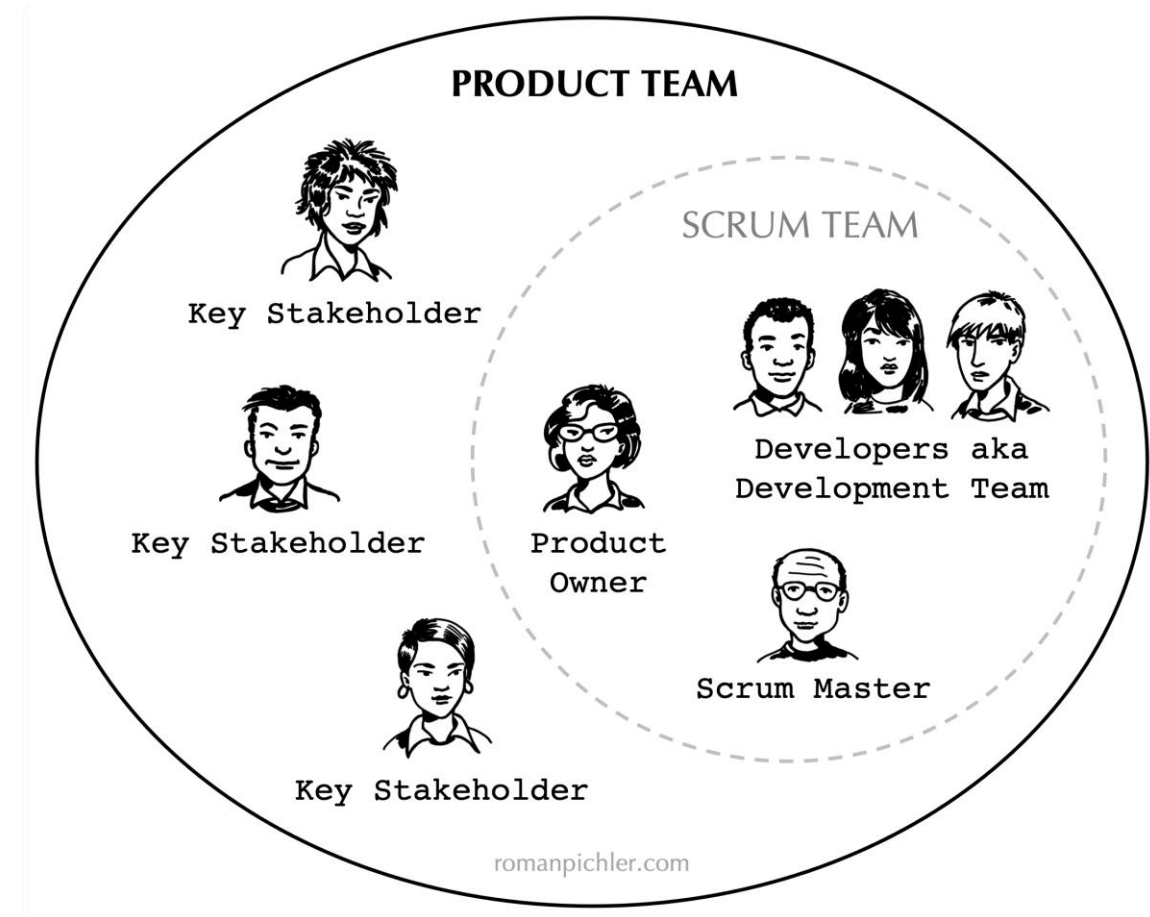
- observation
- experience
- experimenting

Observation

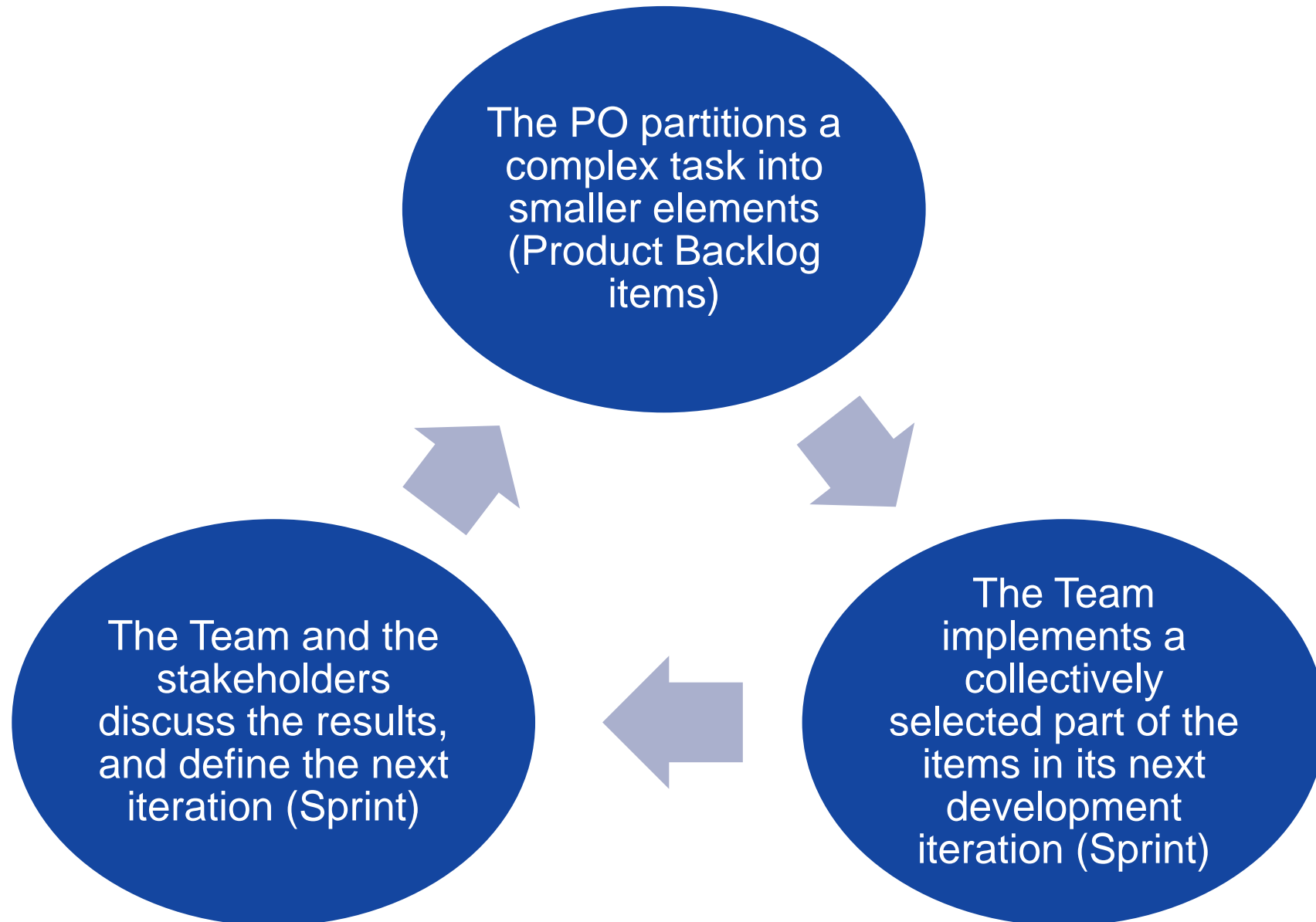
- Continuously following the results of the work

Roles

- Participants (pigs)
 - Product Owner
 - Scrum Master
 - Development Team: 6-9 persons
- Stakeholders (chickens)
 - Management
 - Customers
 - Users
 - Technical experts
- See <https://scrumguides.org/scrum-guide.html>



Steps



Responsibilities of the Product Owner (Tactics)

Identifying the project goals and the product properties (product goal)

- What should be implemented and what should not

Managing the tasks (Product Backlog)

- Creating items (or accepting suggestions)
- Having transparent and understandable tasks

Prioritizing, ordering

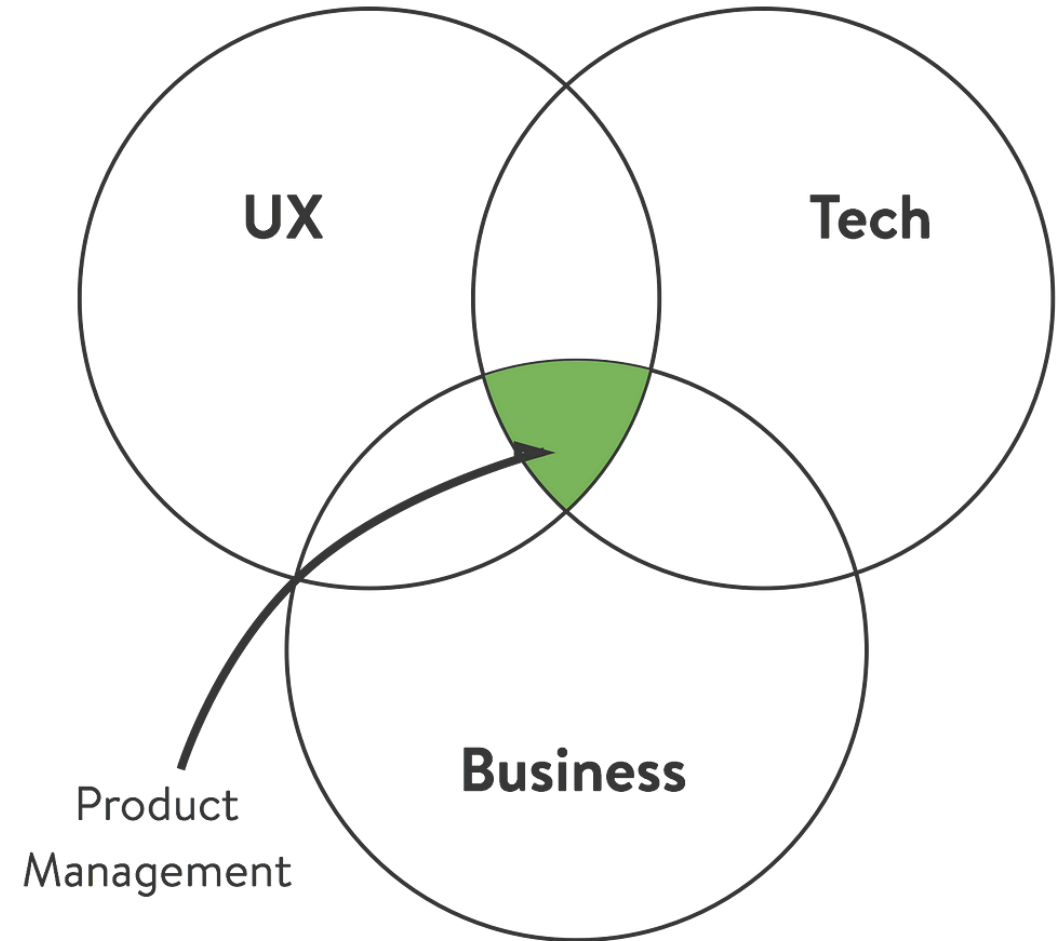
- What is worth for the team to deal with?
- What produces the highest earnings?
- What should be in the next release?

Teamwork

- Involving the team members into the decisions
- **The PO has the final responsibility**
- It is NOT his task to determine, who should finish what in how much time

Responsibilities of the Product Owner (Strategy)

- Preparing a product strategy
- Preparing a product roadmap
- Preparing financial forecasts
- Measuring performance (business and technical)
- Communicating with the stakeholders



<https://blackboxofpm.com/mvpm-minimum-viable-product-manager-e1aeb8dd421>

Scrum Master

Supporting the team

- Education of the individual team members (autonomy, communication)
- Keeping the team focused
- Organising and chairing events

Supporting the work of the PO

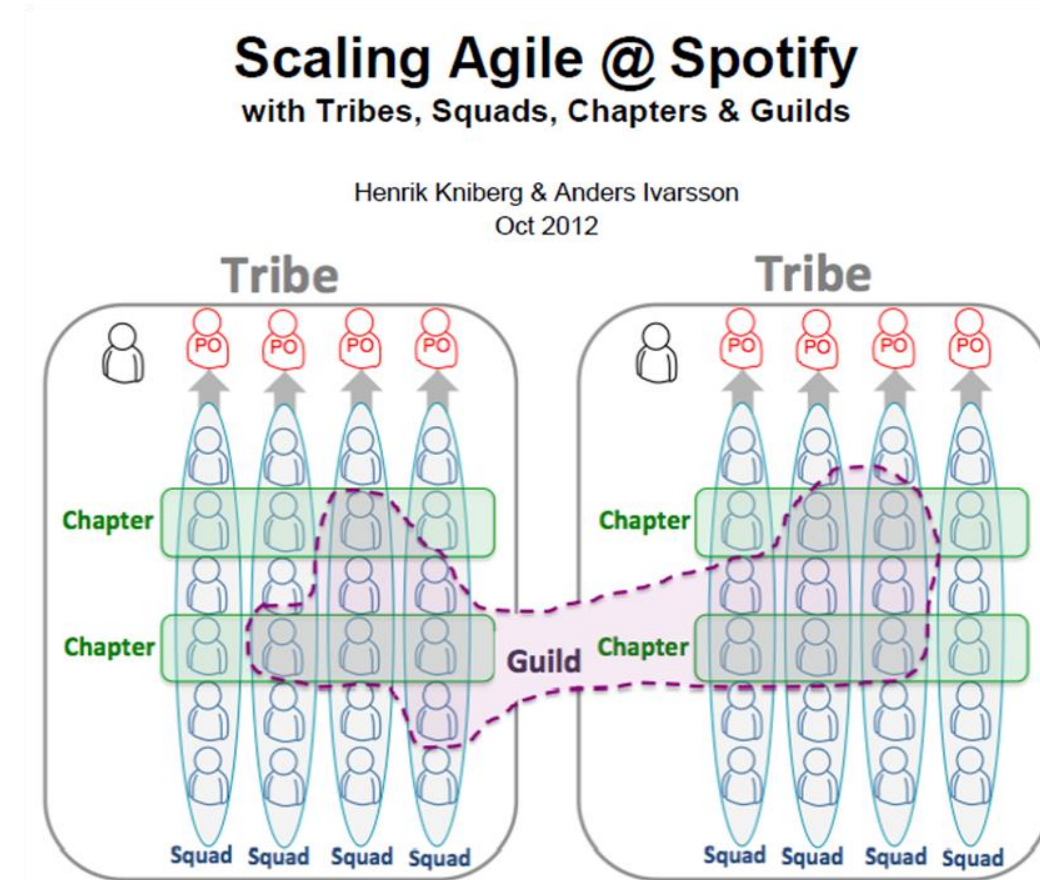
- Defining an efficient project goal
- Efficient backlog management
- Testing the product in a complex environment
- Supporting the communication with the stakeholders

Organisational tasks

- Organising education, trainings, coaching
- Overcoming communication obstacles

Organisational Architecture

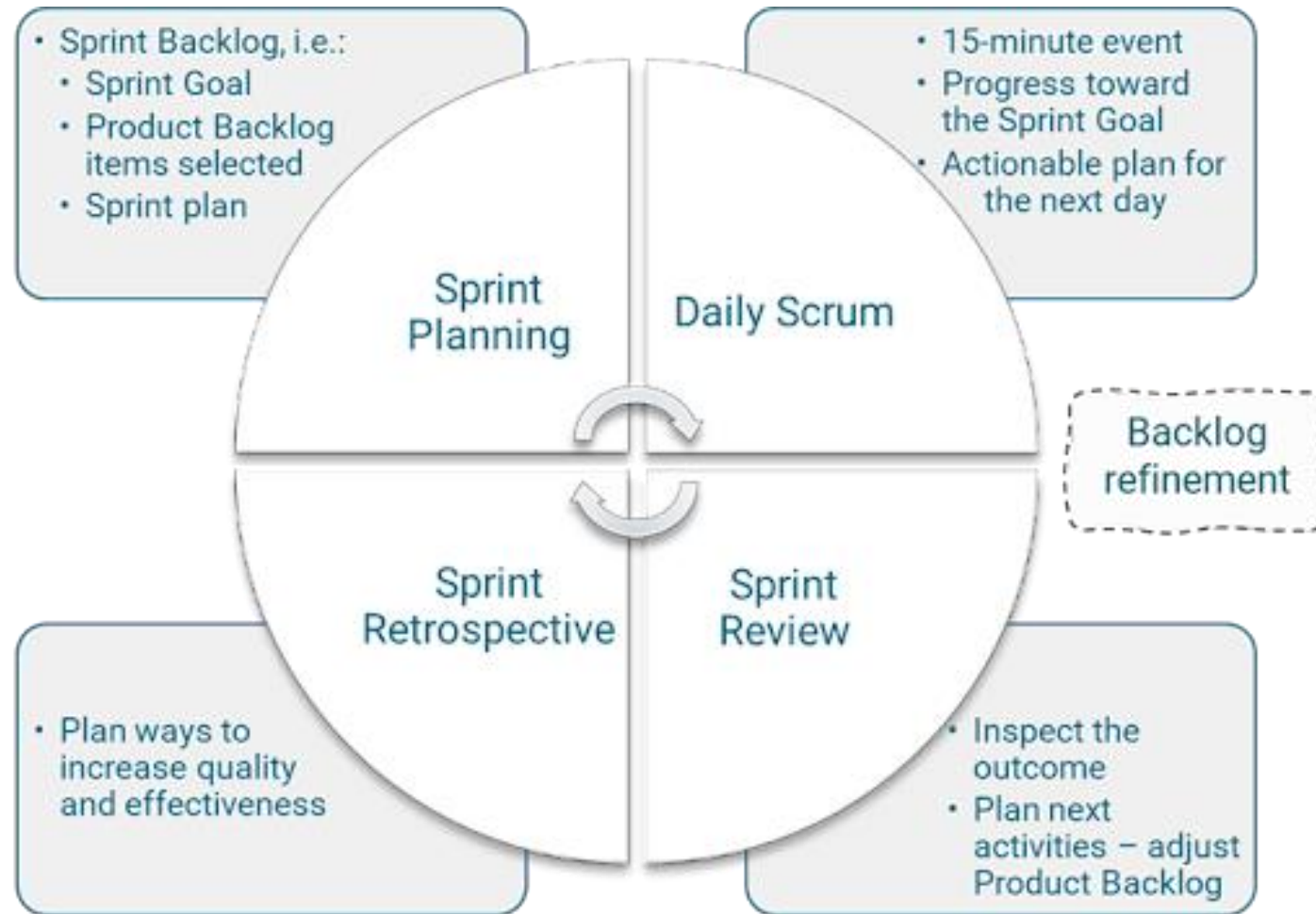
- Squad
 - A single project team
 - Design, development, testing, releases
 - Not only developers!
- Chapter
 - Same function in a team
 - E.g. front-end developers
- Tribe
 - Several teams working on similar projects
- Guild
 - Same interests
 - Knowledge sharing, no specific tasks



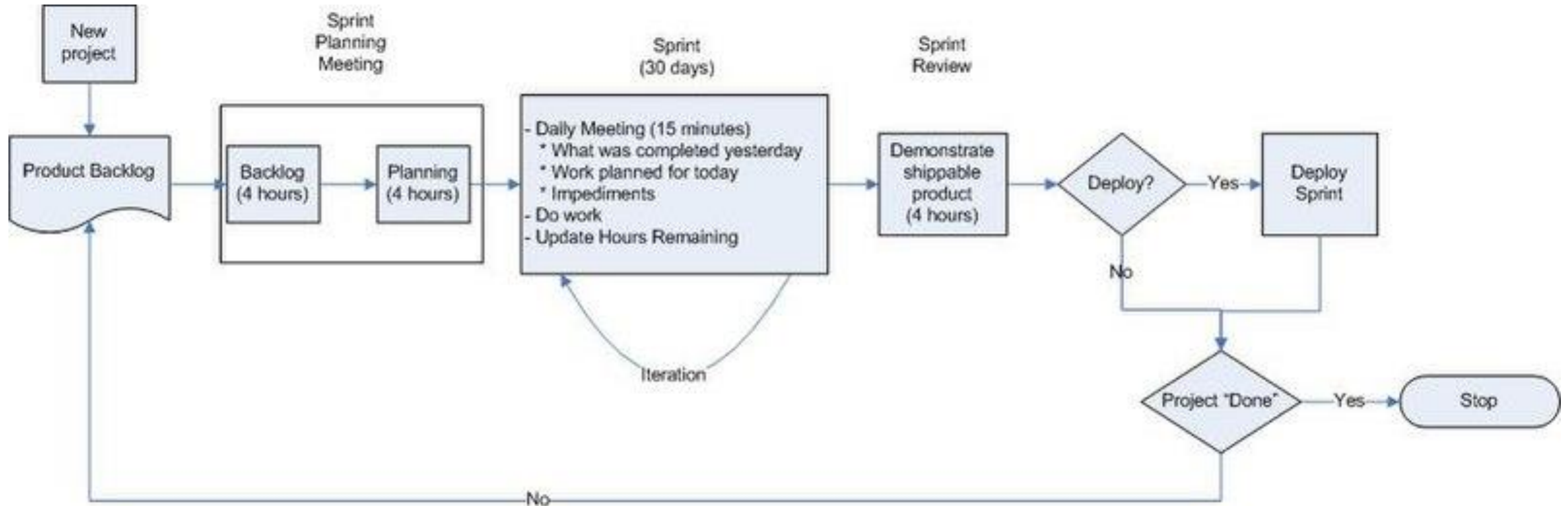
Standup

- Daily regular meeting with each project member
- 3 questions
 - What have you done since the last meeting?
 - What will you do now? (until the next meeting)
 - Are there any obstacles?
- Max. 15 minutes (standing)
 - Details should be discussed separately

Scrum Events



Scrum Process



Story Points

- Elementary functions
 - “I must be able to add a new element to the Favorites”
 - “The expected balance for the end of the month must be presented”
- Measuring complexity
 - Fibonacci series as a scale (1, 2, 3, 5, 8, 13, 21, 34, 55...)
 - In agile methods, typically (1, 2, 3, 5, 8, 13, 20, 40, ~~100~~)
- Determining
 - Complexity
 - Risks
 - Previous similar experiences (reference)
 - „Planning poker“, Storypoint matrix

Example

Because of the risks,
the effort is not linear
to the time

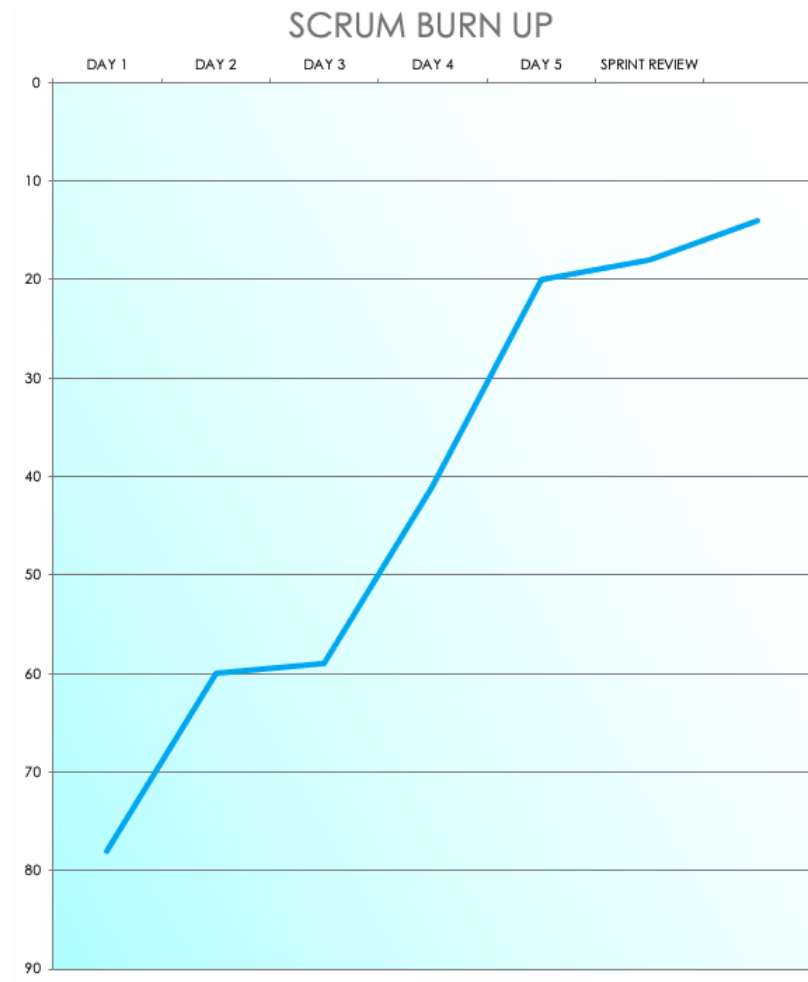
Story point	Amount of effort required	Amount of time required	Task complexity	Task risk or uncertainty
1	Minimum effort	A few minutes	Little complexity	
2	Minimum effort	A few hours	Little complexity	
3	Mild effort	A day	Low complexity	
5	Moderate effort	A few days	Medium complexity	
8	Severe effort	A week	Medium complexity	
13	Maximum effort	A month	High complexity	



<https://www.7pace.com/blog/story-points>

Measurement: Burn-up Chart

BACKLOG TASK & ID	STORY POINTS	ASSIGNED TO	STATUS	ORIGINAL ESTIMATE	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	SPRINT REVIEW
User Story #1	8									
Task			Complete	7	5	3	0	0	0	0
Task			In Progress	3	1	1	5	0	1	0
Task			Not Started	1	0.5	0	3	0	0	0
Task			Needs Review	0.5	1	2	3	1	0	0
User Story #2	1									
Task			Approved	3	3	0.5	0.5	0	0	2
Task			Overdue	3	5	5	1	1	1	0
Task			On Hold	2	2	5	0	1	0	1
Task				5	5	9	5	1	0	1
User Story #3	5									
Task				8	6	0	0	0	0	0
Task				3	1	3	3	3	0	0
Task				1.5	1	0.5	0.5	1	1	0
Task				2	0.5	0	0	0	0	3
User Story #4	8									
Task				9	4	2	2	1	1	0
Task				6	6	3	3	3	1	1
Task				6	2	8	8	1	0	1
Task				0.5	0.5	0.5	0.5	0	0	0
User Story #5	3									
Task				2	1	1	1	0.5	1	1
Task				6	6	6	0.5	3	9	0
Task				9	9	9	4	3	3	3
Task				0.5	0.5	0.5	1	0.5	0	1
TOTAL				78	60	59	41	20	18	14



<https://www.smartsheet.com/content/burn-up-chart-templates>

Backlogs

Product backlog

- Everything needed to reach the project goal
- “Owned” by the PO
- Requirements, bugs, functions
- Everyone may suggest
- „Product Backlog Refinement meeting”
- Estimation: user story
- Measurement: release burndown

Sprint backlog

- Everything needed for the current Sprint
- “Owned” by the development team
- Not changing if the Sprint has already started
- Overviewed in each daily stand-up
- „Sprint Planning meeting”
- Estimation: elementary development steps
- Measurement: sprint burndown

Example

- User story
 - Searching for friends in the address list
- Tasks
 - Create a user interface for the search
 - Design the search conditions
 - Design the communication (data formats, sequences, protocols, ...)
 - Develop the server API
 - Develop the test cases
 - Implement the user side cache
 - ...

Definition of Done

- Together with the acceptance tests
- “Done”
 - Development is finished
 - Tests are executed
 - Each acceptance requirement is satisfied (acceptance test successful)
 - Other requirements: test coverage, code review, ...
- Process of accepting
 - Sprint review meeting
 - Right & Responsibility: Product Owner
- Goal
 - “Done” can be released to the customer

Multi-level DoD

Agile - Definition of Done Manifesto

User Story: Definition of Done Checklist	Sprint: Definition of Done Checklist	Release: Definition of Done Checklist
✓ Code builds with no error	✓ Satisfied DoD for each user story in the sprint	✓ Satisfied DoD for each sprint in the release
✓ Unit testing is complete	✓ Marketing feedback is implemented	✓ Production environment is ready
✓ Code review is complete	✓ Legal / compliance review is complete	✓ CI / CD verified and working
✓ Localization & translation is complete	✓ User help guide created or updated	✓ User help guide localized
✓ Localization testing passed	✓ Training video created or updated	✓ Training video localized
✓ Browser and / or device compatibility testing is complete	✓ Refactoring is complete	✓ Rollback process is documented
✓ Regression testing is complete	✓ Configuration or build changes documented	✓ Smoke testing scenarios are ready
✓ Automation tests are written and passed	✓ Performance testing is complete	✓ Customer Support team is trained
✓ Acceptance criteria is met	✓ Security testing is complete	✓ Release communications are sent
✓ Signed off by Product Owner	✓ Sprint marked as ready for deployment	✓ All stakeholders signed off for the release

Source: Naren's Scratch Pad

<https://medium.com/@narensratchpad/agile-definition-of-done-checklist-for-user-story-sprint-and-release-52b7ad8d49de>

Starting a Project

Agile Inception Deck

Why are we
here?

Elevator pitch

What will it
cost?

NOT list

Get to know
your
neighbours!

Product box

What can we
add? (quality)

What makes
us
sleepless?

Estimate it!

What will be
the (technical)
solution?

Examples:
<https://miro.com/app/board/uXjVNLpfh5o=/#tpicker-content>
[https://miro.com/miroverse/project-kick-off-10-questions/
Mobil application](https://miro.com/miroverse/project-kick-off-10-questions/Mobil_application)

Example: Elevator Pitch



The Elevator Pitch

- For [individual construction teams]
- who [need track road access on the construction site],
- the [Road Closure System (RCS)]
- is a [safety communication tool],
- that [informs crews when roads will be closed].
- Unlike [the current paper-based system]
- our product [is web based and can be accessed
- by all contractors anywhere anytime].

<https://agilewarrior.wordpress.com/2010/11/06/the-agile-inception-deck/>

The Role of Scrum and Kanban

Comparison

- Kanban
 - Concentrates on the amount of work, and on the state of the tasks
- Scrum
 - Project management, roles, communication
- Can be combined
 - The Scrum tasks may “flow” on the board

What Makes Agility Different?

- A general approach
- It is not about methods
 - Originally created **to oppose** traditional inflexible methods
- It is not about formal frameworks
 - 10 minutes, 5 metres, ...
- Organisation of work \succ process + ceremonies
- Communication among teams without reorganisation

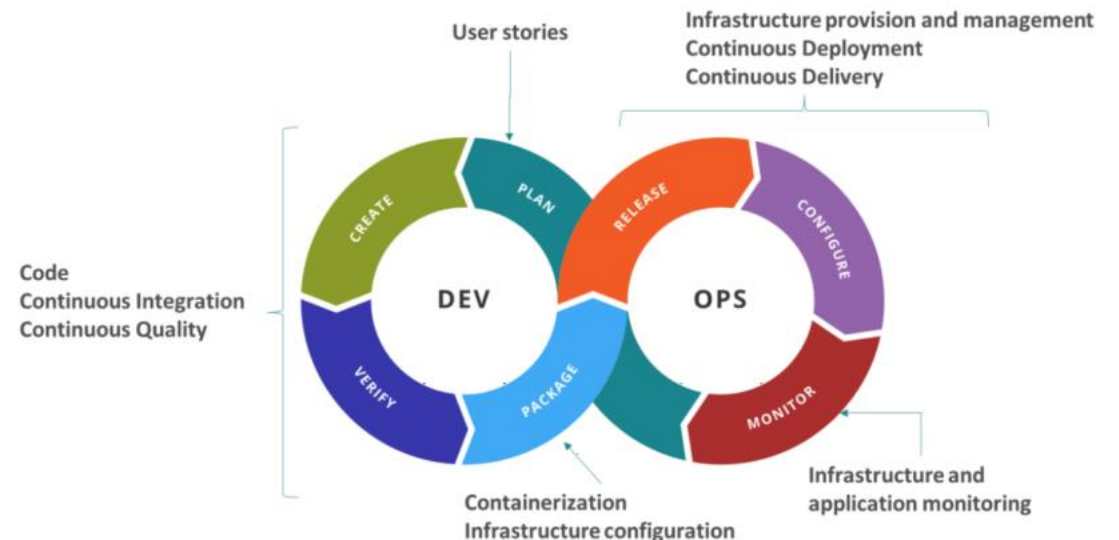
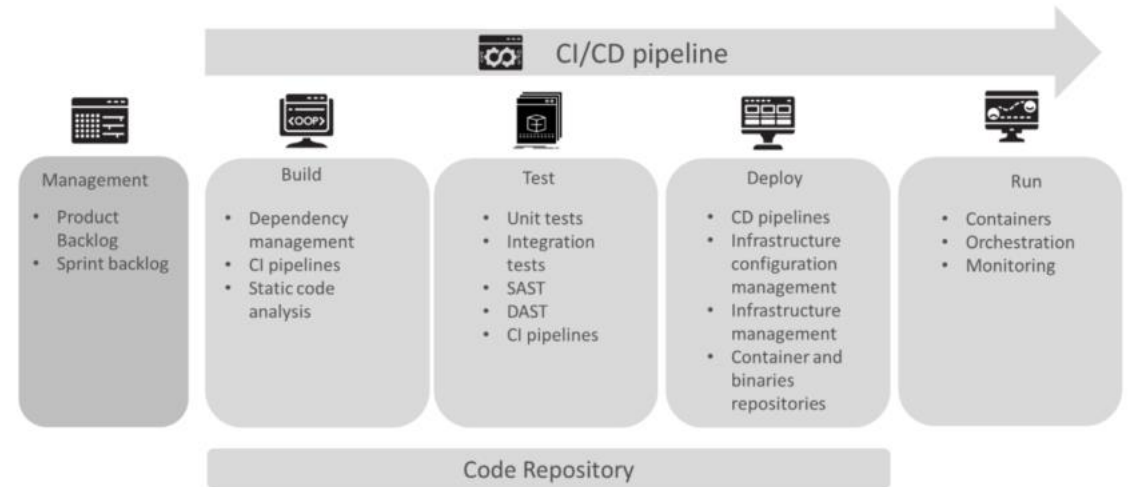
Large-scale Agile Projects

Example

The Development Team(s), led by their Scrum Master, and the Product Owner work on a common **Product Backlog** of tasks to be performed and features to be implemented.

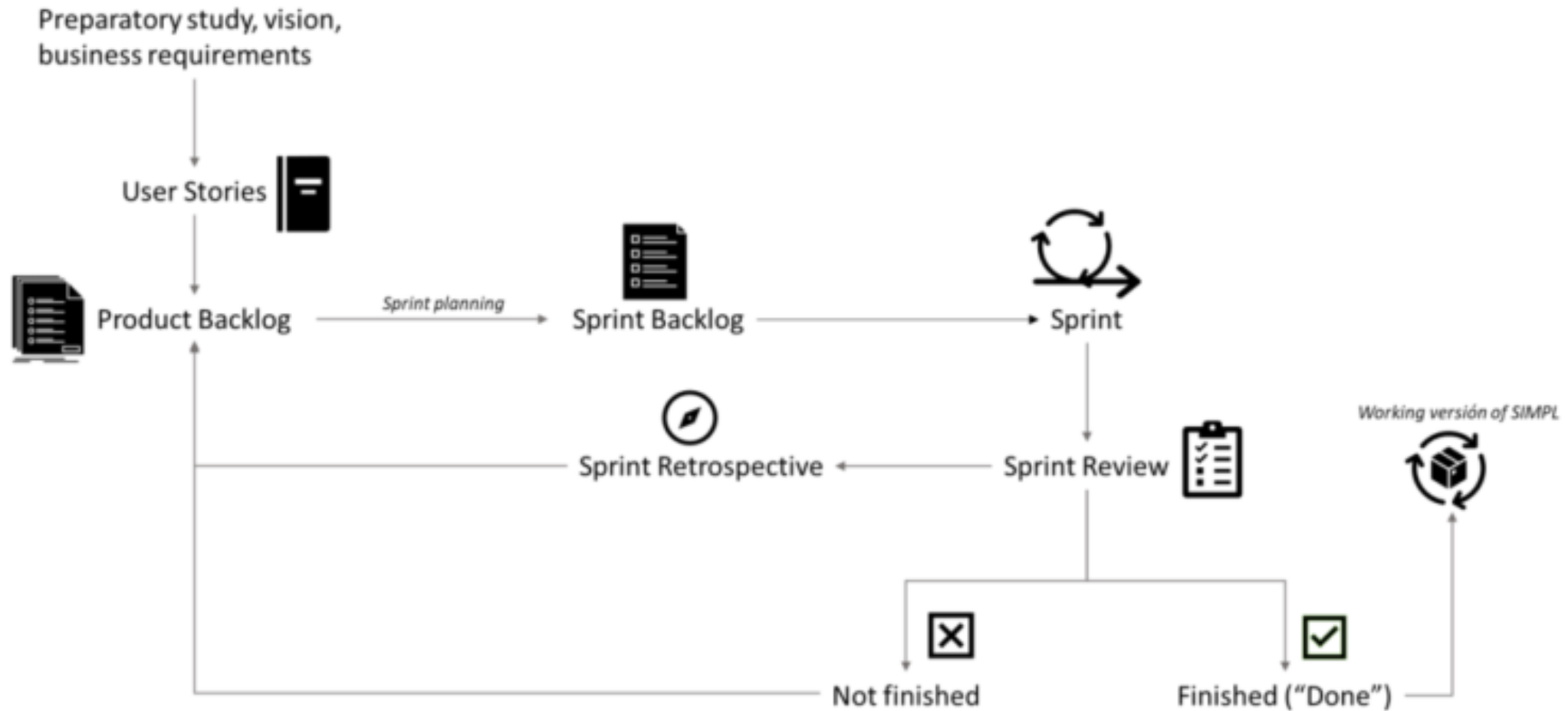
User Stories are objectives defined by the Contracting Authority, broken down into technical Tasks implemented by the Contractor.

For the **scrum master**, an experience of **6 years** in projects of similar complexity is expected. The person should be Scrum Master certified.



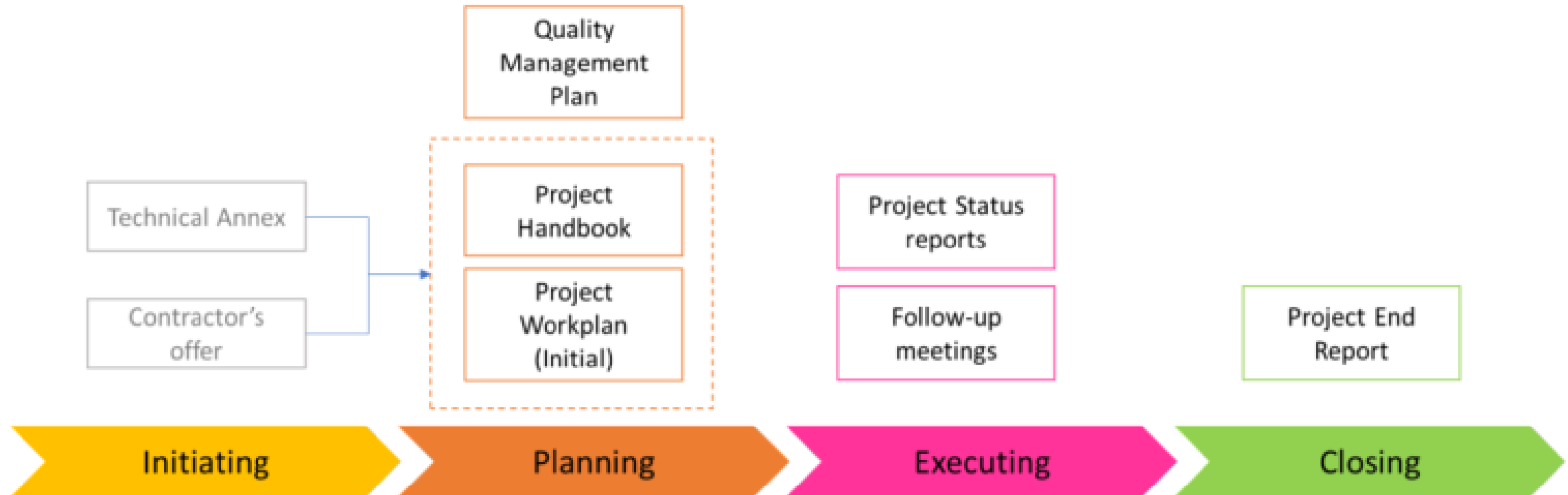
<https://etendering.ted.europa.eu/cft/cft-document.html?docId=139712>

Example – Required Agile Method



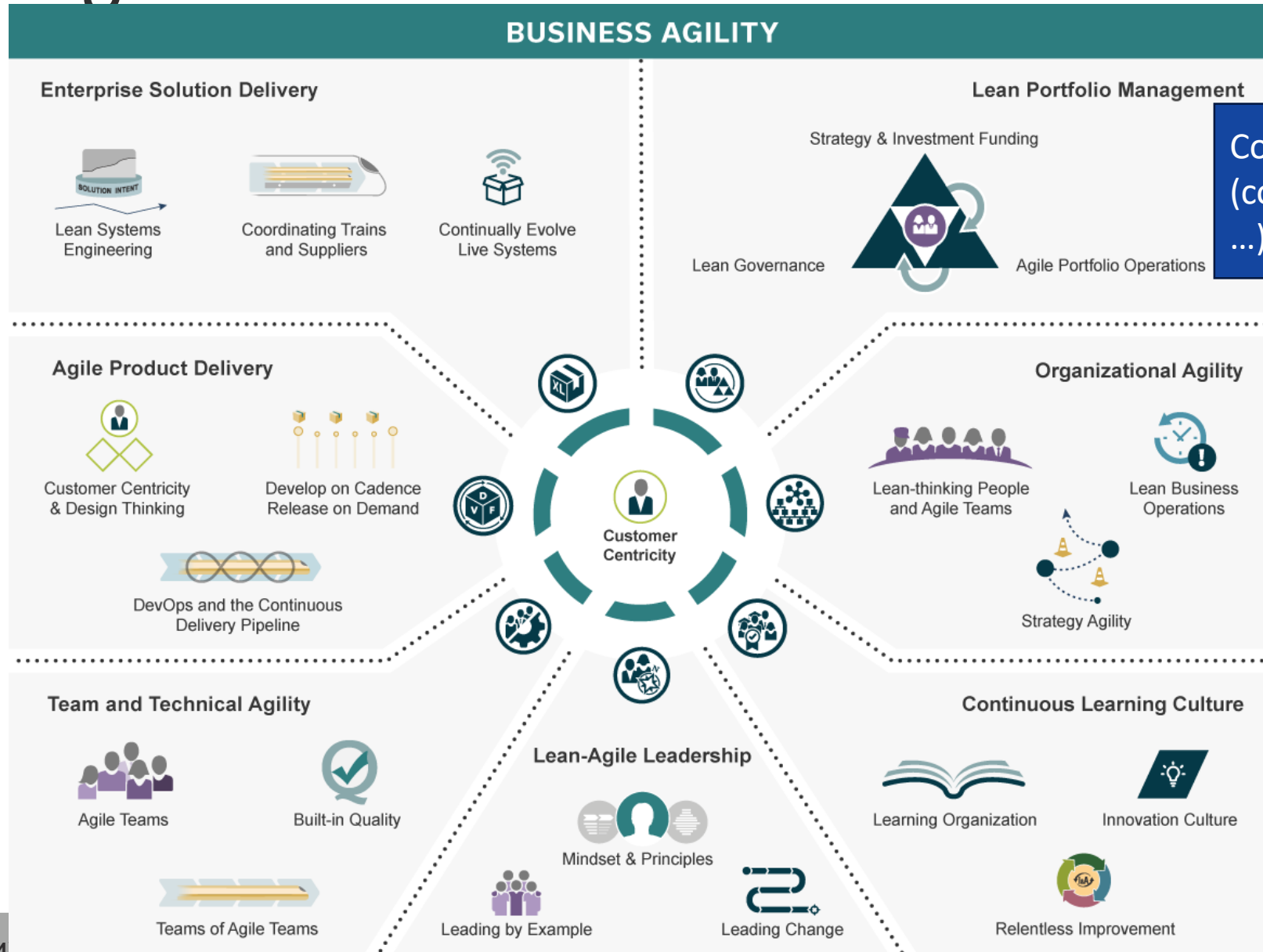
EU Specification – The PM²-Agile Guide

- A project management methodology developed by the European Commission, enabling and supporting the use of Agile practices in any type of project or work activity



<https://op.europa.eu/en/publication-detail/-/publication/ed85debf-decc-11eb-895a-01aa75ed71a1>

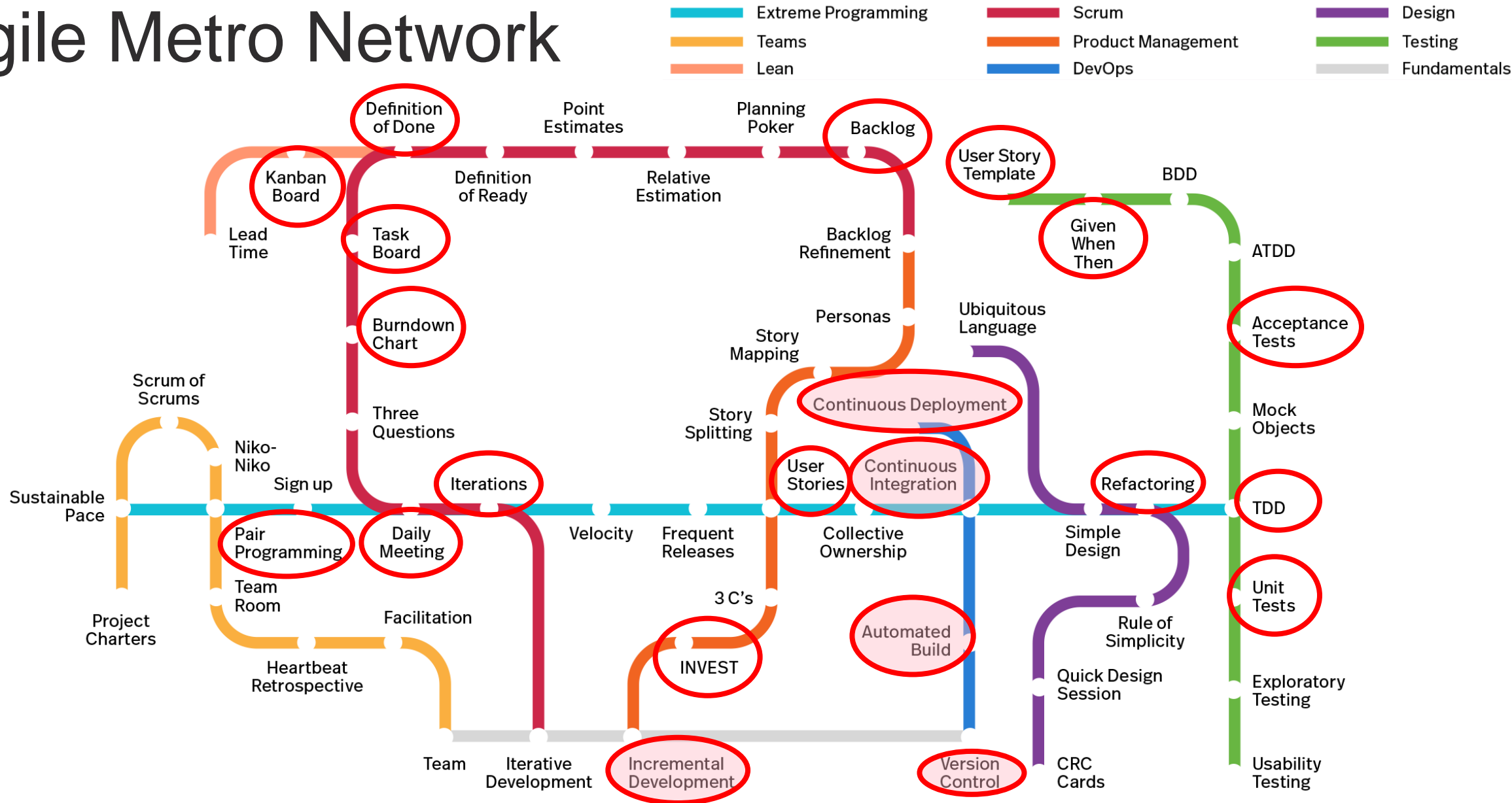
Scaled Agile Framework



Corporate-level agile
(considerable overhead,
...)

Summary

Agile Metro Network



<https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>