tremend

UNIVERSITATEA DIN
BUCUREȘTI
VIRTUTE ET SAPIENTIA

Course 3 - Agile Frameworks

Inginerie Software
An universitar 2022 - 2023

Agenda

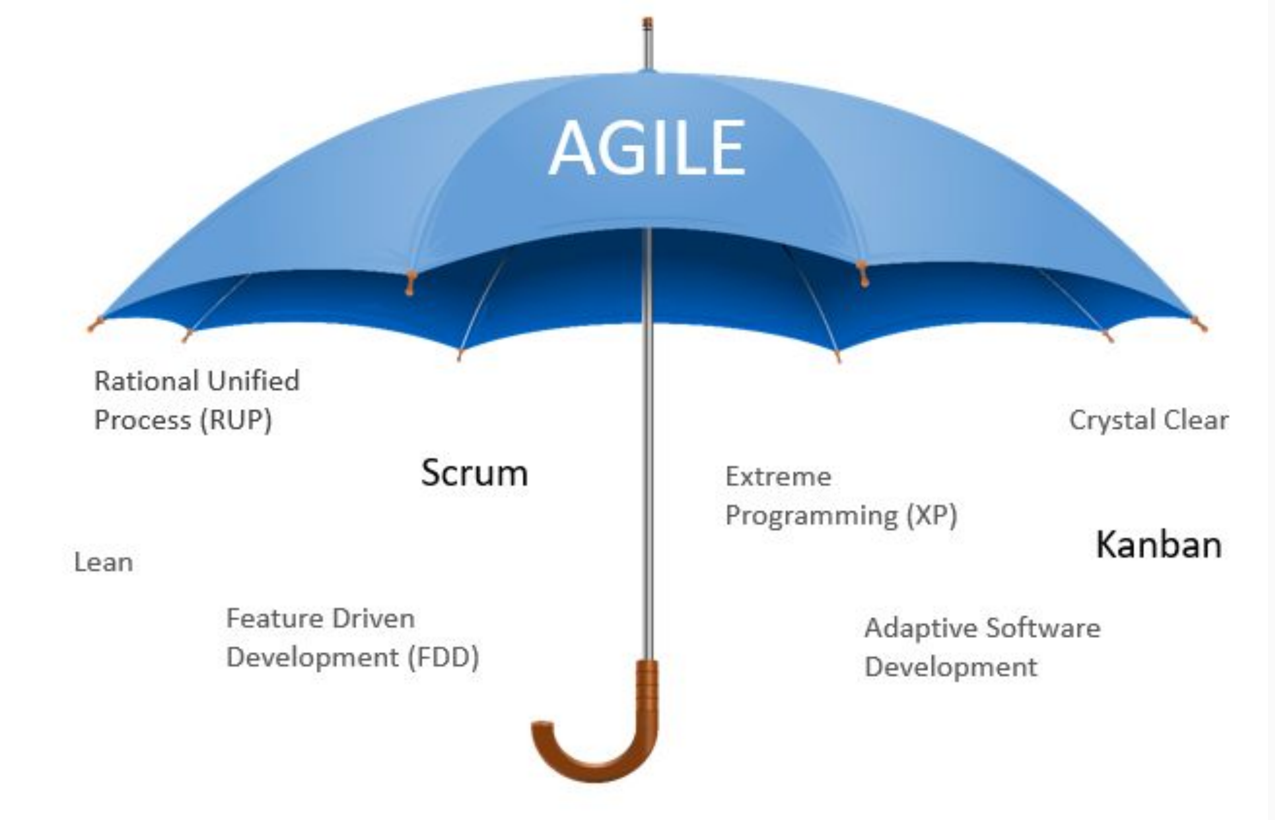
- *Frameworks overview … classic and scaled*
- *Scrum Introduction*
- *Scrum artifacts*
- *Scrum Roles*
- *Scrum events*
    - *Daily Scrum Meeting*
    - *Refinement Meeting*
    - *Sprint Planning Meeting*
    - *Sprint Review Meeting*
    - *Retrospective*
- *Scaled Agile (ex: Safe); Lean Software Development, etc.*
- *Extreme Programming (XP)*
- *Q&A*

Agile Frameworks - Overview

An Agile framework is **a specific approach to planning, managing, and executing work**.

An agile framework is one of many **documented software-development approaches** based on the philosophy articulated in the Agile Manifesto.

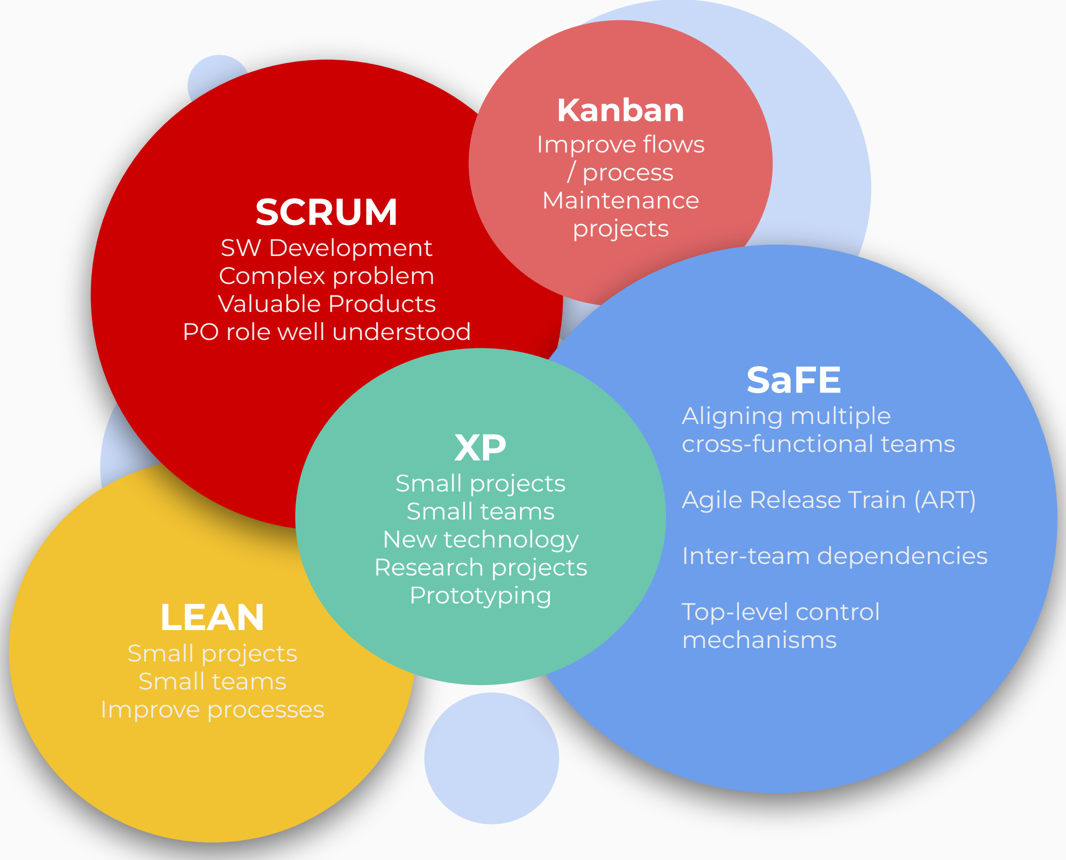
An agile framework incorporates elements of **continuous planning, testing, integration, and other forms of continuous development.**


AGILE
Rational Unified Process (RUP)
Scrum
Extreme Programming (XP)
Crystal Clear
Kanban
Lean
Feature Driven Development (FDD)
Adaptive Software Development

Agile Frameworks - Overview

Agile frameworks typically fall into two categories:

- frameworks designed for teams
- frameworks designed to help organizations practice Agile at scale, across many teams.



SCRUM
SW Development
Complex problem
Valuable Products
PO role well understood

Kanban
Improve flows / process
Maintenance projects

SaFE
Aligning multiple cross-functional teams

Agile Release Train (ART)

Inter-team dependencies

Top-level control mechanisms

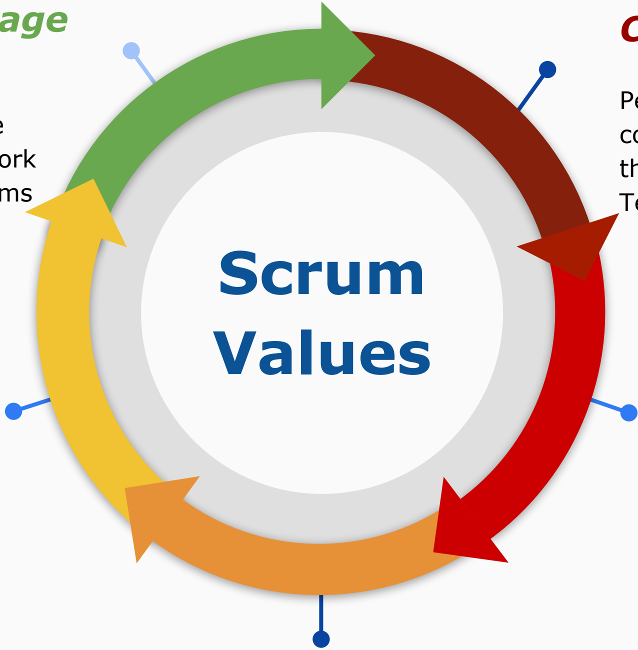
XP
Small projects
Small teams
New technology
Research projects
Prototyping

LEAN
Small projects
Small teams
Improve processes

Scrum Introduction

Scrum is simple. It is the opposite of a big collection of interwoven mandatory components. Scrum is not a methodology. Scrum is an Agile framework that fits perfectly into complex projects where there is a high level of requirements and a very broad scope in the future, where the client's culture is open to change, and where contexts and needs may vary from one moment to another.

Scrum Values



Courage

Scrum Team has courage to do the right thing and work on though problems

Commitment

People personally commit to achieving the goals of the Scrum Team

Respect

Scrum Team members respect each other to be capable, independent people

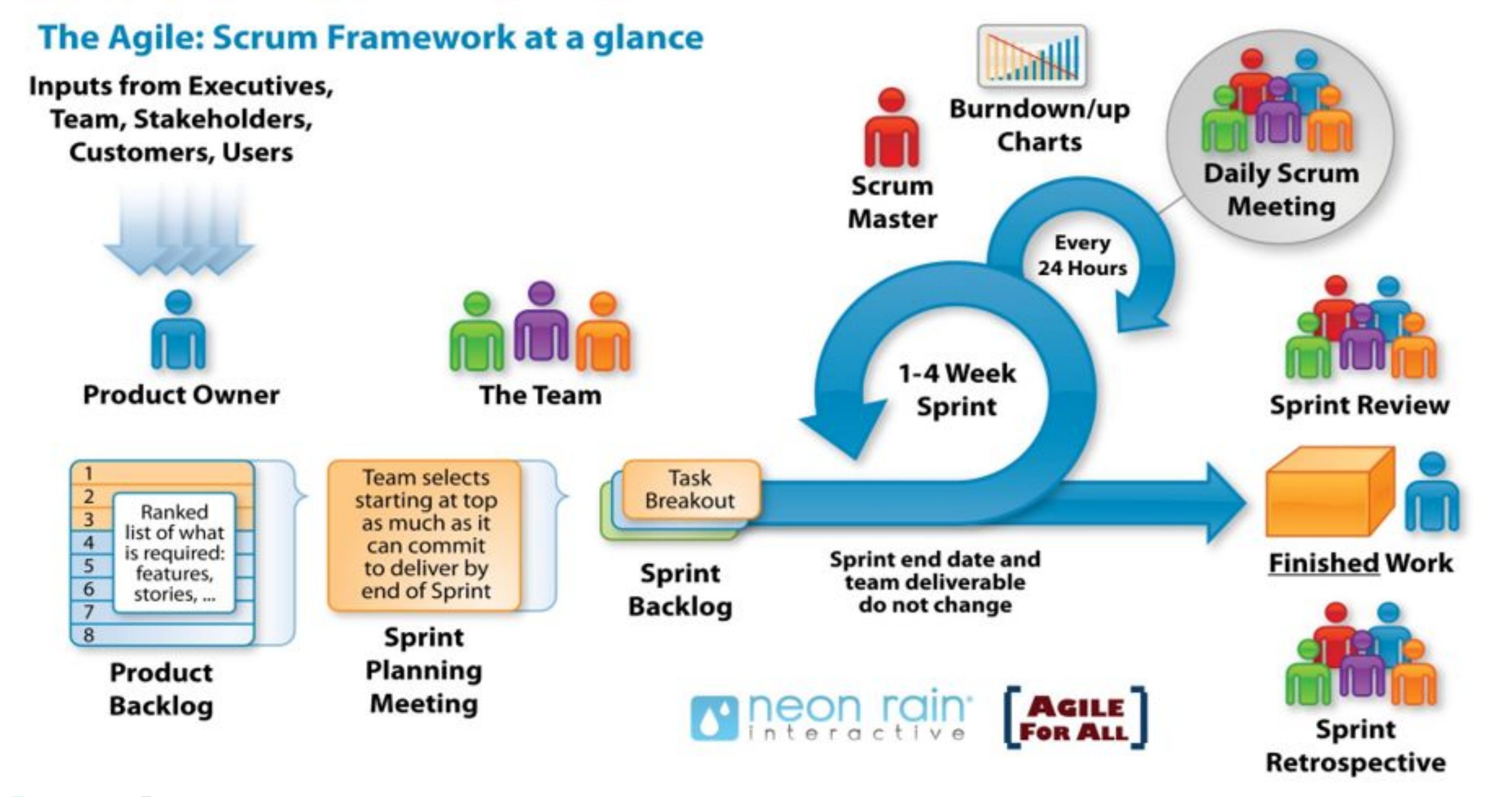
Focus

Everyone focus on the work of the Sprint and the goals of the Scrum Team

Openness

The Scrum team and its stakeholders agree to be opened about the work and challenges with performing the work

Scrum in a nutshell


The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective
neon rain interactive
Agile For All

Scrum artifacts

Agile scrum artifacts are information that a scrum team and stakeholders use to detail the product being developed, actions to produce it, and the actions performed during the product development.

The main agile scrum artifacts are

- **product backlog** - *list of new work items needed to build a product*
- **sprint backlog -** *set of product backlog items that have been promoted to be developed during the next product increment*
- **product increments -** *deliverables that were produced by completing product backlog items during a sprint*

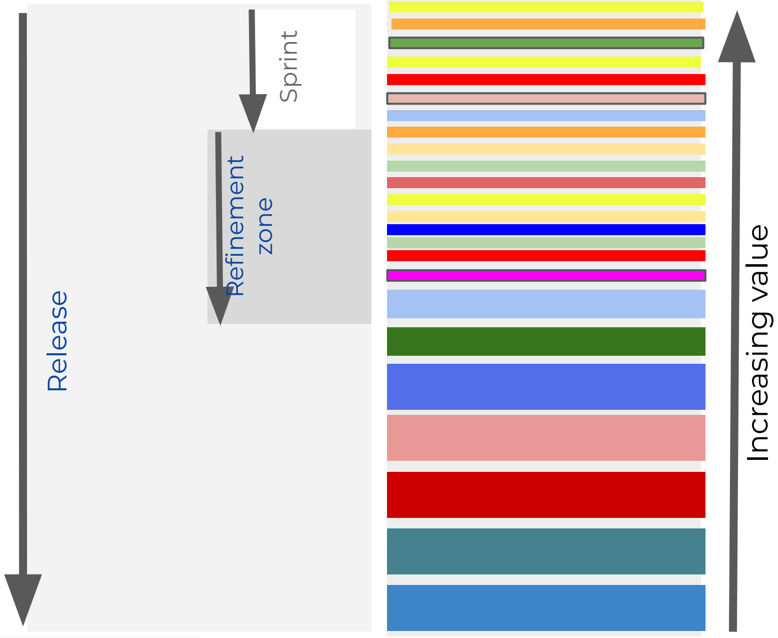
Extended (derived) artifacts

- Product vision
- Definition of ready, definition of done
- Burn-down chart; Velocity
- Sprint goal (…)

Scrum artifacts - Product backlog

Product Backlog *- an ordered list of functional and non-functional requirements that might be needed in the product*

D*etailed Appropriately*
E*mergent*
E*stimated*
P*rioritized*



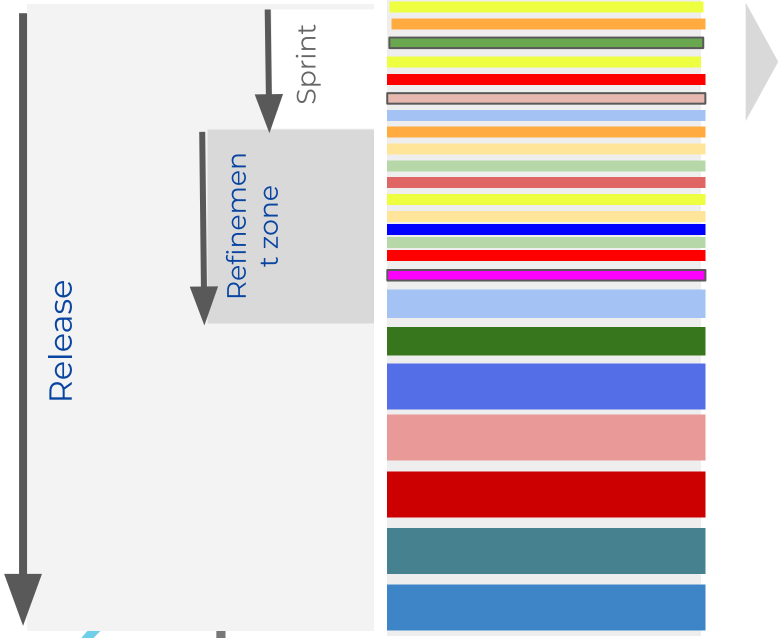
- Prioritized requirement list - Functional & non-functional
- Each Item should be valuable to the users or customers
- Prioritized by the Product Owner

- Ordered by:

> Value / Return in investment
> Strategic fit
> Compliance
> Technical or business risk

Scrum artifacts - Sprint backlog

Sprint Backlog - a subset of the Product Backlog with defined tasks designated to be completed within a Sprint and focused to met the Sprint Goal



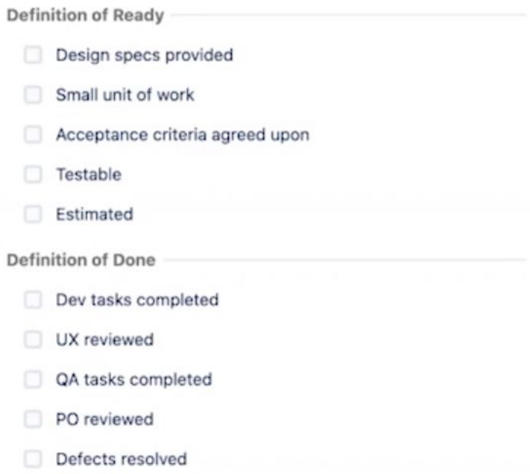
	In progress	Ready for QA	Done
User Story 1	Task Task	Task	Task
User Story 2	Task Task Task	Task Task	
User Story 3		Task Task	Task Task Task

Sprint Backlog

Scrum artifacts - DoR & DoD

➤ The **Definition of Ready** defines the *criteria that a specific user story has to meet* before being considered for estimation or inclusion into a sprint.

➤ This is the **Definition of Done** for the Scrum Team and it is used to assess when work is complete on the product Increment. In short, DoD is a *shared understanding within the Scrum Team on what it takes to make your Product Increment releasable.*



Definition of Ready

□ Design specs provided

□ Small unit of work

□ Acceptance criteria agreed upon

□ Testable

□ Estimated

Definition of Done

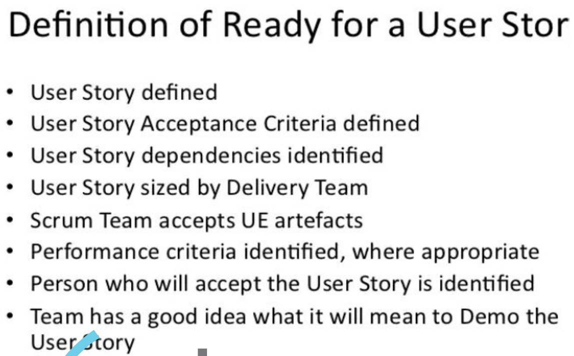
□ Dev tasks completed

□ UX reviewed

□ QA tasks completed

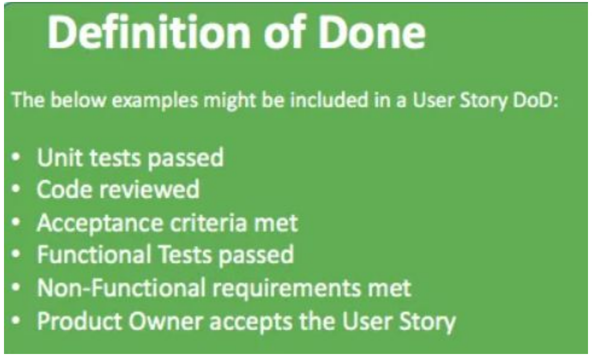
□ PO reviewed

□ Defects resolved



Definition of Ready for a User Stor

- User Story defined
- User Story Acceptance Criteria defined
- User Story dependencies identified
- User Story sized by Delivery Team
- Scrum Team accepts UE artefacts
- Performance criteria identified, where appropriate
- Person who will accept the User Story is identified
- Team has a good idea what it will mean to Demo the User Story

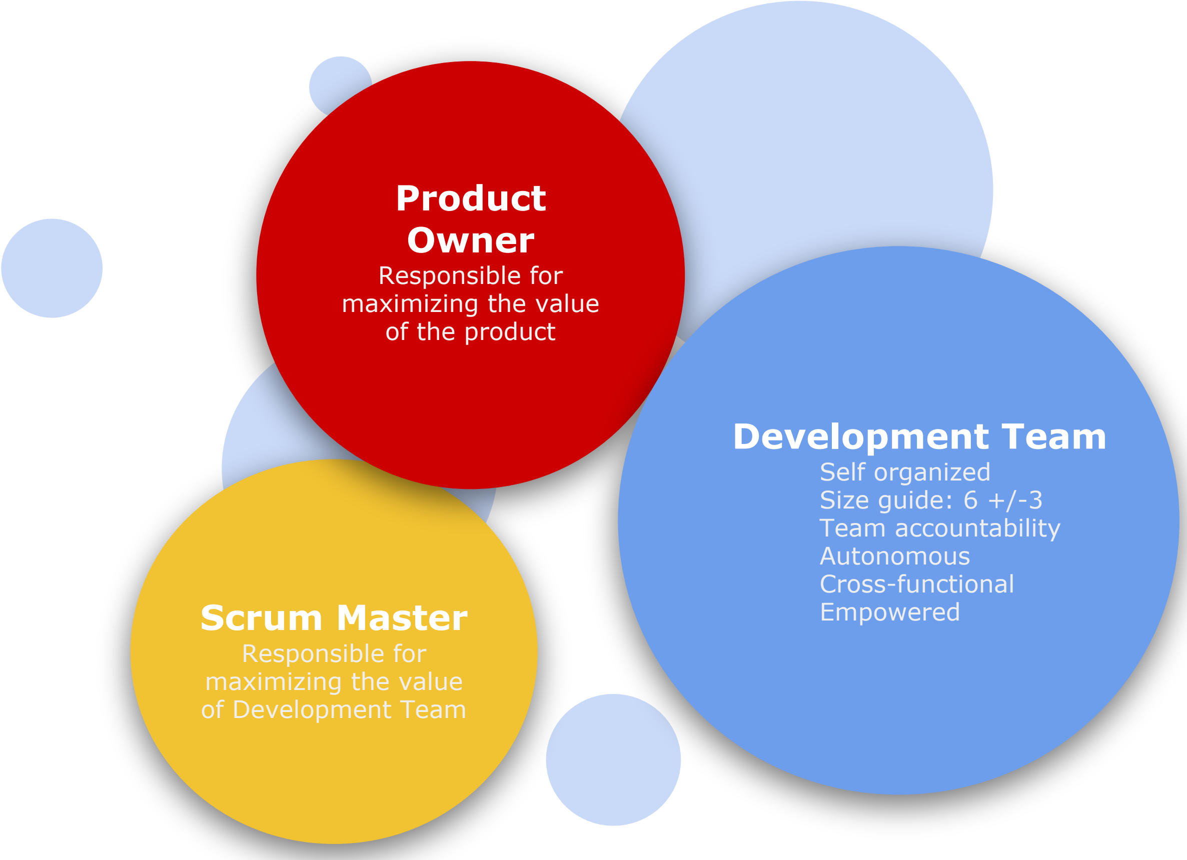


Definition of Done

The below examples might be included in a User Story DoD:

- Unit tests passed
- Code reviewed
- Acceptance criteria met
- Functional Tests passed
- Non-Functional requirements met
- Product Owner accepts the User Story

Scrum Roles - Standard roles



Product Owner

Responsible for maximizing the value of the product

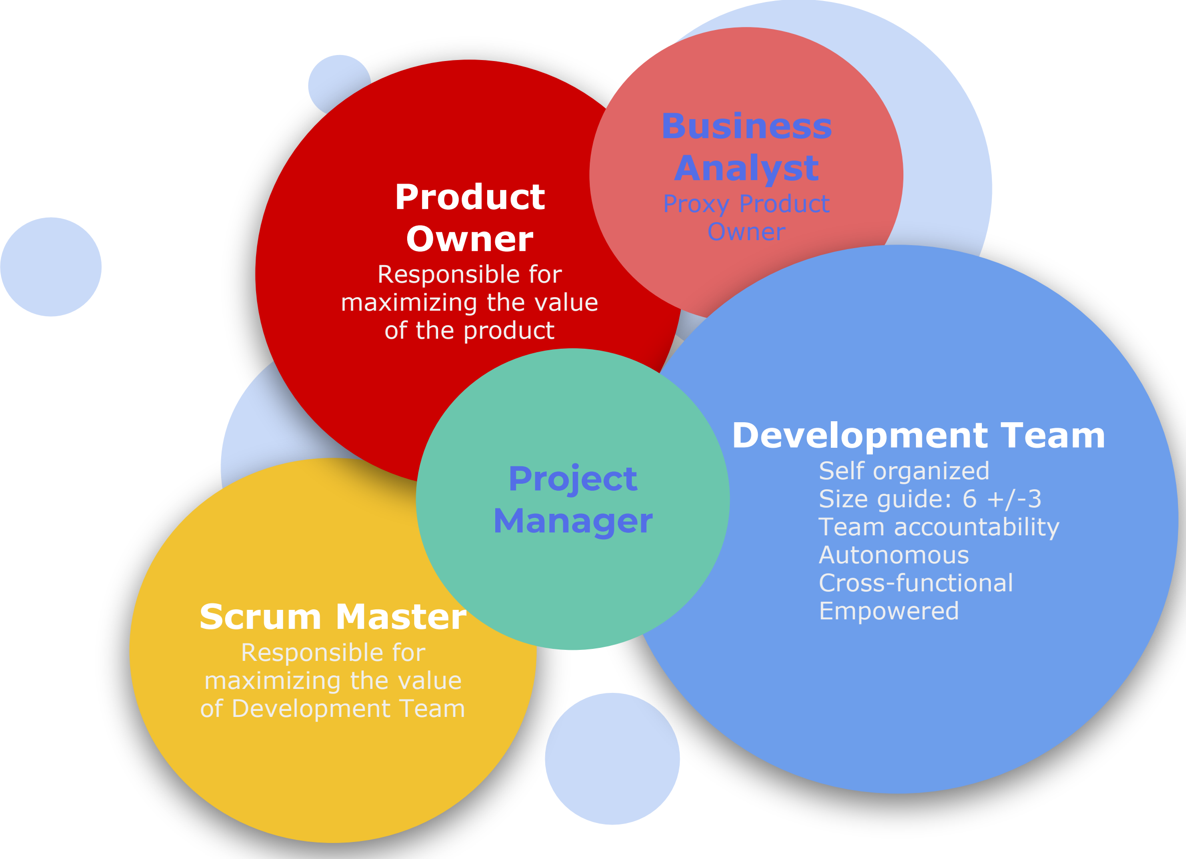
Development Team

Self organized
Size guide: 6 +/-3
Team accountability
Autonomous
Cross-functional
Empowered

Scrum Master

Responsible for maximizing the value of Development Team

Scrum Roles - Extended roles



Product Owner

Responsible for maximizing the value of the product

Business Analyst

Proxy Product Owner

Project Manager

Development Team

Self organized
Size guide: 6 +/-3
Team accountability
Autonomous
Cross-functional
Empowered

Scrum Master

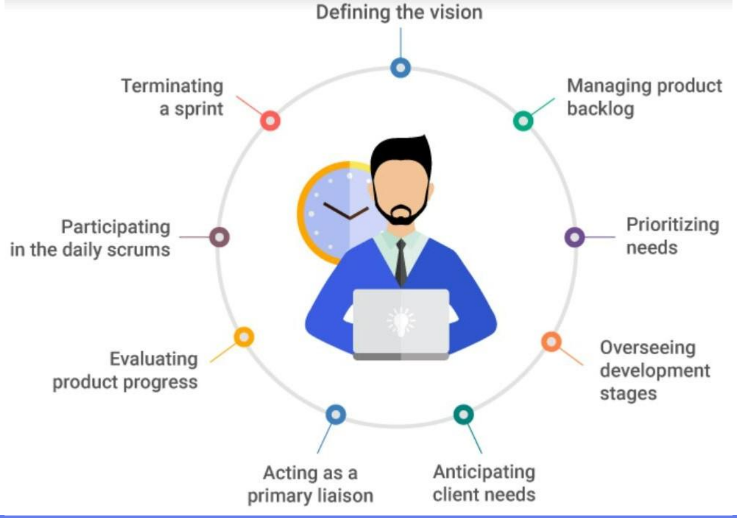
Responsible for maximizing the value of Development Team

tremend

Scrum Roles - PO

The Product Owner

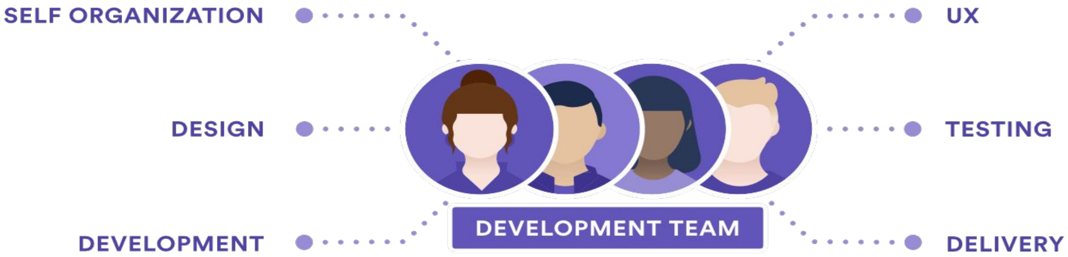
*The business is represented by the **product owner** who tells the development team what is important to deliver.*
***Trust** between these two roles is crucial.*
*The product owner should not only understand the customer, but also **have a vision** for the value the scrum team is delivering to the customer.*
*The product owner also **balances the needs** of other stakeholders in the organization.*


Defining the vision
Managing product backlog
Prioritizing needs
Overseeing development stages
Anticipating client needs
Acting as a primary liaison
Evaluating product progress
Participating in the daily scrums
Terminating a sprint

Scrum Roles - Dev team

The Development Team

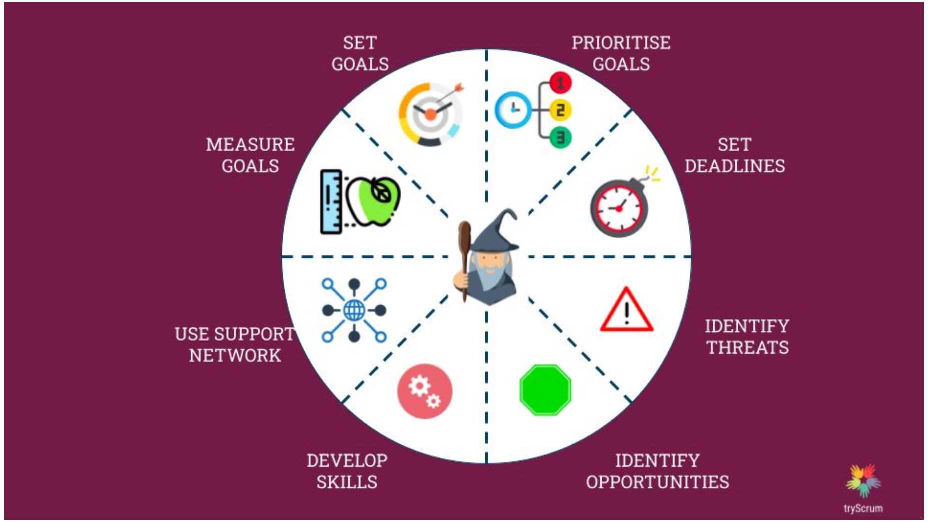
You can think of it in the same way as when you have a house project and you hire a developer.
They develop the project and do the work. Yes, this might mean they lay bricks, do plumbing, even dig holes, but the person is known as a developer.
So, that means the 'developer' role in scrum means a team member who has the right skills, as part of the team to do the work.



Scrum Roles - SM

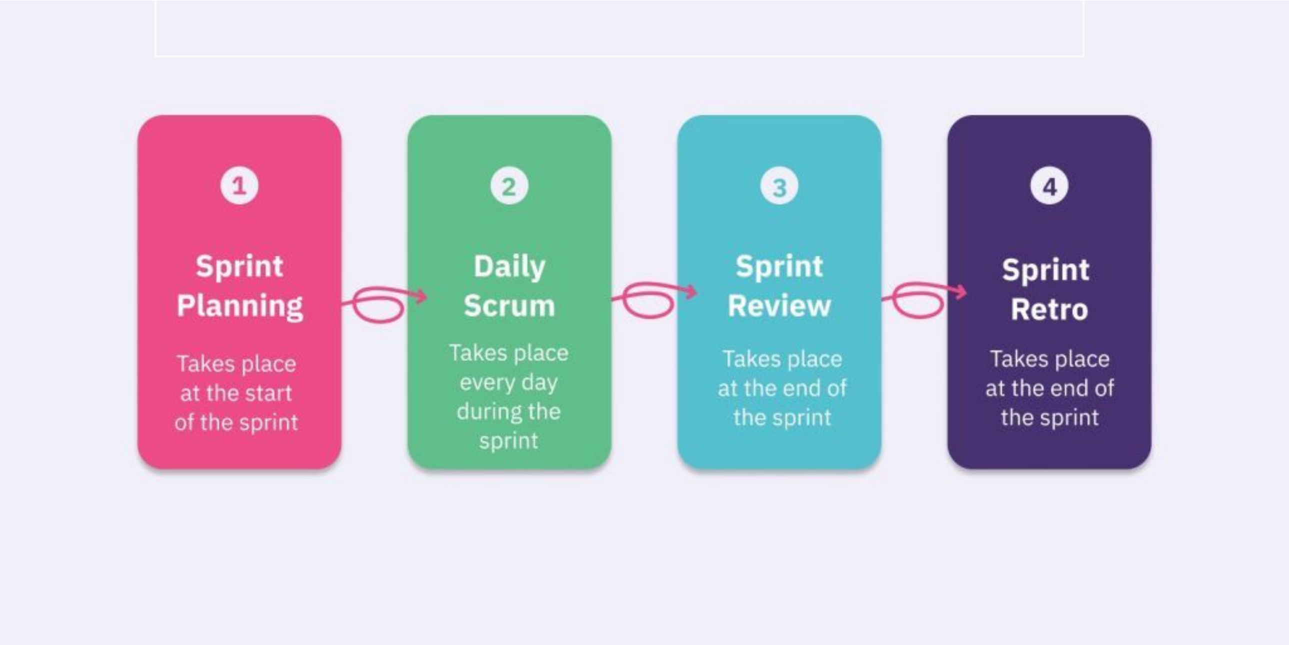
The Scrum Master

Holding it all together, he/she is the role responsible for gluing everything together and ensuring that scrum is being done well.
In practical terms, that means they help the product owner define value, the development team deliver the value, and the scrum team to get better.
The scrum master is a servant leader which not only describes a supportive style of leadership but describes what they do on a day-to-day basis.



Scrum Events

What Are the main Scrum Ceremonies?


1
Sprint Planning
Takes place at the start of the sprint
2
Daily Scrum
Takes place every day during the sprint
3
Sprint Review
Takes place at the end of the sprint
4
Sprint Retro
Takes place at the end of the sprint

Scrum Events - Sprint planning

What:

- Analyse and evaluate Product Backlog
- Select Sprint Goal

How:

- Decide how to achieve sprint goal (desin)
- Create Sprint Backlog (tasks) from Product Backlog items (user stories / features)
- Estimate sprint backlog in hours
- Enable forecast of what can be achieved

→ ***Sprint Goal***

→

Story 1
Task 1 3 hr
Task 2 2 hr
Task 3 4 hr

Sprint Backlog

Inputs	Outputs
• Refined Product Backlog • Team Velocity • Team Capacity • DoD - definition of ready	• Agreed Sprint Goal • Sprint Backlog

Timebox: 8 hours for 1 month sprint - proportionally less for shorter sprint

Scrum Events - Daily Scrum

Makes sure the meeting happens

Responsible for conducting the meeting

May participate in meeting

What:

- A key Inspect and Adapt opportunity

How:

- What did I do yesterday that helped Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

Inputs	Outputs
● Sprint Backlog ● Sprint Goal ● Developments Activities ● Impediments	● Updated Sprint Backlog ● Updated Development activities ● Impediments log

Timebox: daily, same place & same time; 15 minutes or less; stand up

Scrum Events - Sprint review / Demo

What:

- Team and stakeholders review what has been accomplished during the Sprint

How:

- Demo potentially shippable product increment - Sprint Product Backlog items meeting the team’s DoD
- Informal: not a slide presentation
- Seek and discuss feedback
- Changes and new ideas go in the Product Backlog
- Discuss the Goal and candidate Product Backlog items for next Sprint

Inputs	**Outputs**
● Sprint Goal ● Sprint Backlog ● DoD ● Product Backlog ● Business conditions	● Reviewed Product Backlog ● Completion Date Forecast ● Actual Sprint Velocity

Timebox: 4 hours for 1 month sprint - proportionally less for shorter sprint

Scrum Events - Sprint retrospective

What:

- Analyse what went well/not so well
- Identify actions that will improve future Product Increment

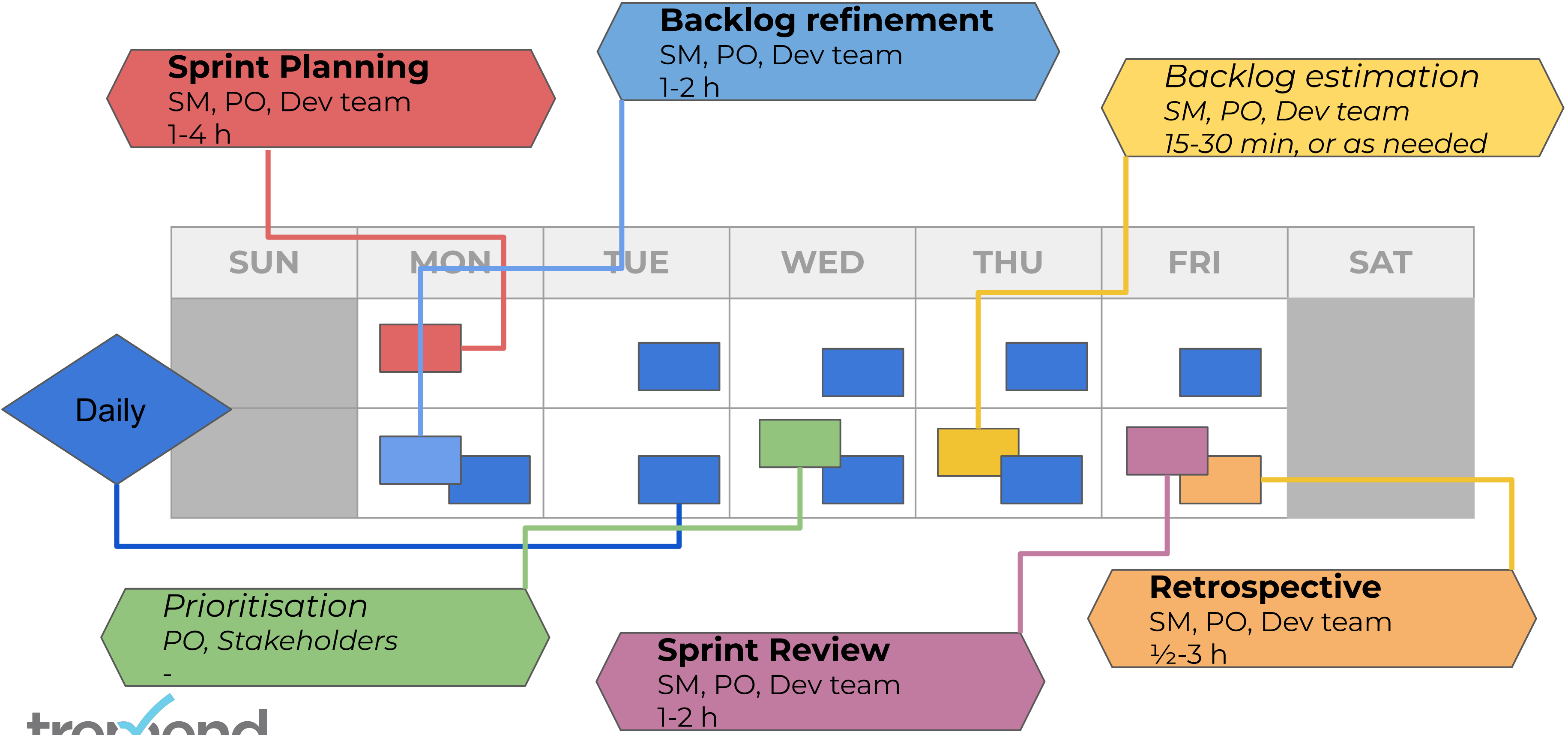
How:

- Drill down to root cause of issues and waste
- Improve process within the Scrum framework and principles
- Exercise driven
- Drive out actions

Inputs	Outputs
● DoD - definition of ready ● Sprint Goal ● Sprint Backlog	● Updated Definition of Done/ Ready ● Actions and plan for improvements: *technical, process, planning or capacity calculation, etc*

Timebox: 3 hours for 1 month sprint - proportionally less for shorter sprint

Scrum Events - extended



Sprint Planning
SM, PO, Dev team
1-4 h

Backlog refinement
SM, PO, Dev team
1-2 h

Backlog estimation
SM, PO, Dev team
15-30 min, or as needed

Prioritisation
PO, Stakeholders
-

Sprint Review
SM, PO, Dev team
1-2 h

Retrospective
SM, PO, Dev team
½-3 h

Scrum Events - Backlog refinement

What:

- **ensure the next few sprints worth of user stories in the product backlog are prepared for sprint planning**

How:

- Review items on product backlog
- prioritise
- make sure that the ones on top respond to next priority and that are ready to be picked up

→ ***Prioritised backlog***

Inputs	Outputs
● Product Backlog ● DoD - definition of ready	● Prioritised product backlog ● Items corresponding to DoD

Timebox: 8 hours for 1 month sprint - proportionally less for shorter sprint

Scaled agile frameworks

What

Scalability is **a property of a system that accentuates the tensions between planning and agility to a new level**. Both scalability and agility seek to accommodate uncertainty.

Why?

*Scaled agile or "agile at scale" is a systematic framework to facilitate big agile implementation. The intent is to provide just the <u>**right amount** of structure and governance</u> necessary to facilitate <u>larger teams working on complex projects</u>.*

How

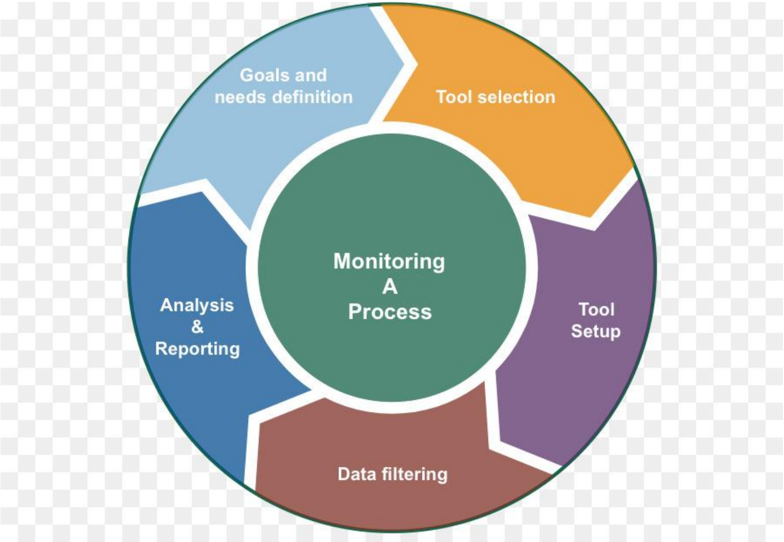
Using **a set of organizational and workflow patterns (Frameworks) for implementing agile practices at an enterprise scale**. The frameworks are body of knowledge that include structured guidance on roles and responsibilities, how to plan and manage the work, and values to uphold.

<u>Scaled</u>

- ★ Scaled Agile Framework (SAFe)
- ★ Scrum@Scale (SaS)
- ★ Large Scale Scrum (LeSS)
- ★ Nexus
- ★ Disciplined Agile (DA)
- ★ Enterprise Kanban, aka Portfolio Kanban.

Lean Software Development

- Lean software development is **a concept** that emphasizes optimizing efficiency and minimizing waste in the software development process.



The Lean model for software development is inspired by "lean" manufacturing practices and principles.

The seven Lean principles (in this order) are:

1. eliminate waste,
2. amplify learning,
3. decide as late as possible,
4. deliver as fast as possible,
5. empower the team,
6. build in integrity and
7. optimise the whole

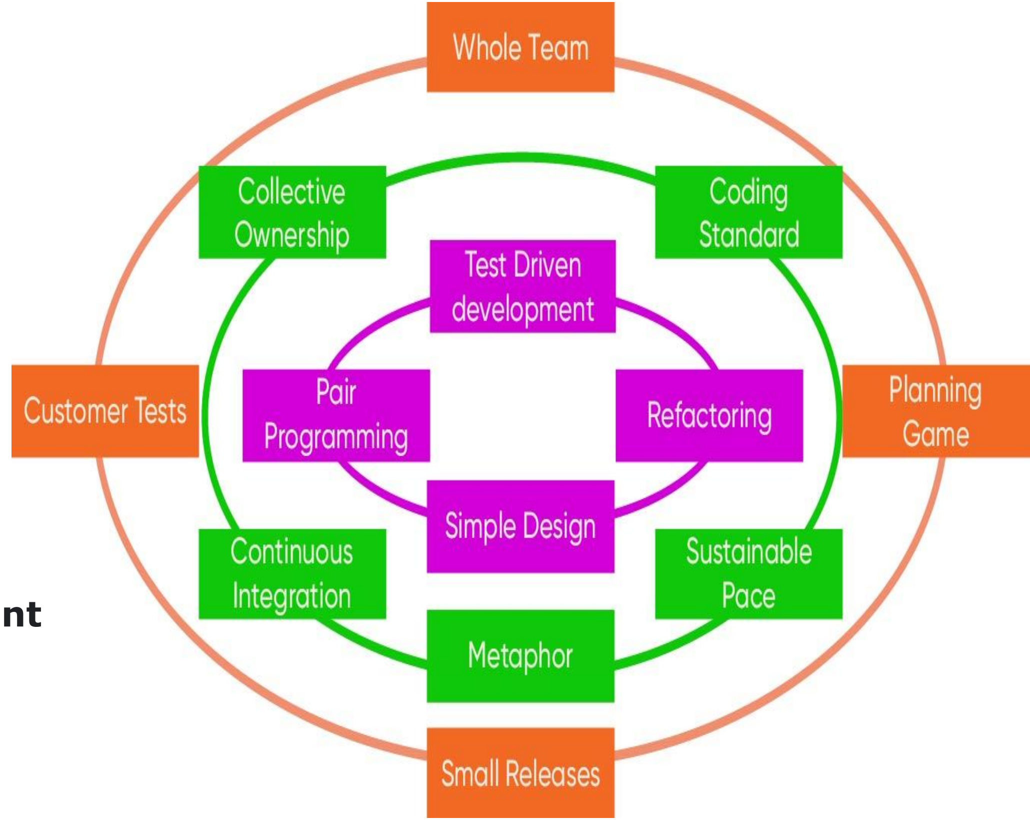
Extreme programming - XP

Extreme Programming (XP) is **an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team**.

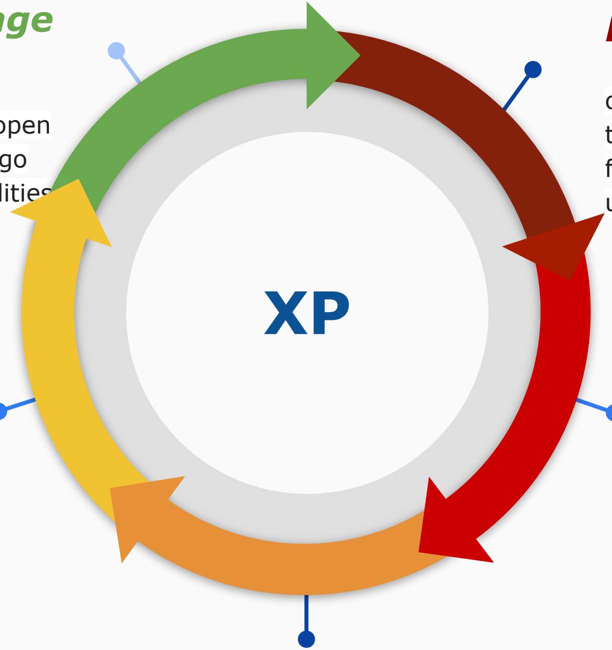
XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

The main characteristics of XP include

- **dynamically changing software requirements;**
- **small, collocated extended development team;**
- **leveraging technology that facilitates automated unit and functional tests**



XP Values



Courage

Team should feel courageous to be open and challenged to go beyond the capabilities

Feedback

opportunity to review the product and get feedback from the users

Simplicity

Easy layout and code that is flexible enough to accommodate changes as the necessities evolve.

Communication is important between the XP team and customers

Respect

As teams and customer work together, sharing success and failure, they respect each other, their agreements and commitments.

Fastest growing tech company in Romania by Deloitte Fast 50

Course 3 - Agile Frameworks

Q&A