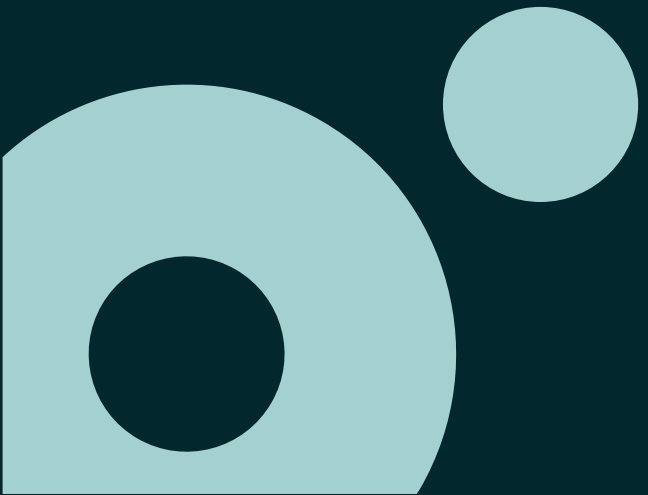
DevOps Fundamentals

Continuous Planning



Agenda

Continuous Planning

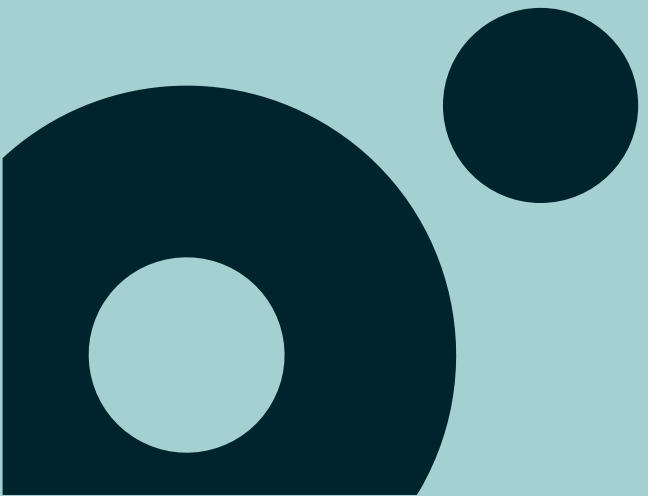
Agile Methodologies

Scrum

Kanban

moOngy.

Continuous Planning



Continuous Planning

Practice that requires planners, architects and agile teams to define their plans working as a whole

On-going plan, based on feedback, integrating new ideas and requirements

Main tasks: Definition, refinement, decomposition, prioritization

Rely on Agile Methodologies to better fit on plan & track requirements

Teams need to be autonomous as possible to take their decisions with real impact

Teams need to be aligned with organization strategy to embrace DevOps Culture

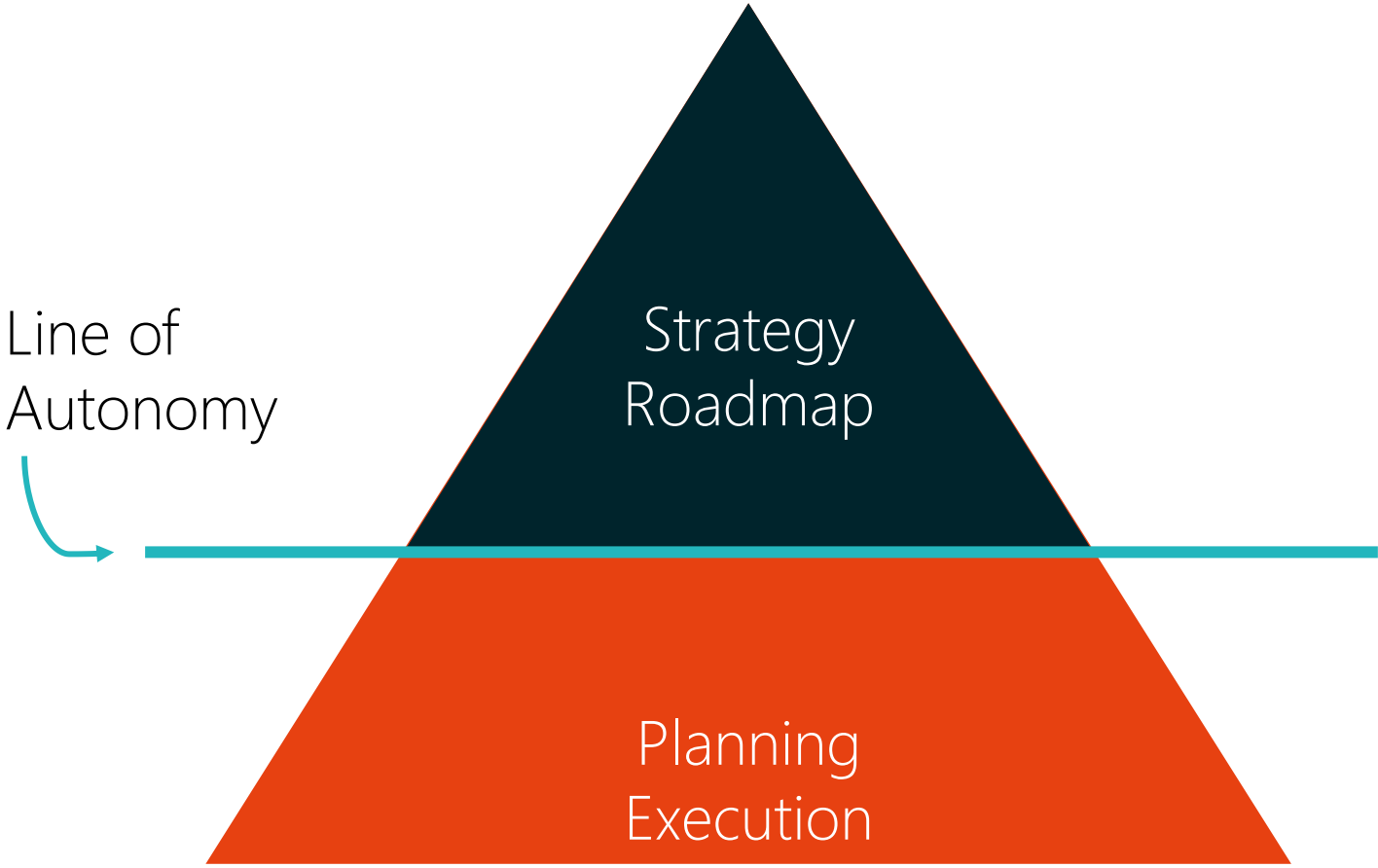
Continuous Planning

Autonomy

- Plan
- Practices

Alignment

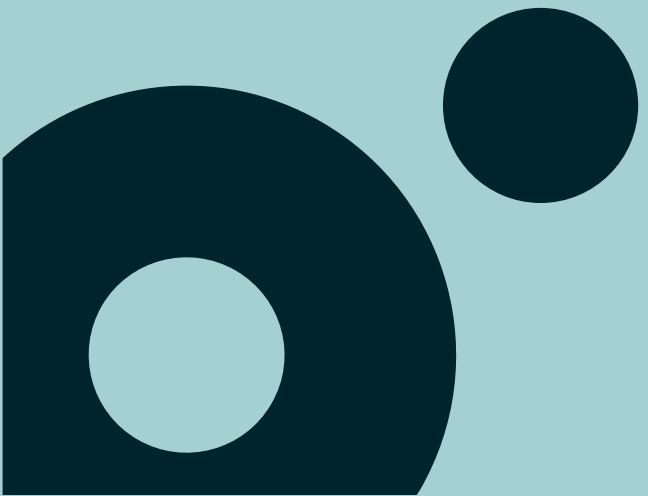
- Organization
- Roles
- Teams
- Cadence
- Taxonomy



Where to put Line of Autonomy?

- Too much alignment, teams are dependent and lack innovation
- Too much autonomy, makes hard to create patterns and culture

Agile Methodologies



Traditional Methodologies

Waterfall process

Sequential and using phases

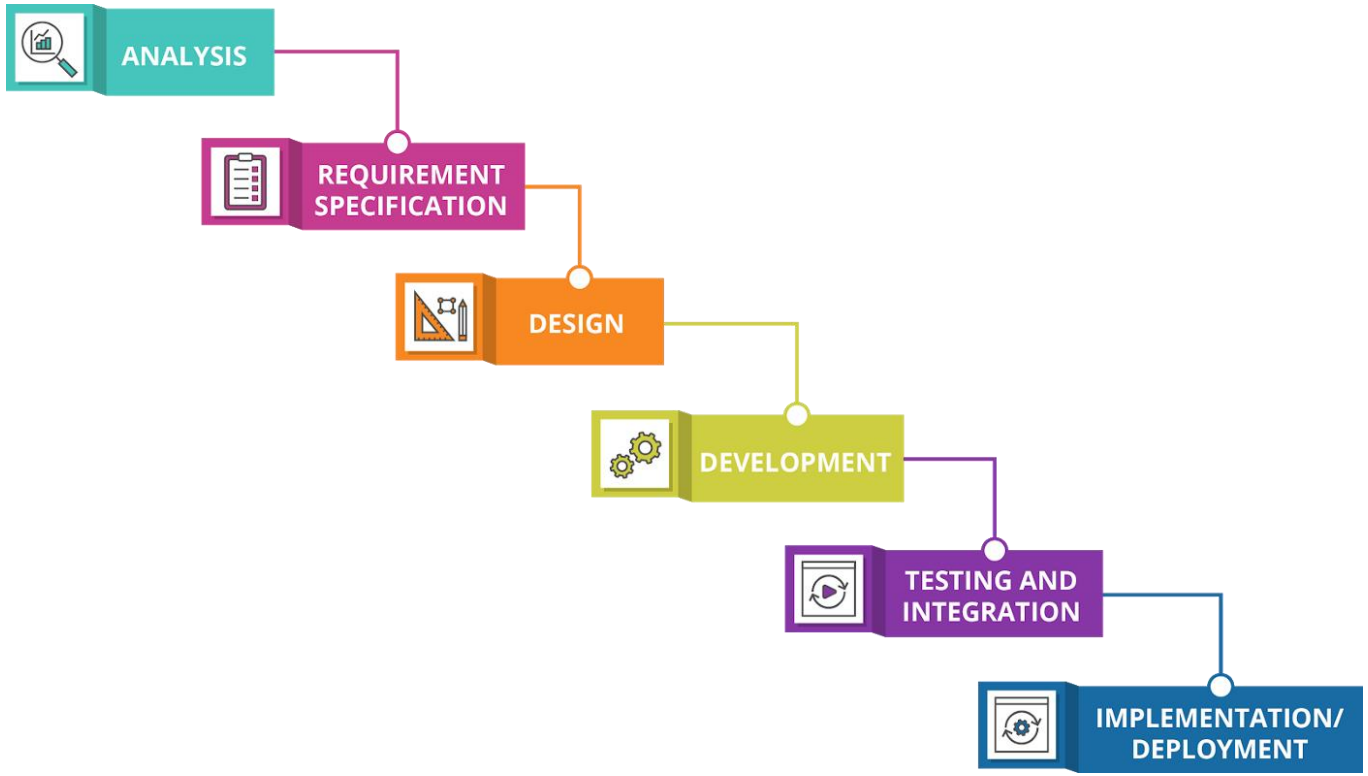
Focus on documentation and processes

No agility

Any change makes you restart the process

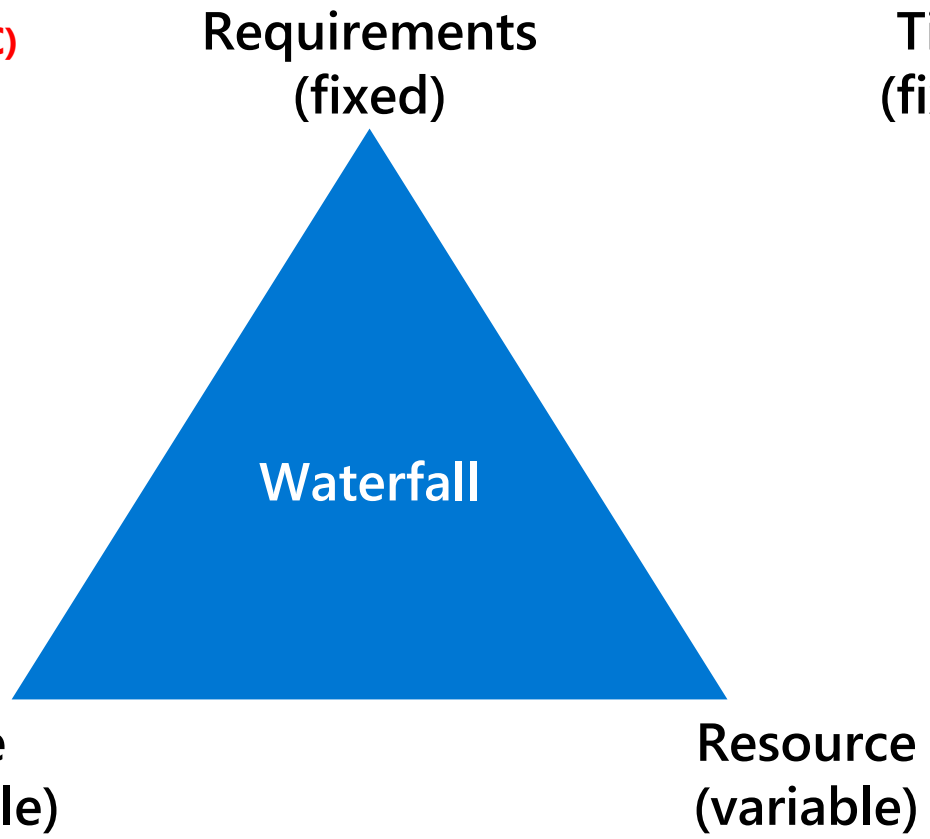
Was the most used process during years

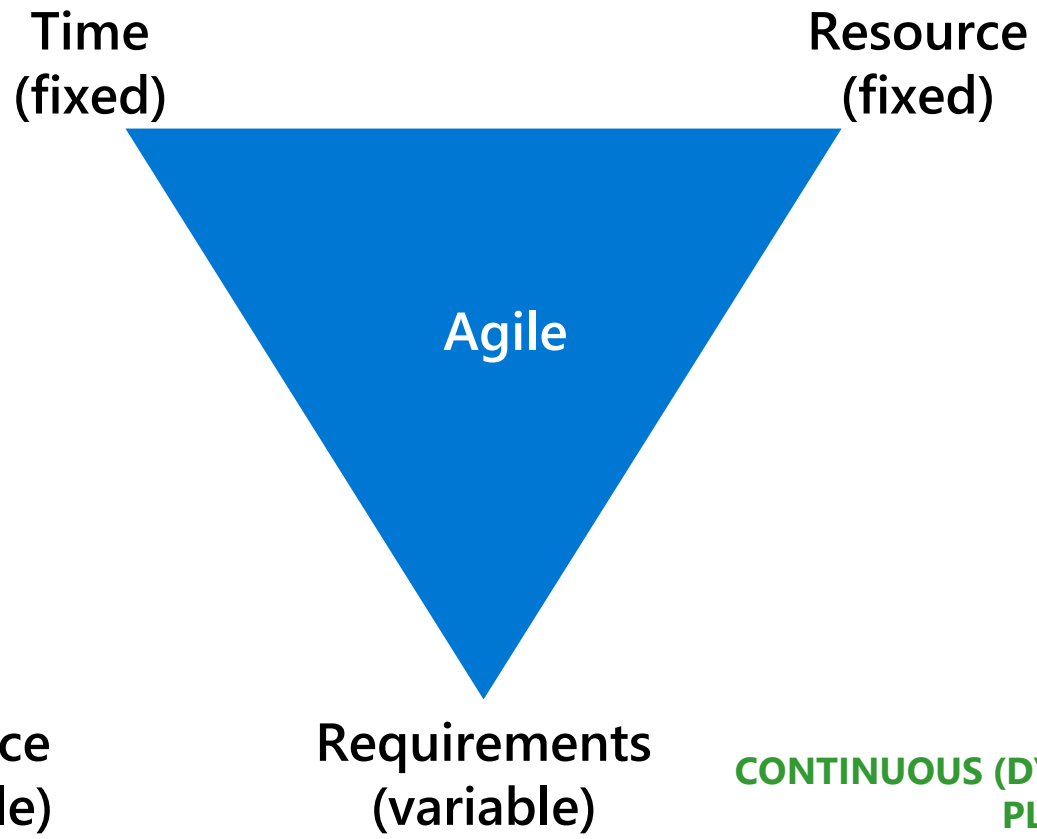
Still in use and for specific scenarios, could be the best choice



Waterfall vs. Agile

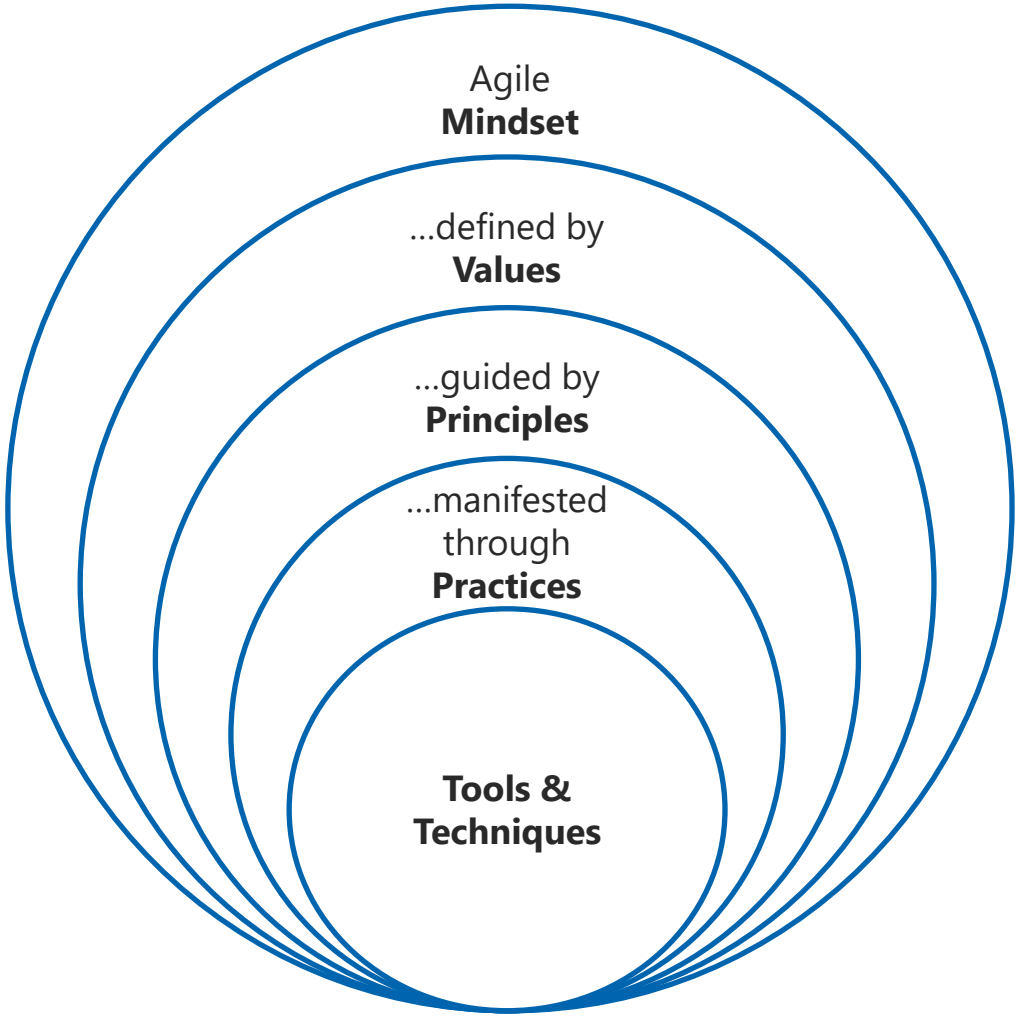
FIXED (STATIC) PLANNING





CONTINUOUS (DYNAMIC) PLANNING

Agile Layers (Agile Onion)

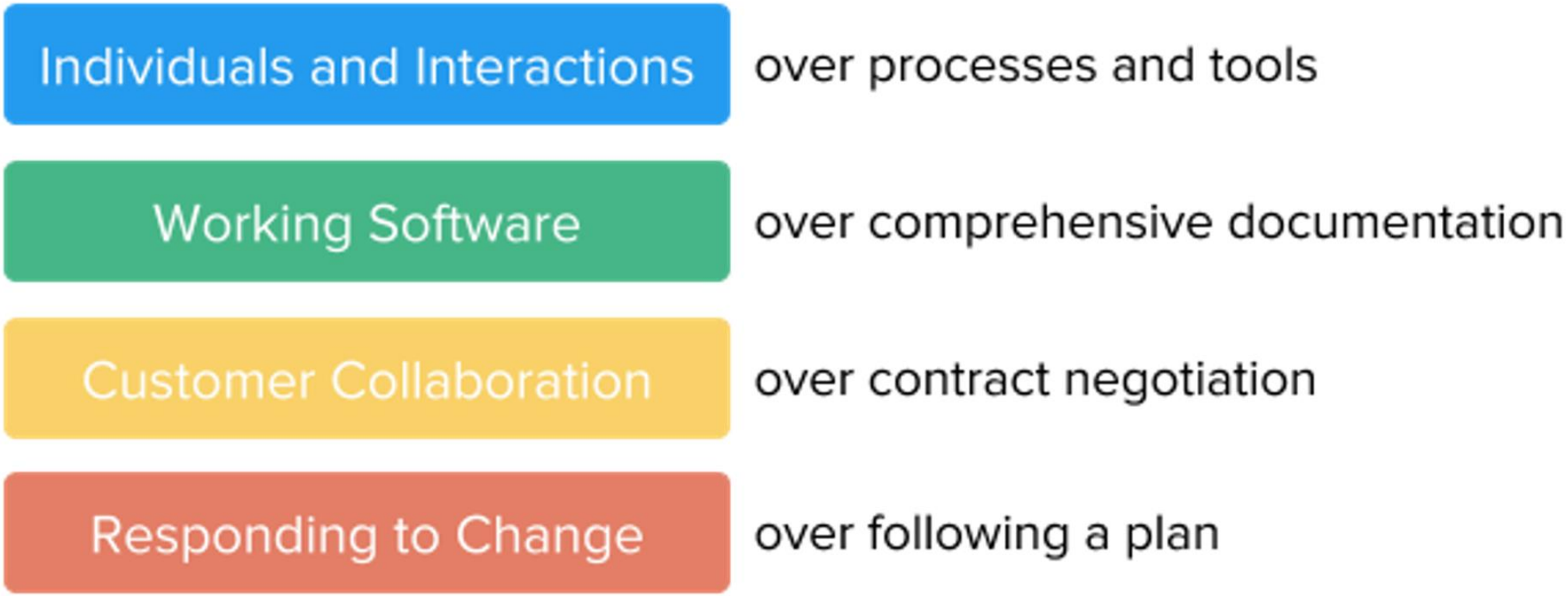


Agile Manifesto

Scrum, SAFe
XP, DSDM, ...

Scrum Board,
Estimation
Techniques, ...

Agile Values



Agile Principles

12 AGILE PRINCIPLES

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
04 Business people and developers must work together daily throughout the project.	05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
07 Working software is the primary measure of progress.	08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	09 Continuous attention to technical excellence and good design enhances agility.
10 Simplicity – the art of maximizing the amount of work not done – is essential.	11 The best architectures, requirements, and designs emerge from self-organizing teams.	12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

[Manifesto for Agile Software Development (agilemanifesto.org)]

moOngy.

Agile Principles

moOngy.

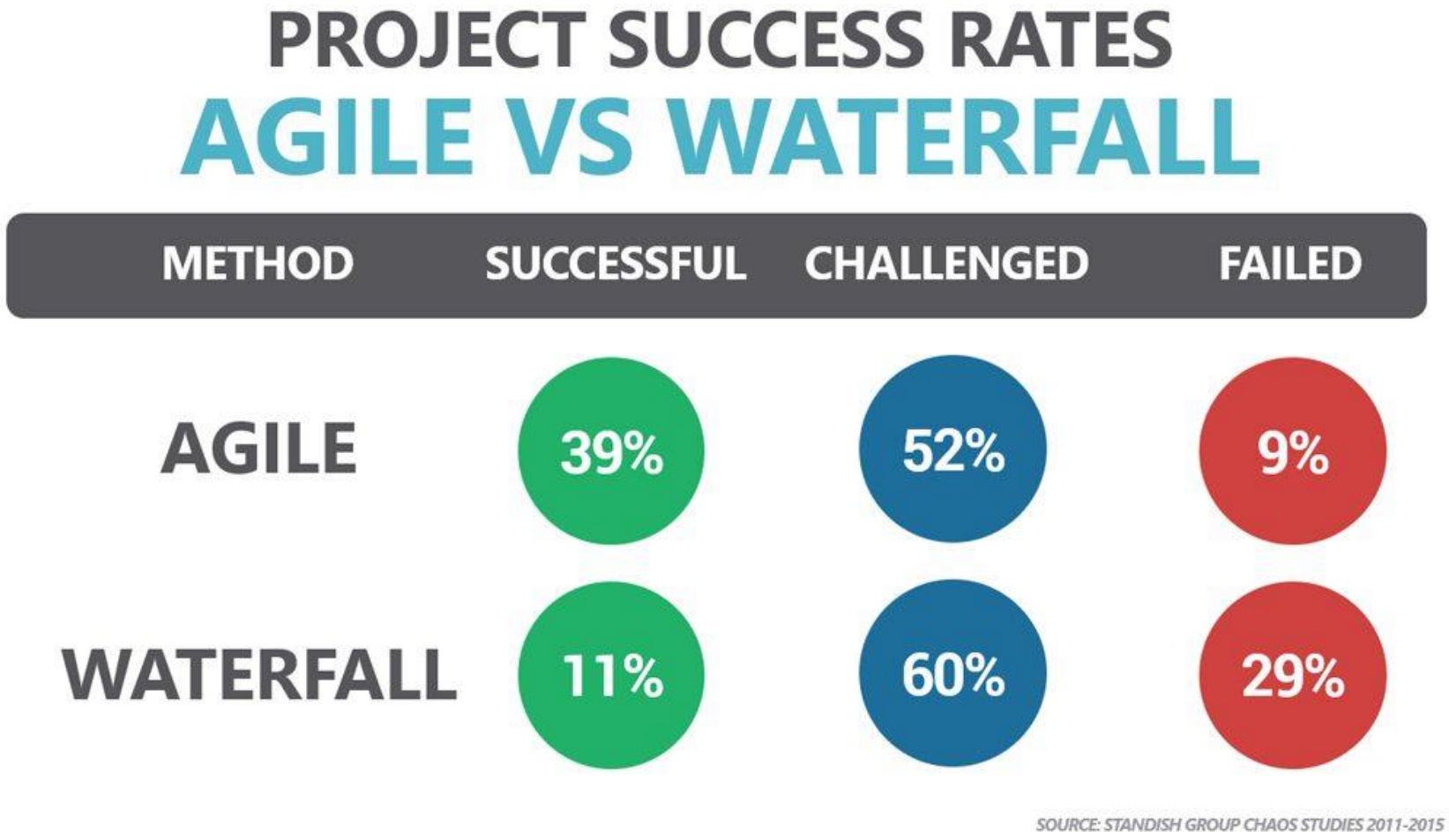
Frameworks

Several frameworks implement these principles defining tools & techniques

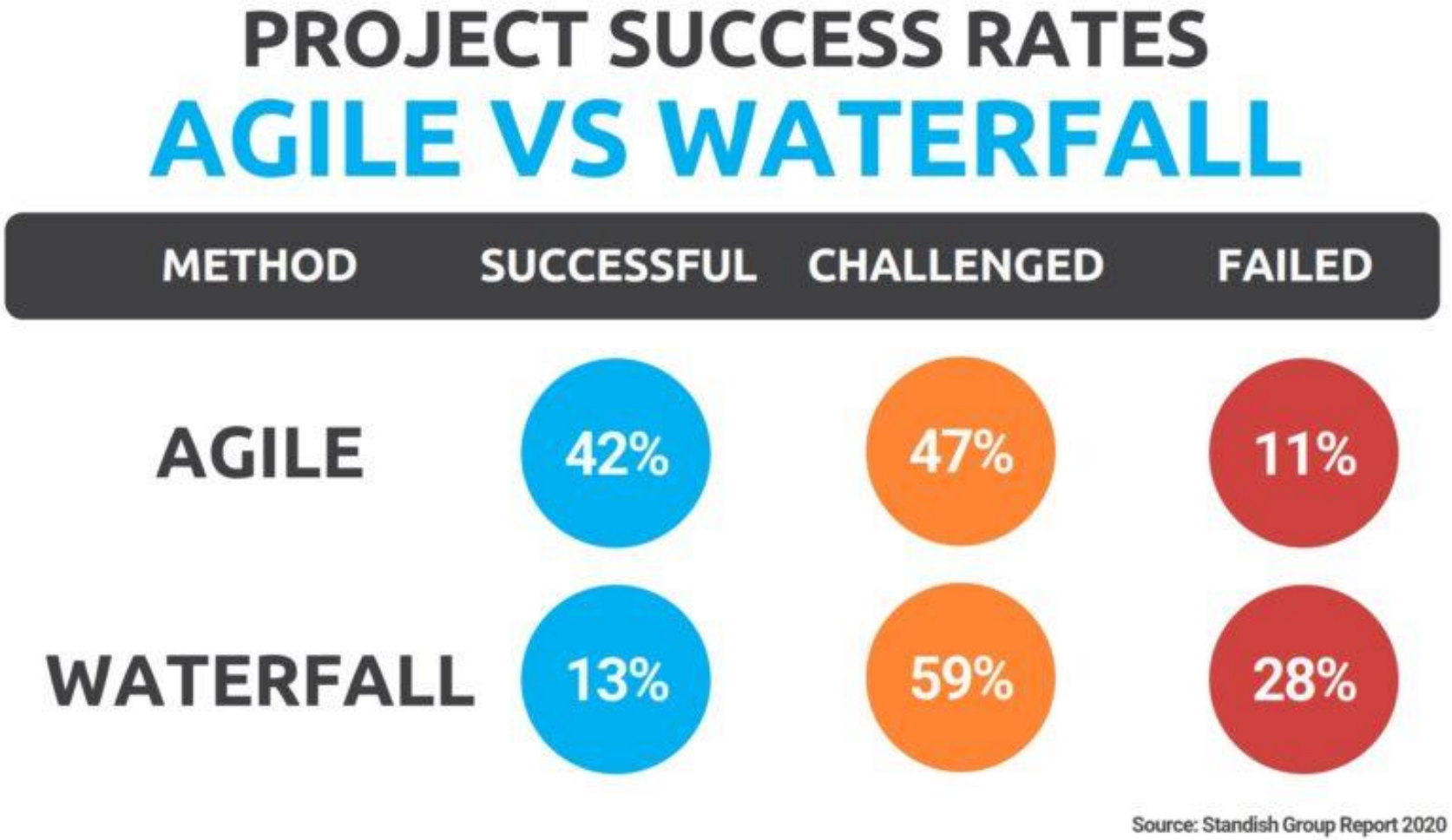
- Scrum
- eXtreme Programming (XP)
- Lean
- Scaled Agile Framework (SAFe)
- Crystal Clear
- Kanban

Nevertheless, the most used framework is “Scrum, but” or “My own implementation of Scrum” ☺

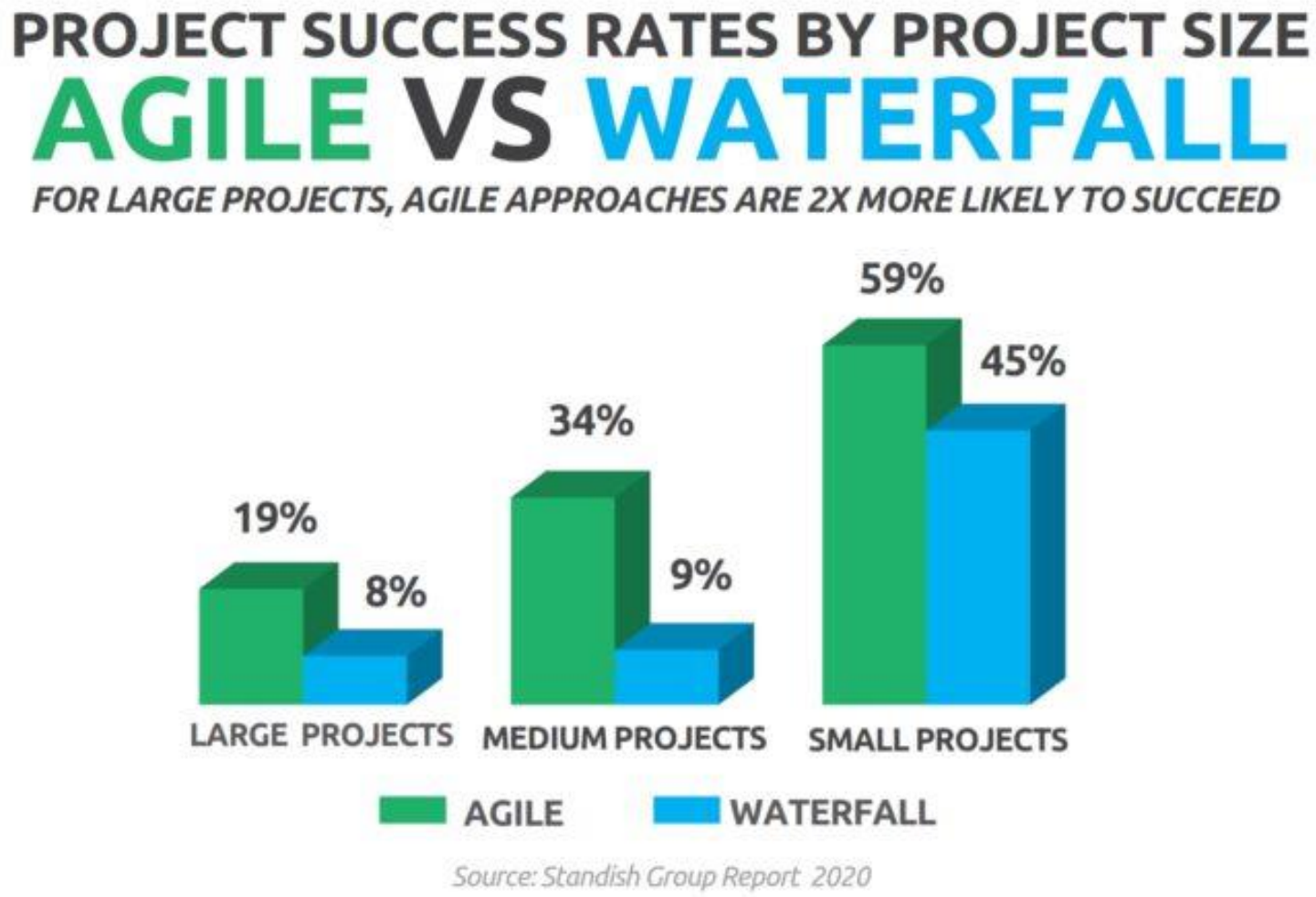
Comparison: Project Success Rates (2015)



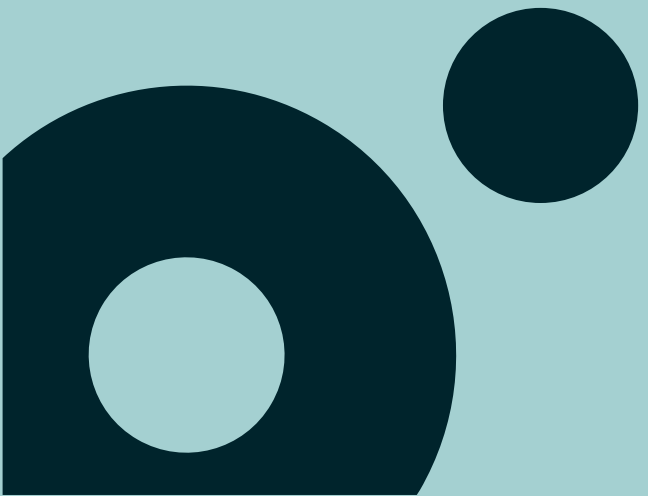
Comparison: Project Success Rates (2020)



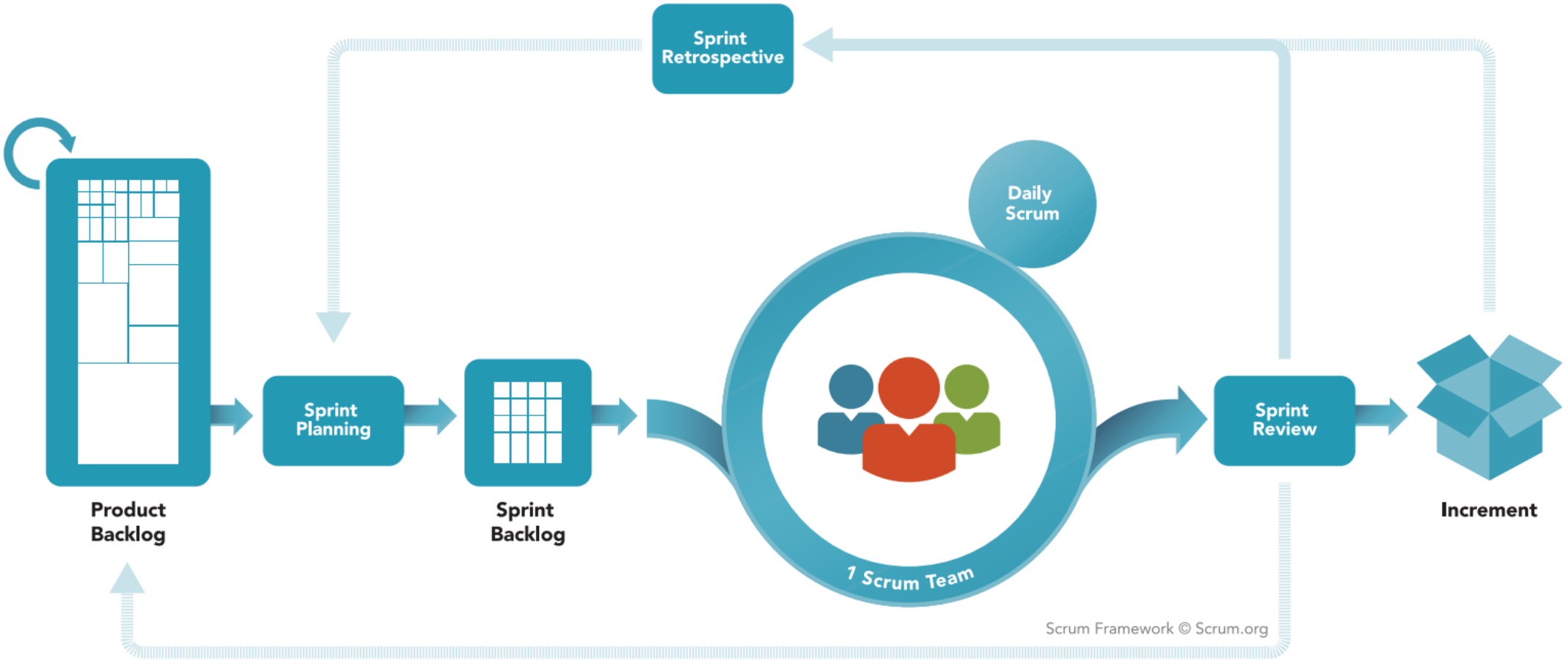
Comparison: Project Success Rates by Size



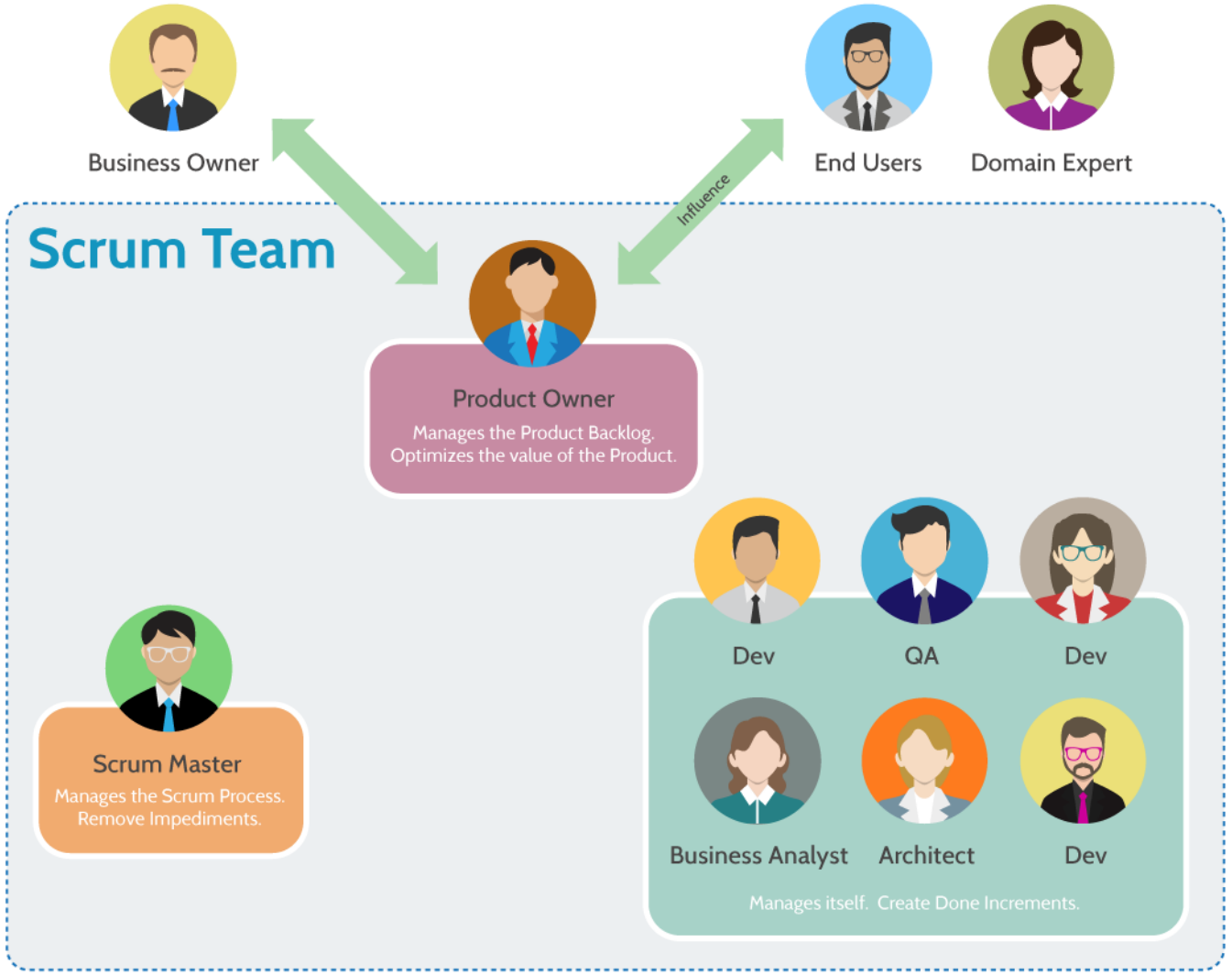
Scrum



Scrum Framework



Scrum Framework: Scrum Team



- Small (up to 10 elements)
- One Product Owner, one Scrum Master and Developers
- No hierarchy
- Self-managed
- Multi-skilled
- Responsible for all activities
- Works in Sprints/Iterations
- Keeps same pace

moOngy.

Scrum Team: Product Owner

Focused on “WHAT” not HOW

The Product Owner is one person, not a committee

The Product Owner may represent the needs of many stakeholders in the Product Backlog

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal
- Creating and clearly communicating Product Backlog items
- Ordering Product Backlog items
- Ensuring that the Product Backlog is transparent, visible and understood

Scrum Team: Product Owner

Focused on “WHAT” not HOW

The Product Owner is one person, not a committee

The Product Owner may represent the needs of many stakeholders in the Product Backlog

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal
- Creating and clearly communicating Product Backlog items
- Ordering Product Backlog items
- Ensuring that the Product Backlog is transparent, visible and understood

Scrum Team: Scrum Master

Focused on ensuring Scrum is applied correctly (IS NOT a project manager)

Scrum Master is on person and may change on each Sprint

The Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done
- Causing the removal of impediments to the Scrum Team's progress
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox

Scrum Team: Scrum Master

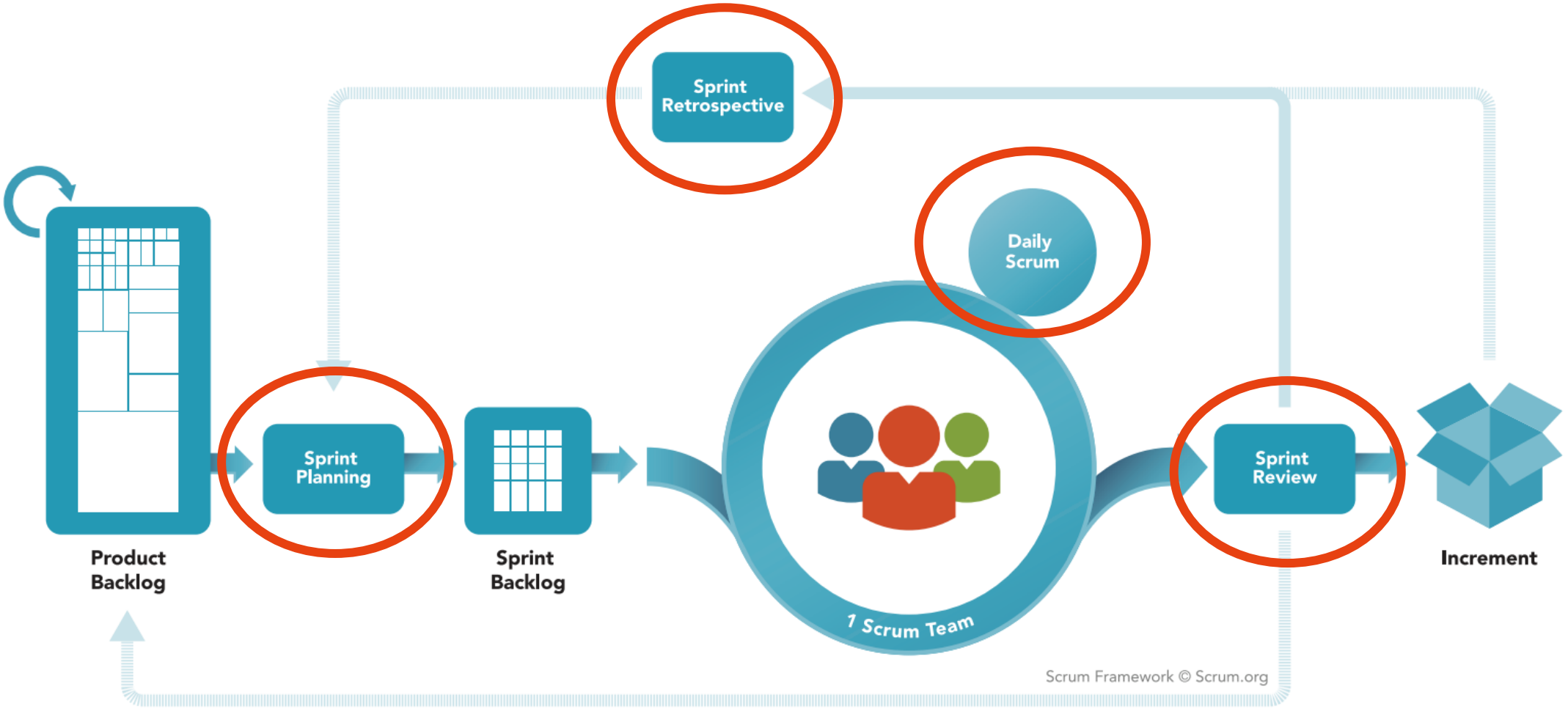
Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint

The specific skills needed by the Developers are often broad and will vary with the domain of work

However, the Developers are always accountable for:

- Creating a plan for the Sprint, the Sprint Backlog
- Instilling quality by adhering to a Definition of Done
- Adapting their plan each day toward the Sprint Goal
- Holding each other accountable as professionals

Scrum Framework: Events



Scrum Framework: Events

Events AKA Ceremonies, are non-negotiable activities

Have a pre-defined duration and must always be executed

They act as collaboration activities inside Scrum Team and help oh self-management

They are defined with clear goals to have all the Scrum Team focused

All of these events occurs inside “master” event called Sprint

Scrum Events: Sprint

Sprints are the heartbeat of Scrum, where ideas are turned into value

They are fixed length events of 2-4 weeks to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal
- Quality does not decrease
- The Product Backlog is refined as needed
- Scope may be clarified and renegotiated with the Product Owner as more is learned

Scrum Events: Sprint Planning

Why is this Sprint valuable?

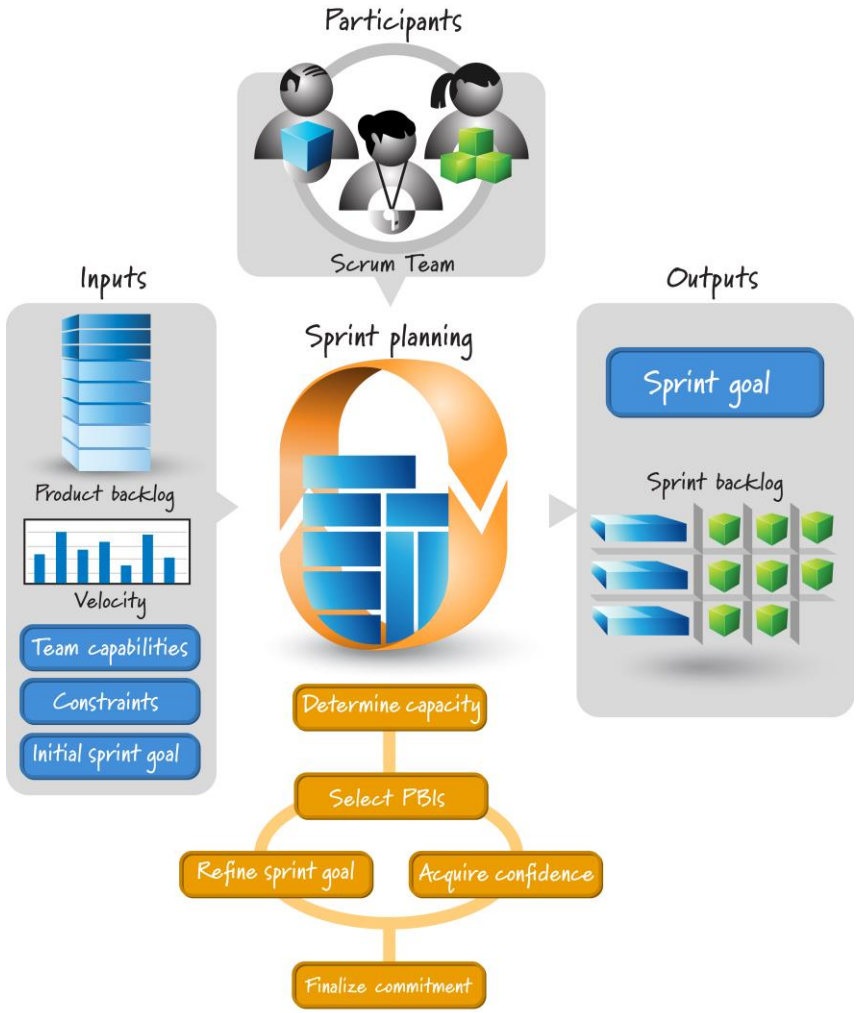
- PO proposes how the product could increase its value
- Scrum Team then collaborates to define a Sprint Goal
- Sprint Goal must be finalized prior to the end of Sprint Planning.

What can be Done this Sprint?

- Scrum Team select items from the Product Backlog. Items may be refined during this phase.
- Selecting how much can be completed within a Sprint may be challenging. Use past performance and upcoming capacity

How will the chosen work get done?

- Developers plan the work necessary to create an Increment based on each selected item
- This is often done by decomposing Product Backlog items into smaller work items of one day or less



Scrum Events: Daily Scrum

Daily meeting (can be standup ☺) to answer 3 questions

- What I did yesterday?
- What will I do today?
- Are there any impediments in my way?

Short session, answering directly to the questions and must be done every day

Ideally, all Scrum Team elements are present

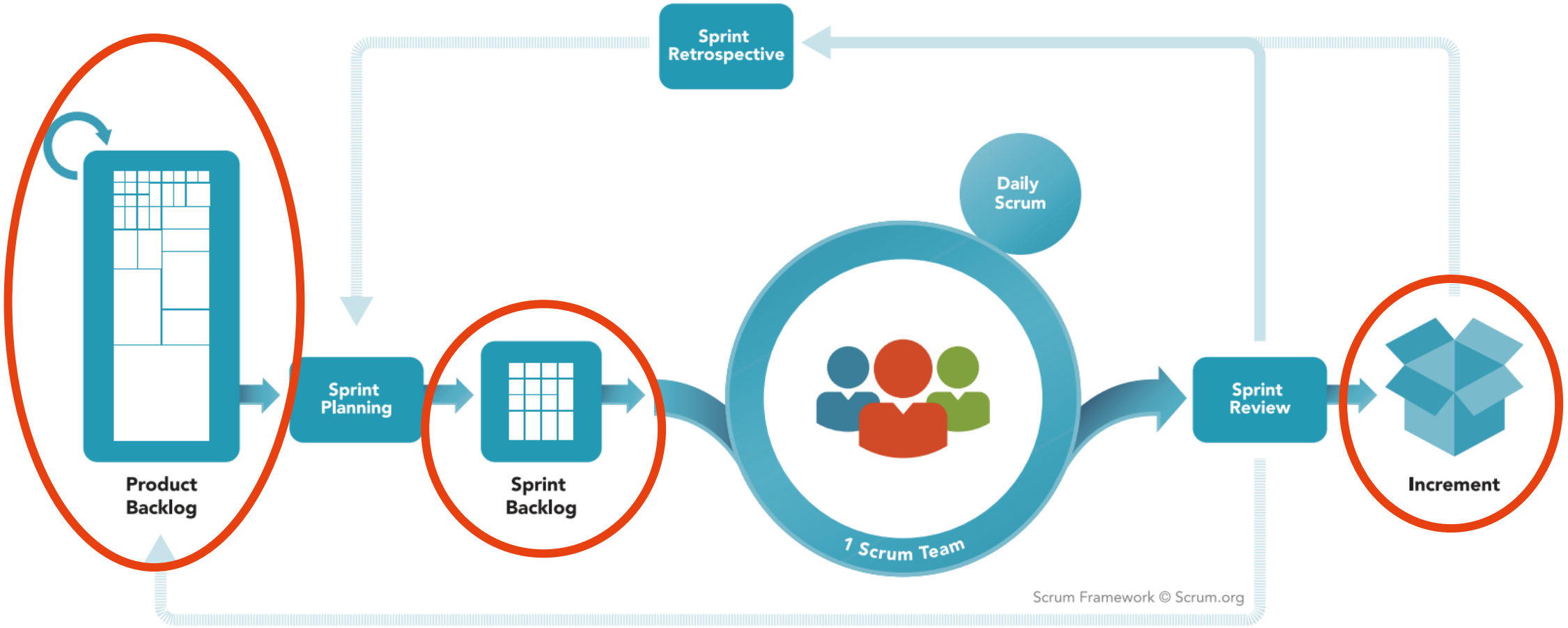
Can be at any time of the day but should always run at the same time

Maximum duration: 15 minutes

Scrum Events: Sprint Review

Scrum Events: Sprint Retrospective

Scrum Framework: Artifacts



Scrum Artifacts: Product Backlog

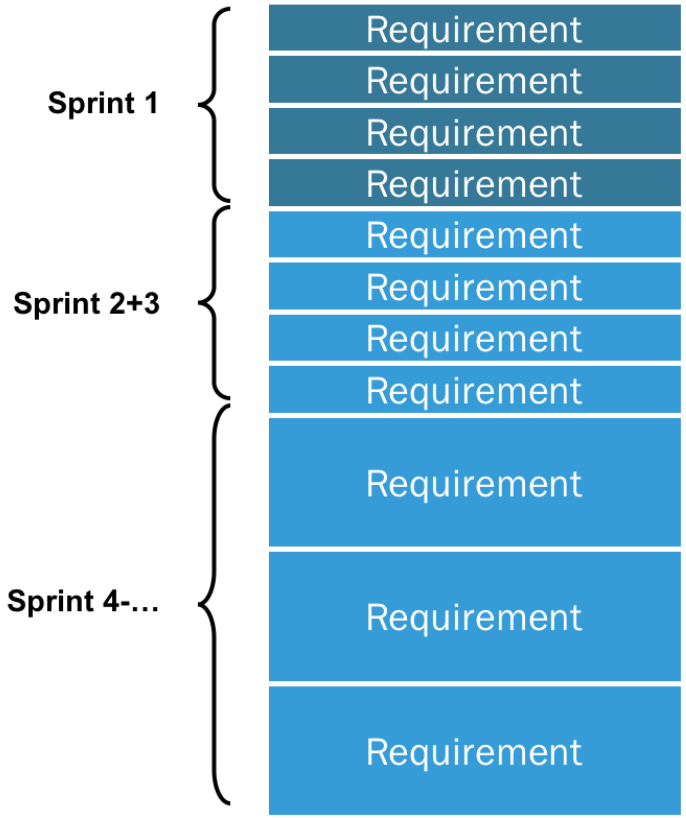
Single source of truth about what will (or not) be done

Each item as a description, priority and estimation of complexity (story points)

Product Owner is responsible to prioritize based on risk, value and effort

Properties of an item on top of Product Backlog

- Well-known by PO and the team
- Clear definition and detailed enough allowing to start work on it now
- With a size that can be fit inside a Sprint



Scrum Artifacts: Sprint Backlog

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

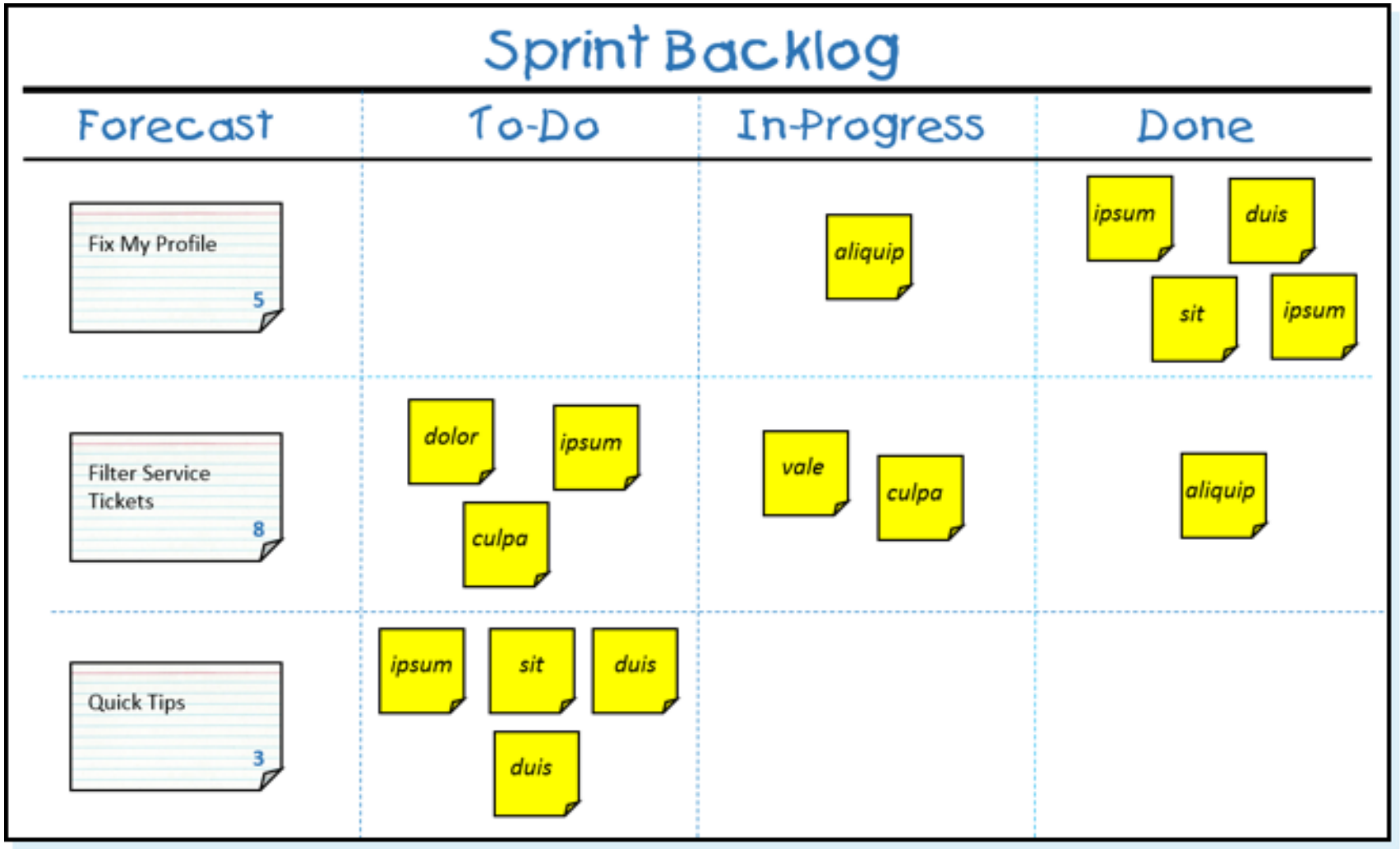
The Sprint Backlog is a plan by and for the Developers.

It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal.

It should have enough detail that they can inspect their progress in the Daily Scrum.

Each item as a description, priority and estimation of effort (hours)

Scrum Artifacts: Sprint Backlog



moOngy.

Scrum Artifacts: Increment

Increment is a concrete step toward the Product Goal

Each Increment is additive to all prior Increments, all Increments must work together, and the Increment must be usable

Represent the work done on items defined on Sprint Backlog

Never included unfinished work.

Work cannot be considered part of an Increment unless it meets the Definition of Done

Scrum Artifacts: Definition of Done (DoD)

Criteria defined and agreed by Scrum Team

Each product/project may have different DoD

DoD clearly defines when a Sprint Backlog item is considered “Done” by the team

DoD doesn’t mean customer acceptance but team acceptance

Example:

- Code complete
- Unit tests code coverage is X% or higher
- Automated build
- QA Tested

Kanban

Kanban

Kanban is a popular Lean workflow management method for defining, managing, and improving services that deliver knowledge work.

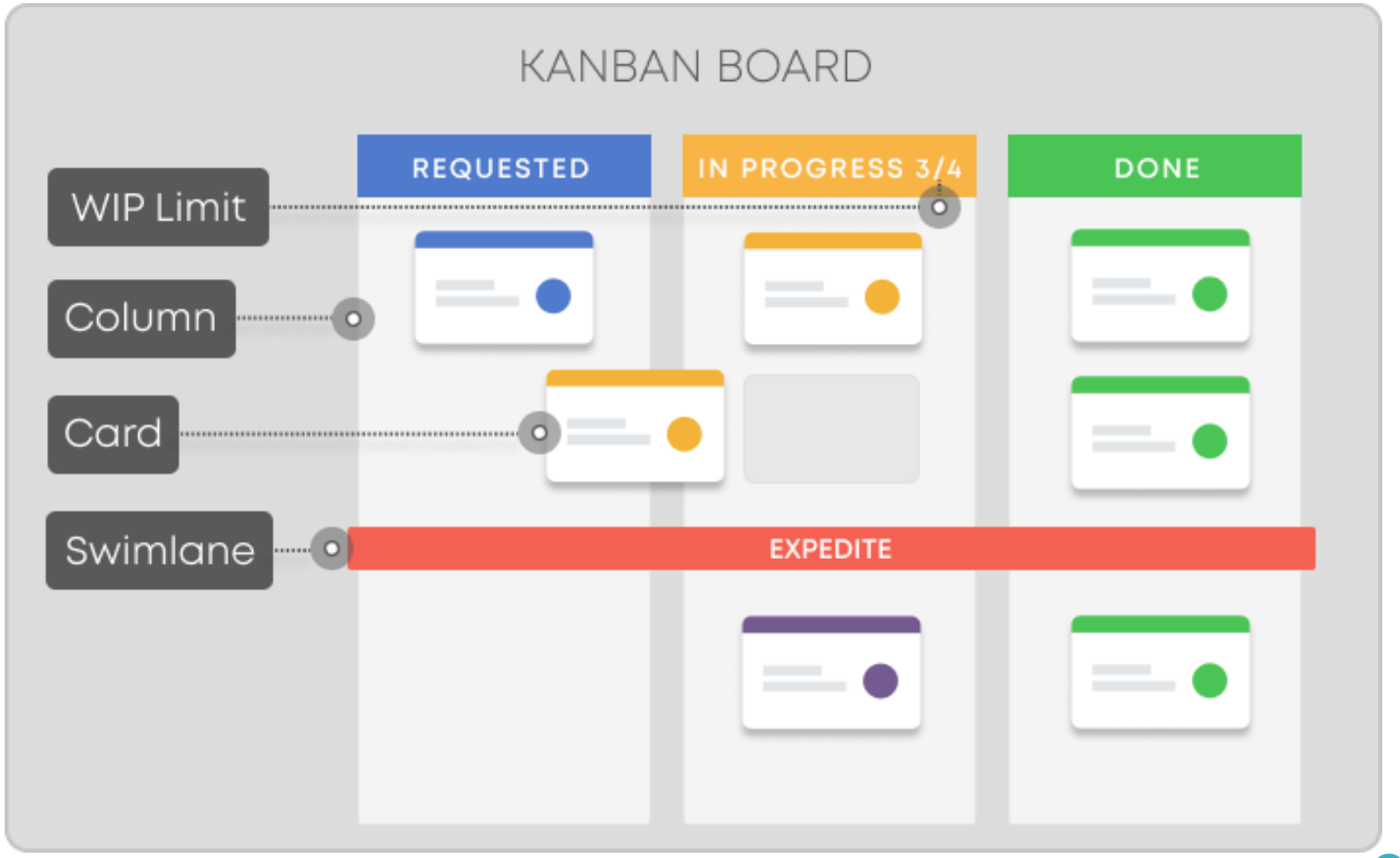
It helps you visualize work, maximize efficiency, and improve continuously.

Work is represented on Kanban boards, allowing you to optimize work delivery across multiple teams and handle even the most complex projects in a single environment

Originating from manufacturing (Toyota), it later became a territory claimed by Agile software development teams.

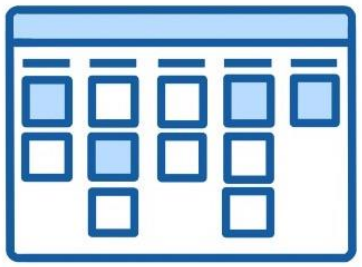
Recently, it started getting recognized by business units across various industries.

Kanban Boards



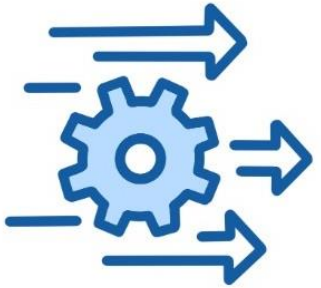
Kanban Boards

Kanban Practices



VISUALIZE

LIMIT WORK
IN PROGRESS



MANAGE
FLOW

MAKE POLICIES
EXPLICIT

IMPLEMENT
FEEDBACK LOOPS

IMPROVE
COLLABORATIVELY

Kanban Principles: Visualize the workflow

To visualize your process with a Kanban system, you will need a board with cards and columns

- Each column on the board represents a step in your workflow
- Each Kanban card represents a work item
- The Kanban board itself represents the actual state of your workflow with all its risks and specifications

The first and most important thing for you is understanding what it takes to get an item from a request to a deliverable product

Recognizing how work flows through your system will set you on the path to continuous improvement by making well-observed and necessary changes.

You may use swimlanes to differentiate your work on different subjects or highlight urgent work items

Kanban Principles: Visualize the workflow

To visualize your process with a Kanban system, you will need a board with cards and columns

- Each column on the board represents a step in your workflow
- Each Kanban card represents a work item
- The Kanban board itself represents the actual state of your workflow with all its risks and specifications

The first and most important thing for you is understanding what it takes to get an item from a request to a deliverable product

Recognizing how work flows through your system will set you on the path to continuous improvement by making well-observed and necessary changes.

Kanban Principles: Limit work in progress (WIP)

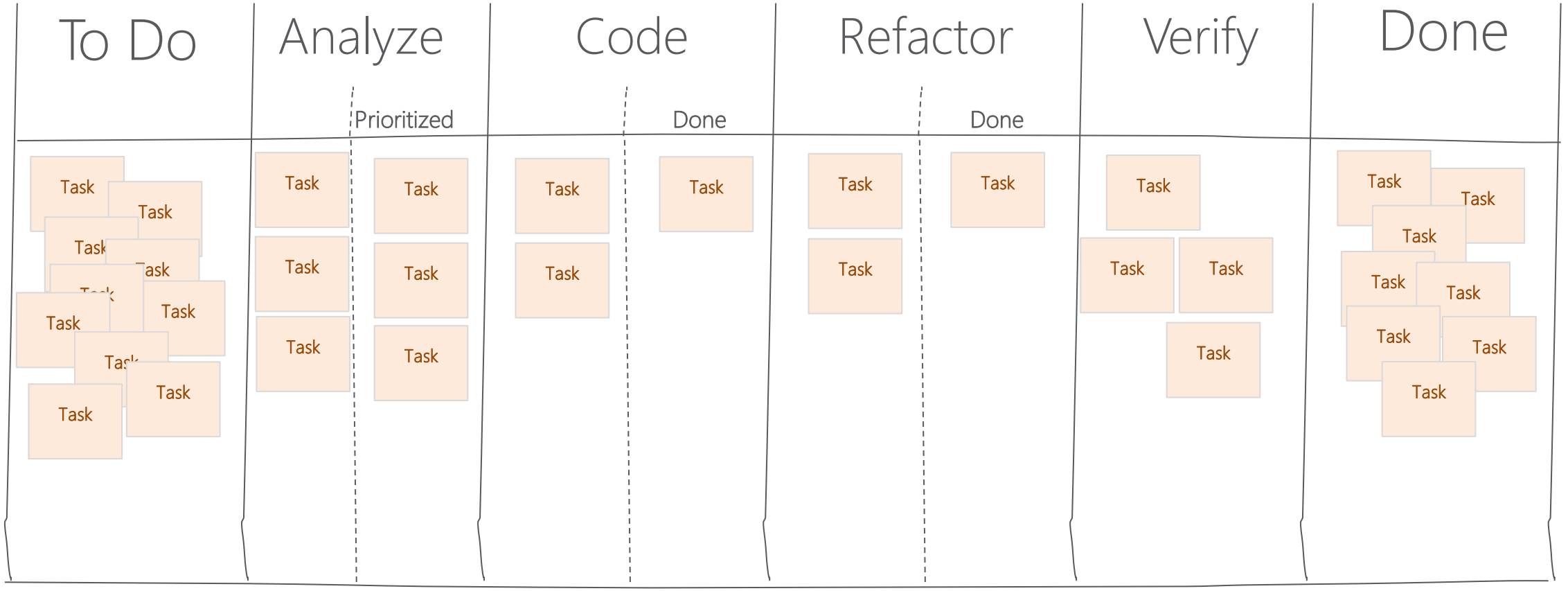
Limiting WIP means implementing a pull system on parts or the complete workflow

Setting maximum items per stage ensures that a card is only “pulled” into the next step when there is available capacity

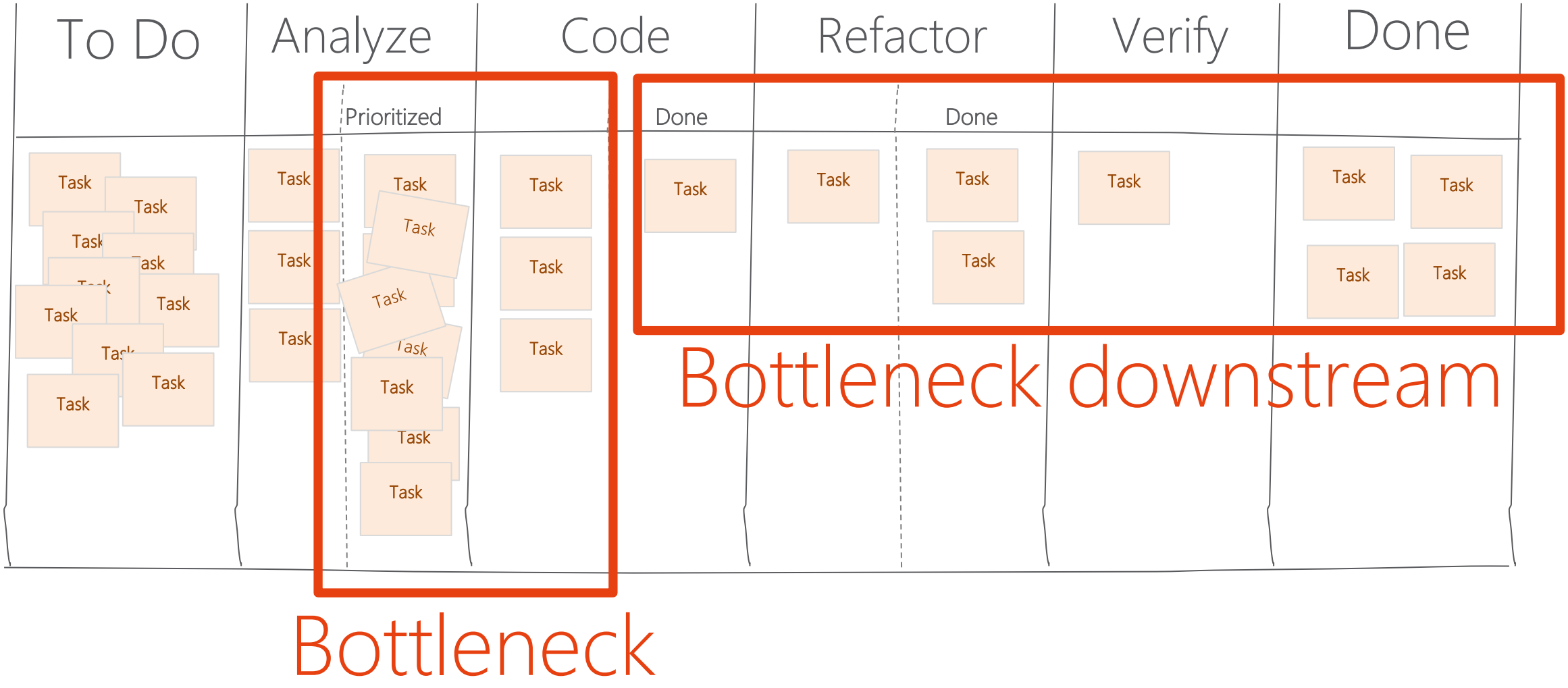
Such constraints will quickly illuminate problem areas in your flow so you can identify and resolve them

If there are no work-in-progress limits, you are not doing Kanban

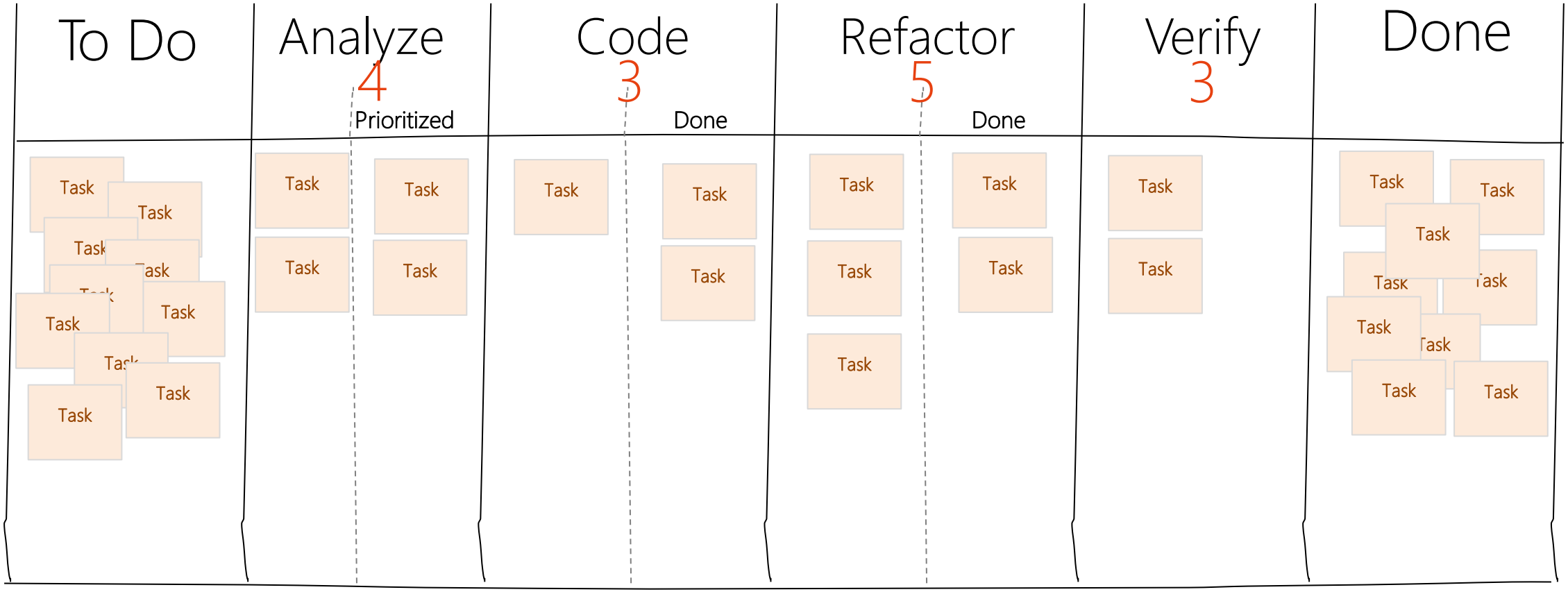
Kanban Principles: Limit work in progress (WIP)



Kanban Principles: Limit work in progress (WIP)



Kanban Principles: Limit work in progress (WIP)



Kanban Principles: Manage Flow

Managing the flow is about managing the work but not the people

By flow, we mean the movement of work items through the production process at a predictable and sustainable pace.

One of the main goals when implementing a Kanban system is to create a smooth, healthy flow

Instead of micro-managing people and trying to keep them busy all the time, you should focus on managing the work processes and understanding how to get that work faster through the system (Autonomy and Alignment)

Kanban Principles: Make process policies explicit

You can’t improve something you don’t understand

Therefore, your process should be clearly defined, published, and socialized. People would not associate and participate in something they do not believe would be useful.

When everyone is familiar with the common goal, they would be able to work and make decisions regarding a positive impact

Sparse, visible, well-defined, and subject to change (if/when needed), work policies have the power to boost people’s self-organization.

Kanban Principles: Implement feedback loops

For teams and companies that want to be more agile, implementing feedback loops is a mandatory step. They ensure that organizations are adequately responding to potential changes and enable knowledge transfer between stakeholders.

Kanban suggests the use of cadences (feedback loops) at a team level

An example of a team-level cadence is the daily Team Kanban Meeting for tracking the status and the flow of work.

It helps to identify available capacity and potential for increasing the delivery pace.

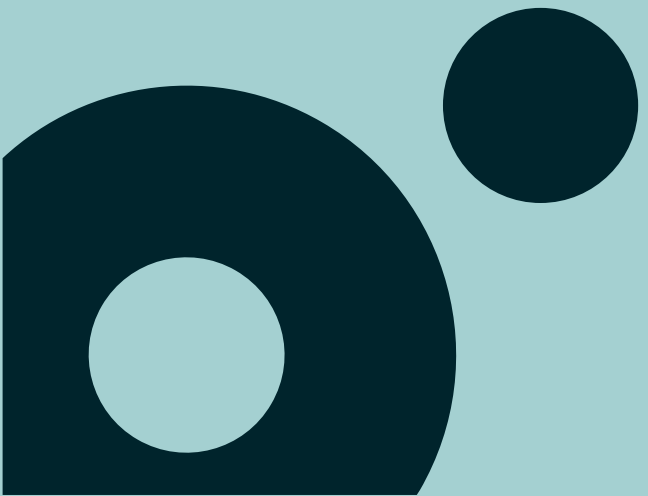
It takes place in front of the Kanban board, and every member tells the others what they did the previous day and what they will be doing today.

Kanban Principles: Improve collaboratively

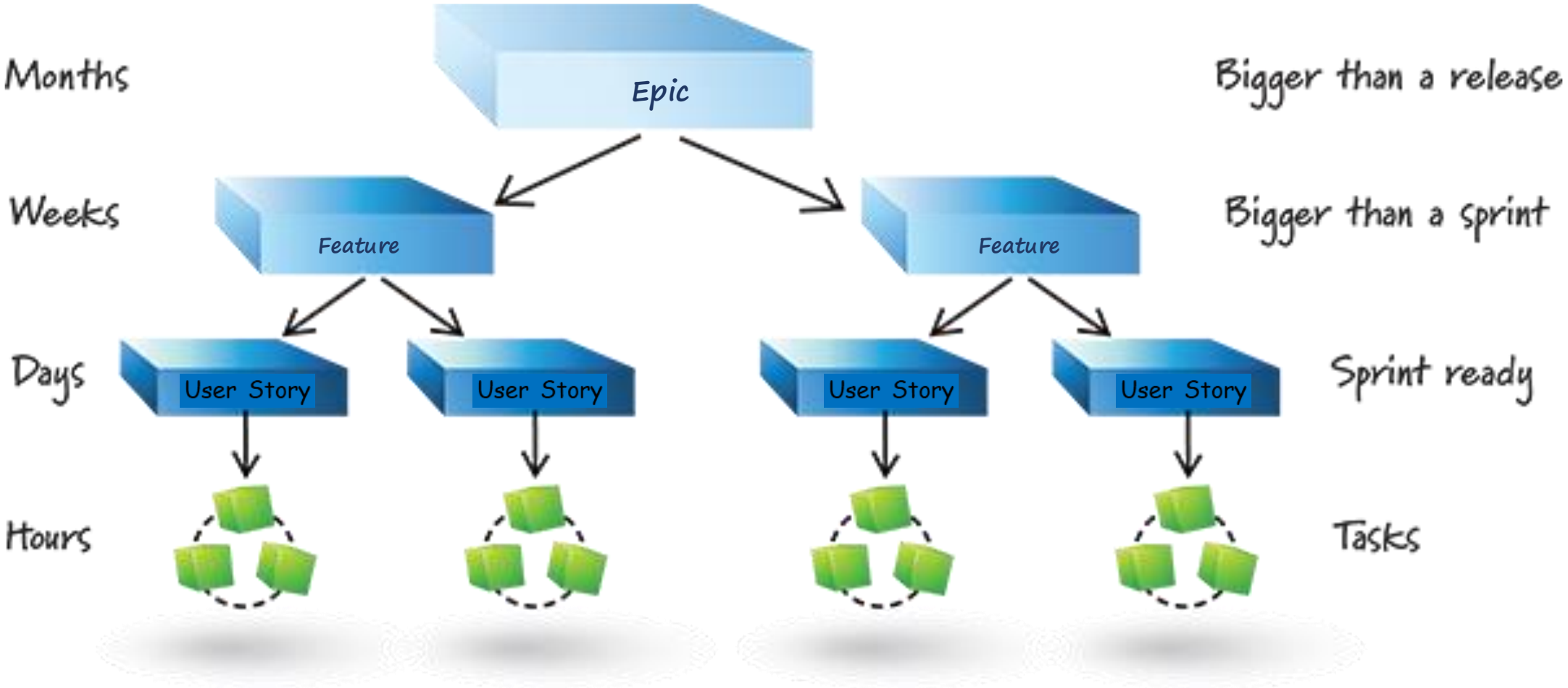
The way to achieve continuous improvement and sustainable change within an organization is through collaboratively implementing changes based on scientifically proven methods, feedback, and metrics.

Cultivating an organizational culture where every hypothesis is proven to have positive or negative results is crucial for developing a mindset focused on improvement through evolutionary change.

Best Practices



Best Practices: Release Planning



Best Practices: Prioritization

MUST HAVE

The most vital things you can't live without

SHOULD HAVE

Things you consider as important, but not vital

COULD HAVE

Things that are nice to have

WON'T HAVE

Things that provide little to no value you can give up on

Best Practices: Estimation

Use story points to estimate complexity (User Stories)

Different classification number: Sequential, Fibonacci, T-Shirt Sizes

Planning Poker: Everyone gives its classification without knowing the others. When everyone shows up, an agreement must be reach

Use hours to estimate effort (Tasks)

Continuous Planning: Best Practices

User Story Definition

INVESTing in well-written user stories
Independent
Negotiable
Valuable
Estimable
Small
Testable

Backlog Grooming

Review backlog to re-prioritize and refinement
Remove what don't make sense anymore

Sprint Lifecycle

Follow Agile Ceremonies
Sprint Planning, Dailies, Sprint Review, Retrospective
Retrospective is the opportunity to improve on planning & tracking process

moOngy.
Minds on the move

Rua Sousa Martins, nº 10
1050-218 Lisboa | Portugal