ALTIMETRIK

SIMPLIFY TECHNOLOGY | AMPLIFY POSSIBILITY

Agile ways of working

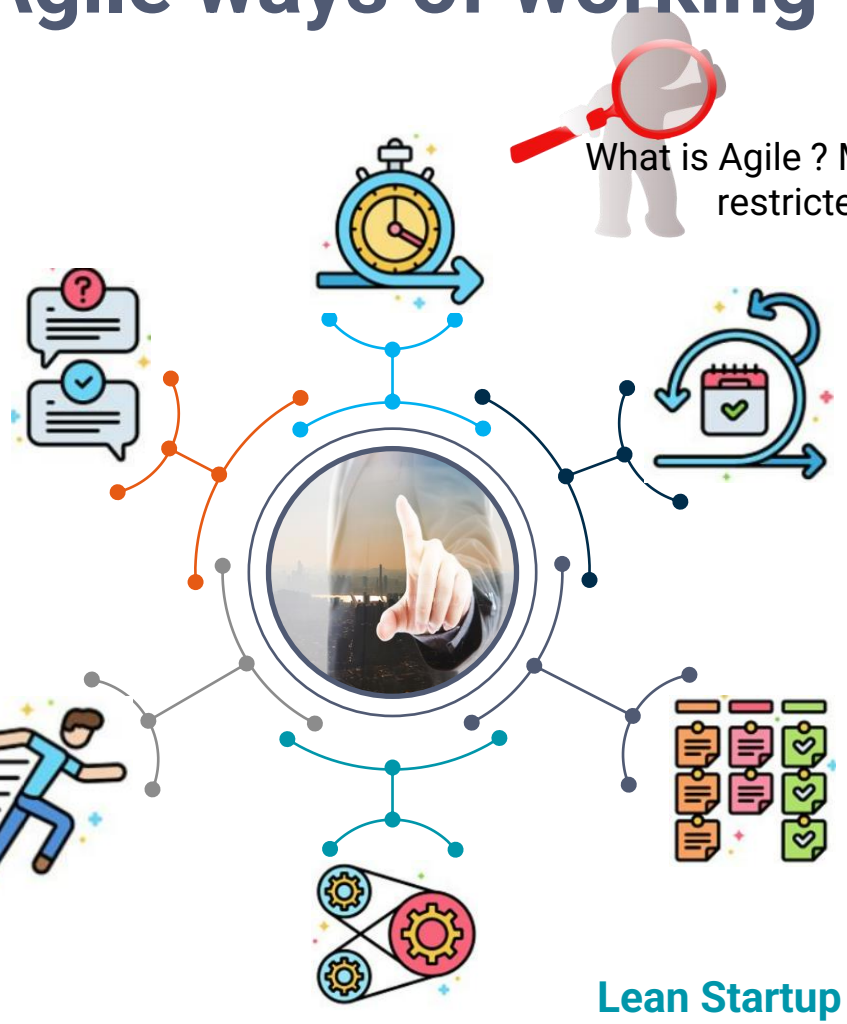
Chandrashekar Reddy

Agile Coach

Feb 2024

Agenda

Agile ways of working



Agile
What is Agile ? Mindset or Framework, is it restricted to only Software

Question and Answers
This is just the beginning of a transformation

Scrum
One of the most commonly used Agile Framework, the 3+5+3 formula

Shifting in behaviours and mind-set
An Agile model consists of cross-functiona teams formed around a certain purpose

Kanban
The simple and most effective ways to get things quickly, you will see this near you

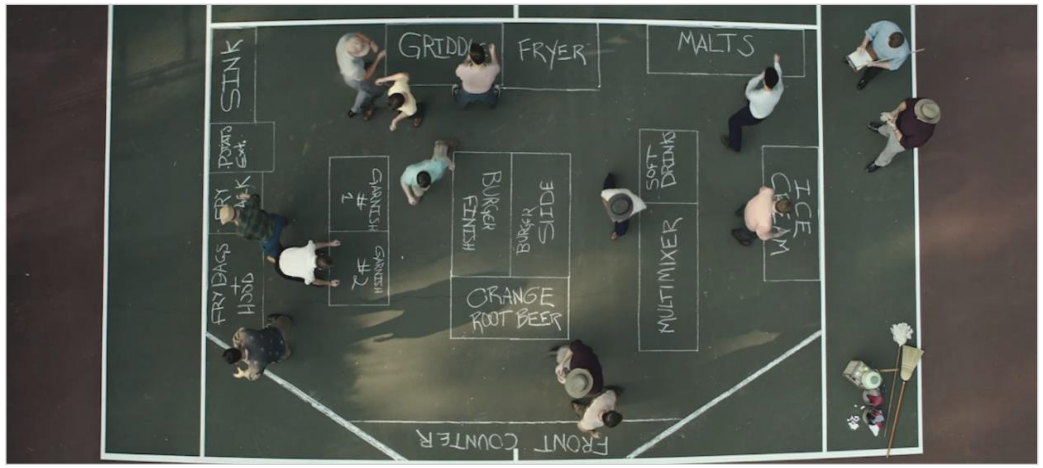
Lean Startup & MVP
A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

The founder

In digital Design Thinking and Agile are intertwined

Design Thinking

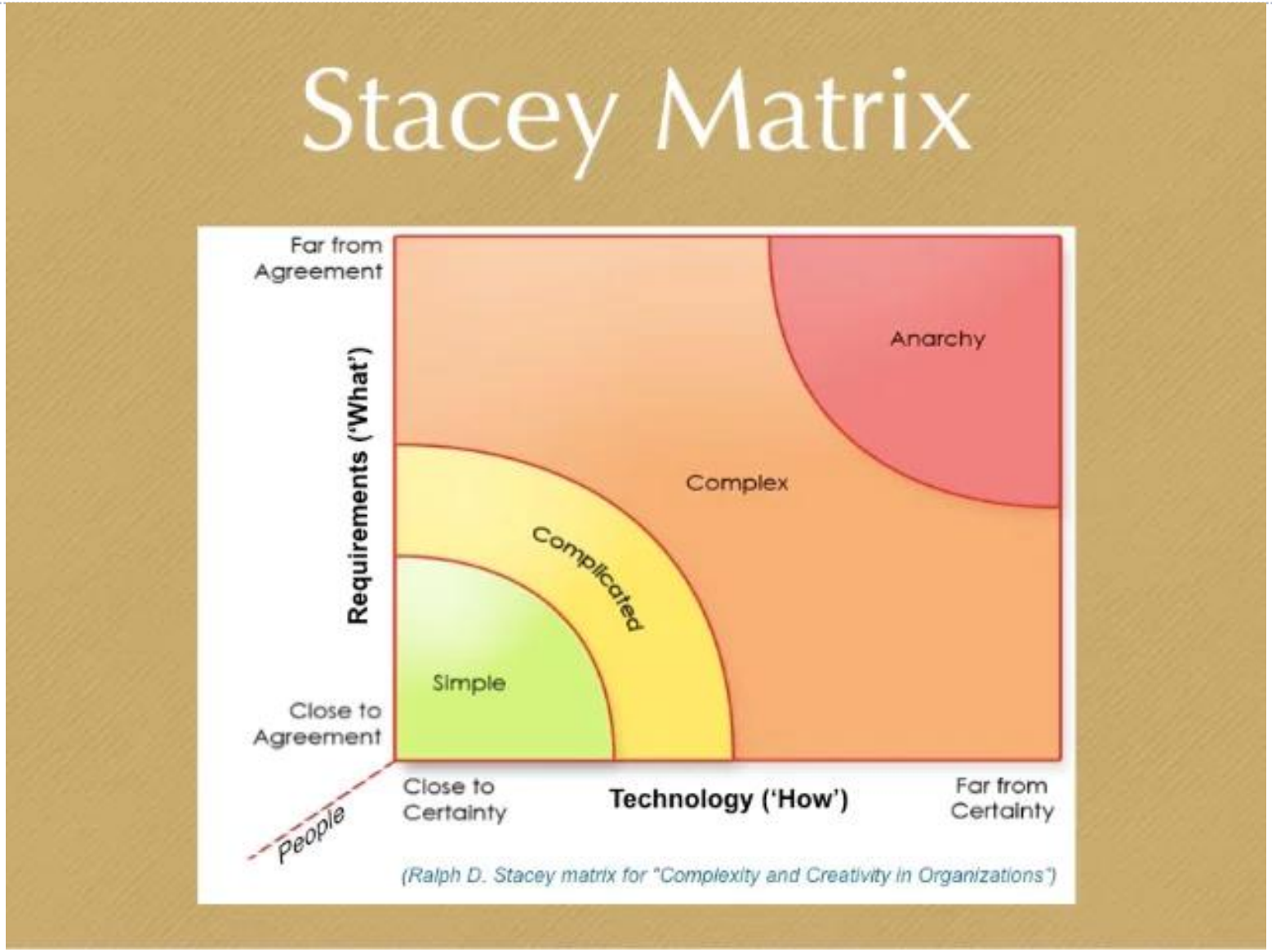
Design Thinking is a methodology for creative problem solving



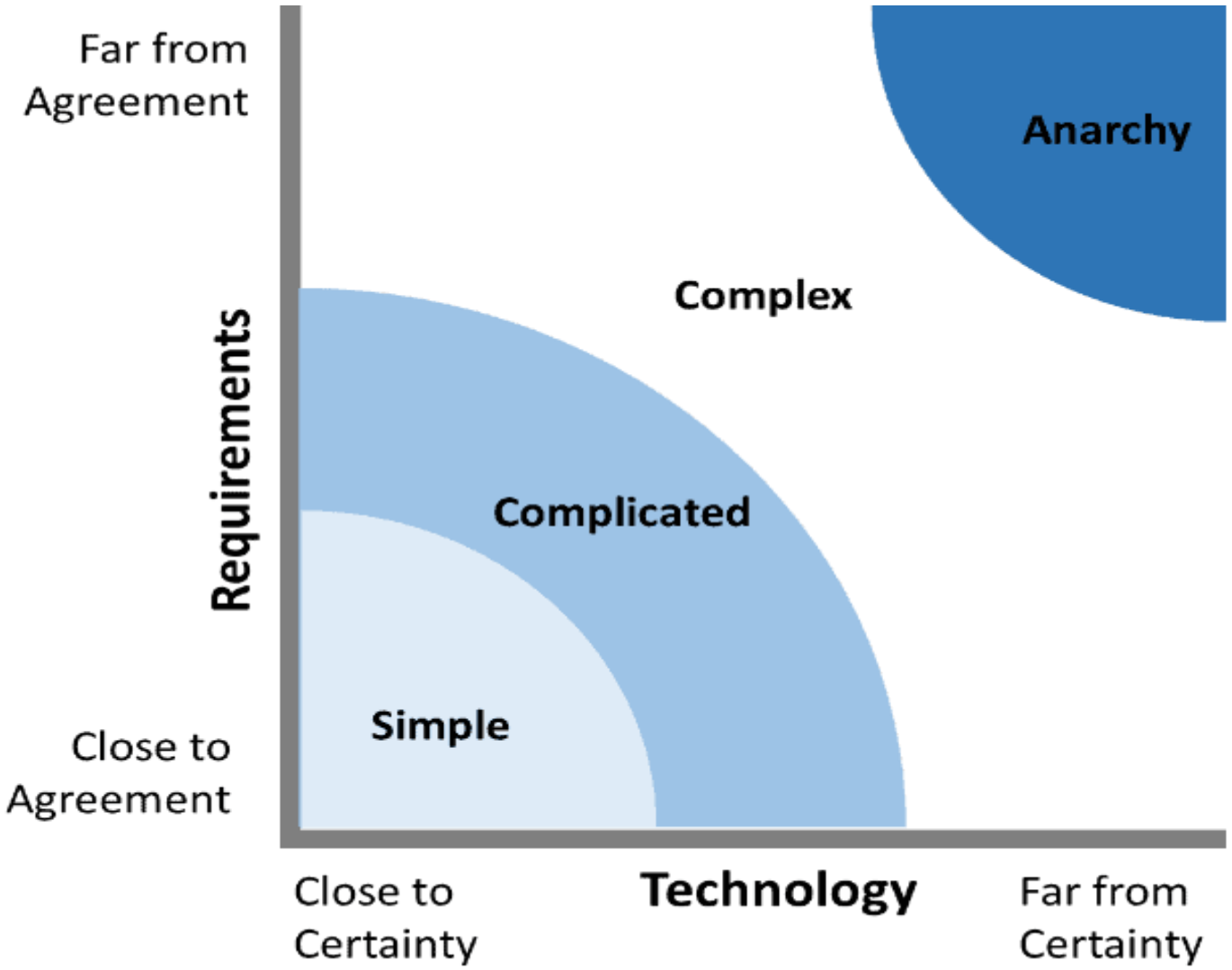
Agile

Ability to create and respond to change – ways of working that enable that ability

Working Agreement



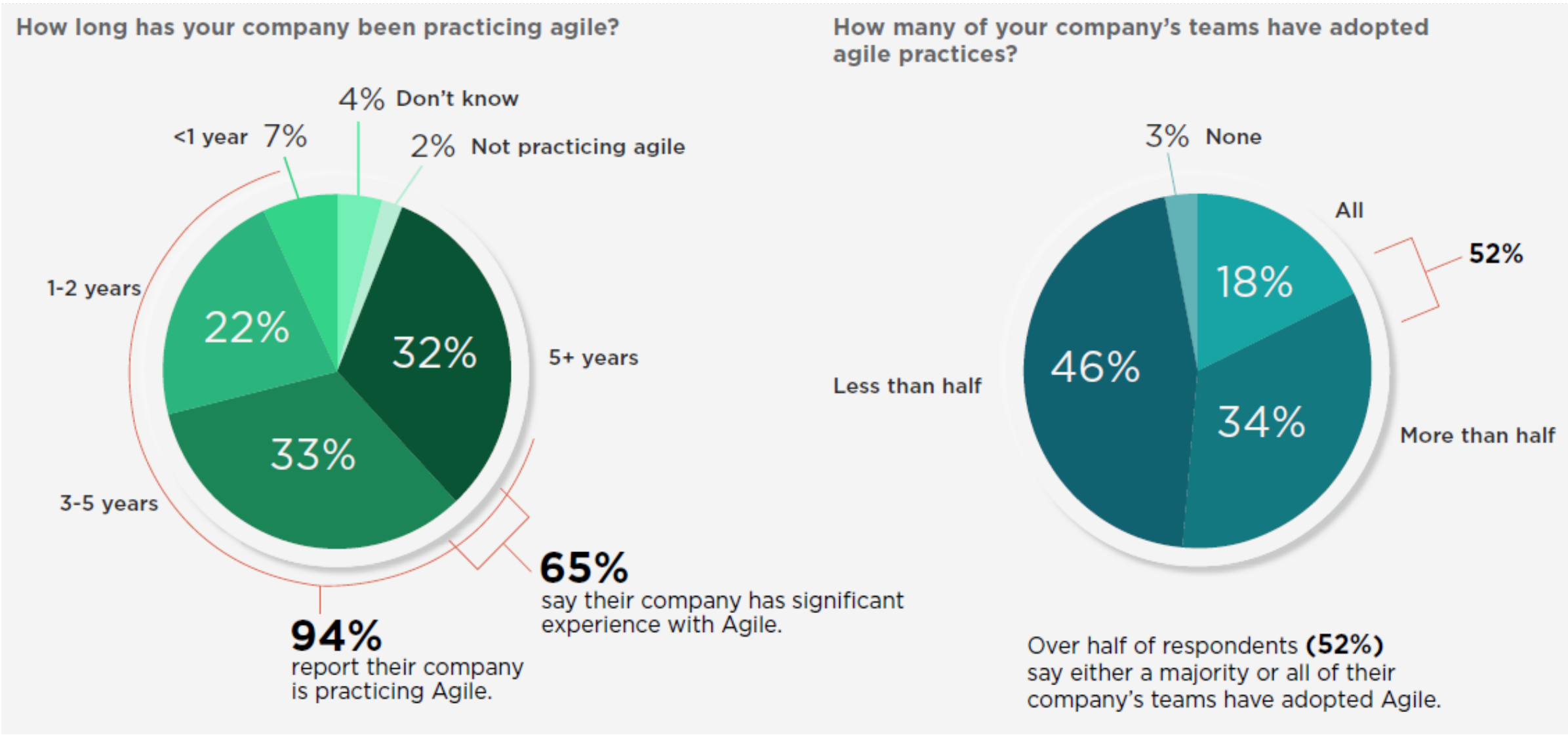
What is the 3rd Dimension to determine the project complexity



Ralph Douglas Stacey was a British organizational theorist and Professor of Management at Hertfordshire Business School, University of Hertfordshire, in the UK and one of the pioneers of enquiring into the implications of the natural sciences of complexity for understanding human organisations and their management.

Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey in *Agile Software Development with Scrum* by Ken Scwaber and Mike Beedle.

Company Experience with Agile

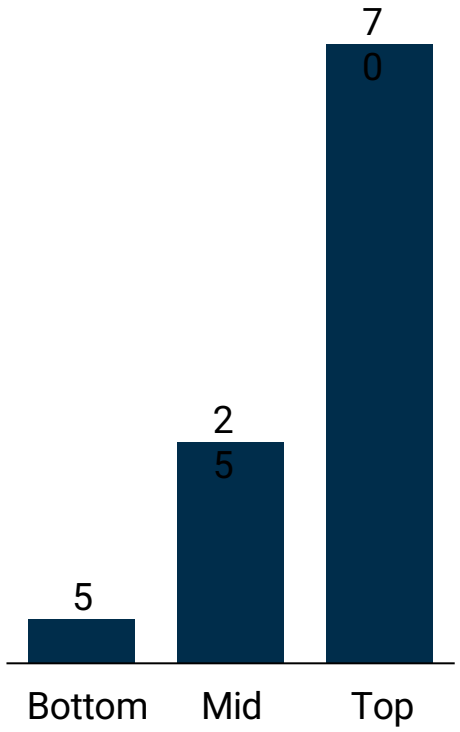


Source:**6** Digital.ai | 15th State of Agile Report

Agile companies show superior health, financial and operational performance

Health

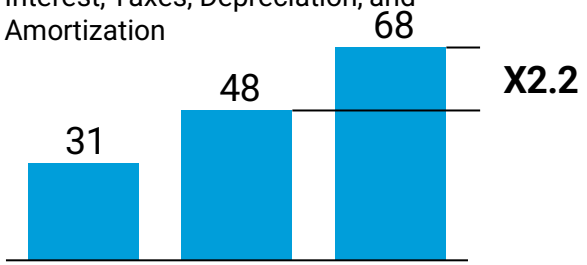
% of agile companies by quartiles of org. health



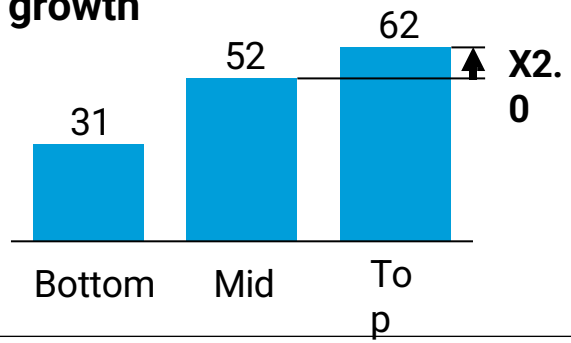
Financial

% of agile companies with performance above median

EBITDA= Earnings Before Interest, Taxes, Depreciation, and Amortization



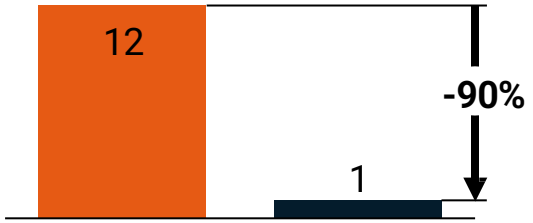
Book value growth



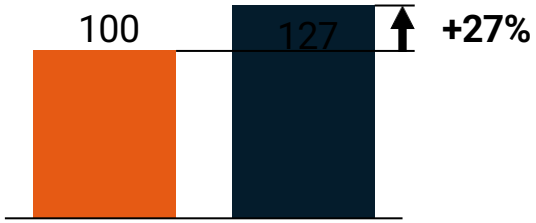
Operational

Companies with operational performance

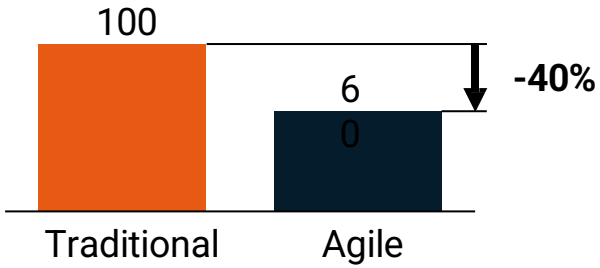
Time to market (months)



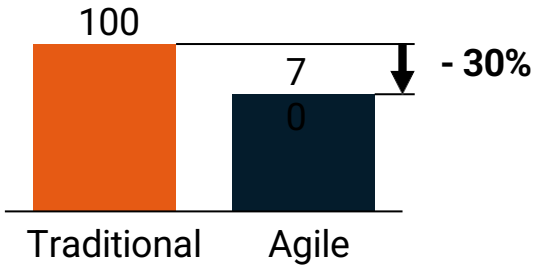
Productivity (percent)



Change cost (percent)



"Change" headcount (percent)



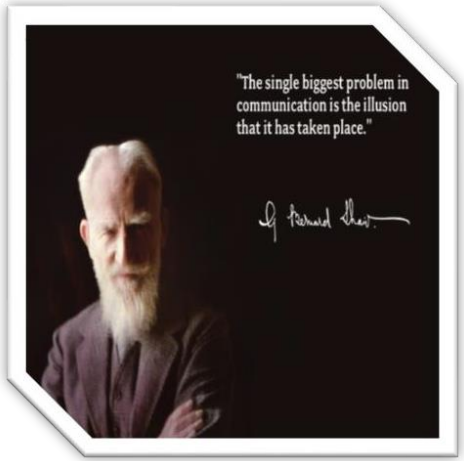
Source: McKinsey Corporate Agility research

Agile has emerged as an answer to several issues that riddle modern organizations

Inefficient structures

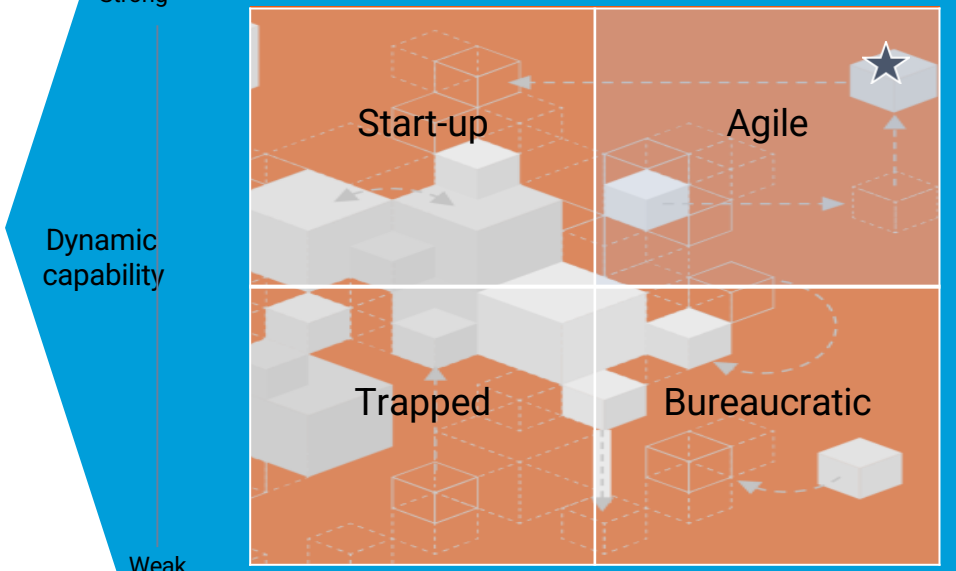
Bureaucratic culture

VUCA environment



Miscommunication

Agile organizations crack the paradox of being both stable and dynamic at the same time



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Agile can be found on 3 different flight levels

	Team level in software delivery	Team level in other business functions	At the organizational level
			
Characteristics	Empowered and self organizing teams Aiming for faster time to market/turn around time and de-risking outcome User and customer centered Business value focused Technical excellence Constantly Improving	Empowered and self organizing teams Aiming for faster time to market/turn around time and de-risking outcome User and customer centered Business value focused Constantly improving	Organization structured to support self organizing teams and scale the chosen team level framework Defines structure of the organization Defines mechanism to insure value delivery (e.g. by Objective Key Results)
Frameworks and Approaches	Scrum (products) Kanban (services) Extreme programming (XP)	Scrum (for products) Kanban (for services)	Spotify engineering culture (aka Spotify model) LeSS SAFe
Examples	Maintenance scheduling app development Production Optimization support apps Predictive HSE alert apps	HR Marketing IT Operations	Spotify Google ING

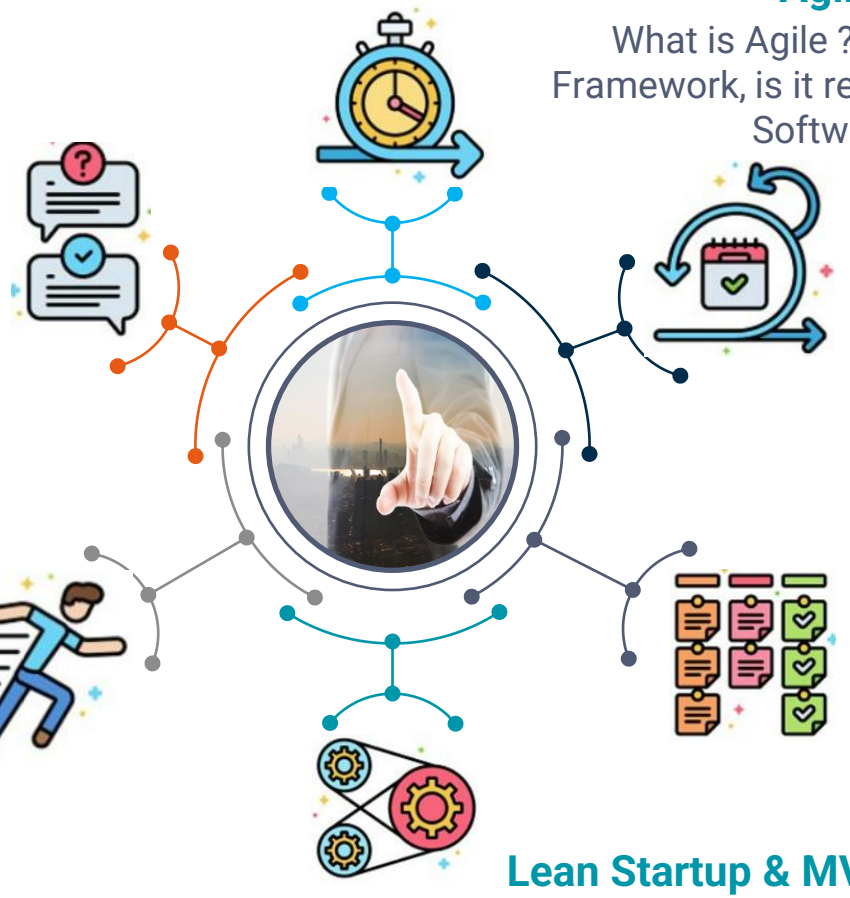
11

Agenda

Agile ways of working

Agile
What is Agile ? Mindset or Framework, is it restricted to only Software

Question and Answers
This is just the beginning of a transformation



Scrum
One of the most commonly used Agile Framework, the 3+5+3 formula

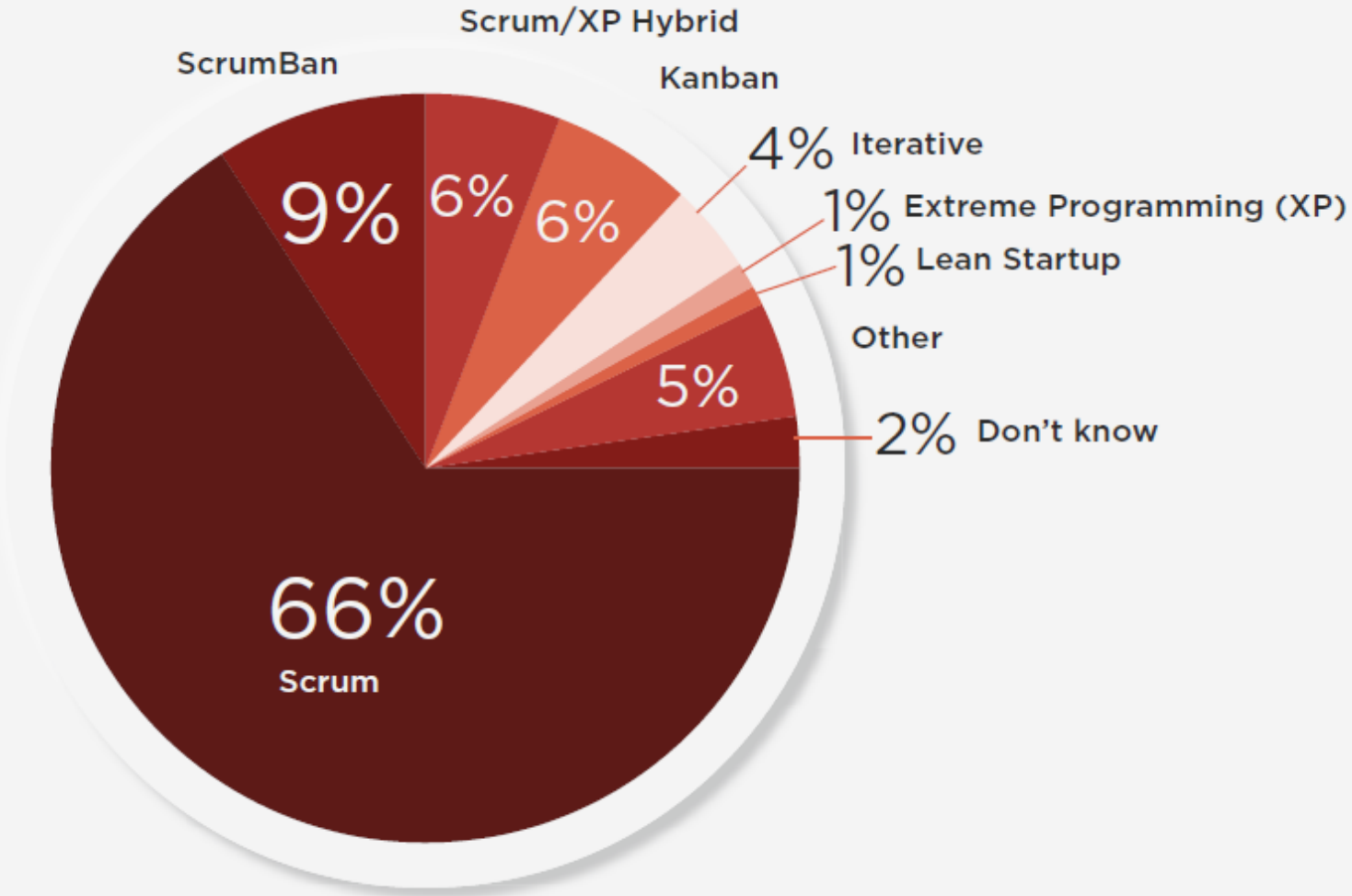
Shifting in behaviours and mind-set
An Agile model consists of cross-functional teams formed around a certain purpose

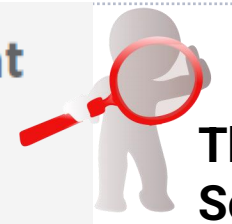
Kanban
The simple and most effective ways to get things quickly, you will see this near you

Lean Startup & MVP
A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

Scrum is the most common agile approach

Which Agile methodology do you follow most closely at the team level?





The survey once again highlighted Scrum as the most popular Agile approach Scrum and Scrum/XP Hybrid (81%)

In terms of agile techniques and practices, at least four out of five respondents cited:

- Daily standups **87%**,
- Retrospectives **83%**, and
- Sprint/iteration planning **83%** were used.

In addition:

- Kanban boards **77%**,
- Task boards **67%**, and
- Spreadsheets **66%** are all widely used for agile planning.

Source: **13** Digital.ai | 15th State of Agile Report

15

Where does Scrum come from? Who invented it?

A **scrum** (short for scrummage) is a method of restarting play in **rugby** that involves **players** packing closely together with their heads down and attempting to gain possession of the ball

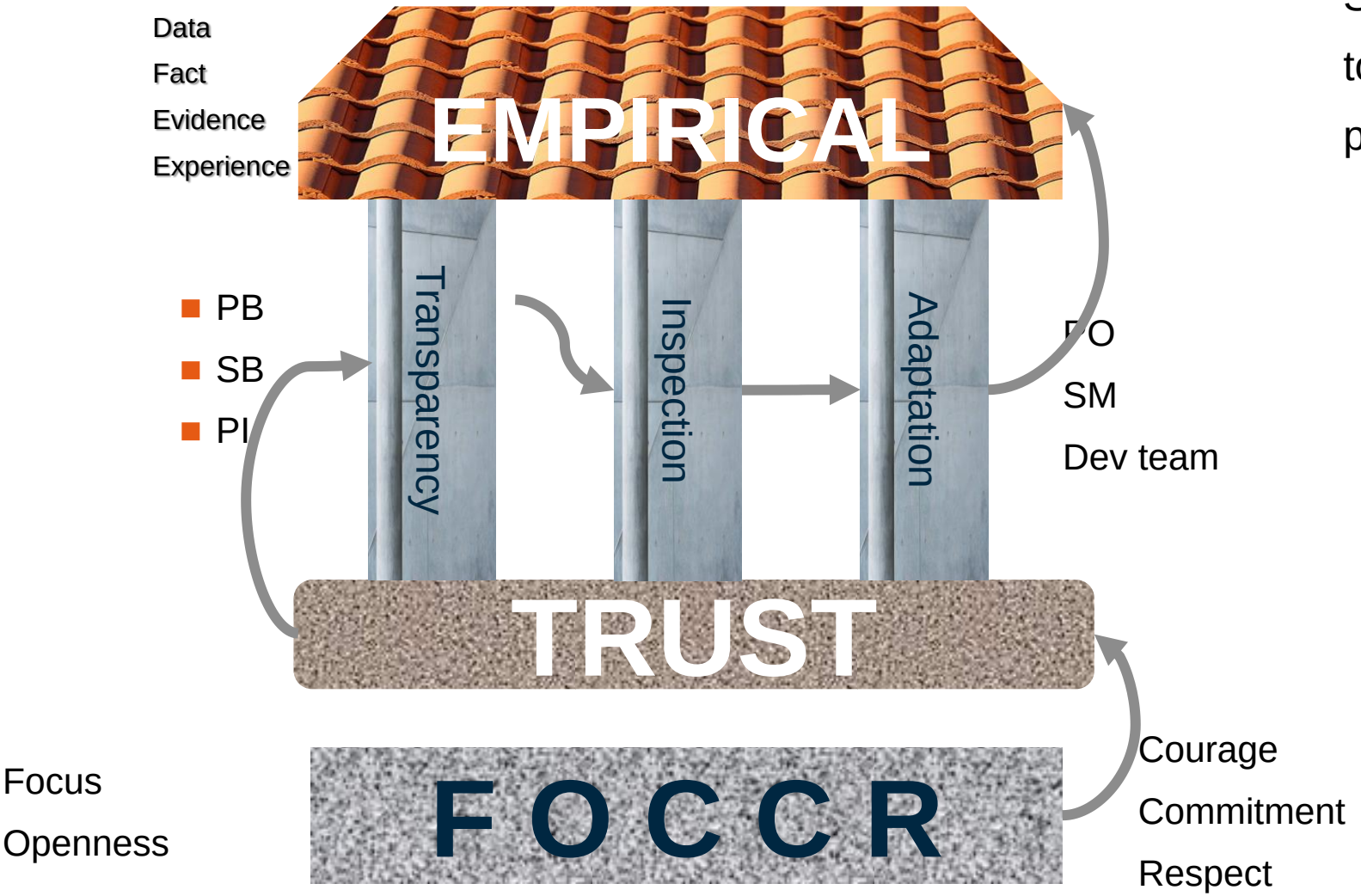
Hirotaka Takeuchi and **Ikujiro Nonaka** introduced the term scrum in the context of **product development** in their 1986 **Harvard Business Review** article, "New New Product Development Game"

In 1995, Sutherland and Schwaber jointly presented a paper describing the Scrum framework at the Business Object Design and Implementation Workshop held as part of **Object-Oriented Programming, Systems, Languages & Applications '95** (OOPSLA '95) in Austin, Texas

Jeff Sutherland

Ken Schwaber

What is Scrum /Empiricism



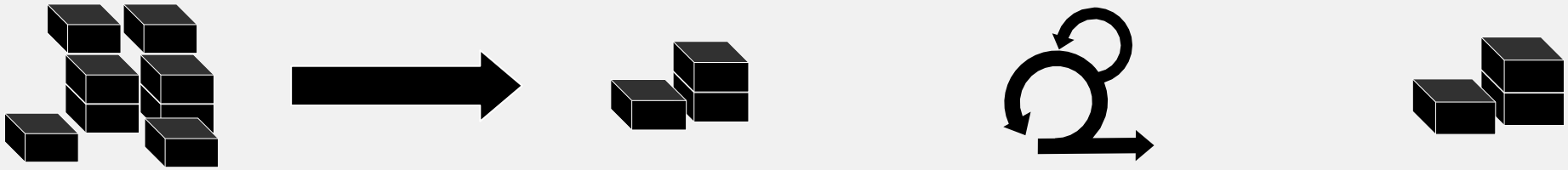
Scrum is a framework to solve complex problems

3+5+3 = 11

a funny Scrum Master movie with Jeff Sutherland

Scrum is a framework within which people can address complex adaptive problems

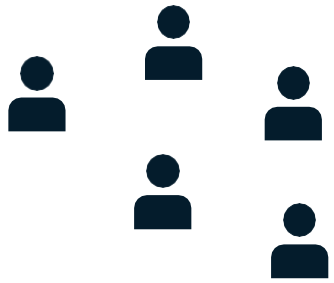
	Development team	Product Owner	Scrum Master		
Roles					
Events	**Sprint Planning** Review product backlog Estimate sprint backlog Commit	**Daily Stand up** What did I do yesterday? What do I plan to do today? What are my impediments?	**Sprint Review** Demo features to all Share key project metrics	**Sprint Retrospective** Done after each sprint Aims to improve the process for next sprint	**Backlog Refinement** During each sprint Socialize up-coming stories with the team
Artefacts	**Product Backlog** User focused prioritized product features	**Sprint Backlog** Features for a Sprint estimated by team Team commitment		**Product Increment** Working software ready for deployment	



Time-boxed “Sprint” Cycles

Scrum knows three different roles

Subject Matter Experts and Stakeholder



Product Owner

Maximize Value

Backlog ordered

Items clearly understood

Roadmap transparent

Scrum Team

Scrum Master

Guide to Scrum Servant

Leader Product Owner

Development Team

Organization

Development Team

Build the product increment

Cross-Functional Self-Organizing

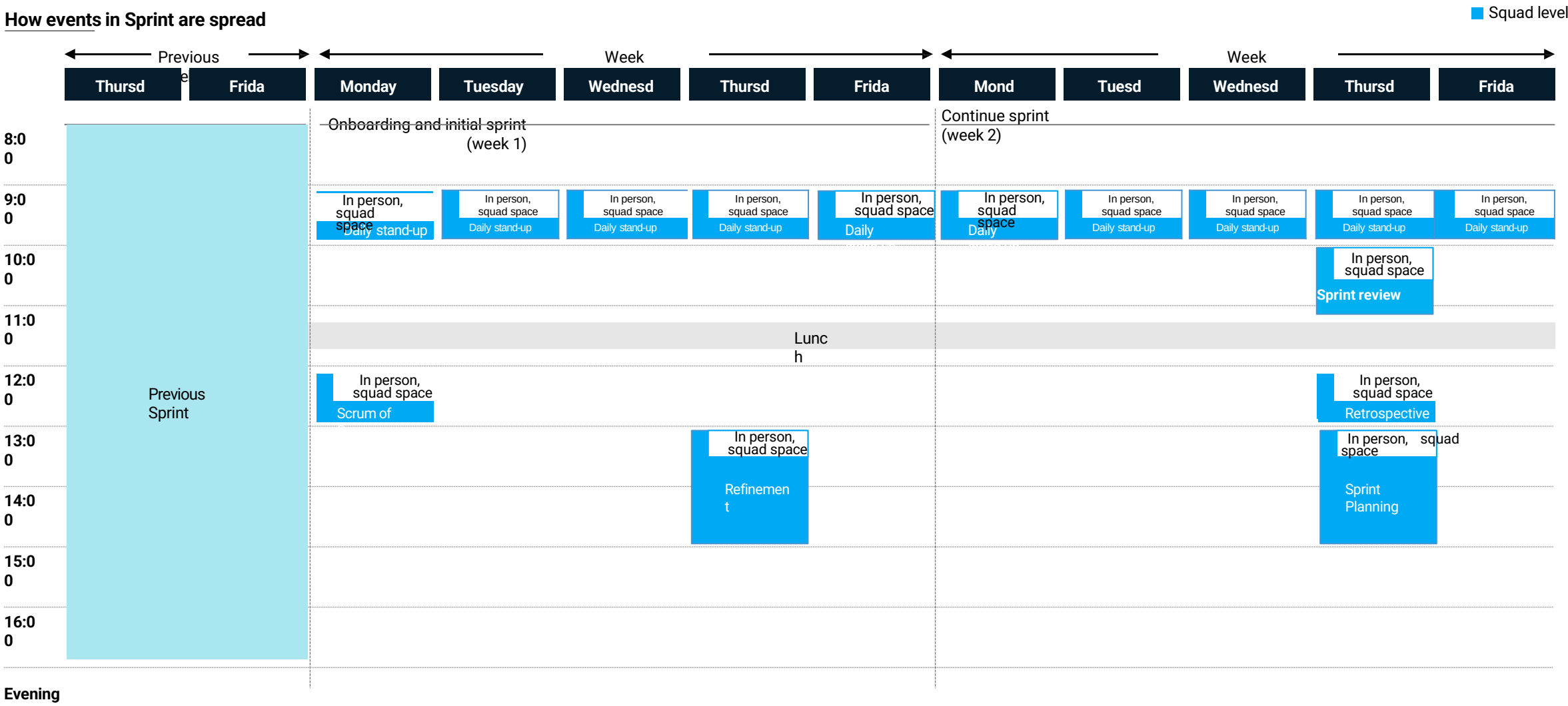
Accountable as a team

Overview of the Scrum events during a sprint

	Sprint Planning	Daily	Review / Demo	Retrospective
Goal	Update the backlog How important? How costly? What's the Sprint goal	Plan the work for the day	Show the accomplished work	How to improve
Frequency	1 per Sprint	Daily, first thing of the day	1 per Sprint	1 per Sprint
Attendees	Development Team Product Owner Scrum Master	Participation of the Product Owner is determined by the team	Development-Team Product Owner Scrum Master, Stakeholders (Asset)	Development Team Product Owner Scrum Master
Time box	2 hours per week of a sprint	15 minutes	1 hour per week of a sprint	1 hour per week of a sprint
Agenda	**Part I = What** Whats in the backlog? Capacity? Changes in Business? Changes in technology? **Part II = How** Discussion	What have you done yesterday? What are you going to do today? Are there any impediments?	Intro Report Demo Discussion – feedback from e.g. field Ops or asset operators	What did we achieve? What do we want to improve? What did we learn? What puzzles us? Define Actions and owners

Scrum events plan for a 2 weeks sprint

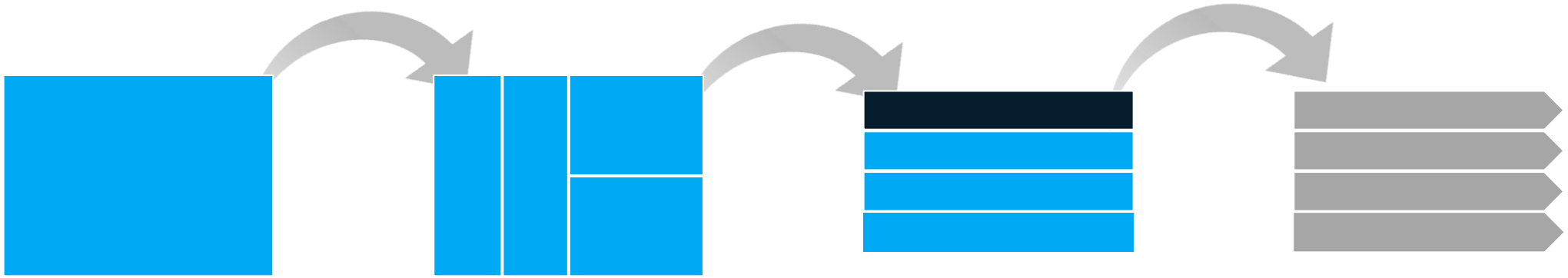
How events in Sprint are spread



Evening

Stable events and meetings that shouldn't be moved and define the Scrum rhythm

Items in a backlog can be structured and categorized in various ways



Product Backlog

Prioritized list of items that is known to be needed in the product

Features list for optimizing planning and logistics of predictive maintenance activities

EPIC

Overarching theme of a set of features

Dynamic maintenance schedule optimization

User Story

The smallest piece of the product that still has a business value

As a maintenance scheduler, I want to see the most impactful jobs ranked

Task

Technical tasks that have no business value, but together deliver a User Story

Display the latest, ranked list of maintenance jobs by impact

$$$** **$$** **$

Business Value → **Single work item**

Agenda

Agile ways of working

Agile
What is Agile ? Mindset or Framework, is it restricted to only Software

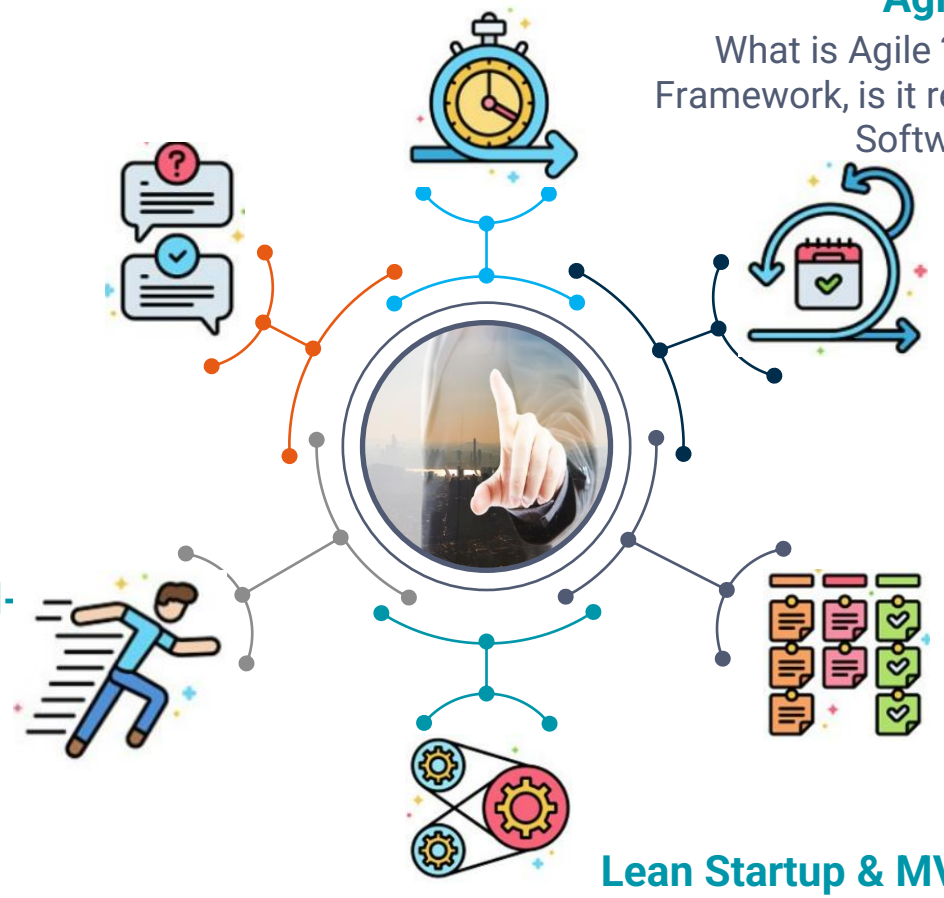
Scrum
One of the most commonly used Agile Framework, the 3+5+3 formula

Kanban
The simple and most effective ways to get things quickly, you will see this near you

Lean Startup & MVP
A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

Shifting in behaviours and mind-set
An Agile model consists of cross-functional teams formed around a certain purpose

Question and Answers
This is just the beginning of a transformation



Kanban definitions

1 Kanban

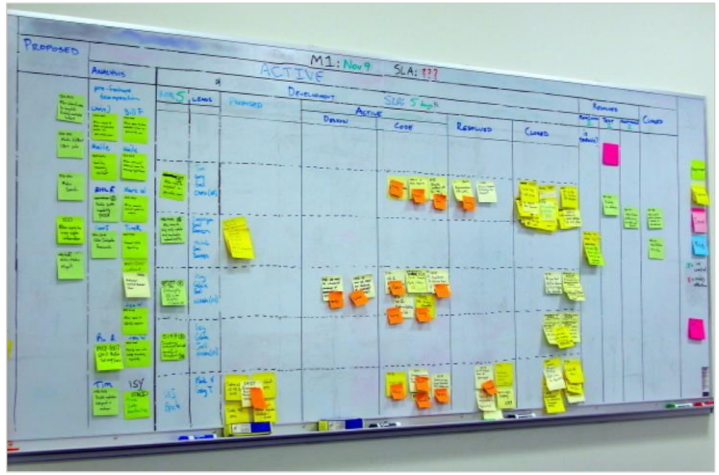
Kanban (看板): Japanese for “visual signal”, “signboard”, or “card”

2 Kanban System

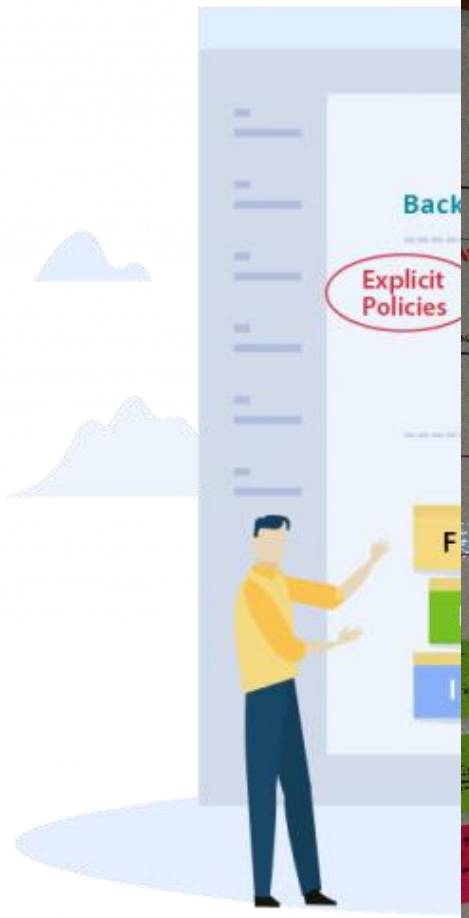
Is a visual system for managing work through a process, e.g. Toyota Production System (TPS) or the Imperial Garden in Tokyo

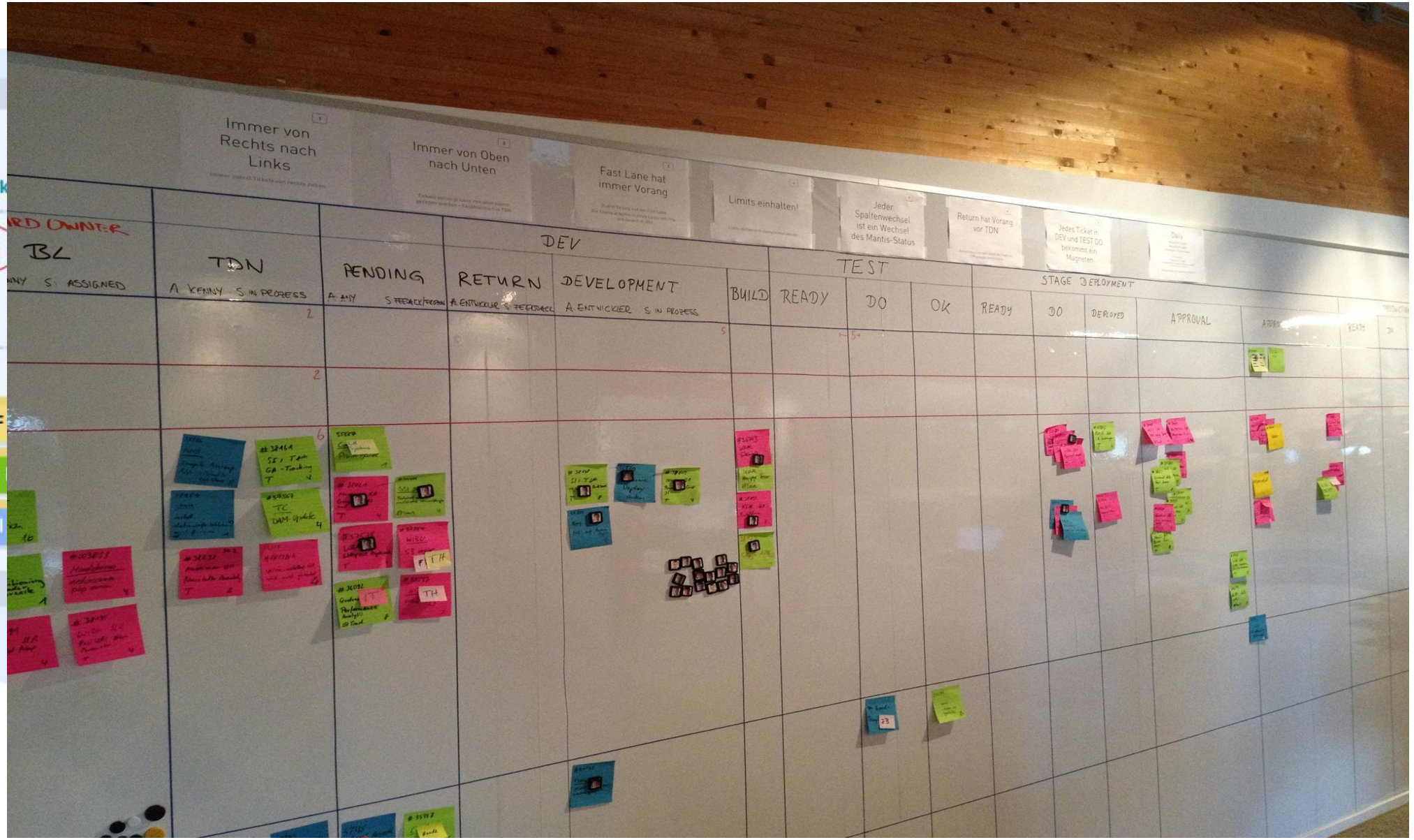
3 Kanban Method

A method that enables teams and organi- zations to visualize their work and thereby reduce bottlenecks and waste



Example Board





Kanban is a pull system where people build what's needed and when its needed as per customer demand

Unlike a "push" system, where work is given to a person and put onto a massive "to-do" list, pull systems allows the person doing the work to **pull in work as they are ready** based on customer demand

This prevents people from feeling overloaded and **forces teams to prioritize value added work** whilst eliminating wasted time or effort.

Push vs. Pull

Complete everything I tell you to do, in the time frame I give you

Take a new item of work as soon as you are done with the current one

Kanban practices

Visualize

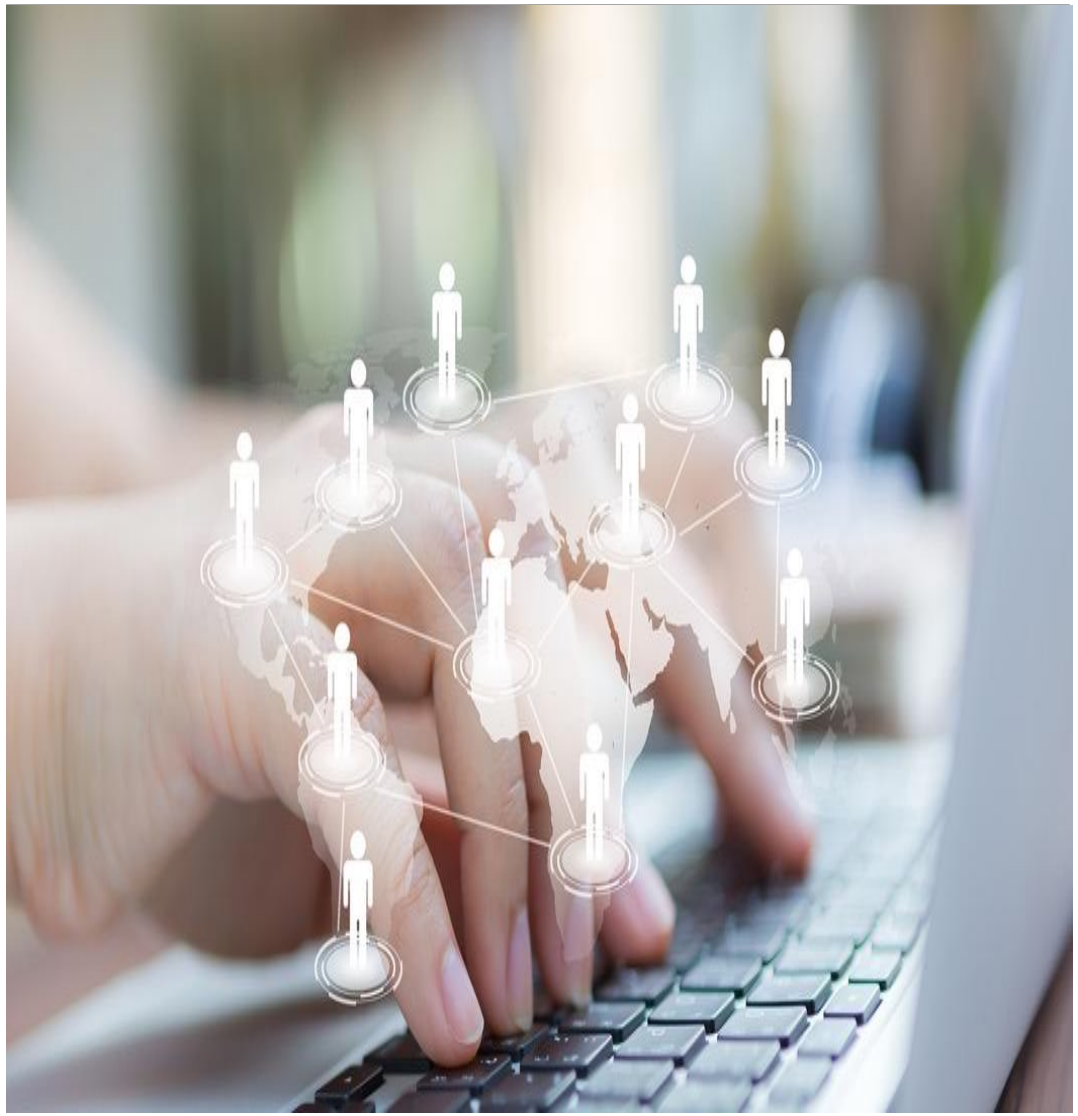
Manage flow

Implement feedback loops

Limit work in progress (WIP)

Make policies explicit

Improve collaboratively. Evolve experimentally



29

There are no prescribed roles in Kanban but the Service Request Manager & Service Delivery Manager have emerged

Service Manager

Provide transparency around the business value of items to support the decision making

Defining and maintaining process policies, e.g. by defining class of services

Service Delivery Manager (SDM)

Optimize work flow efficiency

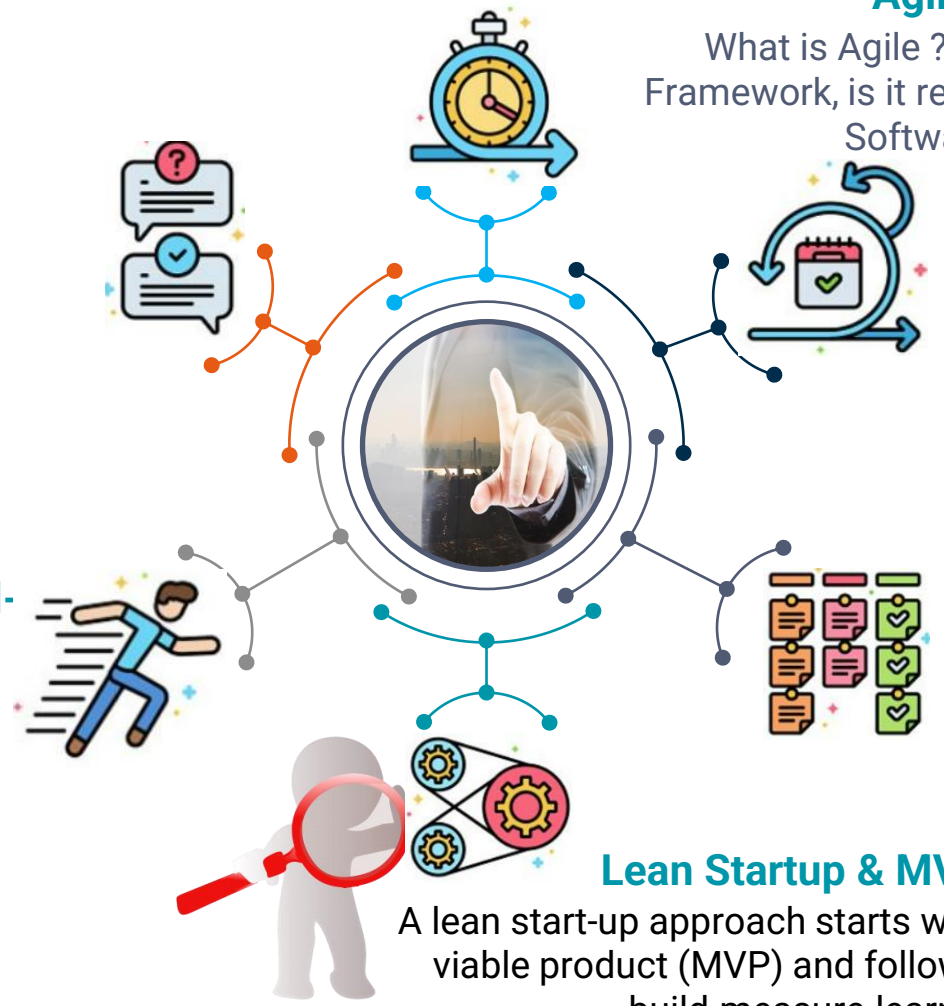
Owns monitoring and charts (e.g. Cumulative Flow Diagram, Cycle Time Histogram, Aging Chart)

Agenda

Agile ways of working

Agile
What is Agile ? Mindset or Framework, is it restricted to only Software

Question and Answers
This is just the beginning of a transformation



Scrum
One of the most commonly used Agile Framework, the 3+5+3 formula

Shifting in behaviours and mind-set
An Agile model consists of cross-functional teams formed around a certain purpose

Kanban
The simple and most effective ways to get things quickly, you will see this near you

Lean Startup & MVP
A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

A lean startup approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

Traditional approach

This is a big and long project, let's make sure we capture everything, otherwise we are never going to get it

What customers could theoretically want

Lean startup approach

Start with the minimum viable product

" Minimum Viable Product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort" – Eric Ries

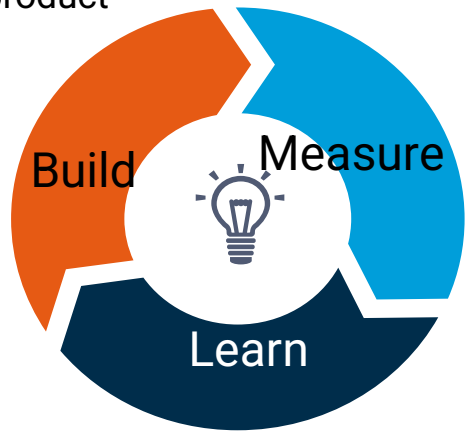
Evolve over time

iterating product by including more detailed features corresponding to customer feedback

An MVP can save time and money by avoiding product features that do not resonate with customer needs

Build-measure-learn cycle

Turn ideas into products fast as MVP Change product according to feedback In each iteration improve product

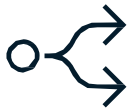


Learn from customers' feedback by evaluating data

Test product

Measure customer s' reaction

The MVP concept is highly debated concept which inspired alternatives



Faster

More complete

MVP alternatives

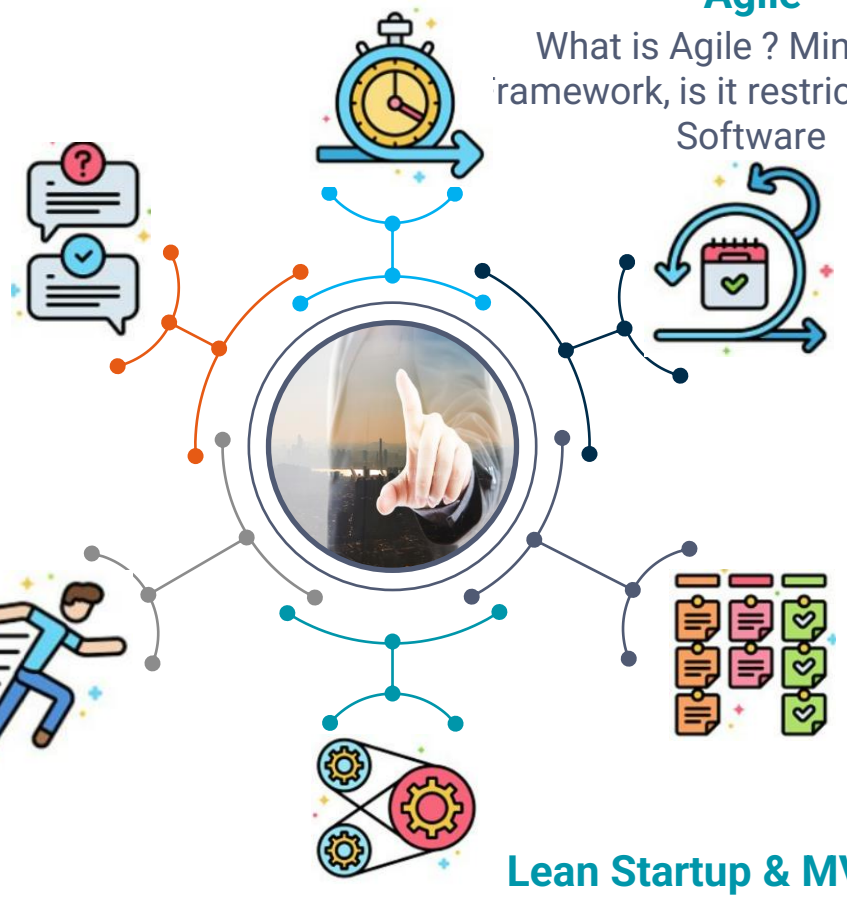
Acronym	Name	Description
RAT	Riskiest Assumption Test	Test your assumptions
MVE	Minimum Viable Experiment	Test the central premise of a business idea
MTP	Minimum Testable Product	First version of your product you can test
MVP	Minimum Viable Product	Sweet spot of maximized learning and effort
MMP	Minimum Marketable Product	Focus on something marketable

Agenda

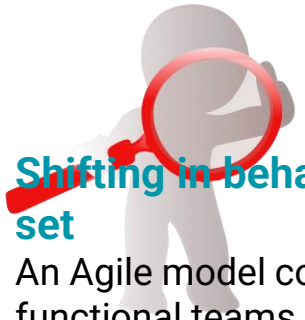
Agile ways of working

Agile
What is Agile ? Mindset or framework, is it restricted to only Software

Question and Answers
This is just the beginning of a transformation



Scrum
One of the most commonly used Agile Framework, the 3+5+3 formula



Shifting in behaviours and mind-set
An Agile model consists of cross-functional teams formed around a certain purpose

Kanban
The simple and most effective ways to get things quickly, you will see this near you

Lean Startup & MVP
A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

An Agile model consists of cross-functional teams formed around a certain purpose, instead of traditional organizational silos

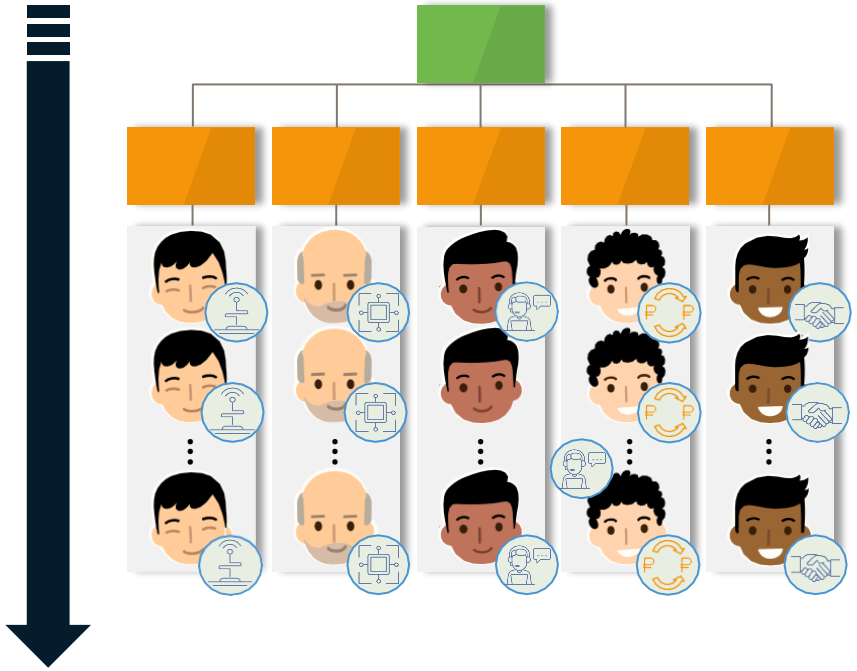
Structure

Processes

People

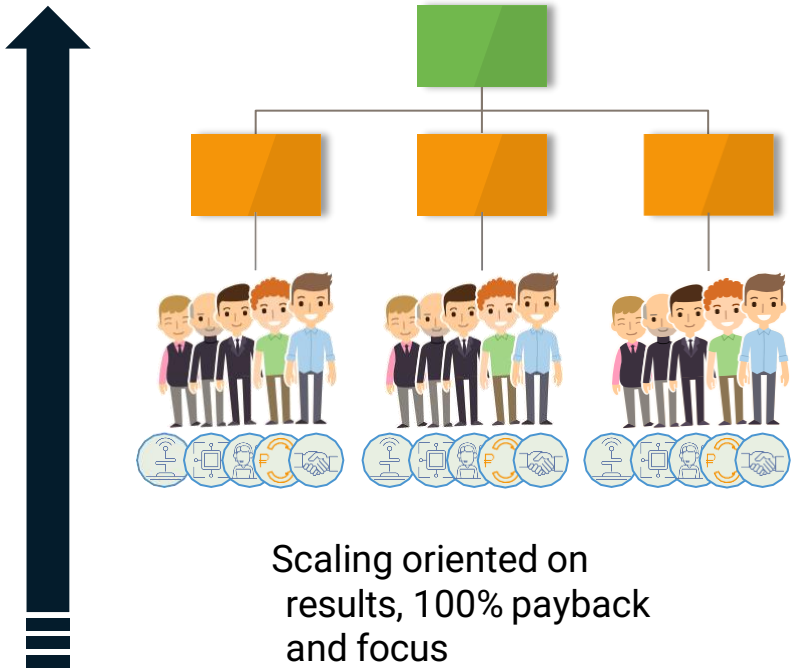
Traditional model

Organization structure divided in functional silos and managed top-down



Agile Model

Cross-functional teams formed bottom-up around client needs



Scaling oriented on results, 100% payback and focus

Team size is limited to ensure effective social interaction and team members are allocated to the team permanently

Structure

Processes

People

Traditional model

Many employees constantly switching between multiple part-time activities

Agile Model

Dedicated team of 7+-2 full time employees dedicated to a single purpose[1]

Maximized focus and engagement

1 The team might attract experts, who can bring expertise, in order to achieve goals of the stream (e.g. Architectures)

Teams are collectively responsible for their product, with one of the team members being the product owner

Structure

Processes

People

Traditional model

The hierarchy of bosses who delegate tasks

Agile Model

Product owner is a person empowered to make decisions on the product

De-bossing

Team members are located as close as possible to each other, ideally in the same room, contrary to the traditional model

Structure

Processes

People

Traditional model

Project team members located in different offices, cities and countries

Agile Model

Transparency allows teams to collaborate better in distributed environment. Keeping your colleagues informed

Distributed

Product development and continuous improvement are done incrementally and iteratively

Structure

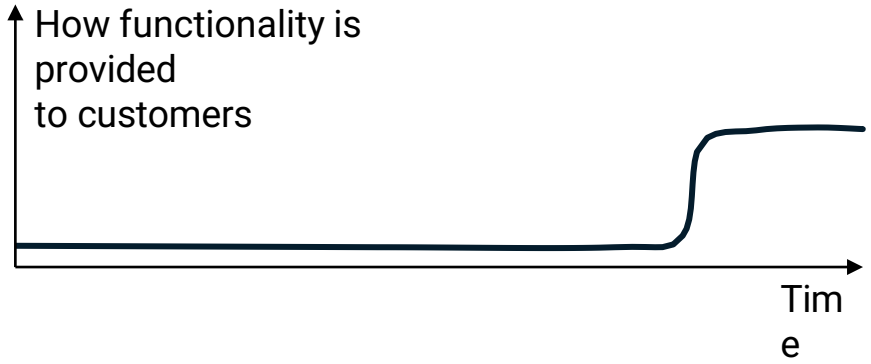
Processes

People

Traditional model

Non-iterative and non-incremental development

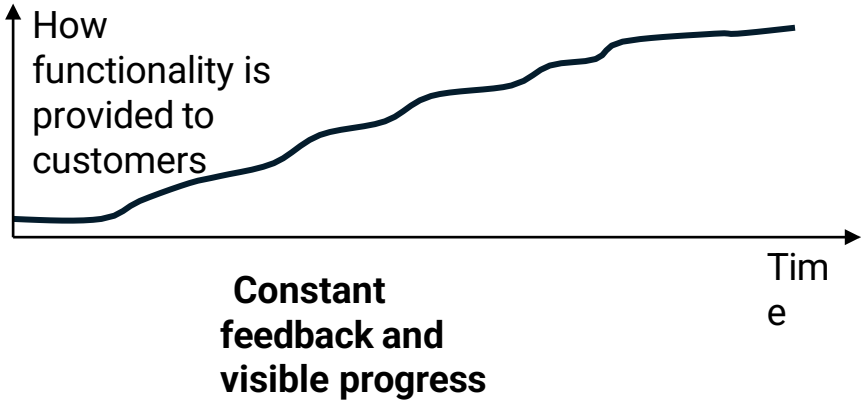
1 2 3 4 5



Agile Model

Iterative and incremental development

1 2 3 4 5



Constant feedback and visible progress

Steering and control based on bureaucratic processes are replaced by frequent demonstrations every couple of weeks

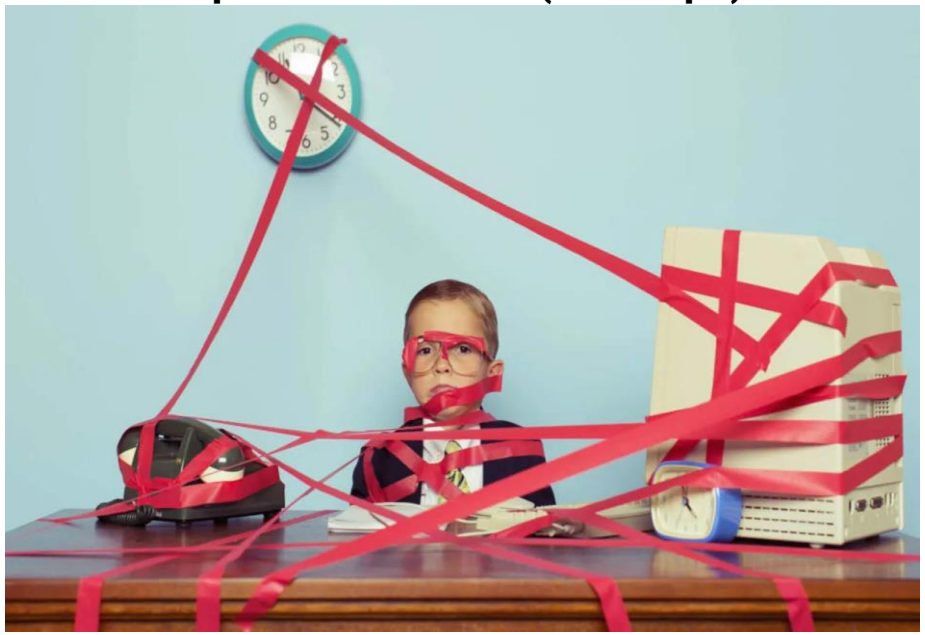
Structure

Processes

People

Traditional model

The long bureaucratic process of requirements and results harmonization with multiple stakeholders (Red Tape)



Agile Model

Demonstration of team results (running product) and collecting feedback from key stakeholders

Initiative and engagement

Gantt charts with fixed sequence of activities are replaced by visual boards like Scrum Board or Kanban

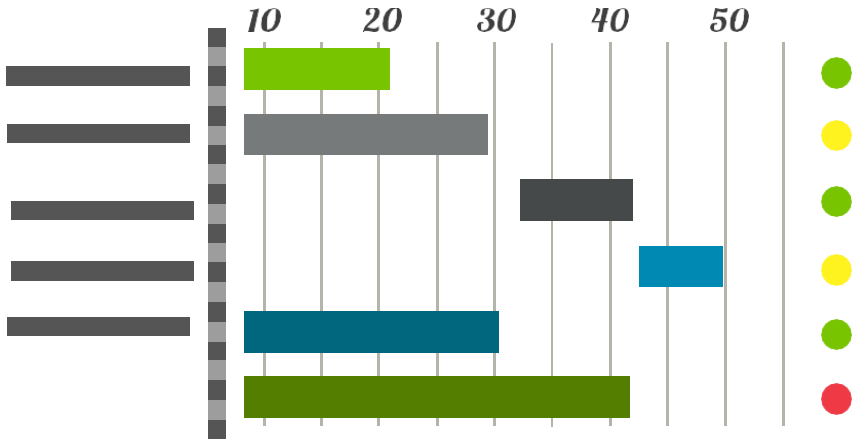
Structure

Processes

People

Traditional model

Gantt charts with fixed sequence of activates and progress reporting



Agile Model

Kanban enables flexible management of priorities while reaching the final goal

Flexible methods of obtaining the final result

39

Changes are welcome at all times and allow continuous improvement

Structure

Processes

People

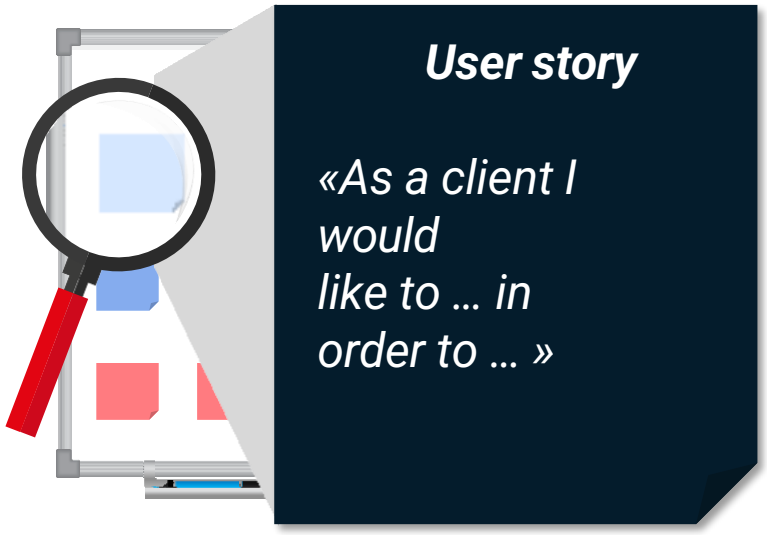
Traditional model

Business requirements are fixed and documented upfront, based on inputs from stakeholders

Documents: policies of project management, business requirements, etc.

Agile Model

User stories are created according to collected feedback from customers/end-users



Flexible methods of obtaining the final result

Experiments are highly encouraged and mistakes are the measure of organization's learning

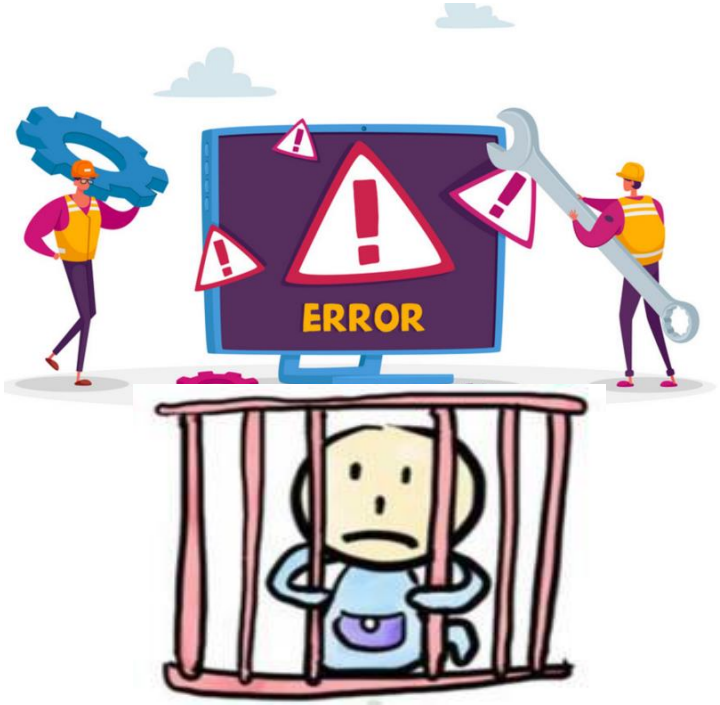
Structure

Processes

People

Traditional model

Error is a failure. We try to avoid errors by creating a set of rules



Agile Model

Error is a sign of success. We encourage experimenting and do not punish colleagues even if they fail

Trial and error

Agenda

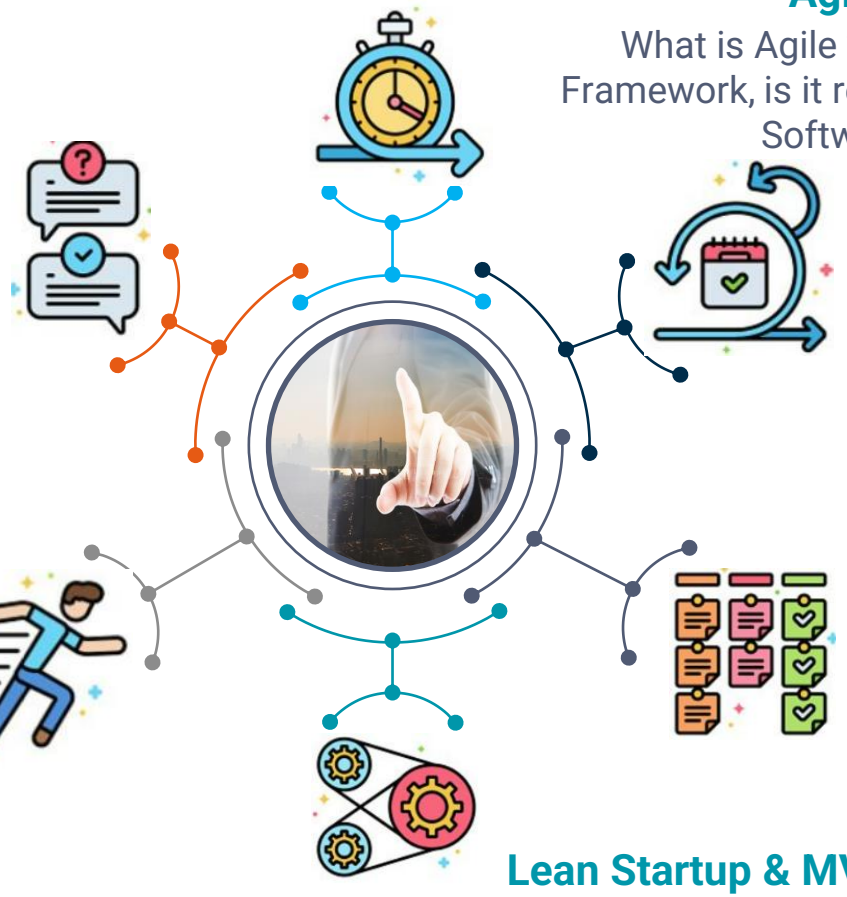
Agile ways of working

Question and Answers

This is just the beginning of a transformation

Agile

What is Agile ? Mindset or Framework, is it restricted to only Software



Scrum

One of the most commonly used Agile Framework, the 3+5+3 formula

Shifting in behaviours and mind-set

An Agile model consists of cross-functional teams formed around a certain purpose

Kanban

The simple and most effective ways to get things quickly, you will see this near you

Lean Startup & MVP

A lean start-up approach starts with a minimum viable product (MVP) and follows a cycle of build-measure-learn

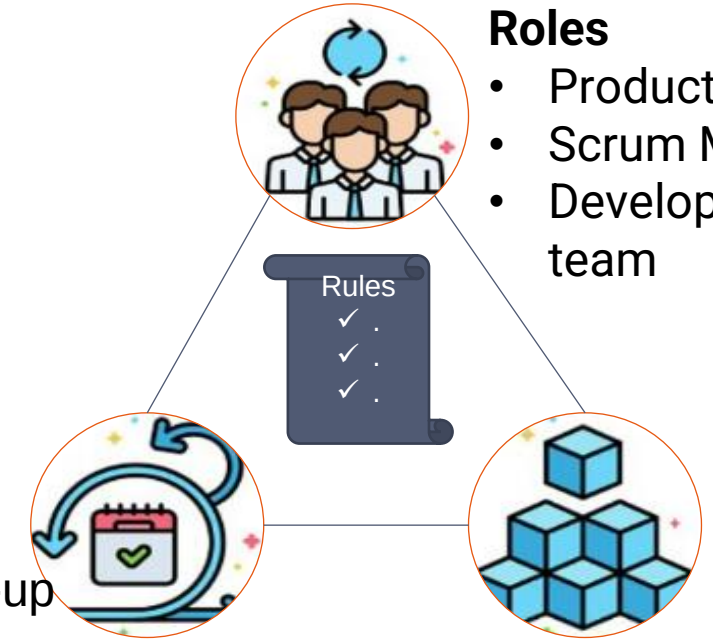
The customer journey teams will embrace Agile by adopting the Scrum framework as their preferred delivery approach

Definition

Scrum is a **framework** of practices aimed to address complex and changing problems, while **productively** and **creatively delivering products** of **the highest possible value**

- **Lightweight** Framework
- **Simple** to understand
- **Difficult** to master

Elements of the Scrum framework



Roles
- Product Owner
- Scrum Master
- Development team

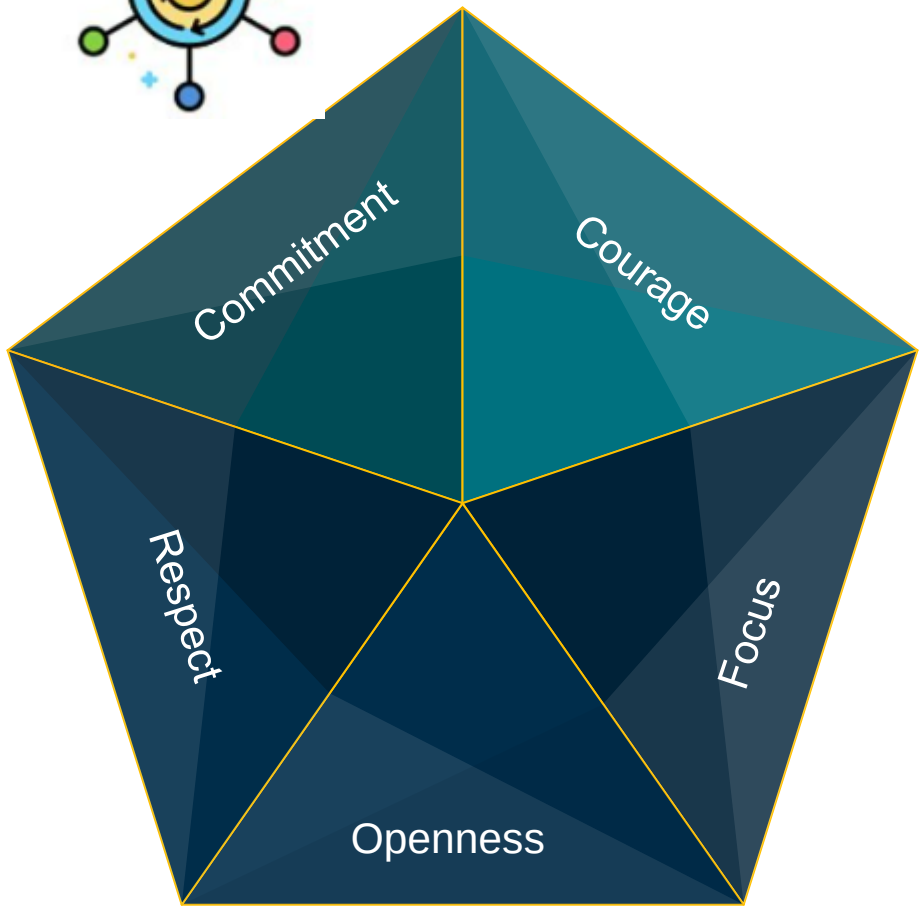
Events
- Daily Stand-up
- Review
- Retrospective
- Planning
- Refinement

Artifacts
- Product Backlog
- Sprint Backlog
- Shippable product increment

Overview of Scrum Valu

Commitment

literally means “**joined together**.” It involves sharing our sincere intent to act, and then accepting responsibility for following through on that intended action. In Scrum, **people personally commit**to achieving the shared **goals**of the Scrum Team.



Courage

means “from the heart.” It involves acting in alignment with our beliefs, especially when that is hard. Scrum Team members have **courage to do the right thing** and work on tough problems.

Focus

comes from the Latin word for “domestic hearth,” which was the location of the fire at the center of the home. The focus was literally the thing that brought people together. In Scrum, **everyone focuses on the work of the Sprint and the goals** of the Scrum Team.

Respect

means “to look or view again.” Respect involves taking a second look at how we view others, to develop a sincere appreciation for the unique capabilities that they contribute. Scrum Team members **respect each other to be capable, independent people**.

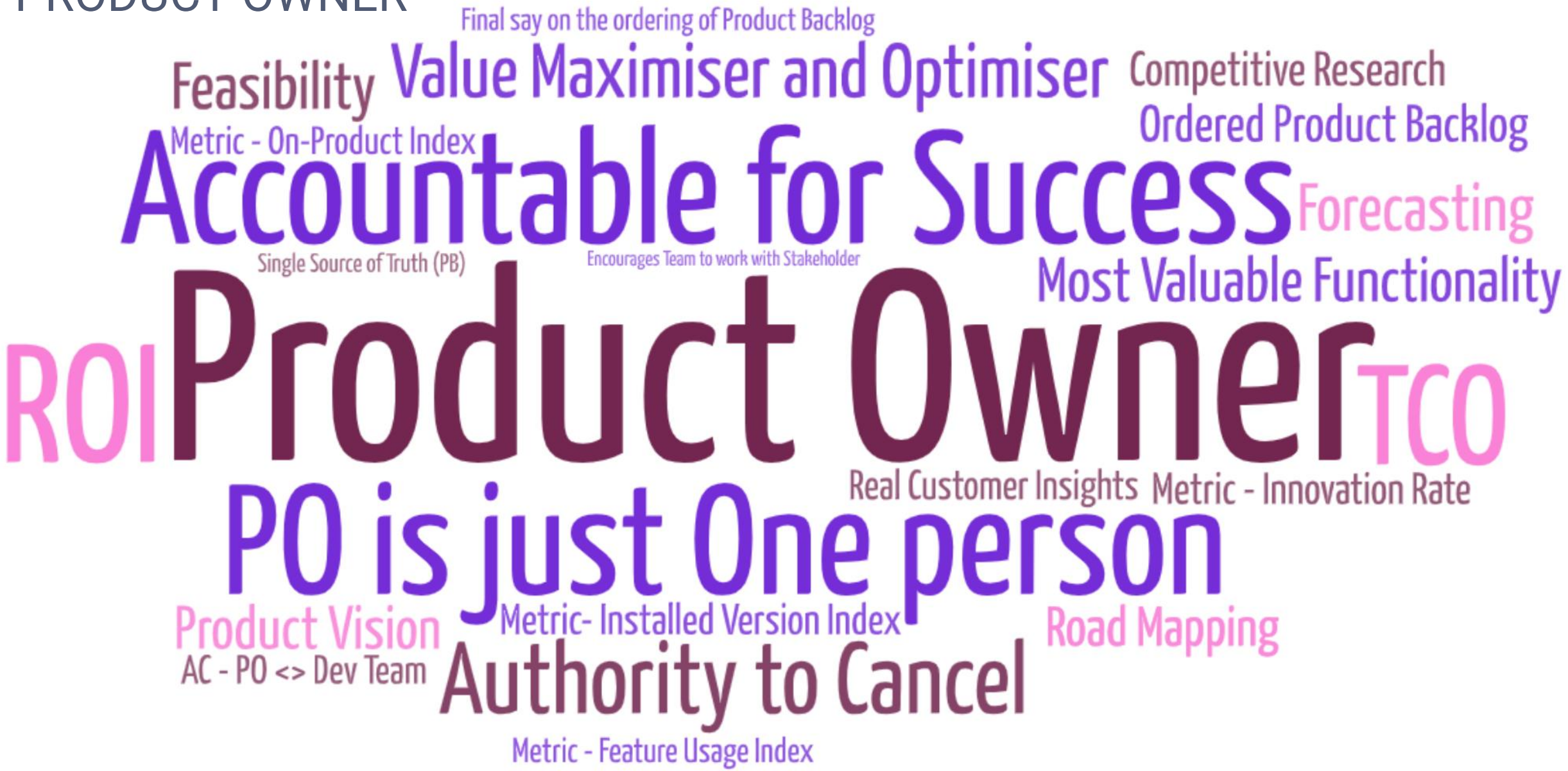
Openness

means “exposed or evident.” The Scrum **Team and its stakeholders agree to be open about all the work and the challenges** with performing the work. Openness is closely related to the empirical pillar of Transparency.

2. How to work in Scrum ?

- Demonstration of how the Scrum ceremonies fit together
- Deep dive into best practices for: Sprint Planning, Daily Stand-up, Sprint Review, Sprint Retrospective, Backlog Refinement
- Definition of the key Scrum roles: Product Owner and Scrum Master
- Which additional roles are needed to support the classic Scrum configuration

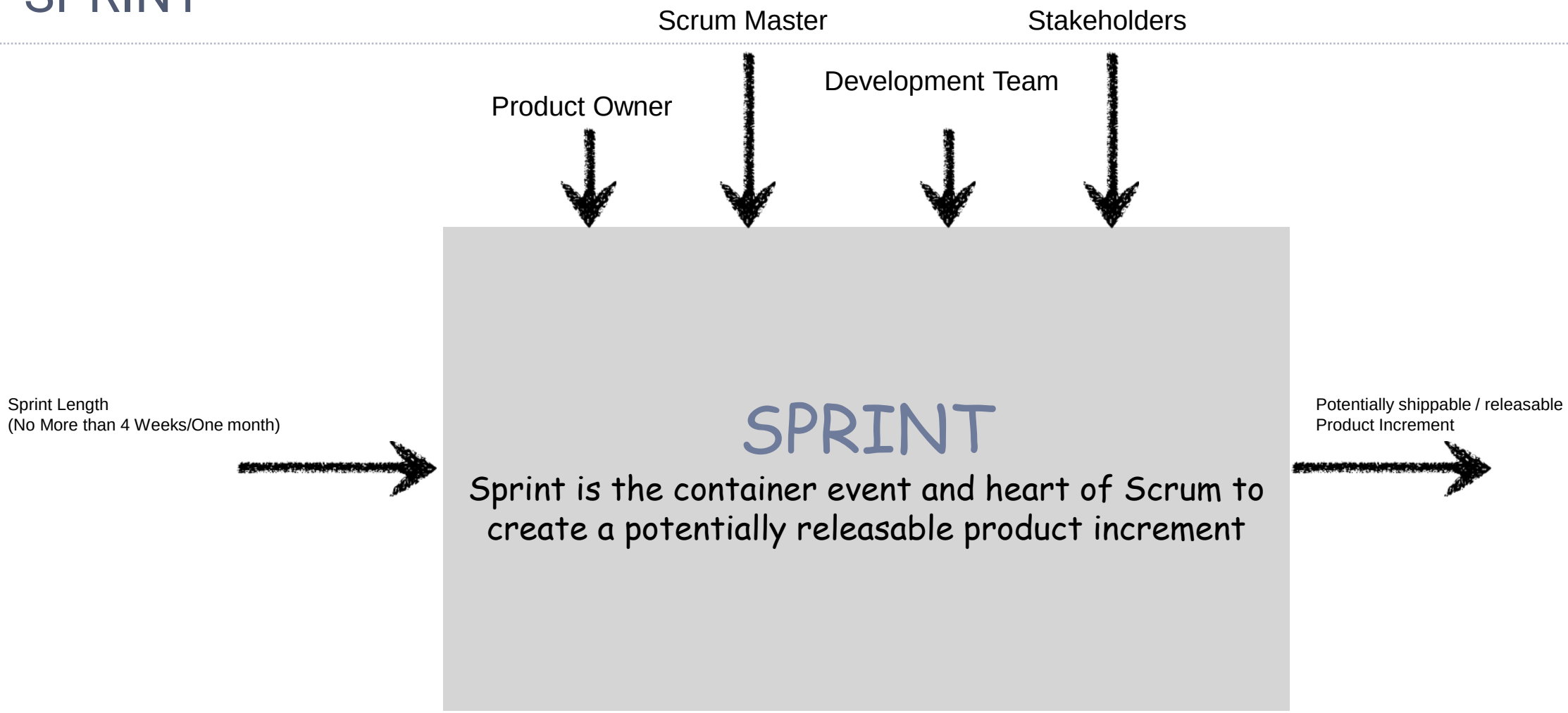
PRODUCT OWNER

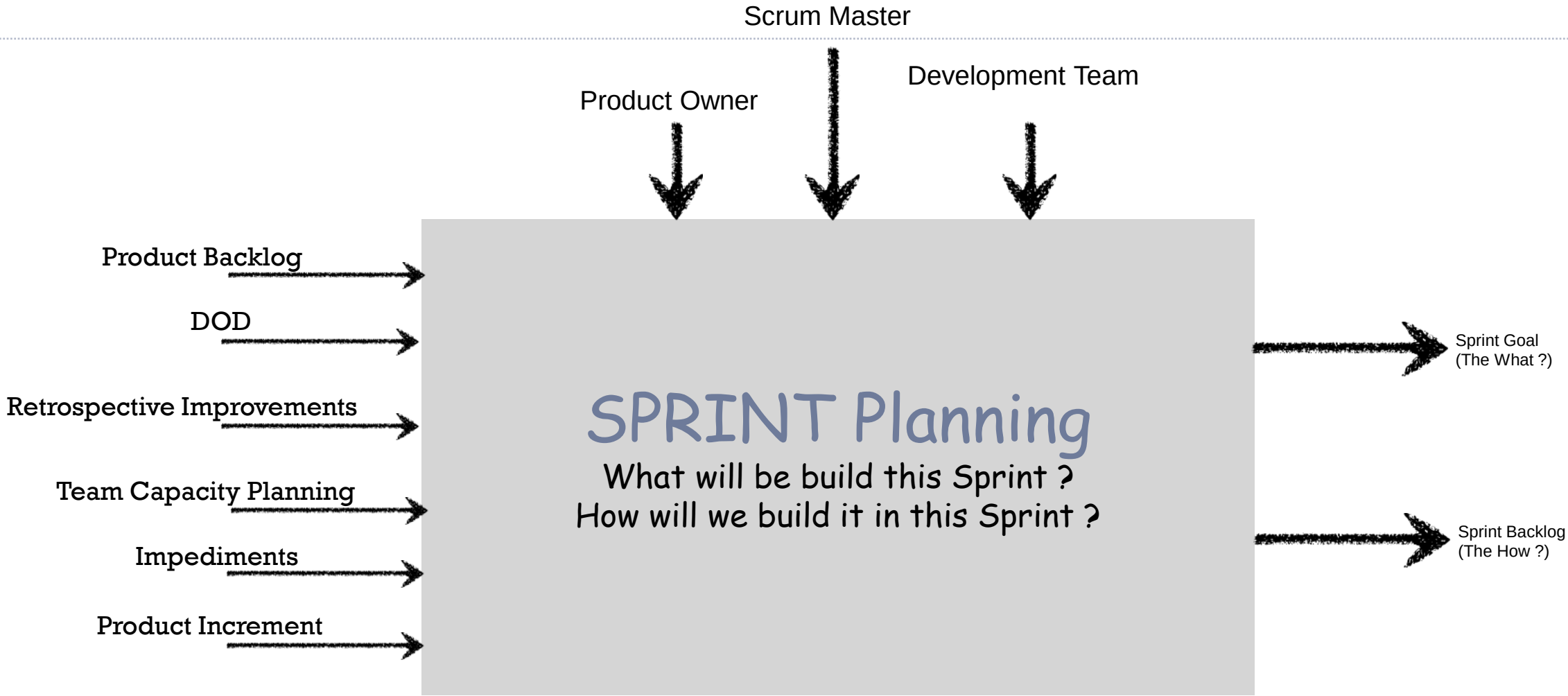


SCRUM MASTER

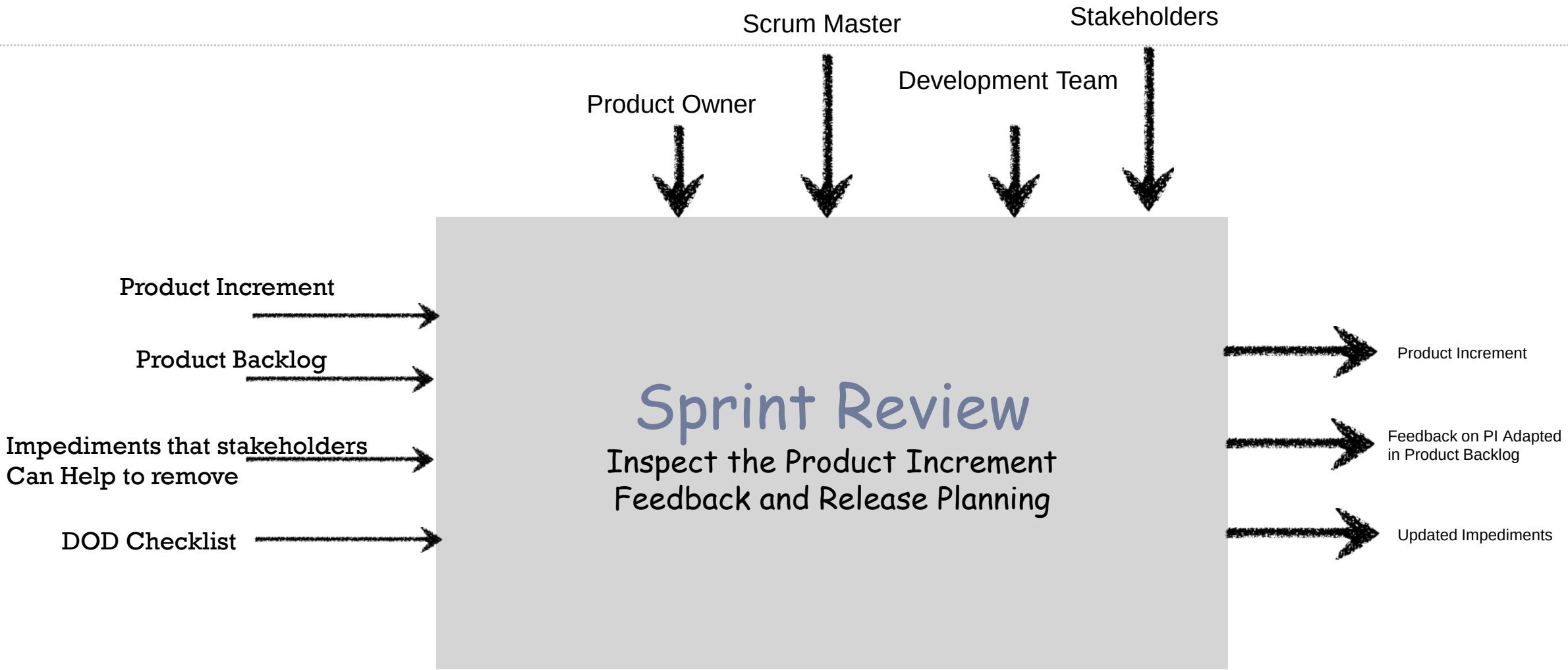
DEVELOPMENT TEAM

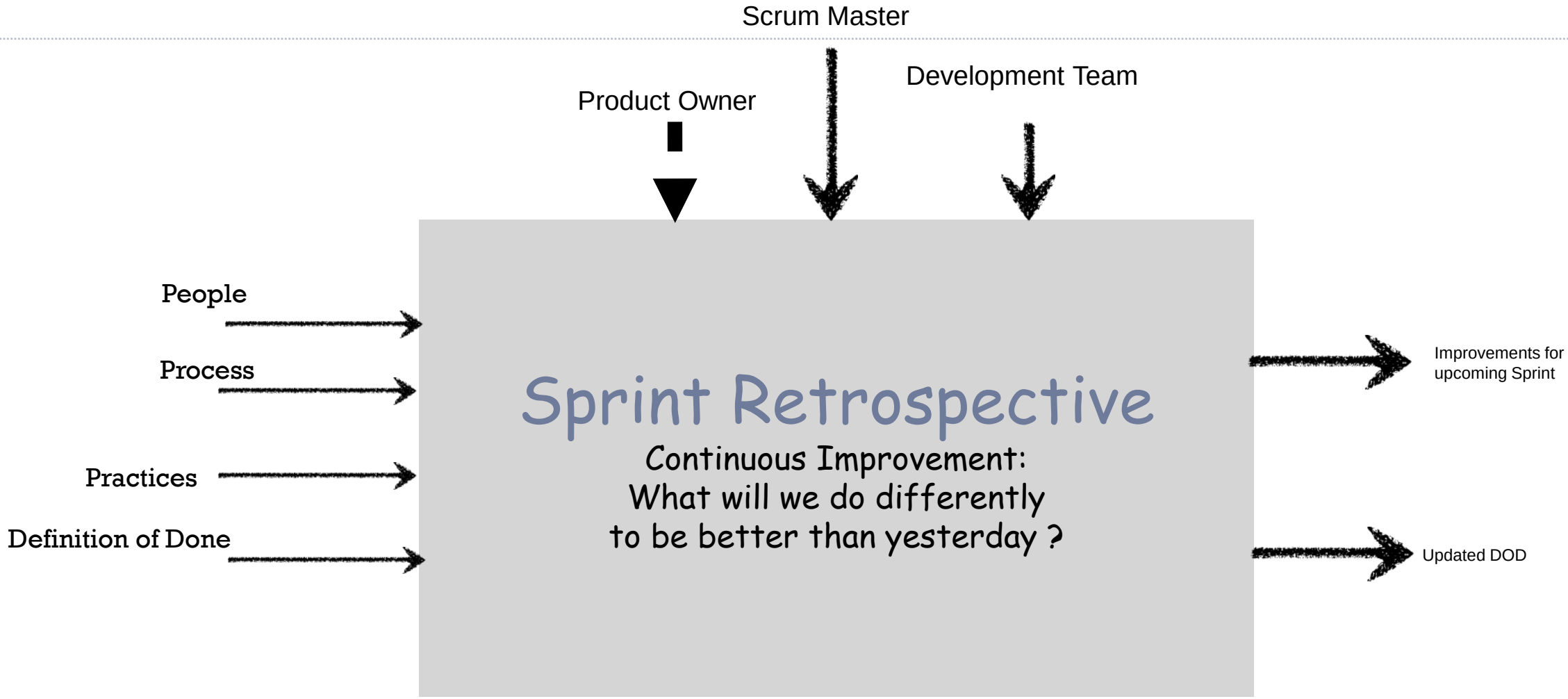
SPRINT


Scrum Master
Stakeholders
Development Team
Product Owner
Sprint Length
(No More than 4 Weeks/One month)
SPRINT
Sprint is the container event and heart of Scrum to create a potentially releasable product increment
Potentially shippable / releasable
Product Increment


Scrum Master
Product Owner
Development Team
Product Backlog
DOD
Retrospective Improvements
Team Capacity Planning
Impediments
Product Increment
SPRINT Planning
What will be build this Sprint ?
How will we build it in this Sprint ?
Sprint Goal
(The What ?)
Sprint Backlog
(The How ?)


Scrum Master
Product Owner
Development Team
Sprint Goal
Sprint Backlog
Open Impediments
Information Radiator
Daily Scrum
Are we on track to achieve our forecasted Sprint Goal ?
Updates Sprint Goal
Updated Sprint Backlog
Updated Impediments
Updates Information Radiator

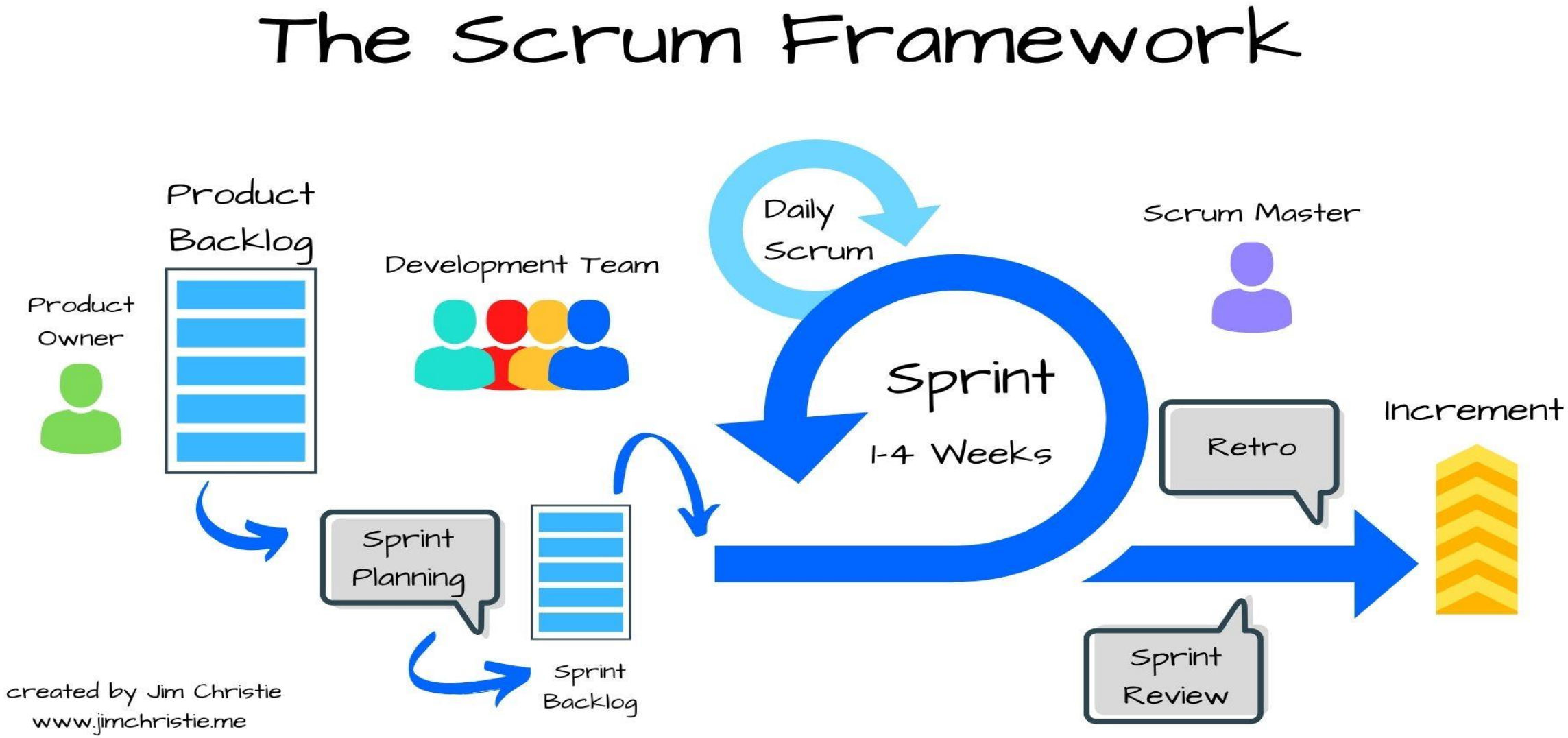

Scrum Master
Stakeholders
Development Team
Product Owner
Product Increment
Product Backlog
Impediments that stakeholders
Can Help to remove
DOD Checklist
Sprint Review
Inspect the Product Increment
Feedback and Release Planning
Product Increment
Feedback on PI Adapted
in Product Backlog
Updated Impediments


Scrum Master
Product Owner
Development Team
People
Process
Practices
Definition of Done
Sprint Retrospective
Continuous Improvement:
What will we do differently
to be better than yesterday ?
Improvements for
upcoming Sprint
Updated DOD

6 STEP Guidelines for effective Sprint Retrospective

1. Start with appreciation
2. Make improvements from last retrospective
3. Set the stage
4. Gather data
5. Generate insights
6. Identify the root cause and an improvement plan

Overview page of Scrum



Sprint Planning

- Time boxed event (max 4 hours for 2 week Sprint)
- The input to this meeting is the Product Backlog
- Scrum Master ensures that the event takes place
- Sprint Planning answers the following:
  - **What increment is feasible to be delivered in the next Sprint?**
  - **How will this work be achieved? What are the specific tasks that will be required?**
- The Sprint Goal is defined, which provides guidance to the Development Team on why it is building the Increment
- Tasks are defined for each user story, generally down to 2 4h long tasks that can be tracked on a Scrum board

Daily Stand-up

- Time boxed event (15 minutes)
- Held everyday, typically in the morning before the bulk of the day begins
- Scrum Master ensures that the event takes place
- Standard format consists of 3 questions:
  - **What did I do yesterday?**
  - **What will I do today?**
  - **Are there any impediments/blockers stopping me from completing my work for today?**
- The team inspects progress toward the Sprint Goal
- The team members try to minimize work in progress (i.e. work in “doing” status)
- The Daily should not have ANY problem solving, any discussions should be noted and taken offline: “hey that’s a great point, looks like we need to have a

Sprint backlog Refinement

- Typically 1-hour and can occur multiple times a sprint at any point in the sprint
- Goal is to update the backlog and prioritize the backlog items
- Refinement sessions should include input from the full team, which mean inputfrom technical team members on complexity and effort (i.e. sizing)
- In a refinement session, the PO describes any new use stories and answers any clarifying questions related to it
- If necessary, technical or business investigations can be launched to clarify elements of the user story (e.g., do we have an API already for this? Are there any major legal requirements we're not considering?)
- If there are reprioritizations due to newly available information, the PO should make the change in the backlog and explain the rationale (e.g., we got updated metrics and we think a different feature will improve customer acquisition instead)
- The output of the refinement session is an updated Product Backlog

Sprint Review

- Time-boxed event (max 3 hours for 2-week Sprint)
- Held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed
- The Sprint Review includes the following elements:
    - **Attendees include the Scrum Team and key stakeholders invited by the Product Owner**
    - **The PO presents the original Sprint Goal and the progress completed towards it**
    - **The PO presents any new decisions or important updates (e.g., result from business investigations, meetings with senior stakeholders)**
    - **The Development Team demonstrates the work that it has "Done" and answers questions**
    - **Stakeholders note feedback down (e.g., on a sticky) and mentions it at the end of the session, with detailed problem solving avoided (should be explored after the Review)**
- A good Sprint Review will help key stakeholders understand the functionality built andallow them to be able to provide necessary signoffs (potentially with a follow up session)

Sprint Retrospective

- Time-boxed event (max 2 hours for 2-week Sprint)
- It is the last meeting of the Sprint
- Opportunity for the Scrum Team to inspect itself and create a plan for improvements
- The purpose of the Sprint Retrospective is to:
- The PO presents the original Sprint Goal and the progress completed towards it
  - **Inspect how the last Sprint went with regards to people, relationships, process, and tools**
  - **Identify and order the major items that went well and potential improvements**
  - **Create a plan for implementing improvements to the way the Scrum Team does its work**
- The Retrospective should be fun! Therefore different and creative ways to facilitate are encouraged

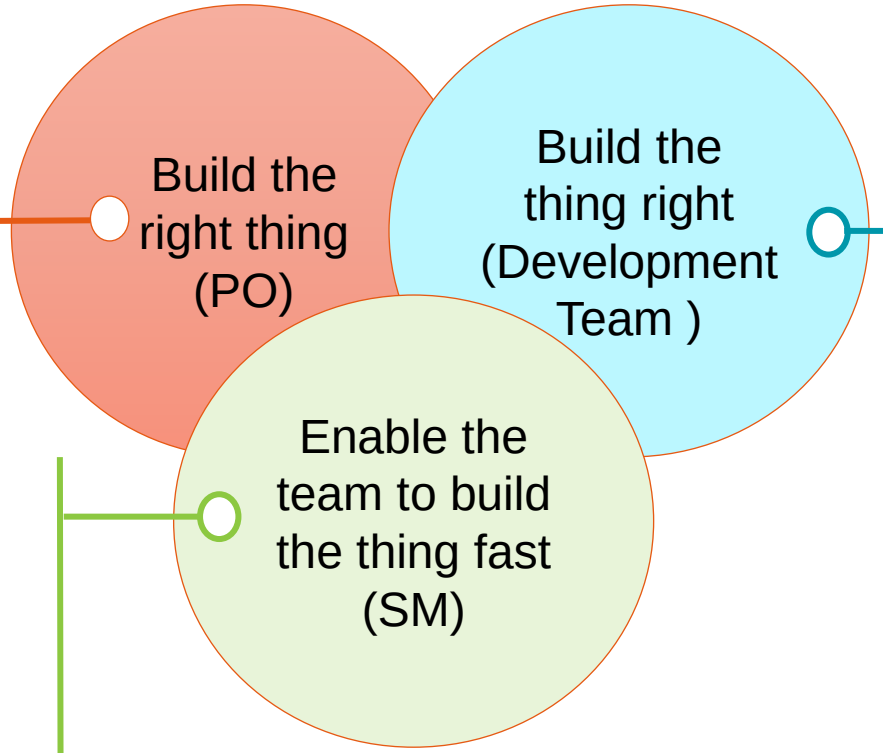
Scrum teams should be led by a trio of roles: Product Owner, Scrum Master and Development Team

Product Owner

- Responsible for "building the right thing" –realizing product vision, goals and business value; ensures product quality
- Takes or elevates key product decisions
- Responsible for building reusable solutions

Scrum Master

- Facilitates the agile processes and practices; supports and coaches the team
- Promote healthy team environment and remove impediments
- Ensure team is delivering/aligned on project vision and goals



Development Team

- Delivers potentially shippable software in every sprint (quality built in)
- Works as a cross-functional team made up of all necessary skillsets
- Is self-organizing and empowered
- Drive continuous improvement of the development process

The Product Owner

Characteristics	Responsibilities
• Is a single person, not a committee • Represents a committee of stakeholders, users, business owners, etc. • Is a member of business or the user community, e.g., product manager	• Maximizes the value of the product and the work of the Development Team • Sets a clear vision for the product • Accountable for the Product Backlog • Empowered to make decisions for the product • Maximizes ROI by identifying and prioritizing the product features by business value; can change features and priority every sprint • Defines the features for the release • Develops, prioritizes, and communicates the Product Backlog • Provides stories and acceptance criteria • Accepts or rejects work product • Coordinates involvement of users and business stakeholders

The Product Owner

Characteristics	Responsibilities
• Could be anyone with the servant leadership mindset (typically not an authority role) • Is NOT the project manager but rather a facilitator and a coach.	• Ensures Scrum is understood and enacted • Oversees the Scrum process and coaches the team; facilitates transparency, inspection and adaptation • Participates in daily Scrum, sprint review, and planning meetings • Facilitates Scrum meetings to ensure full team engagement • Removes impediments(eg.eg., cultural barriers, logistic challenges) • Enables close cooperation across all team members • Facilitates communication and collaboration inside and outside the team • Shields the team from external interference and ensures the team is fully functional and productive • Fosters self organization of the team

The Product Owner

Characteristics	Responsibilities
• Self organizing • Typical members of the team include developers, testers, architects, and representatives from UX, Infrastructure, etc. • The optimal size of team is 7, +/ 2 team members • Composed of cross functional members that includes all skills necessary to deliver the project • Structured and empowered by the organization to organize and manage their own work	• Defines the goal of the sprint with PO and PO proxy • Has the right to do everything within the boundaries of the project guidelines to reach the Sprint goal • Work as one to meet the Sprint goal • Flags any impediments/blockers to the SM • Provides effort estimates and decides what can be completed in a sprint as per the business priority • Organizes itself and its work • Demos work results to the Product Owner • Focuses on continuous improvement

On top of the classic Scrum Roles, our teams also have additional roles to support the journey delivery

Shaping Team

Role	Examples of responsibilities
Product Owner	• Responsible for "building the right thing", ensures product quality • Own and continuously manage product backlog • Take or elevate key product decisions • Responsible for building reusable solutions • Ensure incremental releases deliver on business KPIs
Business Analyst(s)	• Write and refine user stories with support of P.O.
SME(s)	• Bring knowledge of functions' processes • Solve problems in the solution design and executes implementation
CX /UX Researcher (s)	• Design customer experience strategy for channel • Lead customer centric design, user engagement plan, and conducts user testing
UI Designer (s)	• Support creation, publication, and management of content across platforms • Design new process in respective area of expertise

Delivery Team

Role	Examples of responsibilities
Business Analyst	• Have oversight and escalate external dependencies and risks • Manage and upkeep the delivery plan (e.g., 30/60/90)Take or elevate key product decisions • Answer questions on user stories for delivery team
Scrum Master	• Ensure team is delivering/aligned on project vision and goals • Promote healthy team environment and remove Impediments • Facilitate the agile processes and practices • Support and coach the team
Tech Lead	• Lead technical vision and direction for product • Responsible for stability, sustainability & code quality • Co responsible for building reusable Developments
Developers (Front End and back end)	• Develop code, writes unit tests and drive integrations • Drive continuous improvement to the dev process

3. Creating and Managing backlog

- What is a Product Backlog and the hierarchy of items within it: Epics and User Stories
- Best practices for writing user stories, including guiding principles (INVEST)
- How to write behaviour driven development (BDD) styled acceptance criteria
- What is a vertical story slice and why it is important
- Techniques for story point sizing
- Definition of Ready (DoR) and Definition of Done (DOD)

The whole Delivery team is encouraged to capture ideas in the Product Backlog that could enhance the Product and the User experience

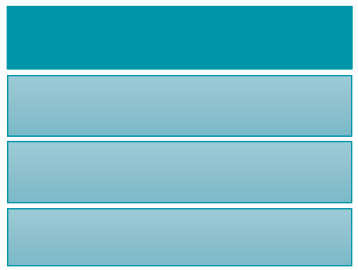
The Product Owner, with the help of the Team, should find the best way to organize the Product Backlog, prioritize and break all items into User Stories level

Product Backlog

prioritized list of items that is known to be needed in the product and should be frequently delivered

Epic

the smallest piece of the product that still has a business value



User Story

the smallest piece of the product that still has a business value

Task

Technical tasks that have no business value, but contribute together to deliver one or many User Stories

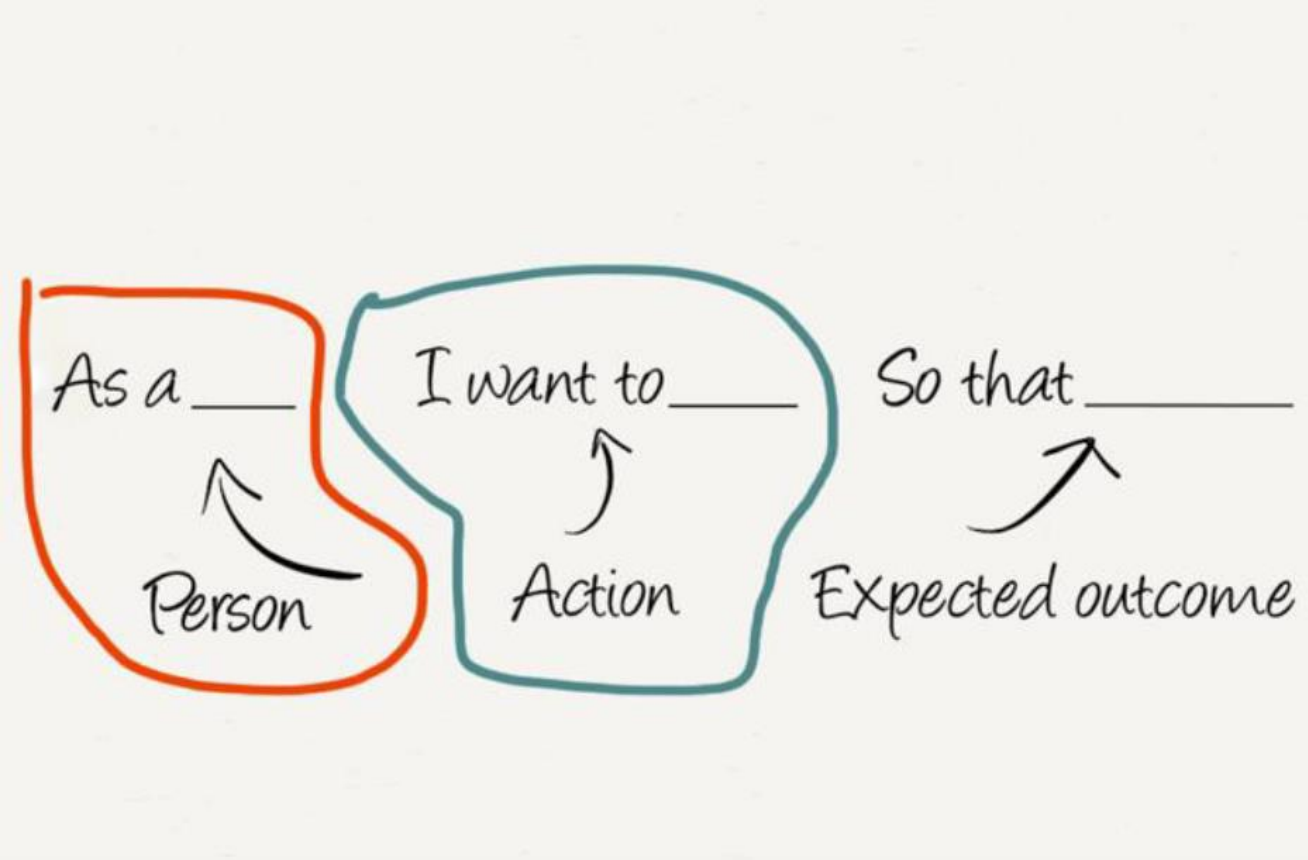
$$$ $$ $

Business Value

Simplicity

In Agile, we capture the Product's requirements always from the user's perspective using User Stories

A user story is a *promise* for a future *conversation*



	• The story itself • A promise to have a conversation at the appropriate time
Conversation	• The requirements themselves communicated from the Product Owner to the Dev Team via a conversation • Write down what is agreed Upon
Confirmation	• The Acceptance Criteria for the story • How the Dev Team will now they have completed the story (DoR)

INVEST Criteria

Free of dependencies on any other stories and self-contained. Stories are able to be completed in any order

Avoids too much detail and allows room for improvisation and
innovation. Stories that are negotiable can always be changed with

Stakeholders, customers and end users get some kind of value from
up to date information before you begin working on it
the work delivered

Clearly understood and contains enough detail to estimate how long it
will take to be delivered

Able to be completed (”done”) in one iteration. Large stories are
difficult to estimate and plan for

Contains key results or acceptance criteria. How will you demonstrate
it? Show me! What has changed in the world now that the story has

BDD approach to Acceptance Criteria

- **Acceptance Criteria** are the conditions (pass/fail scenarios) that a **software product must satisfy to be accepted by a user, customer or client**, or in the case of a system level functionality, the consuming system
- Behaviour Driven Development (BDD), follows a specific format, and breaks an acceptance criterion into the three sections below:
- **Given** clause describes the state of the world before you begin the behaviour you're specifying in this scenario. You can think of it as the pre-conditions to the test
- **When** clause indicates when/where new behaviour is expected
- **Then** clause describes the changes you expect due to the specified behaviour

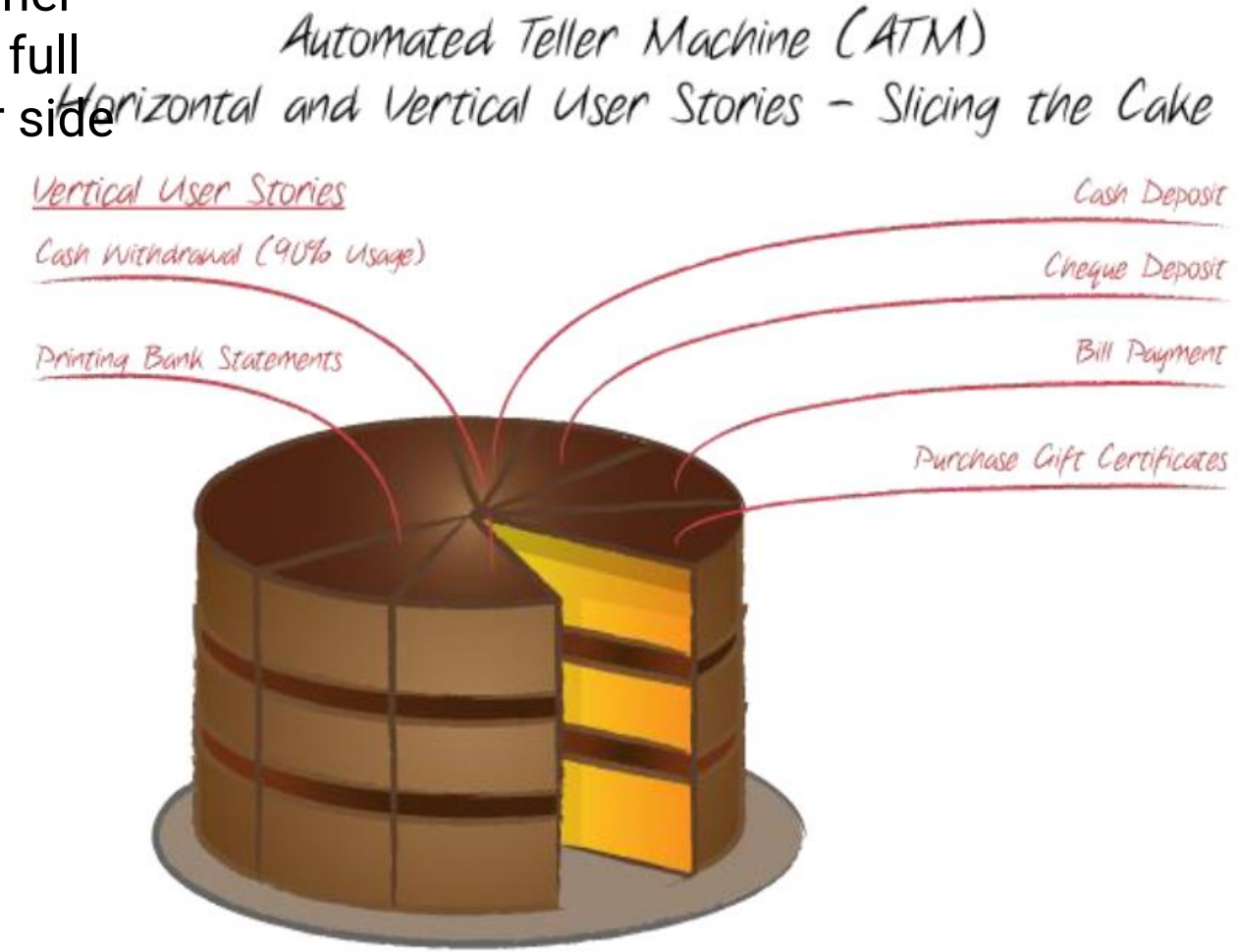
Vertical Story Slicing allows for a reduced project risk and earlier delivery of working software…

What is a vertical story slice?

- End to end functionality that meets a customer need and requires implementation across a full set of tech layers (e.g. user interface, server side API, middleware, backend systems)

Key outcomes

- Short feedback loops
- Reduced Risk
- Small E2E functionality
- Easier to prioritize
- Early integration
- Early value delivery



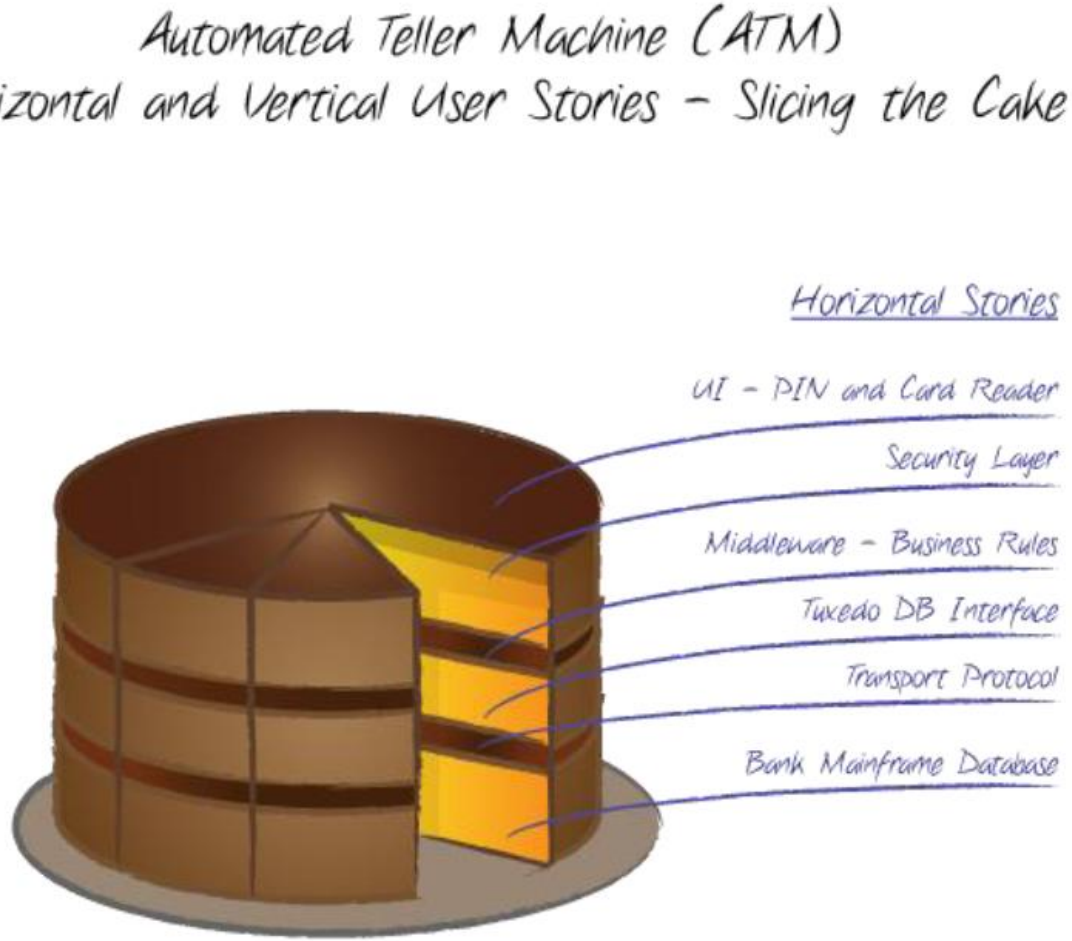
…whilst Horizontal Story Slicing, in which 'layers' are implemented separately increases risk and delivery time

What is a horizontal story slice?

- A technical functionality that on its own does not deliver customer value and is most often a single technical layer (e.g., middleware enterprise bus, user interface screens)

Key outcomes

- Long feedback loops
- Increased Risk
- "Entire cake baked"
- Difficult to prioritize
- Late integration
- Late value delivery



Purpose of user story sizing

- Sizing of the user stories in individual releases allows the complexity of the releases to be described
- User stories are sized using ‘story points’ and this allows the velocity (number of story points completed in a sprint) to be tracked
- Once sized, user stories can be more accurately committed during sprints and ‘swapped’ in and out for similar-sized stories
- Once the team(s)’ has been estimated, and gradually refined, this will allow a date range for releases to be calculated, to help stakeholders plan ahead (not this is a ranged estimate not a commitment)
- This helps us de-risk scope of uncertain size, complexity and effort

How to size (process)

Team agrees on estimation system to use (e.g., Fibonacci series: 0, 1, 2, 3, 5, 8, 13, 20)

1. Each team member independently estimates the level of effort in story points relative to a baseline user story
2. A team member provides a quick overview of the user story to besized

3. Team votes and members with maximum/minimum scores provide rationale for score to other team members

4. Team votes again and once consensus is reached story points are assigned

•Relative precision is critical, e.g., if feature A is twice as difficult as feature B it should get twice as many points
•Evaluators should consider the comprehensive effort of implementing the feature (e.g.,

Definition of ready and definition of done –actual criteria to be developed with the team collectively

A **Definition of Ready** (DoR) ensures that Product Backlog Items are ”ready” to be worked on in the next Sprint.

Ready stories are the output of the Product Backlog Refinement session.

Some examples for reference (NOT an exhaustive list):

- User Story has been sized by the Delivery (Dev) Team
- Wording has been finalized and made available to Dev
- User Story has no more than 5 Acceptance Criteria
- User Story is small enough to be completed in a week
- No outstanding questions remain that stop the team from working on the User Story
- All wireframes relevant to a User Story have been finalized and attached accordingly (where appropriate)
- User Story has been defined (in User Story format)

A **Definition of Done** (DoD) ensures that members of the team have a shared understanding of what it means for work to be.

Done stories are the output of a Sprint!

Some examples for reference (NOT an exhaustive list):

- All acceptance criteria are met and can be demonstrated to the entire team
- The Product Owner has reviewed and signed-off the User Story
- Code coverage on the repository is always above 85%
- Code has been reviewed by peers
- Integration tests completed with mock ups (or existing systems if available) and executed successfully
- Any refactoring required has been completed
- Completed User Story has been deployed on the test environment without issues

4. Estimating and Planning

- Guidelines for Agile planning, including the Cone of Uncertainty
- Steps to create a release plan
- Techniques for prioritization (MoSCoW)
- Techniques for story point sizing
- Delivery planning examples

Agile Planning and Forecasting –Key Principles

Work incrementally, plan incrementally

- Scrum teams work in increments of scope
- User stories are also detailed ‘just in time’ ahead of sprints
- Teams working in Agile understand the “Cone of Uncertainty” and know not to plan the entire project in detail on day 1

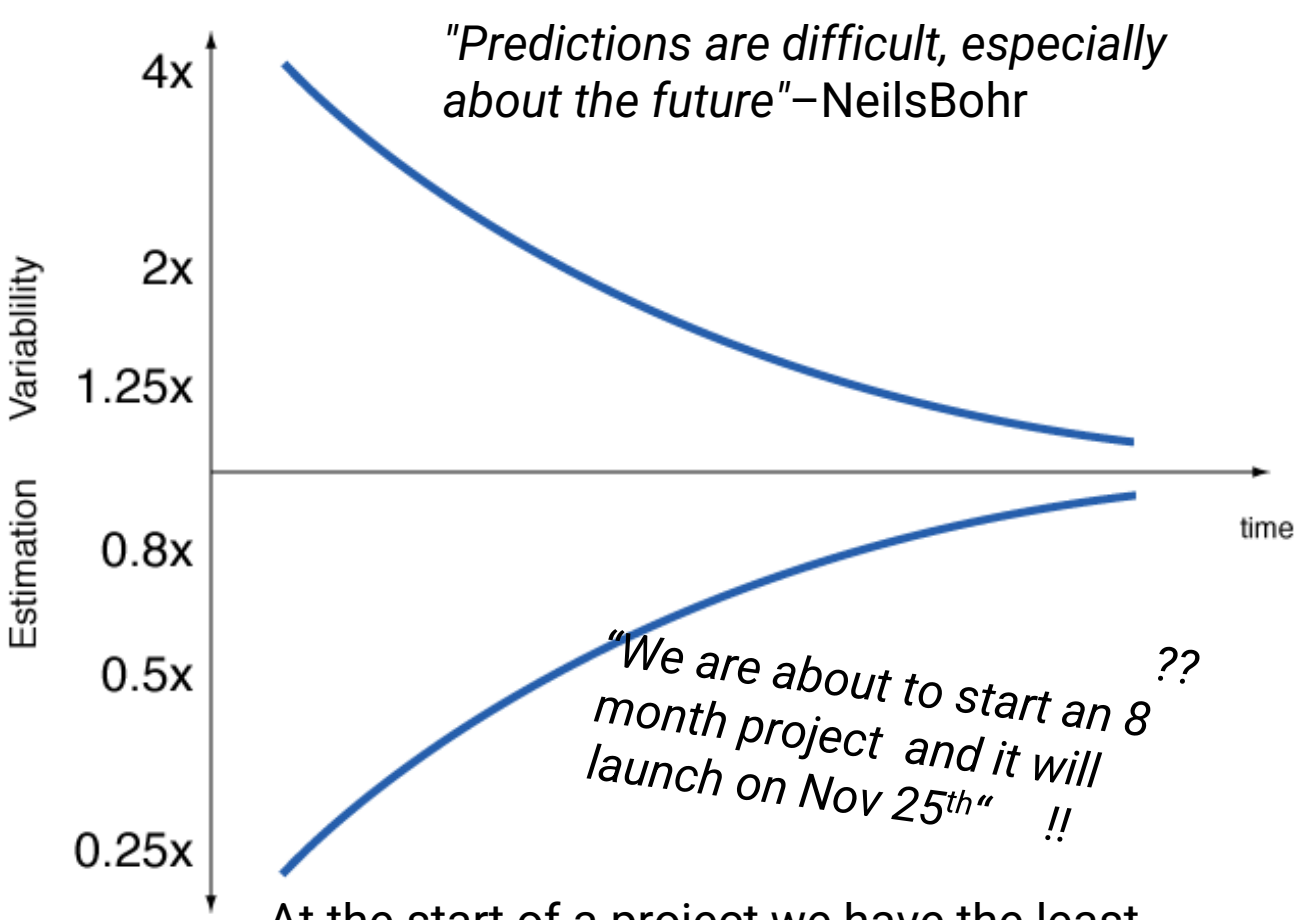
Don’t give fixed dates

- Release plans should use the latest team velocity information to predict delivery timelines
- At the start of a project, only rough velocity ranges are known
- Delivery dates should be only articulated as ranges to avoid false sense of certainty

Fix either time or scope, not both

- Waterfall projects typically fix both time and scope, leading to compromises made (e.g., quality) to hit go live dates
- Agile planning only fixes one of time or scope and uses the latest team velocity to estimate delivery ranges

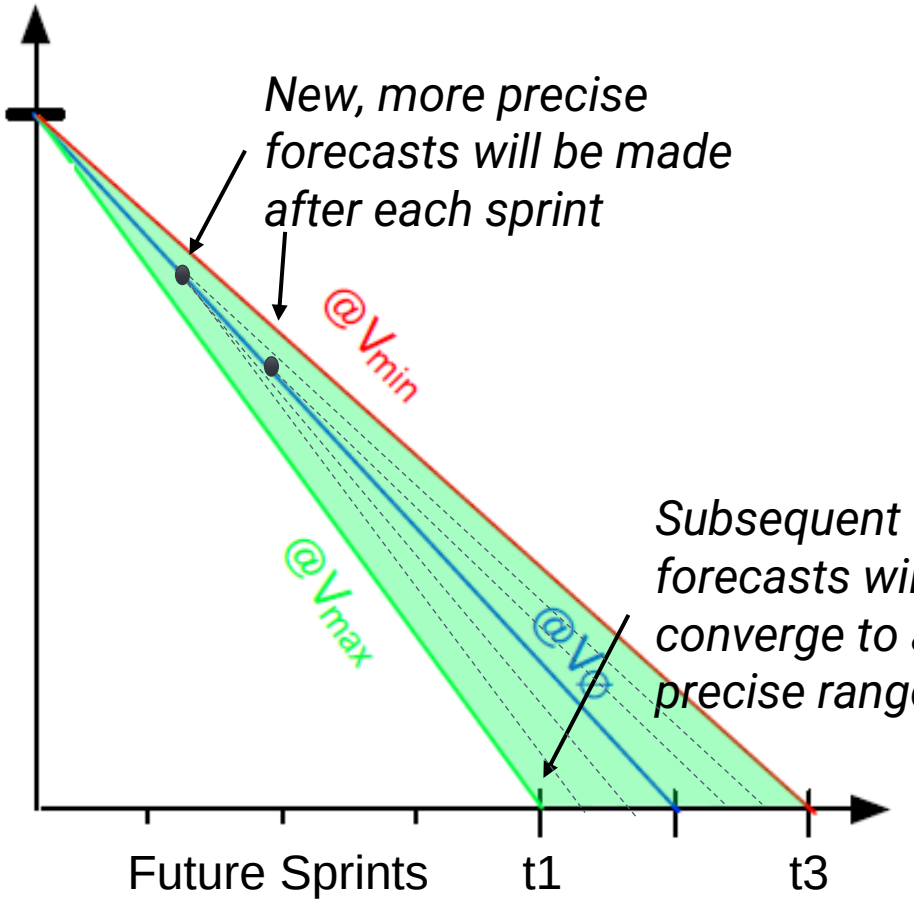
Agile Planning and Forecasting -The Cone of Uncertainty



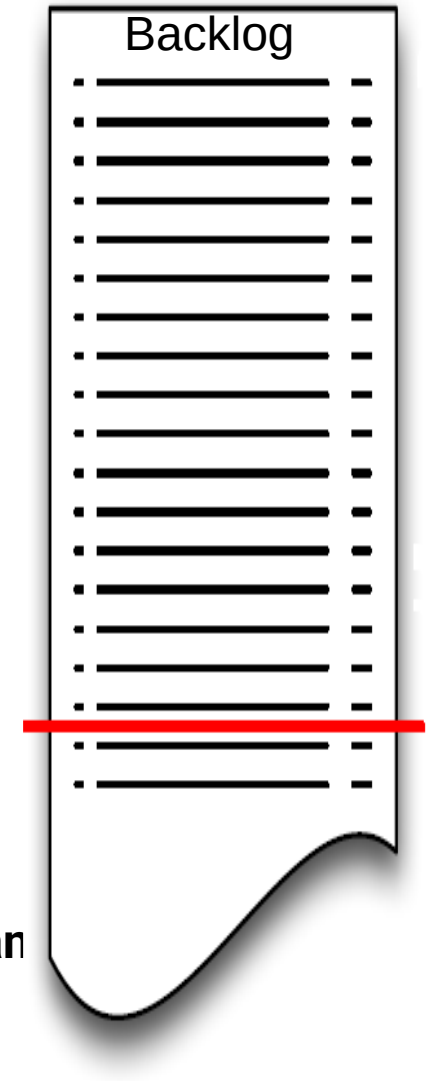
At the start of a project we have the least information. Making precise commitments at this time will lead to poor expectation management.

- At time zero the least is known uncertainty is high
- Knowledge builds as time progresses, reducing uncertainty
- Early predictions will have a high level of variability
- The further out the prediction, the less precise it will be
- Think of a car journey and predicting the ETA. The longer the journey, the less precise the ETA.

Planning and forecasting -with fixed scope

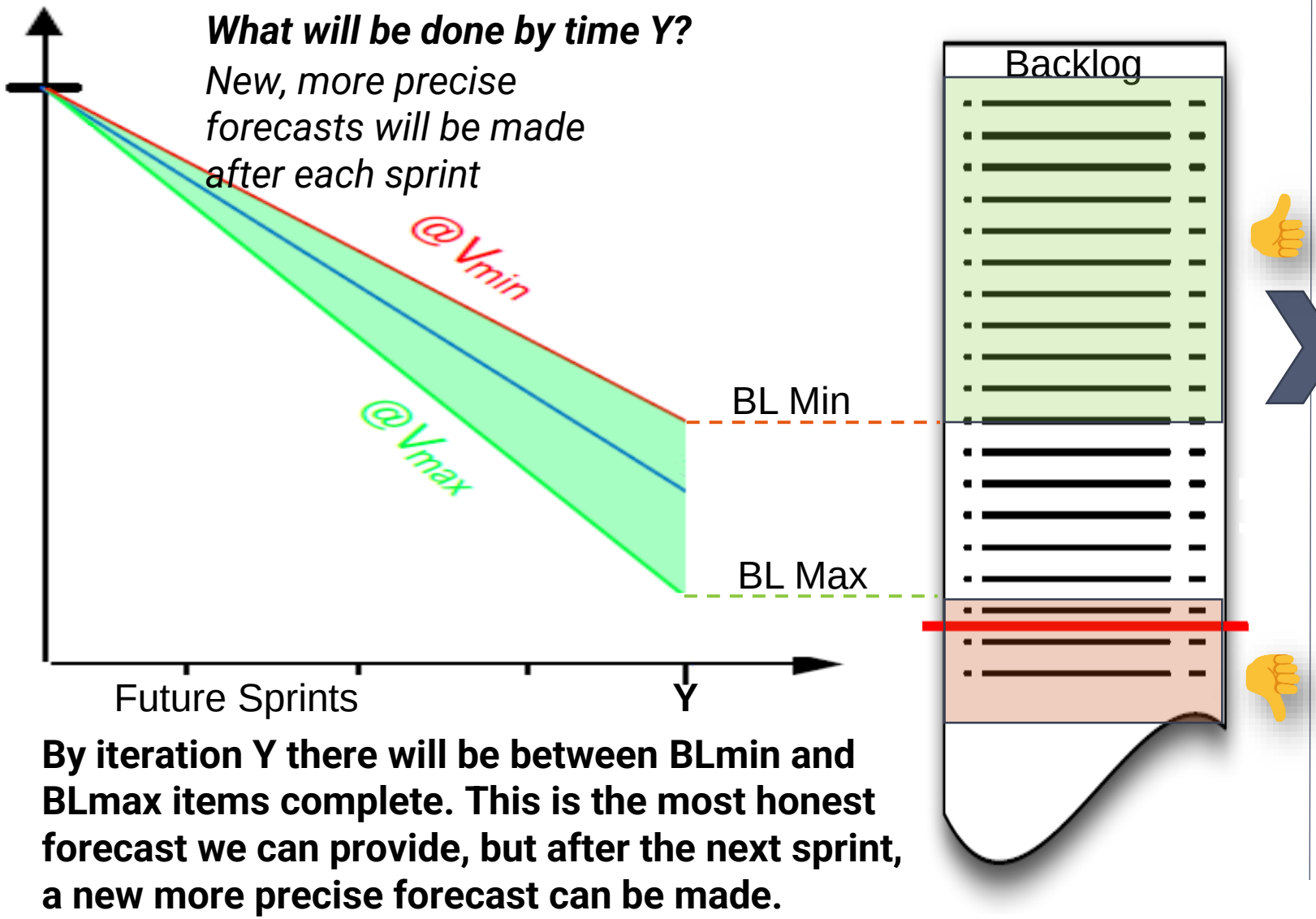


X will be completed between t1 and t2 sprints from now. This is the most honest forecast we can provide, but after the next sprint, a new more precise forecast can be made.



- Create forecast using Vminand Vmax. E.g. If X=100 and Vmin=8 and Vmax=12. We can calculate t1=100/12= 9 iterations and t2=100/8= 13 iterations.
- Therefore, the forecast is; we can definitely have all of X done in 13 iterations from now, but will not be done in less than 9 iterations from now. It will most likely be somewhere in between 9 and 13 iteration[[[ns.
- The forecast is a range of dates, not a single date
- This is the most honest forecast that can be made and is based on what is known at the time.
- However, after the next iteration a new,
- more precise forecast can be

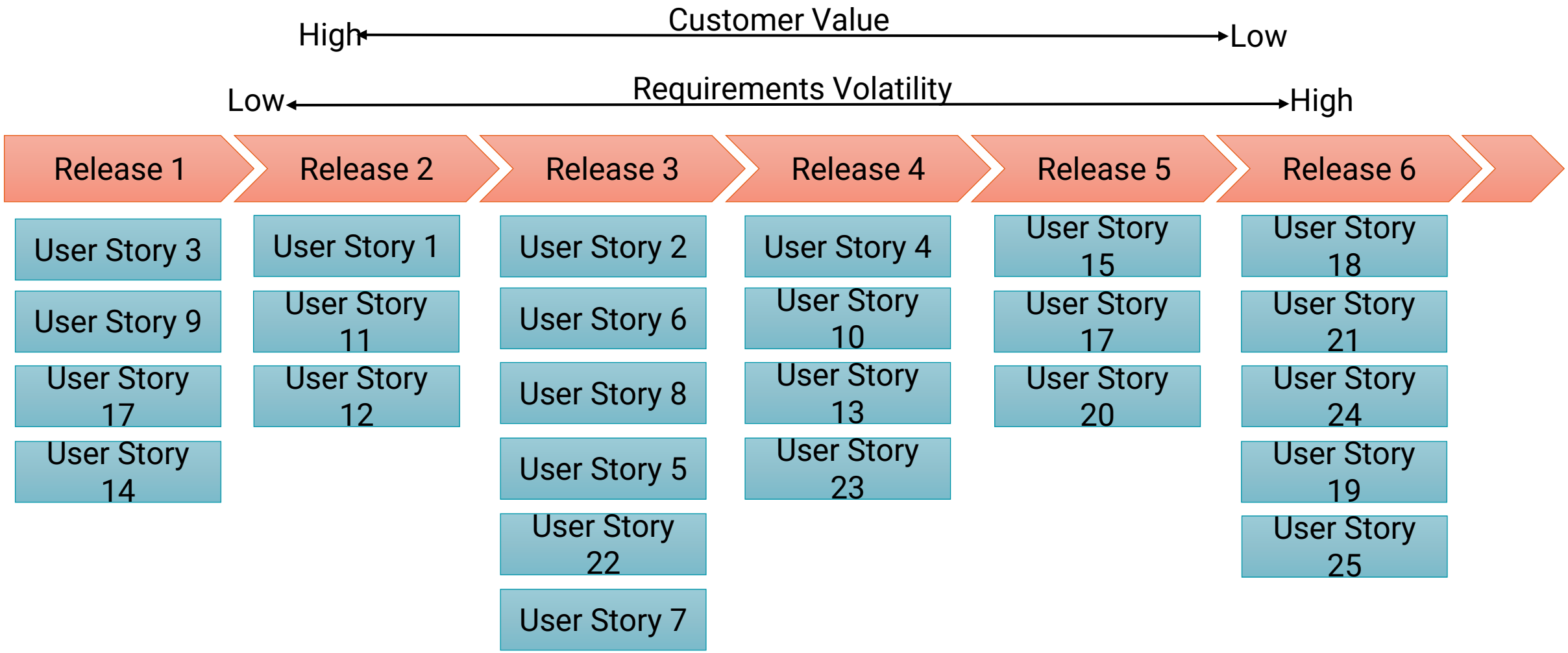
Planning and forecasting -with fixed time



By iteration Y there will be between BLmin and BLmax items complete. This is the most honest forecast we can provide, but after the next sprint, a new more precise forecast can be made.

- Create forecast using Vmin and Vmax. E.g. If Y=4 (iterations away) and Vmin=8 and Vmax=12. We can calculate BLmin=8x4= 32 backlog (BL) items and BLmax=12/4= 48 backlog (BL) items.
- Therefore, the forecast is; we can definitely have 32 BL items done by sprint Y, but no more than 48 BL items. It will most likely be somewhere in between 32 and 48 items.
- The forecast is made as a range on the backlog, not a single point.
- This is the most honest forecast that can be made and is based on what is known at the time.
- However, after the next sprint a new, more precise forecast can be made

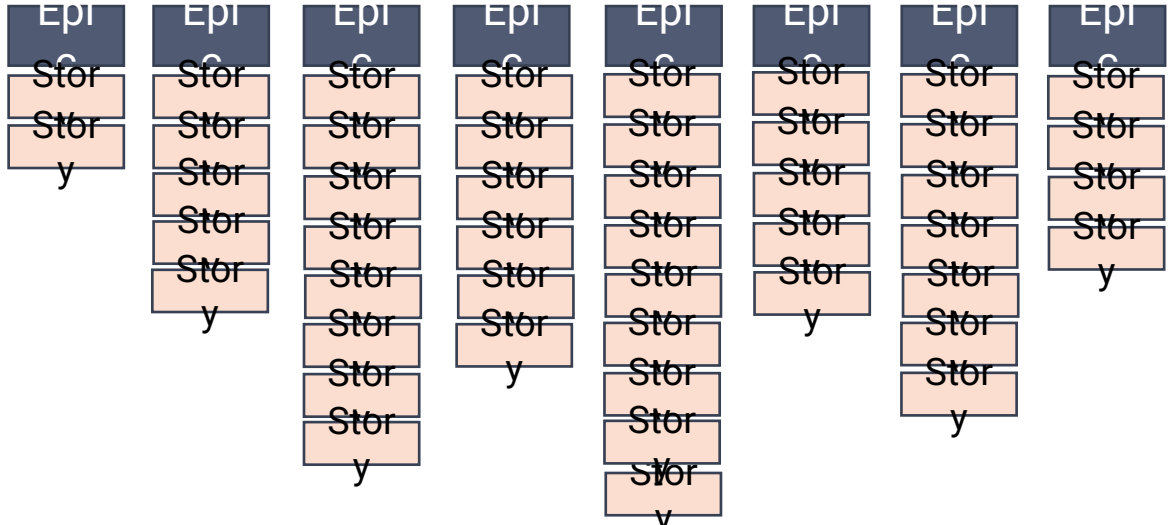
A “Release Plan” defines the product road map in the form of the first release and subsequent evolutions



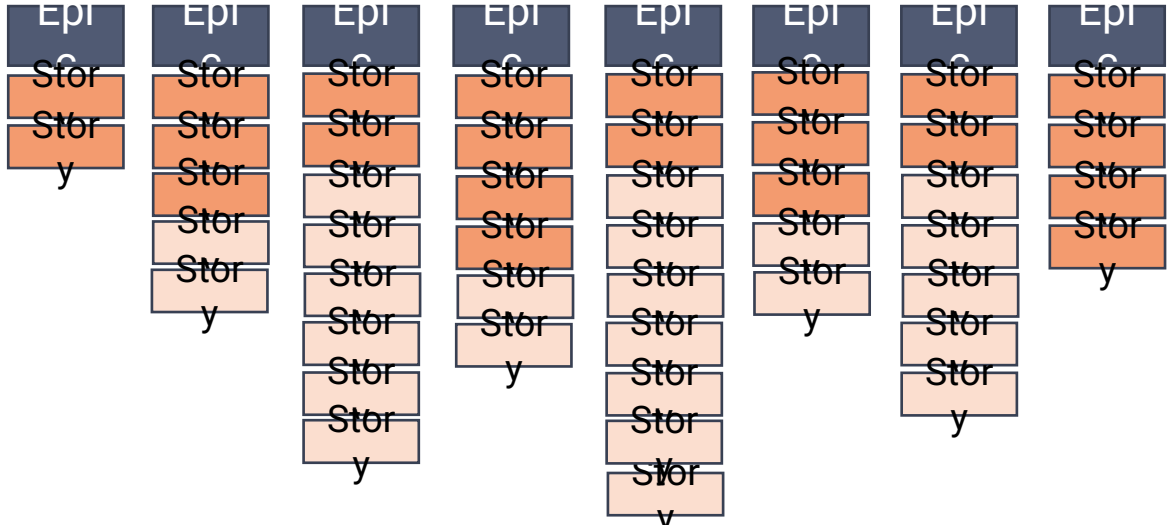
- The first version of the release plan is based on estimated priorities and dependencies and captured generally in an online tool for distributed teams or on a project wall with physical cards for collocated teams.
- It is revised at the end of every sprint based on new information and priorities

How to release plan in 4 steps

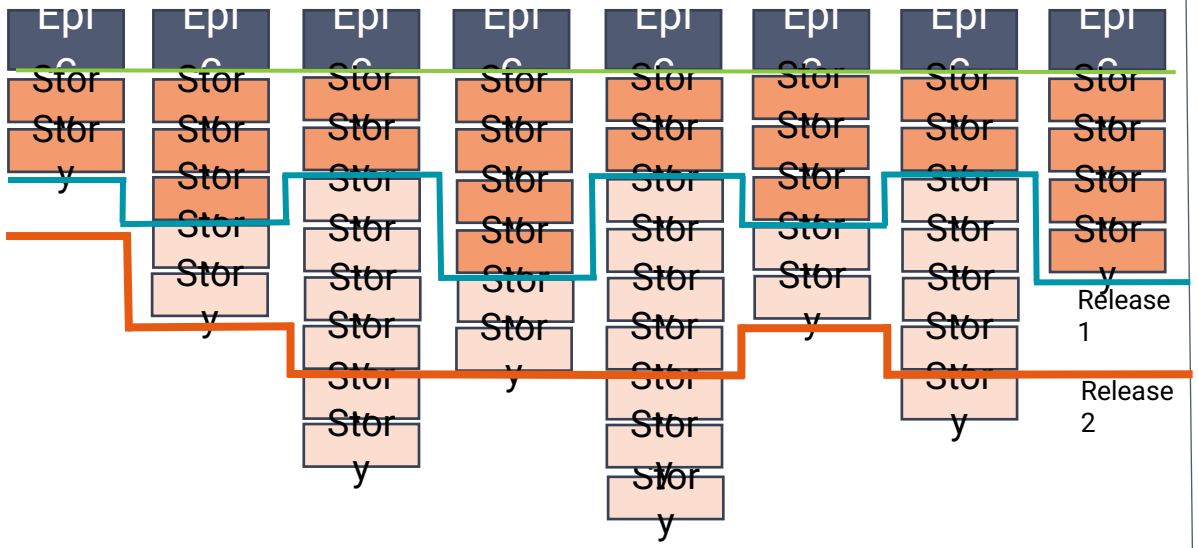
Step 1: Go through list of epics/stories to check completeness



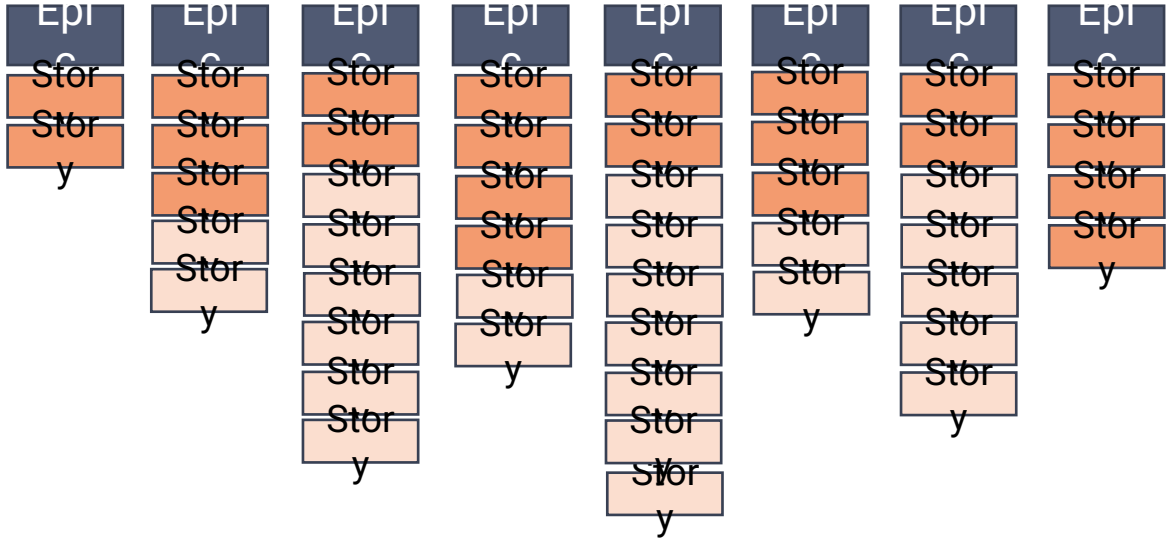
Step 2: Prioritise user stories based on necessity for release



Step 3: Decide what is necessary for each release



Step 4: Size release 1 stories using poker planning cards



ALTIMETRIK
SIMPLIFY TECHNOLOGY | AMPLIFY POSSIBILITY

Feedback

Thank you

What next ?