Talent Academy
Empowering future leaders
An initiative of excelencia

excelencia
People First IT Services Ecosystem

EPIC

Effective Professional Inspiring Competent

Project Management

Fundamentals

Agile Concepts

Agenda

- Predictive vs Adaptive
- The Agile Manifesto
- Agile Frameworks
- Scrum Framework

Predictive vs Adaptive

Which Project Approach is this?

Agile

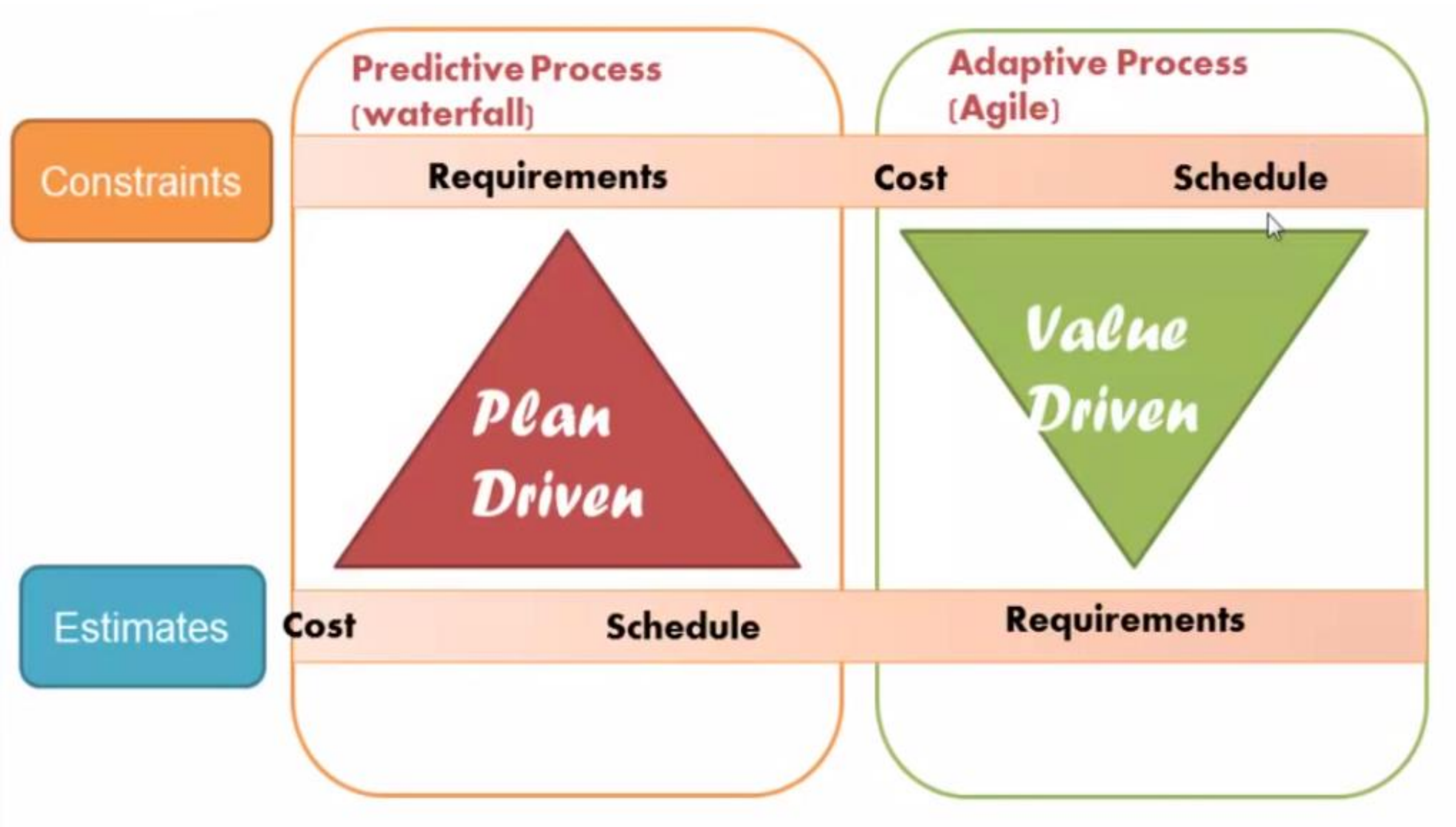
Waterfall

Hybrid

Value Driven Approach

Fixed

Variable



Predictive Process (waterfall)

Adaptive Process (Agile)

Constraints

Requirements

Cost

Schedule

Plan Driven

Value Driven

Estimates

Cost

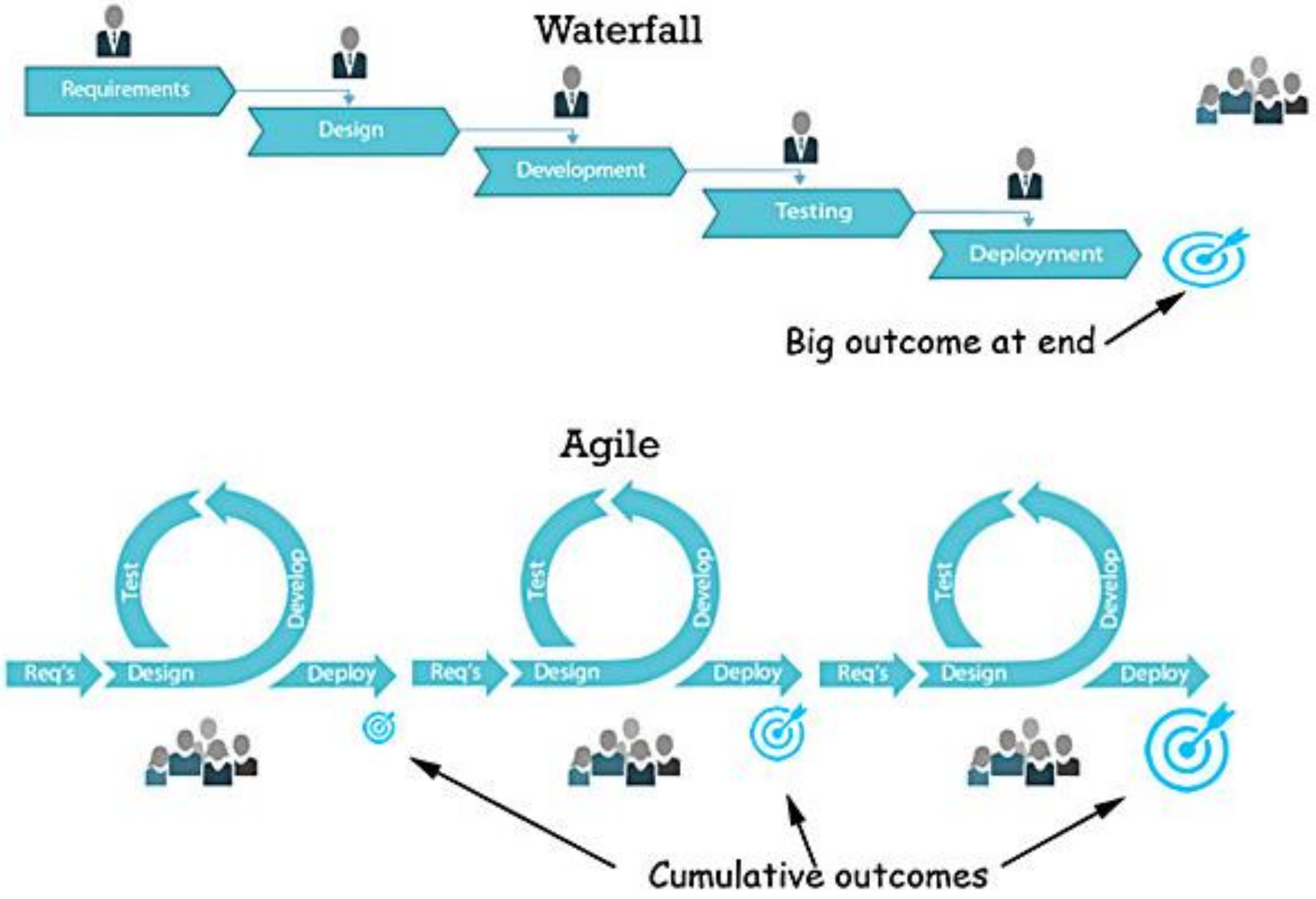
Schedule

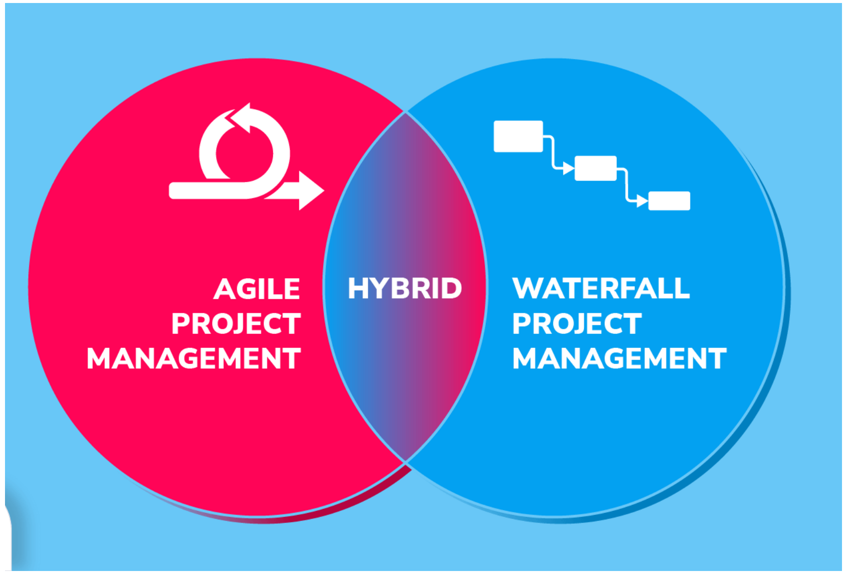
Requirements

Comparison

	Delivery Model	Requirement Clarity	Customer Involvement	Change	Risks
Traditional	Phases	High	Low	Formally controlled	Identified & managed
Agile	Time-boxed cadence	Evolving	High	Inherent	High tolerance
Hybrid	Tailored development approach, combining these elements				

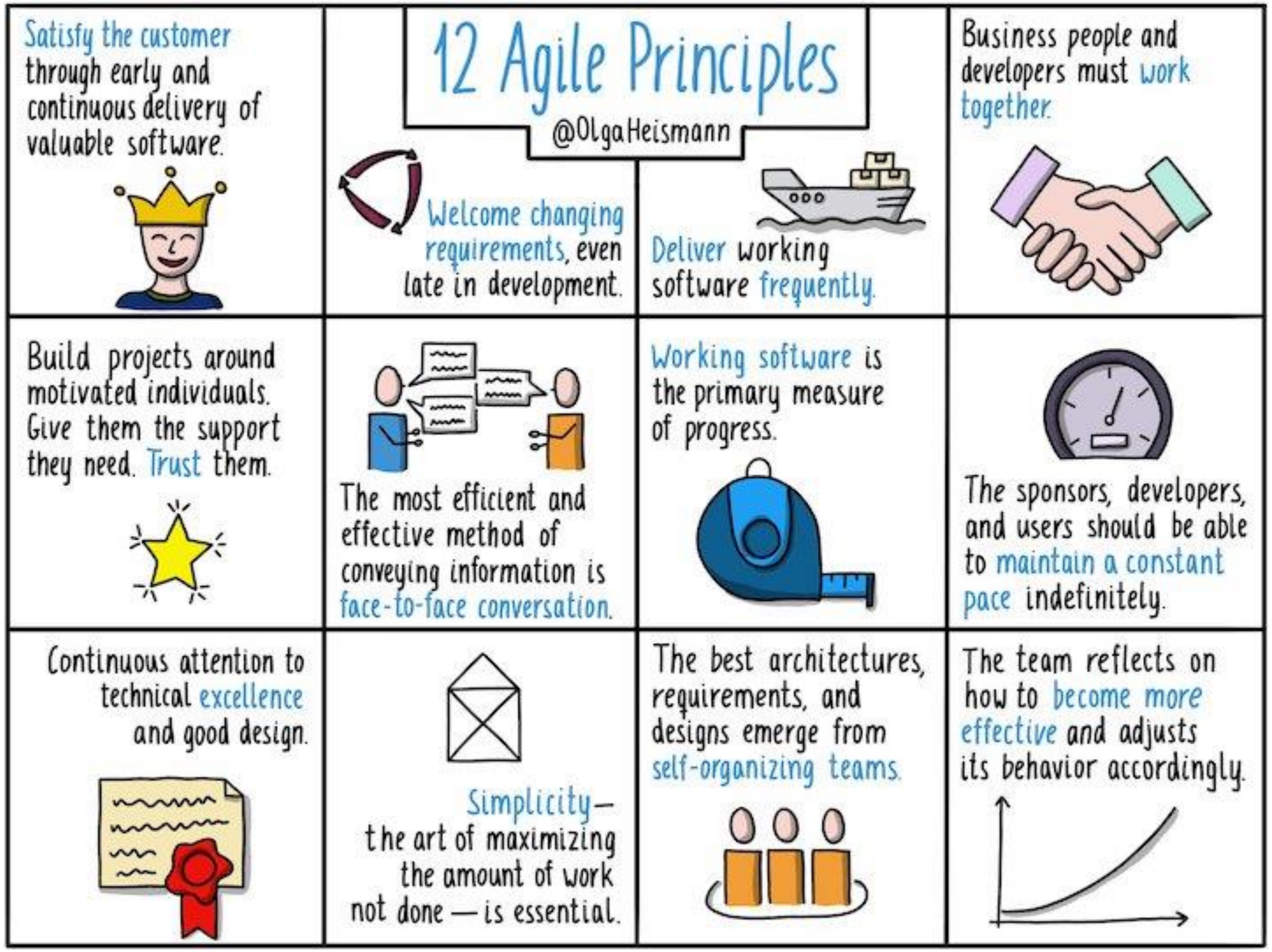
Comparison


Waterfall
Requirements
Design
Development
Testing
Deployment
Big outcome at end
Agile
Req's
Design
Test
Develop
Deploy
Req's
Design
Test
Develop
Deploy
Req's
Design
Test
Develop
Deploy
Cumulative outcomes


AGILE PROJECT MANAGEMENT
HYBRID
WATERFALL PROJECT MANAGEMENT

The Agile Manifesto

The Agile Manifesto



12 Agile Principles

@OlgaHeismann

Satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development.

Deliver working software frequently.

Business people and developers must work together.

Build projects around motivated individuals. Give them the support they need. Trust them.

The most efficient and effective method of conveying information is face-to-face conversation.

Working software is the primary measure of progress.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design.

Simplicity—the art of maximizing the amount of work not done—is essential.

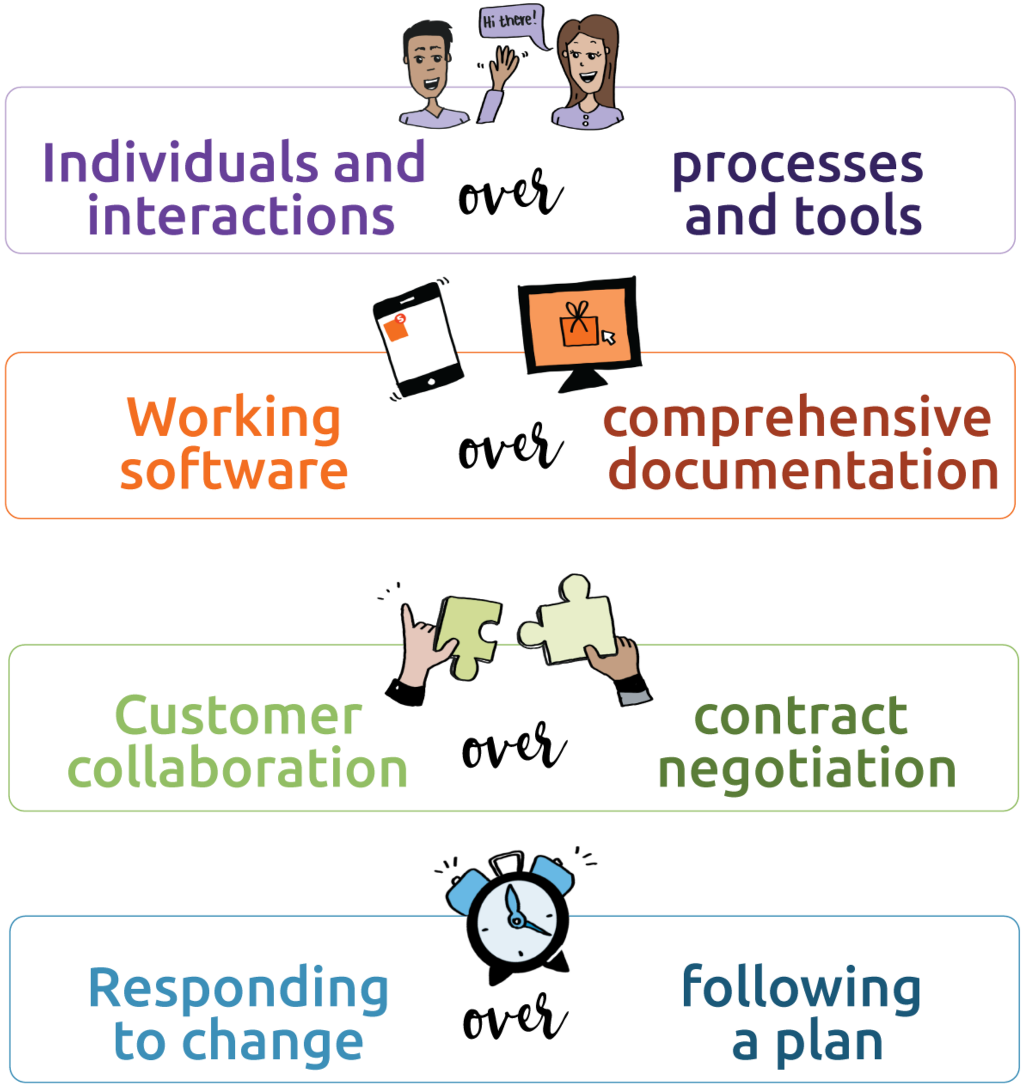
The best architectures, requirements, and designs emerge from self-organizing teams.

The team reflects on how to become more effective and adjusts its behavior accordingly.

The Agile Manifesto

“ 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

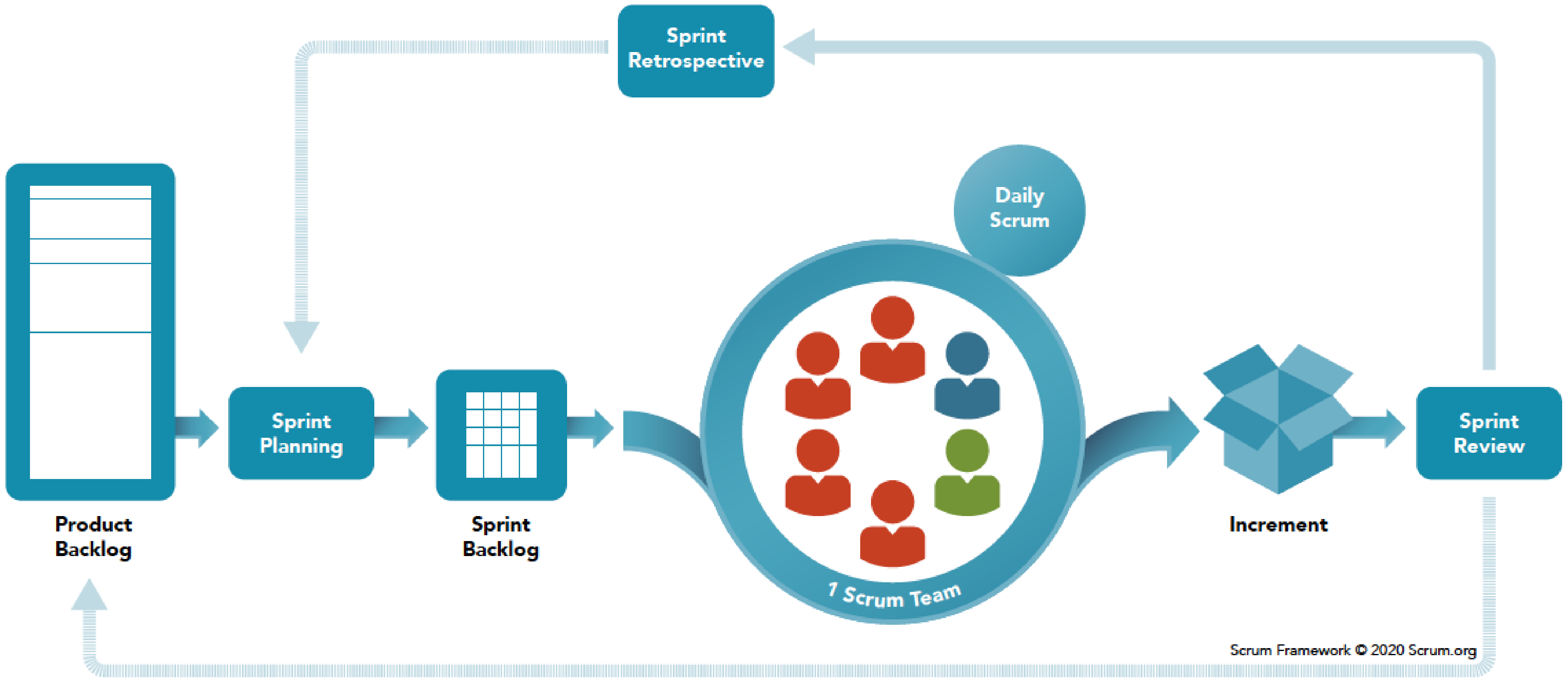
Responding to change over following a plan

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

Agile Frameworks

Scrum Framework

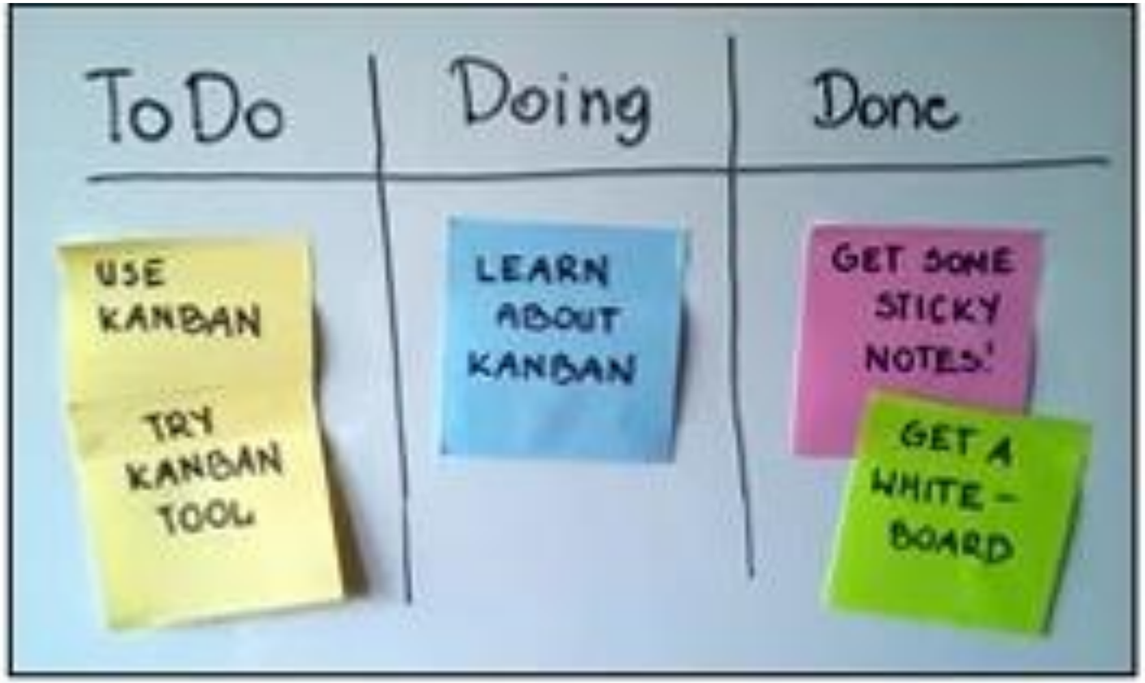
Focuses on time-boxed iterations to
implement the highest priority requirements
by a cross functional team



Scrum Framework © 2020 Scrum.org

Kanban Framework

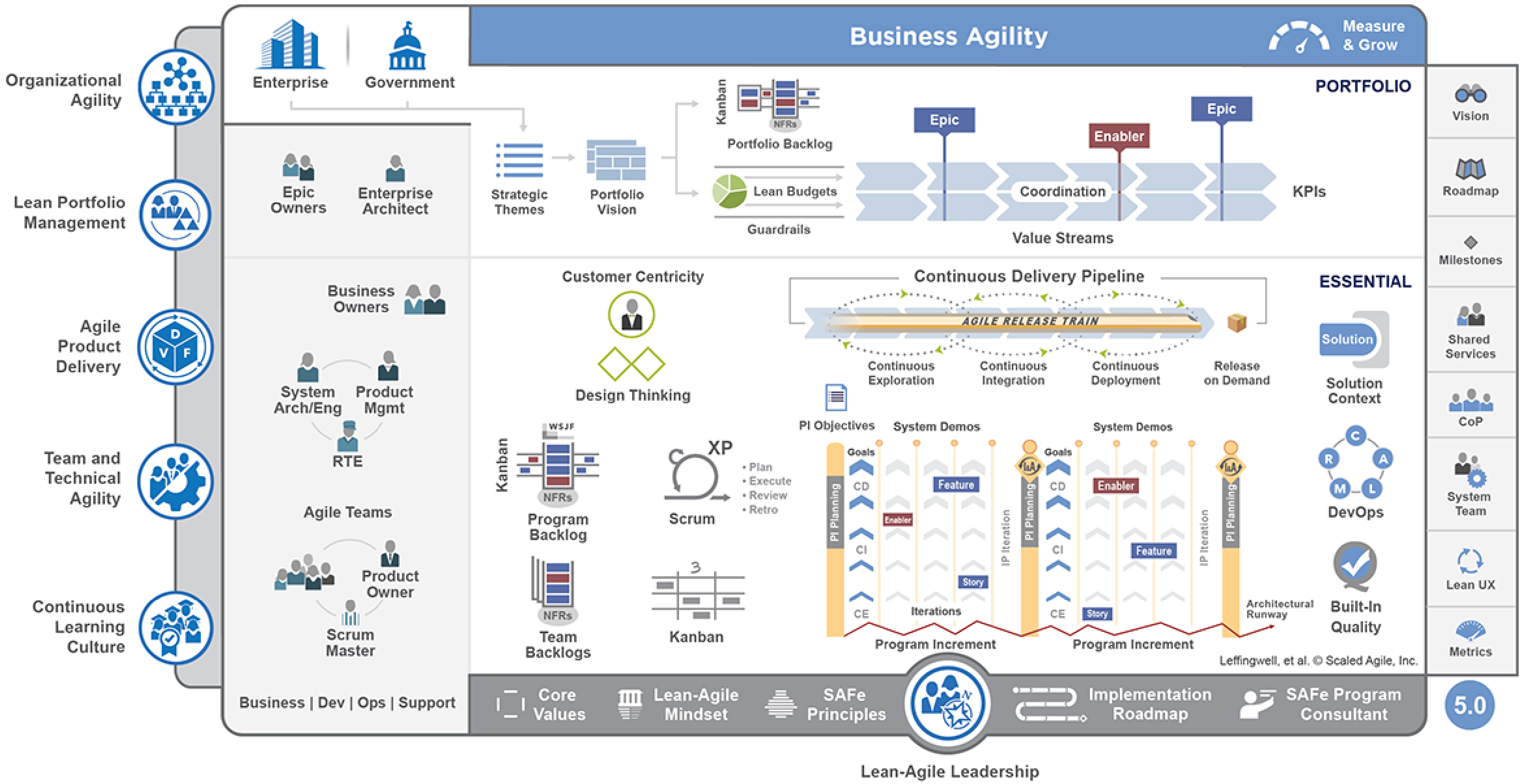
Focuses on visualizing the entire project on boards in order to increase project transparency and collaboration between team members



Kanban Board

Scaled Agile Framework® (SAFe®)

Organizational and workflow patterns for implementing agile practices at an enterprise scale



Organizational Agility

Lean Portfolio Management

Agile Product Delivery

Team and Technical Agility

Continuous Learning Culture

Enterprise

Government

Business Agility

Measure & Grow

PORTFOLIO

Epic Owners

Enterprise Architect

Strategic Themes

Portfolio Vision

Kanban

NFRs

Portfolio Backlog

Lean Budgets

Guardrails

Epic

Enabler

Epic

Coordination

Value Streams

KPIs

Vision

Roadmap

Milestones

Shared Services

CoP

System Team

Lean UX

Metrics

ESSENTIAL

Business Owners

System Arch/Eng

Product Mgmt

RTE

Agile Teams

Product Owner

Scrum Master

Business | Dev | Ops | Support

Customer Centricity

Design Thinking

Continuous Delivery Pipeline

AGILE RELEASE TRAIN

Continuous Exploration

Continuous Integration

Continuous Deployment

Release on Demand

Solution

Solution Context

Kanban

WSJF

NFRs

Program Backlog

XP

Scrum

- Plan
- Execute
- Review
- Retro

NFRs

Team Backlogs

Kanban

PI Objectives

System Demos

System Demos

PI Planning

Goals

CD

CI

CE

Feature

Enabler

Story

Iterations

IP Iteration

PI Planning

Goals

CD

CI

CE

Enabler

Feature

Story

IP Iteration

PI Planning

Program Increment

Program Increment

Architectural Runway

DevOps

C

A

L

M

R

Built-In Quality

Leffingwell, et al. © Scaled Agile, Inc.

Core Values

Lean-Agile Mindset

SAFe Principles

Implementation Roadmap

SAFe Program Consultant

5.0

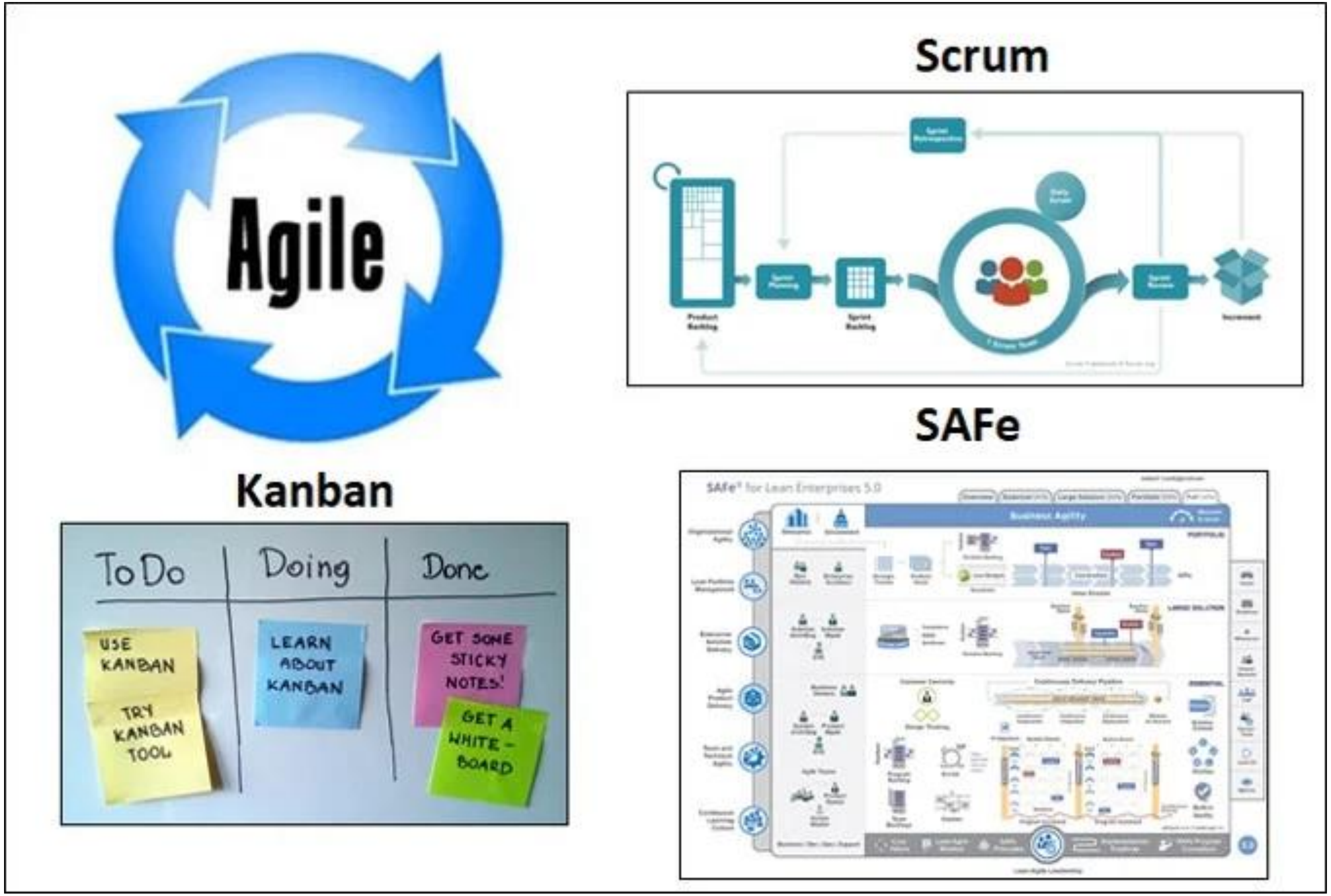
Lean-Agile Leadership

Talent Academy

Empowering future leaders

An initiative of excelencia

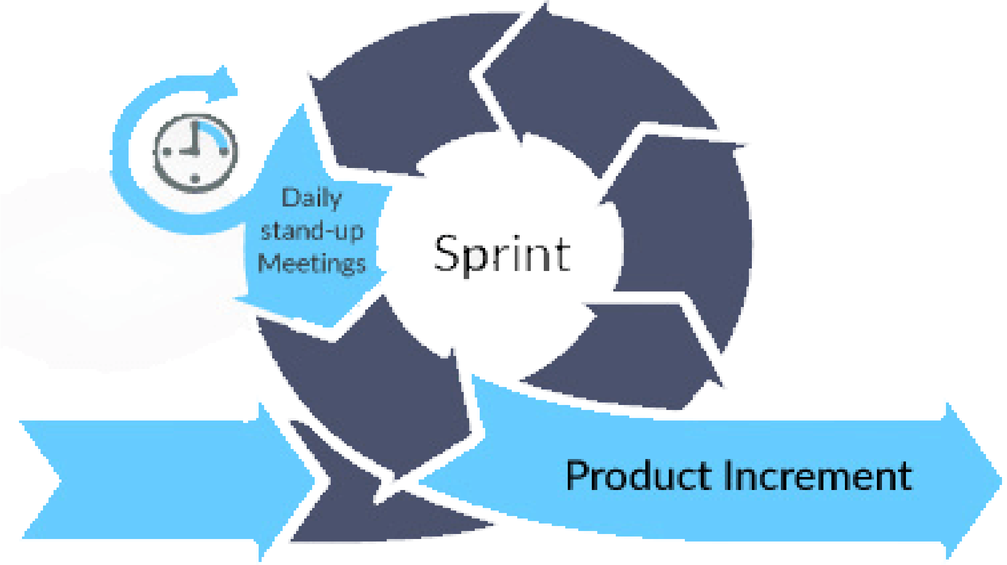
Summary


Agile
Scrum
Sprint Retrospective
Daily Scrum
Product Backlog
Sprint Planning
Sprint Backlog
Sprint Review
Increment
SAFe
SAFe® for Lean Enterprises 5.0
Business Agility
Kanban
To Do
Doing
Done
USE KANBAN
TRY KANBAN TOOL
LEARN ABOUT KANBAN
GET SOME STICKY NOTES!
GET A WHITE-BOARD

Scrum Framework

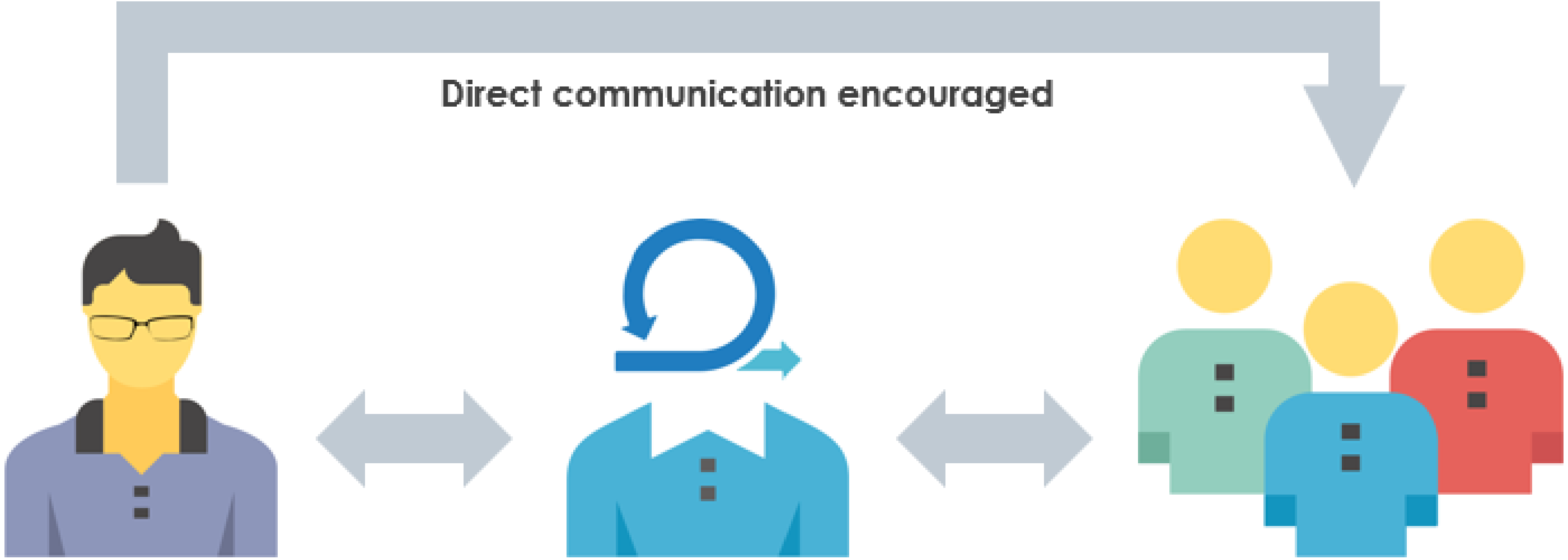
Scrum Framework

Get work done as a **team** in small pieces at a time, with **continuous** experimentation and **feedback** loops along the way to learn and **improve** as you go



Scrum Teams

Group of collaborators, typically between five and nine individuals, who work toward completing projects and delivering products



Direct communication encouraged

Product Owner:
Owns "what" is desired
And "why" it's desired

Scrum Master:
Keeper of Scrum
Process, facilitator

Scrum Team Members:
Owns "how" and "how quickly"
work is delivered

Project Team Formation

Self-organizing team

A cross-functional team in which people fluidly assume leadership as needed to achieve the team’s objectives.

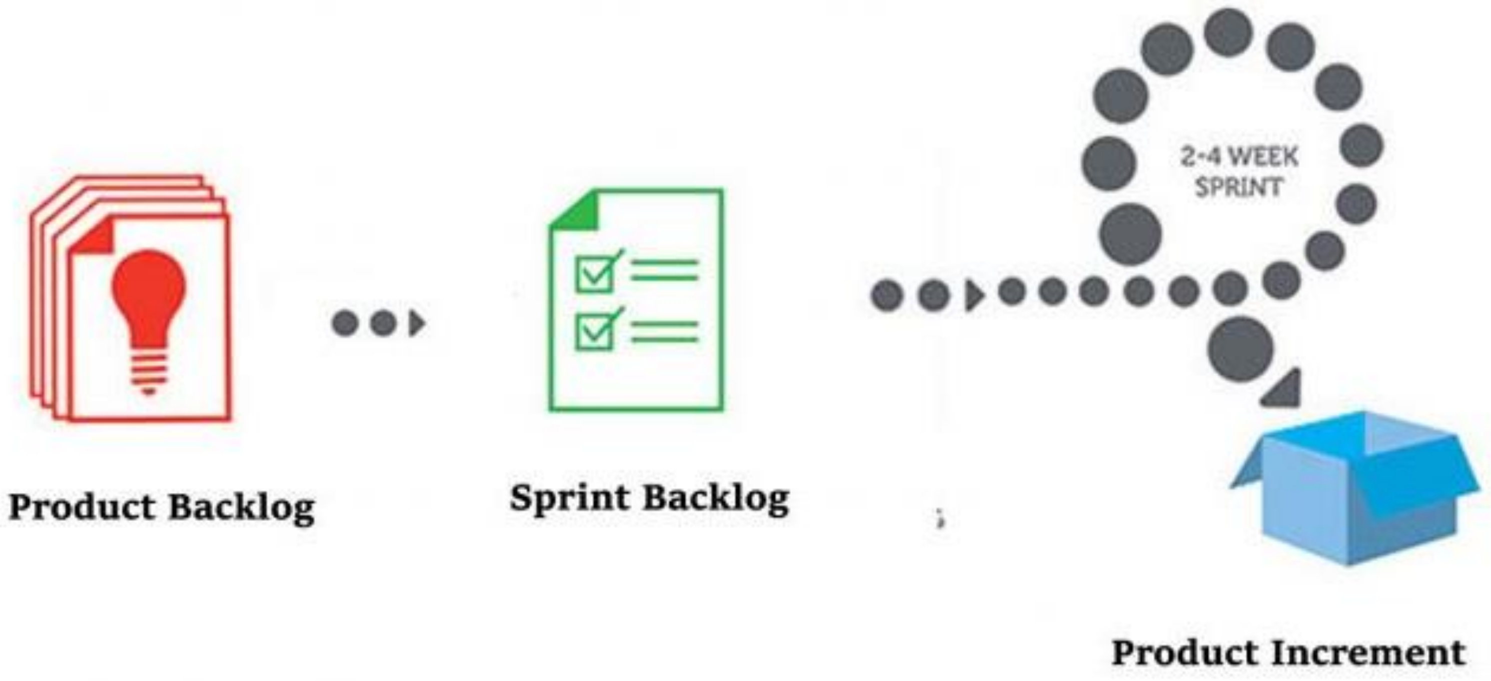
Cross-functional team

Teams that have all the capabilities to deliver the work they’ve been assigned. Team members can specialize in certain skills, but the team is capable of delivering what they’ve been called on to build. See also “self-organizing teams.

These concepts can be applied in any kind of project team.

Scrum Artifacts

Information that a scrum team and stakeholders use to detail the product being developed, actions to produce it, and the actions performed during the project

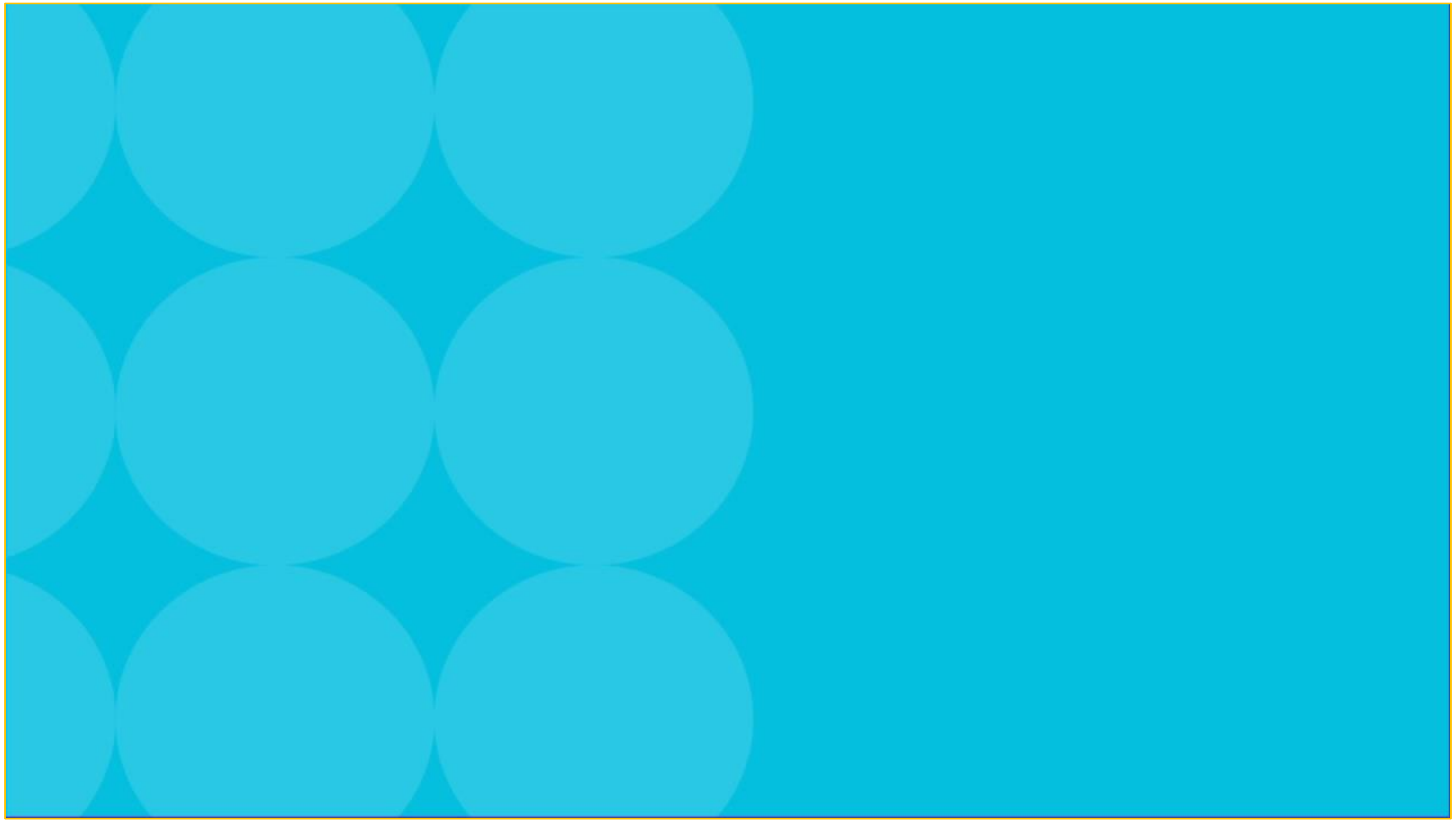


Product Backlog

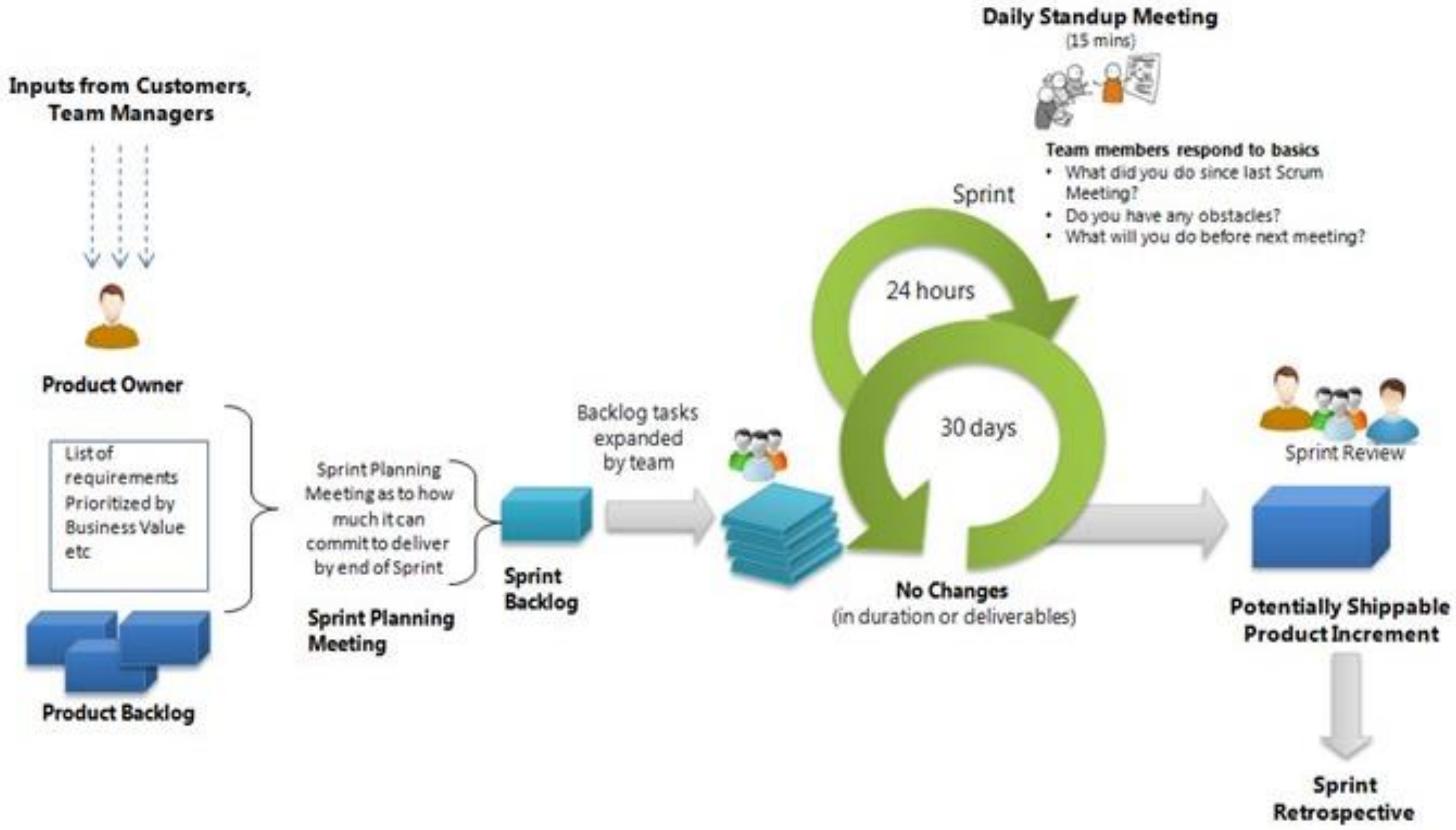
Sprint Backlog

Product Increment

Iterative Way of Working: Video



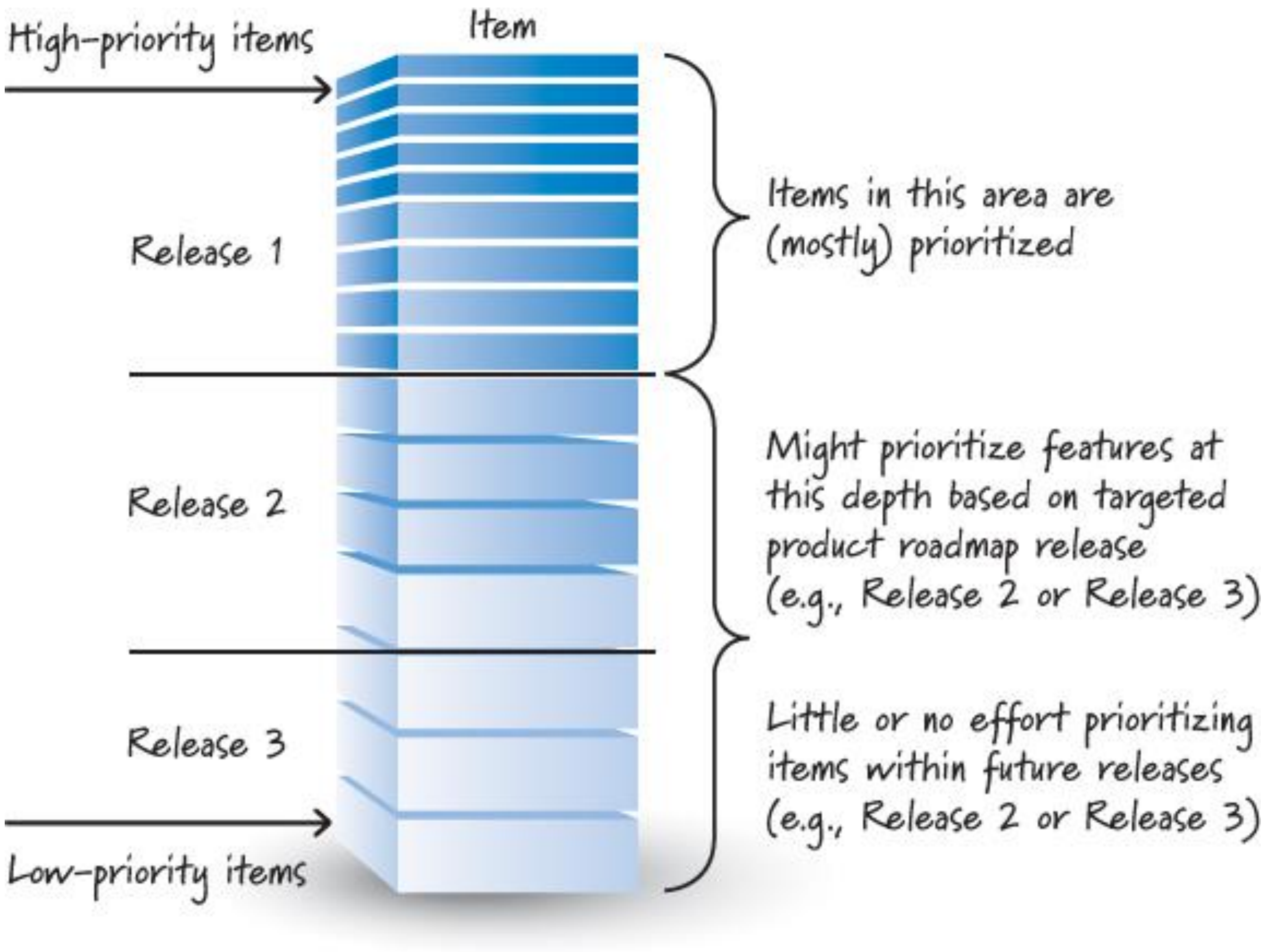
Summary


Inputs from Customers, Team Managers
Product Owner
List of requirements Prioritized by Business Value etc
Product Backlog
Sprint Planning Meeting as to how much it can commit to deliver by end of Sprint
Sprint Planning Meeting
Sprint Backlog
Backlog tasks expanded by team
Daily Standup Meeting
(15 mins)
Team members respond to basics
• What did you do since last Scrum Meeting?
• Do you have any obstacles?
• What will you do before next meeting?
Sprint
24 hours
30 days
No Changes
(in duration or deliverables)
Sprint Review
Potentially Shippable Product Increment
Sprint Retrospective

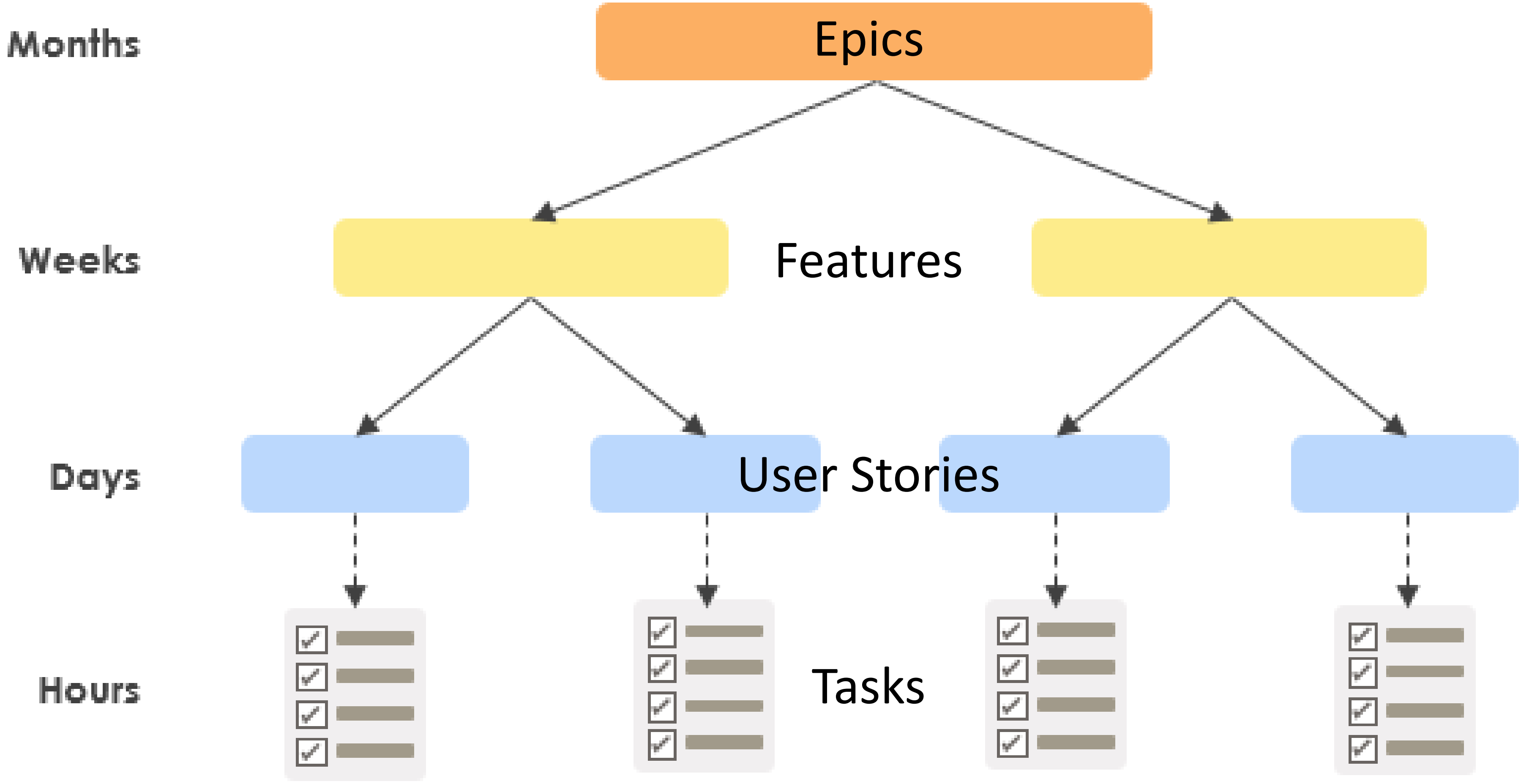
Scope Management

Product Backlog

- List of ALL the requirements / desired work on the project (similar to project scope)
- Prioritized by the Product Owner
- Reprioritized at the start of each sprint


High-priority items
Item
Release 1
Release 2
Release 3
Low-priority items
Items in this area are (mostly) prioritized
Might prioritize features at this depth based on targeted product roadmap release (e.g., Release 2 or Release 3)
Little or no effort prioritizing items within future releases (e.g., Release 2 or Release 3)

Breaking down the work



Requirement Documentation

EPIC

A **large body of work** that can be broken down into smaller pieces—features and user stories. Epics can **take months** to complete.

FEATURE

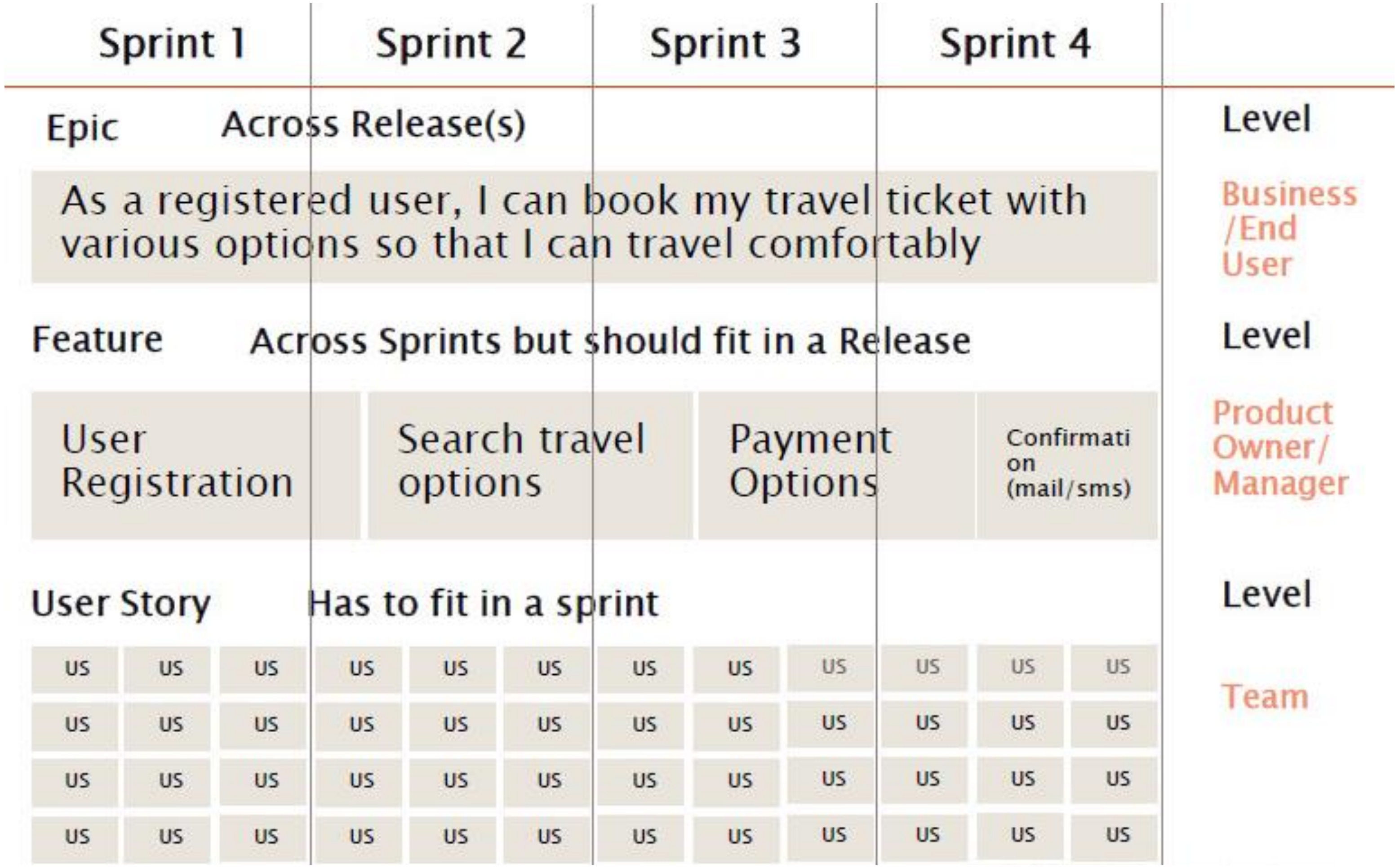
A set of **related requirements** that allows the user to **satisfy a business objective** or need.

USER STORY

•A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

•The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

Requirement Documentation



Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4

Epic Across Release(s) — Level: Business /End User

As a registered user, I can book my travel ticket with various options so that I can travel comfortably

Feature Across Sprints but should fit in a Release — Level: Product Owner/ Manager

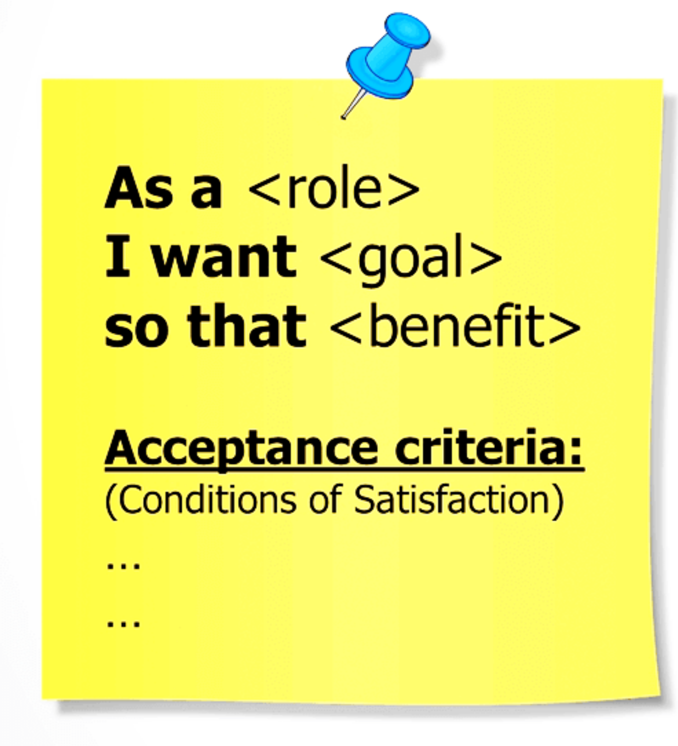
User Registration | Search travel options | Payment Options | Confirmation (mail/sms)

User Story Has to fit in a sprint — Level: Team

US US US US US US US US US US US US
US US US US US US US US US US US US
US US US US US US US US US US US US
US US US US US US US US US US US US

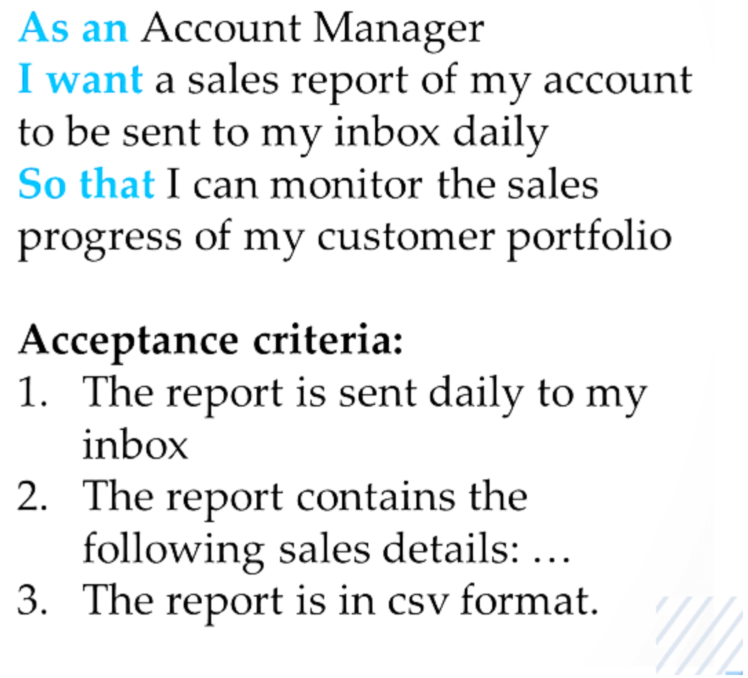
User Story

Requirements are recorded as User Stories in Scrum Framework



As a <role>
I want <goal>
so that <benefit>

Acceptance criteria:
(Conditions of Satisfaction)
…
…



As an Account Manager
I want a sales report of my account to be sent to my inbox daily
So that I can monitor the sales progress of my customer portfolio

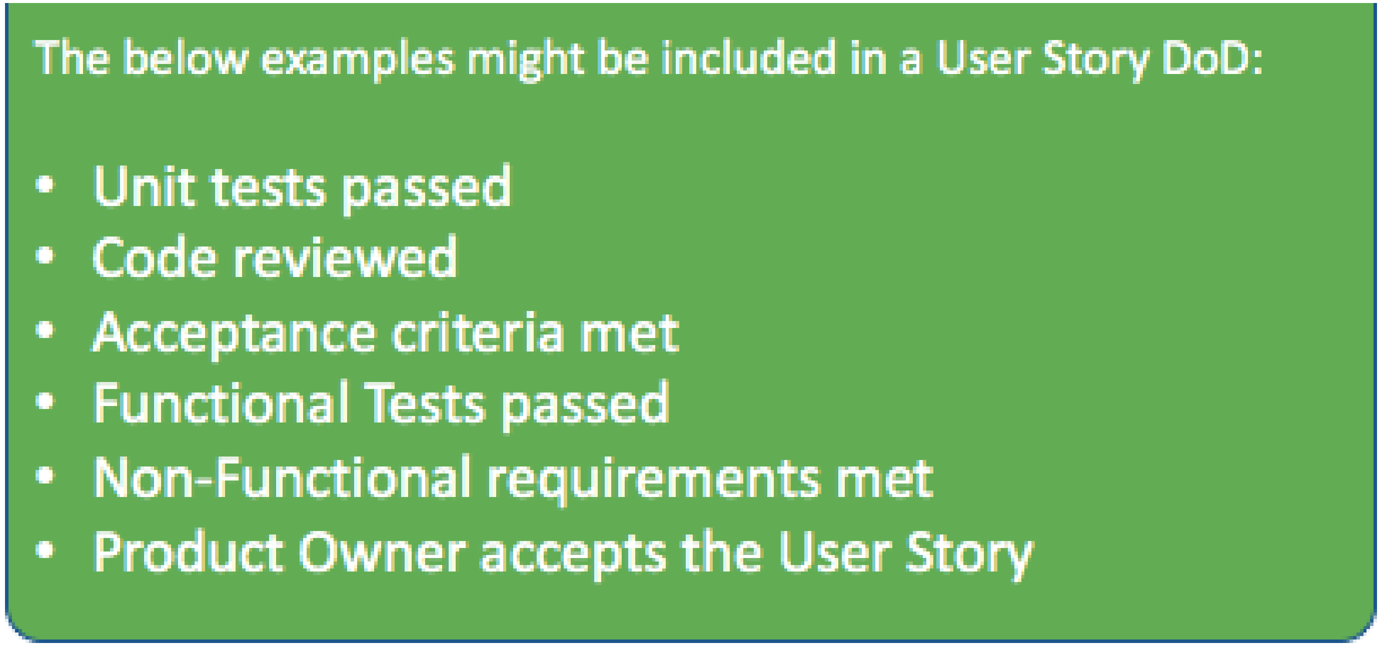
Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: …
3. The report is in csv format.

Can have Acceptance Criteria & must comply to Definition of Done (DoD)

Definition of Done (DoD)

A team's checklist of all the criteria required to be met so that a deliverable can be considered ready for customer use.



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

Schedule Management

Release and Iteration Planning

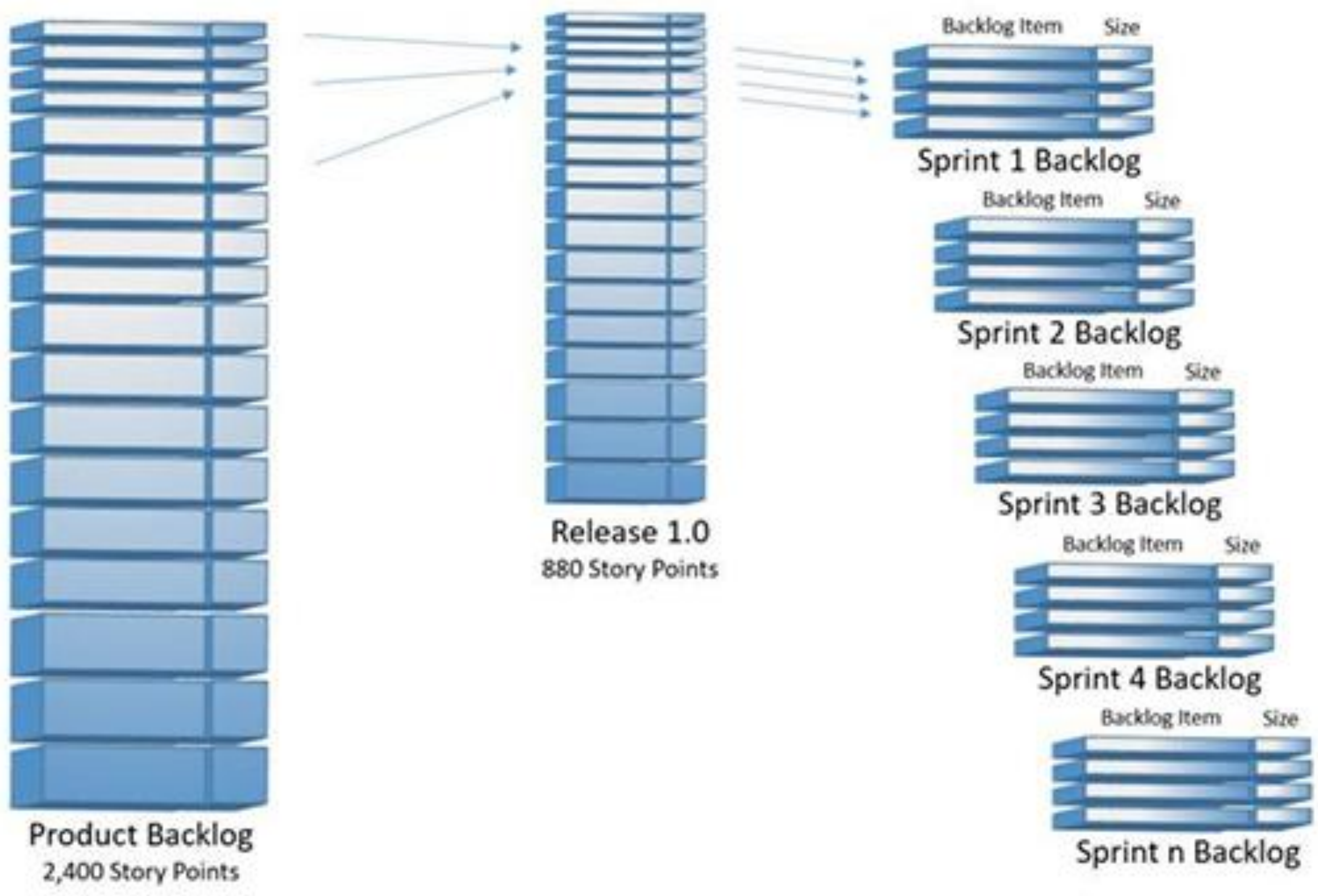
Release Planning

- Number of iterations or sprints needed to complete each release
- Features contained in the release
- Goal dates of each release
- Usually for 3-6 months

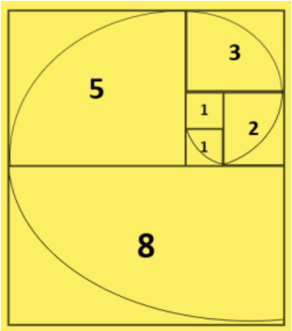
Sprint Planning

- Review the highest prioritized user stories or key outcomes
- Ask questions
- Agree on effort required to complete the user story in the current iteration
- Determine the activities required to deliver iteration objectives
- Time-boxed iterations or sprints typically last 1 - 4 weeks

Release & Iteration Planning



Measure Effort, Not Time



Relative sizing

- Compares effort of multiple user stories through assignment of values (XS, S, M, L, XL)
- *Use common **t-shirt sizes** to assign values to user stories.*

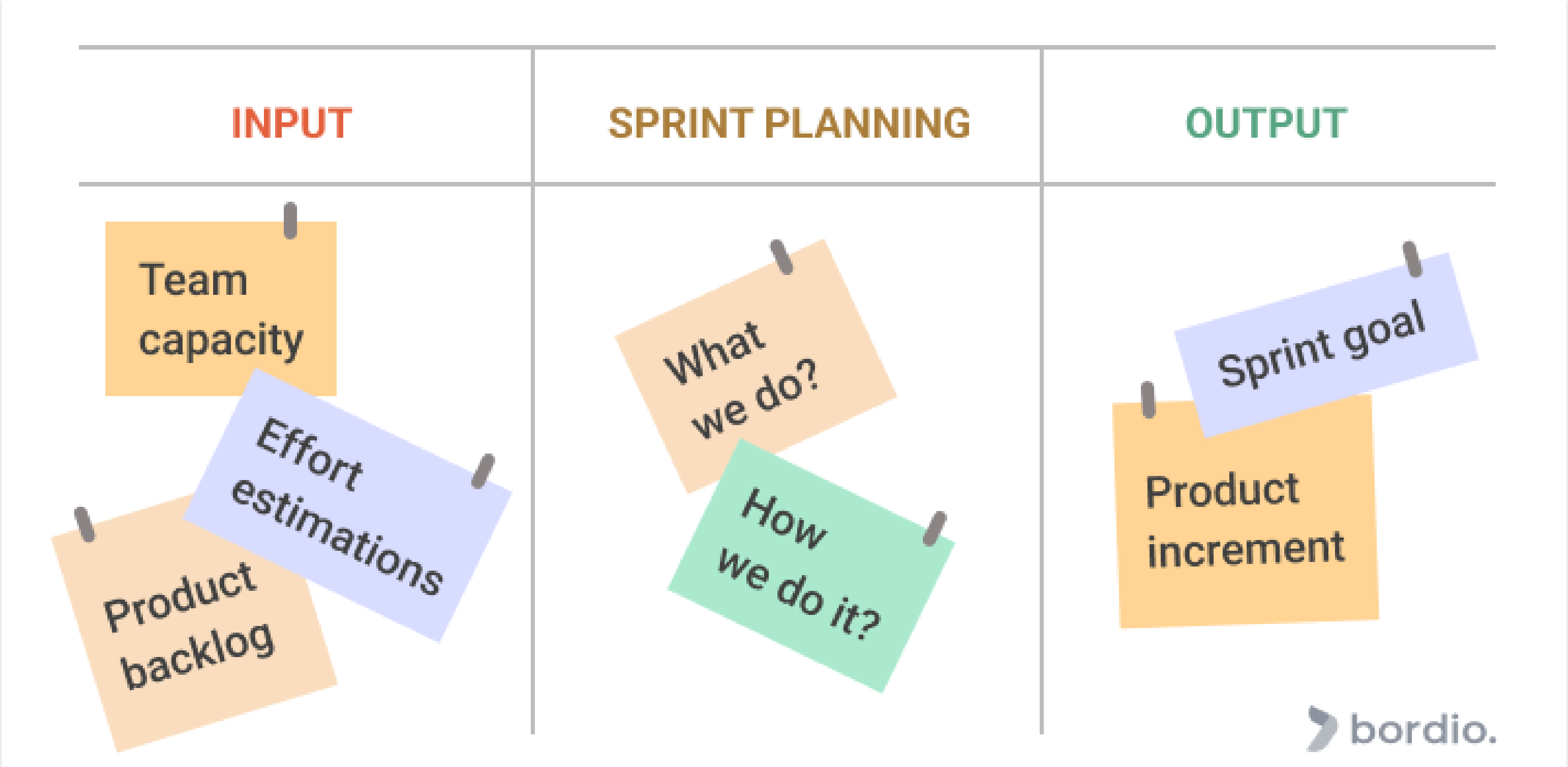
Story points

- Uses a relative measure – e.g., numbers in the **Fibonacci sequence** – to identify the level of difficulty or complexity of a user story or task

Planning poker

- Estimates effort or relative size of development effort
- Uses a deck of cards with modified Fibonacci numbers to vote on user stories

Summary

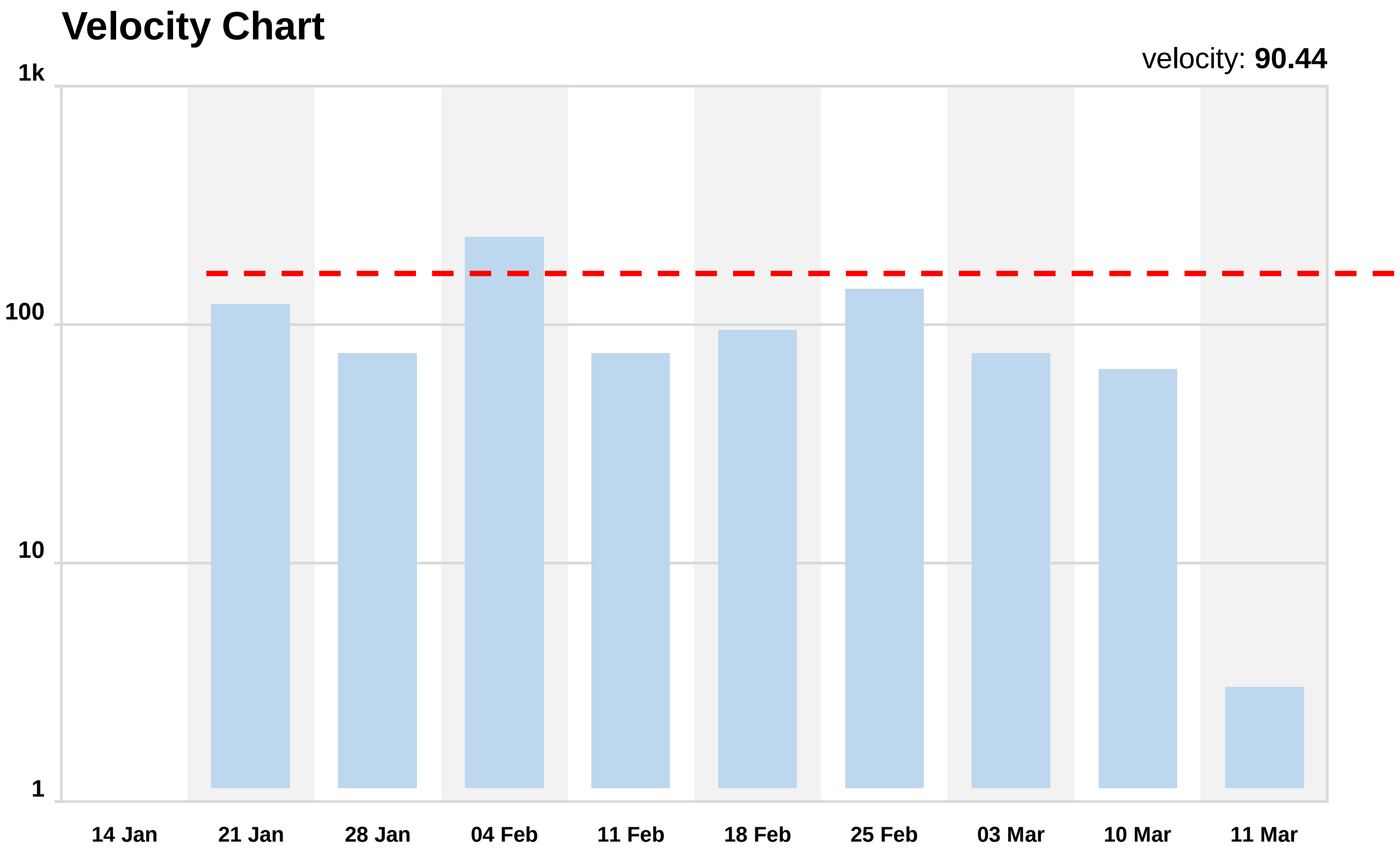

INPUT
SPRINT PLANNING
OUTPUT
Team capacity
Effort estimations
Product backlog
What we do?
How we do it?
Sprint goal
Product increment
bordio.

Metrics

Velocity

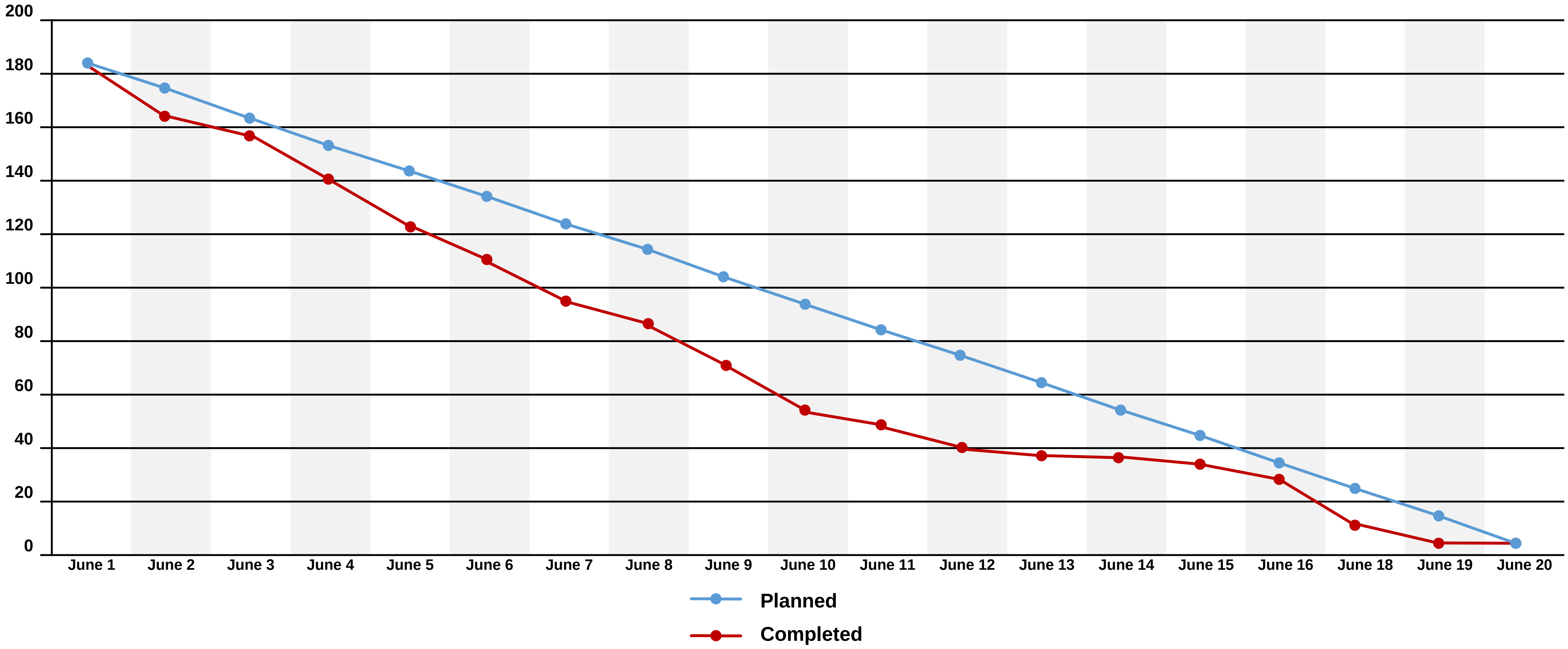
Team’s estimated rate of progress of completed work

- Calculate by estimating number of story points that can be completed during an iteration
- Then modify during subsequent iterations
- Goal: Achieve constant velocity from one iteration to the next
- *Velocity is a unique metric to a project; it can’t be used to compare the performance of teams.*



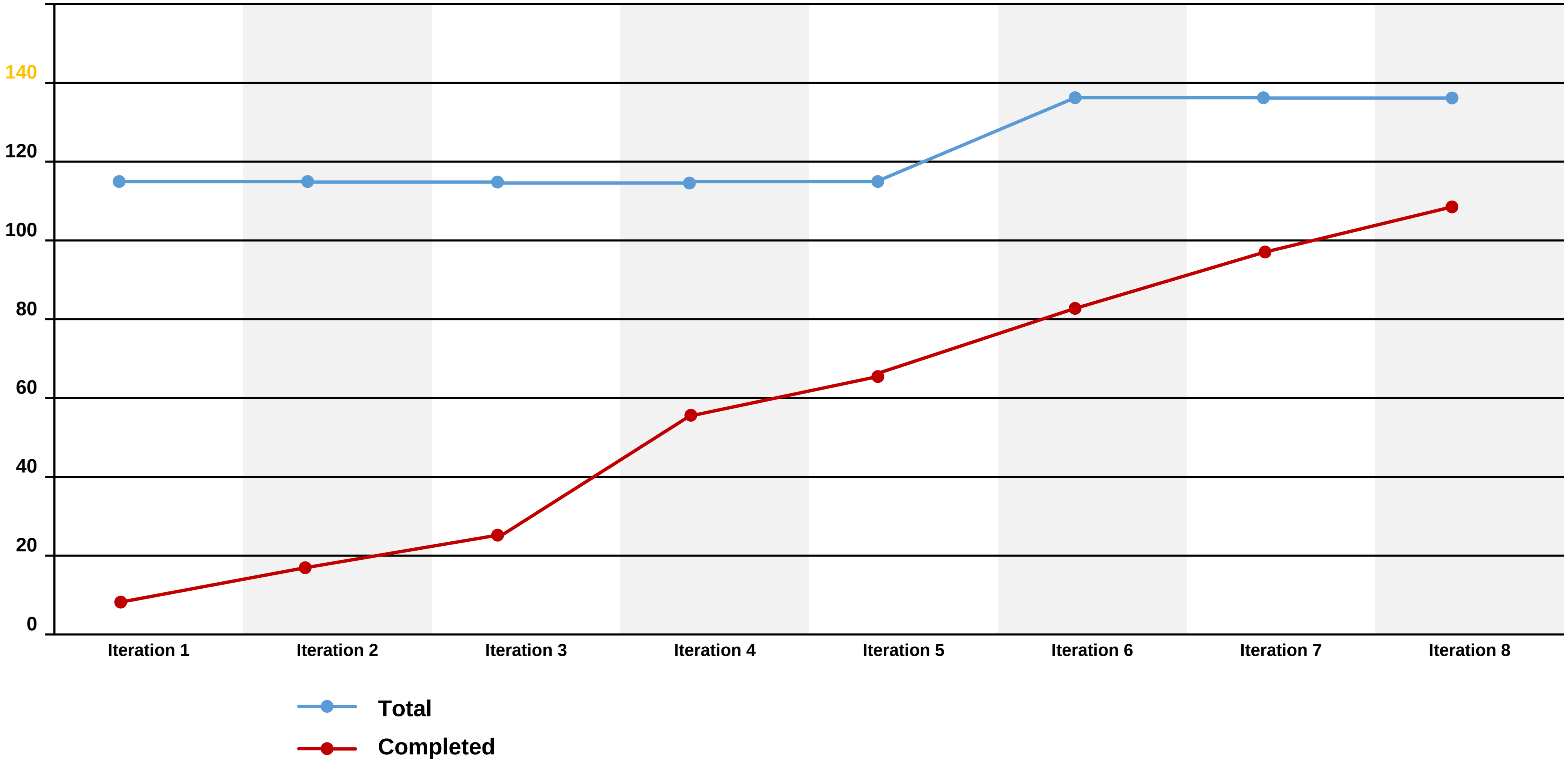
Burn Down Chart

Tracks the planned vs actual work to be completed in a sprint



Burn Up Chart

Show accumulated progress of completed work



Key Take-aways

Note down the top 3 Key Take-aways for you from this session

Thank You

Talent Academy

Empowering future leaders

An initiative of excelencia

https://www.linkedin.com/company/talent-academy-taualpha/