Agile Scrum Foundation

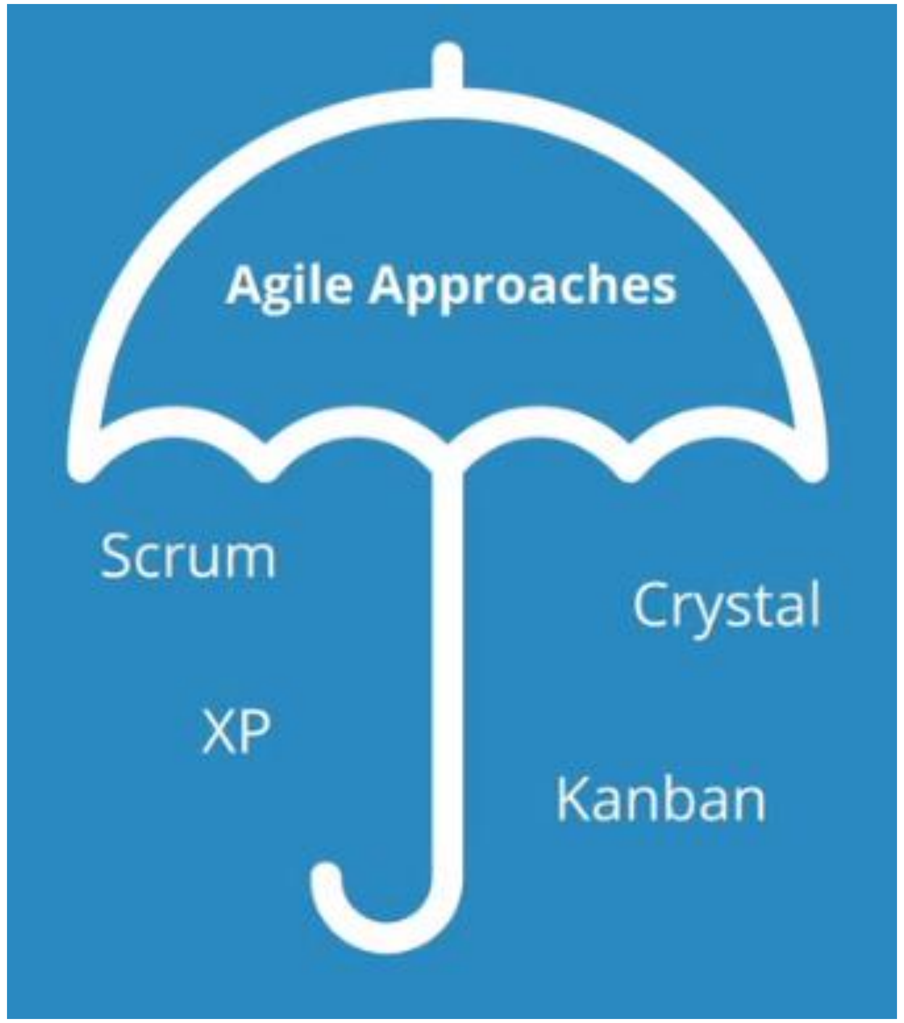
Lesson 0—Course Introduction

Objectives

- Understand the importance of the Agile Scrum Foundation Course
- Identify the course content
- Understand the ASF Certification Exam requirements

Agile Approaches

A large number of organizations worldwide, not just those involved in software development, are embracing the Agile Project Management Approach.

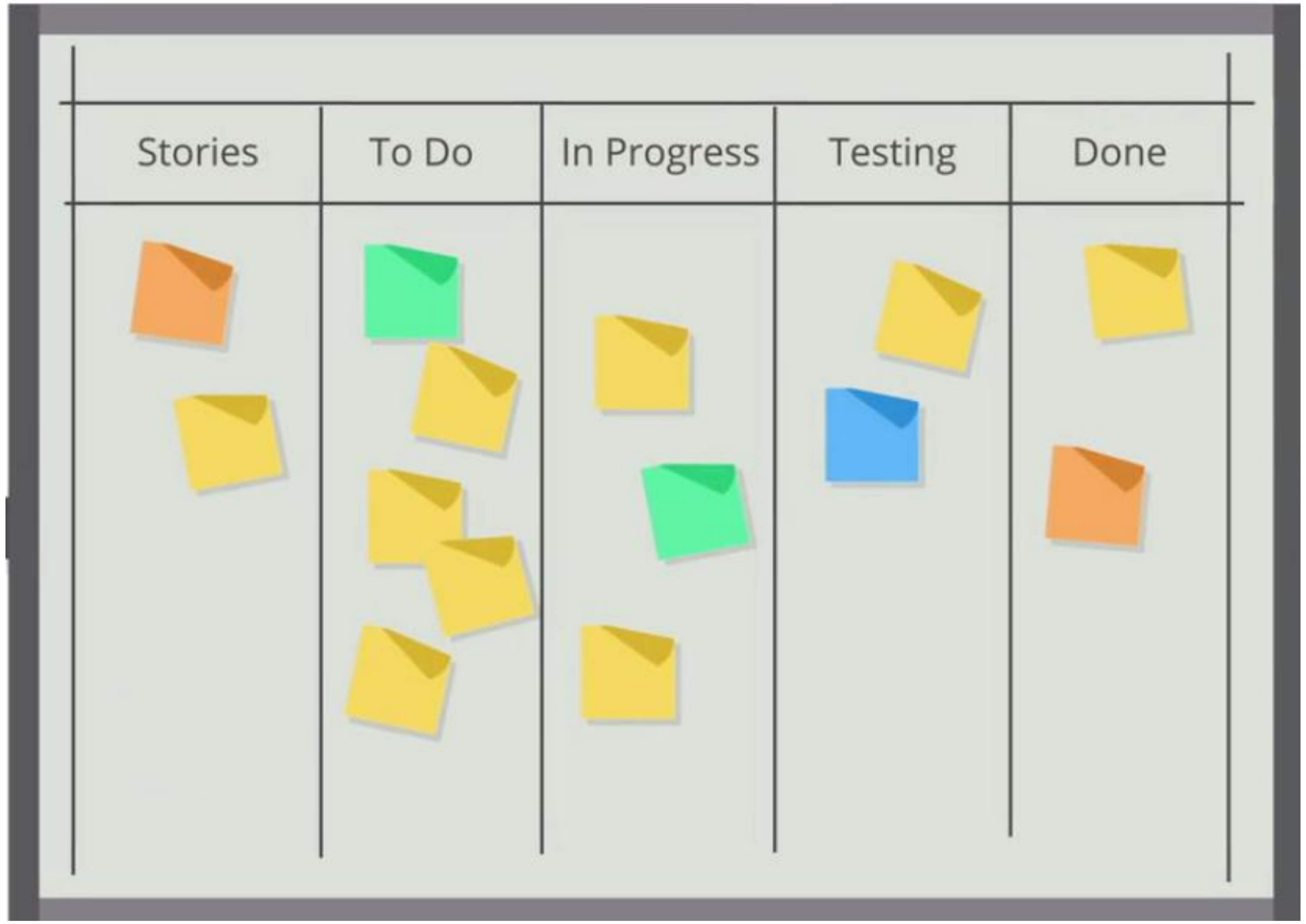


Family of Agile methods

Scrum

Of all Agile methods, Scrum is the most popular.

In fact nearly two thirds of all Agile projects use the Scrum method.



Value of ASF to Professionals

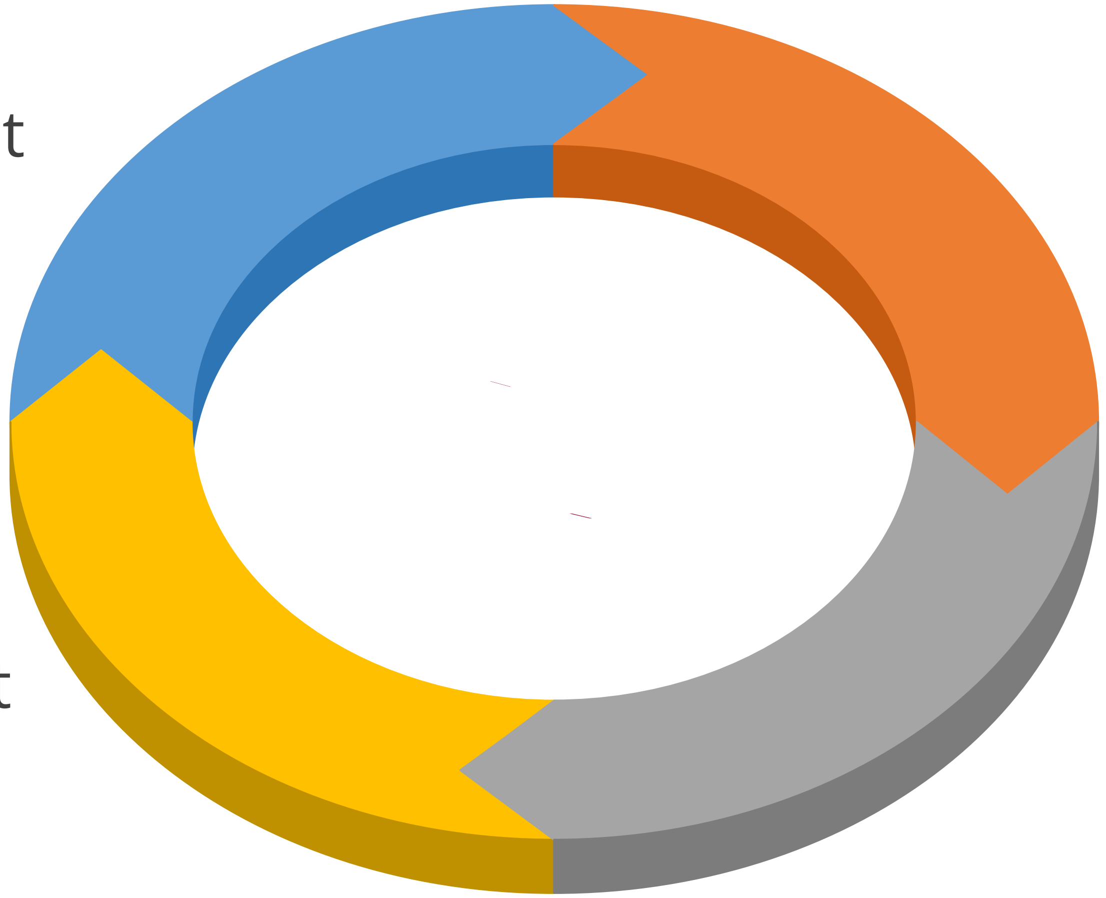
Simplilearn's EXIN Agile Scrum Foundation (ASF) Course covers the key concepts of Agile and Scrum specifically.

EXIN's Agile Scrum Foundation certification validates a professional's combined knowledge of Agile methods and Scrum practices.

Scrum practices include establishing cross-functional and self-managed teams and producing a working deliverable at the end of each iteration or Sprint.

Target Audience

The target audience for this course is professionals looking to keep their knowledge up-to-date, particularly professionals participating in projects such as:



Project Management

Software Development

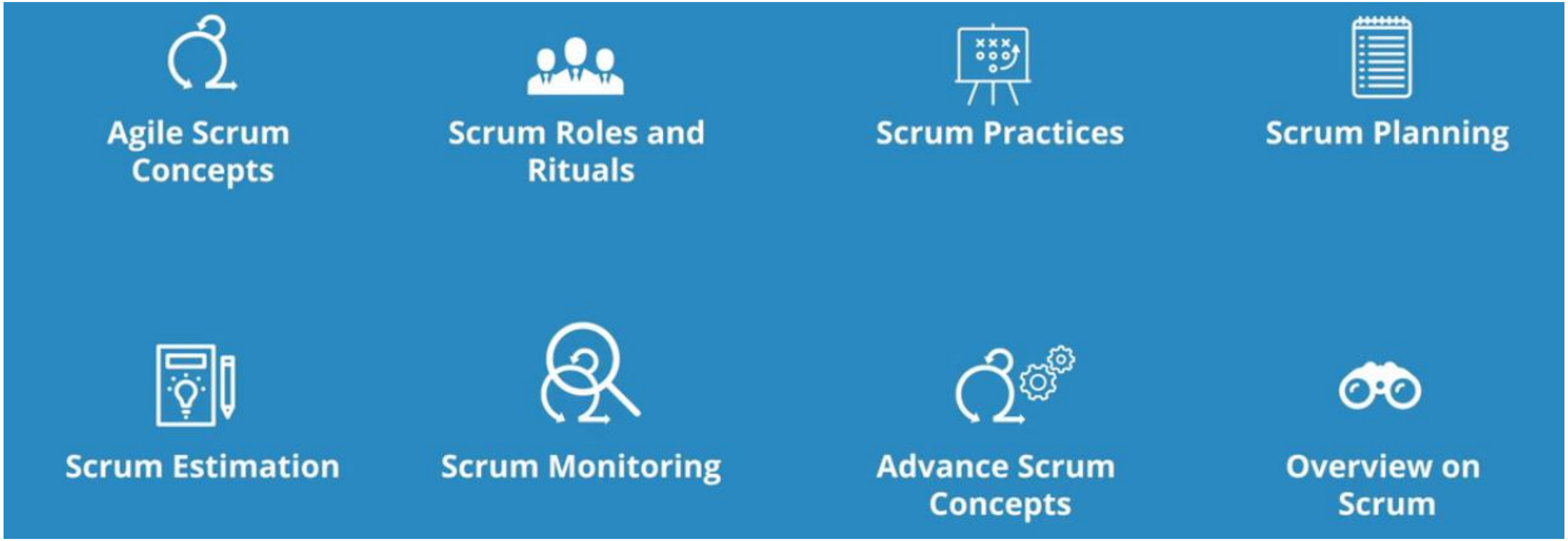
IT Service Management

Business Management

There are no course prerequisites.

About This Course

The Agile Scrum Foundation course is divided into eight lessons.



Agile Scrum Concepts

Scrum Roles and Rituals

Scrum Practices

Scrum Planning

Scrum Estimation

Scrum Monitoring

Advance Scrum Concepts

Overview on Scrum

Exam Details

Number of Question: **40**

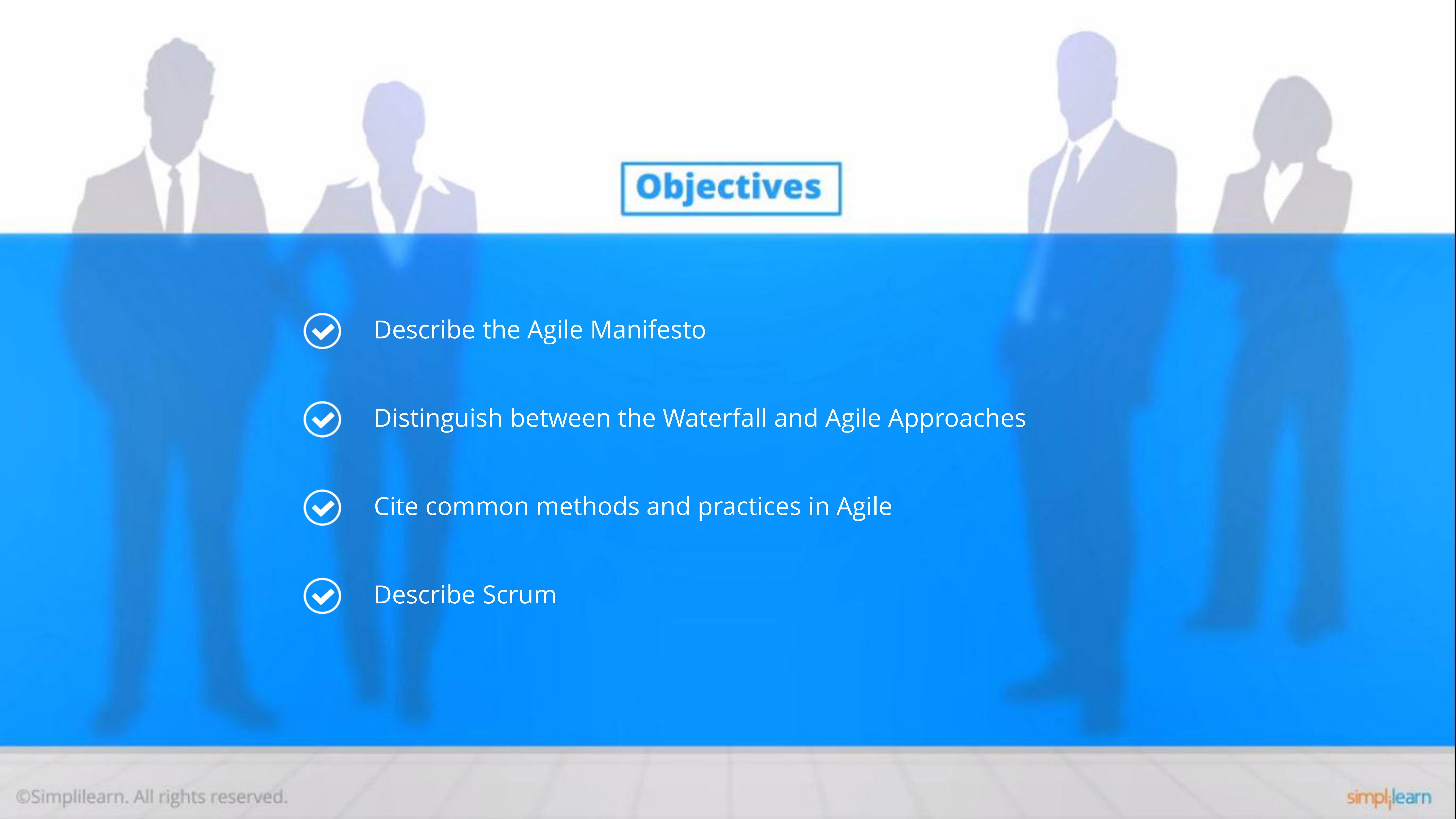
Pass Mark: **65%** (26/40 correct answers)

Open Book/Notes: **NO**

Electronic Equipment Permitted: **NO**

Agile Scrum Foundation

Lesson 1—Agile Scrum Concepts

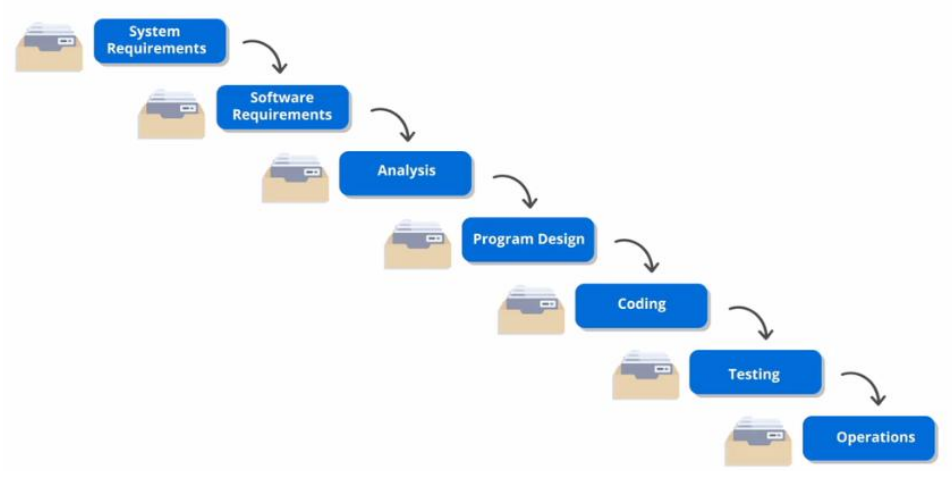


Objectives

- ✓ Describe the Agile Manifesto
- ✓ Distinguish between the Waterfall and Agile Approaches
- ✓ Cite common methods and practices in Agile
- ✓ Describe Scrum

Project Management Approaches

There are two main approaches to Project Management.



Waterfall Approach

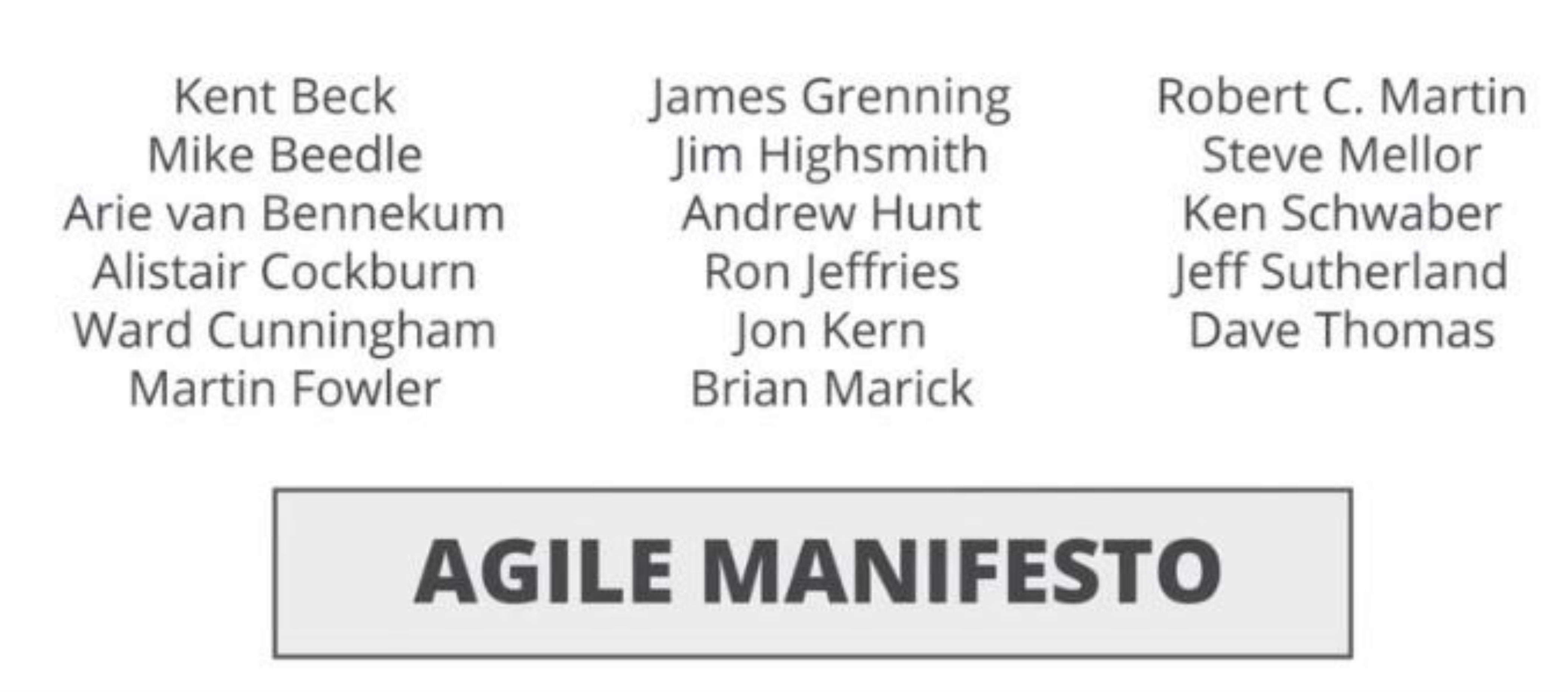
Agile Approach

Agile Approach

The Agile approach was devised in the late 1990s to address the need for an alternative to a heavyweight, documentation-driven process.

Agile Manifesto

The Agile Manifesto was written in February 2001 in Park City, Utah, by seventeen leading software developers.



Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

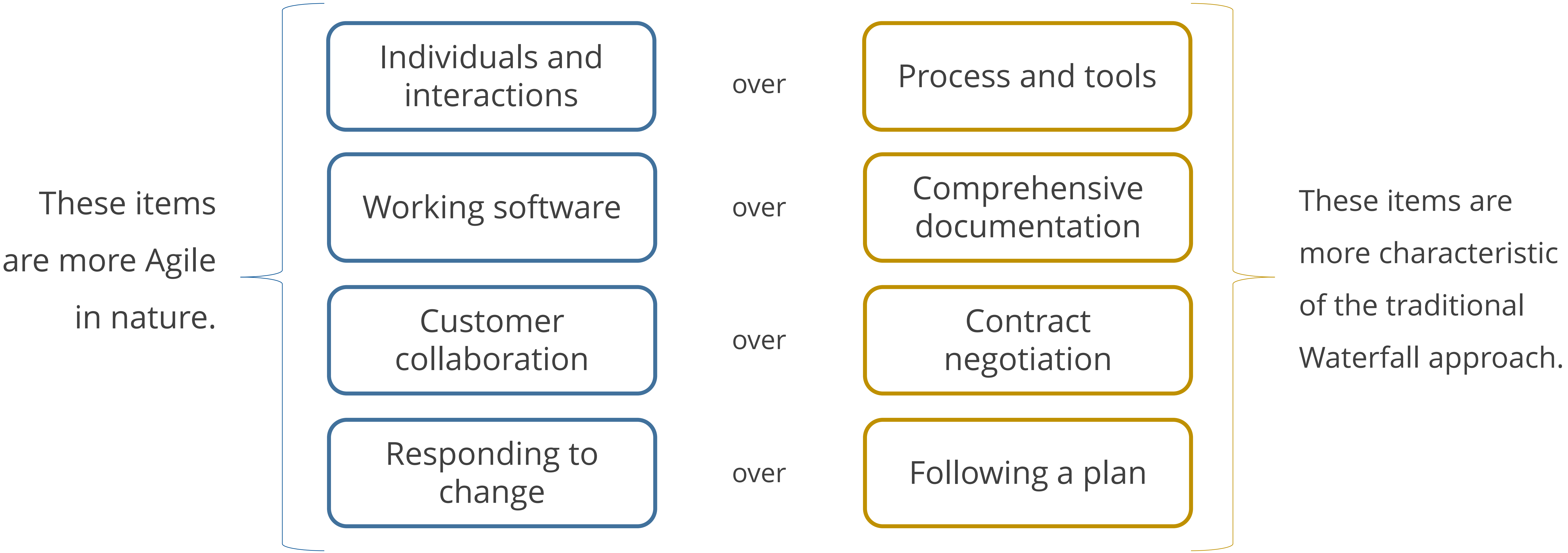
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

AGILE MANIFESTO

Agile Manifesto for Software Development

Statement of value



These items are more Agile in nature.

Individuals and interactions over Process and tools

Working software over Comprehensive documentation

Customer collaboration over Contract negotiation

Responding to change over Following a plan

These items are more characteristic of the traditional Waterfall approach.

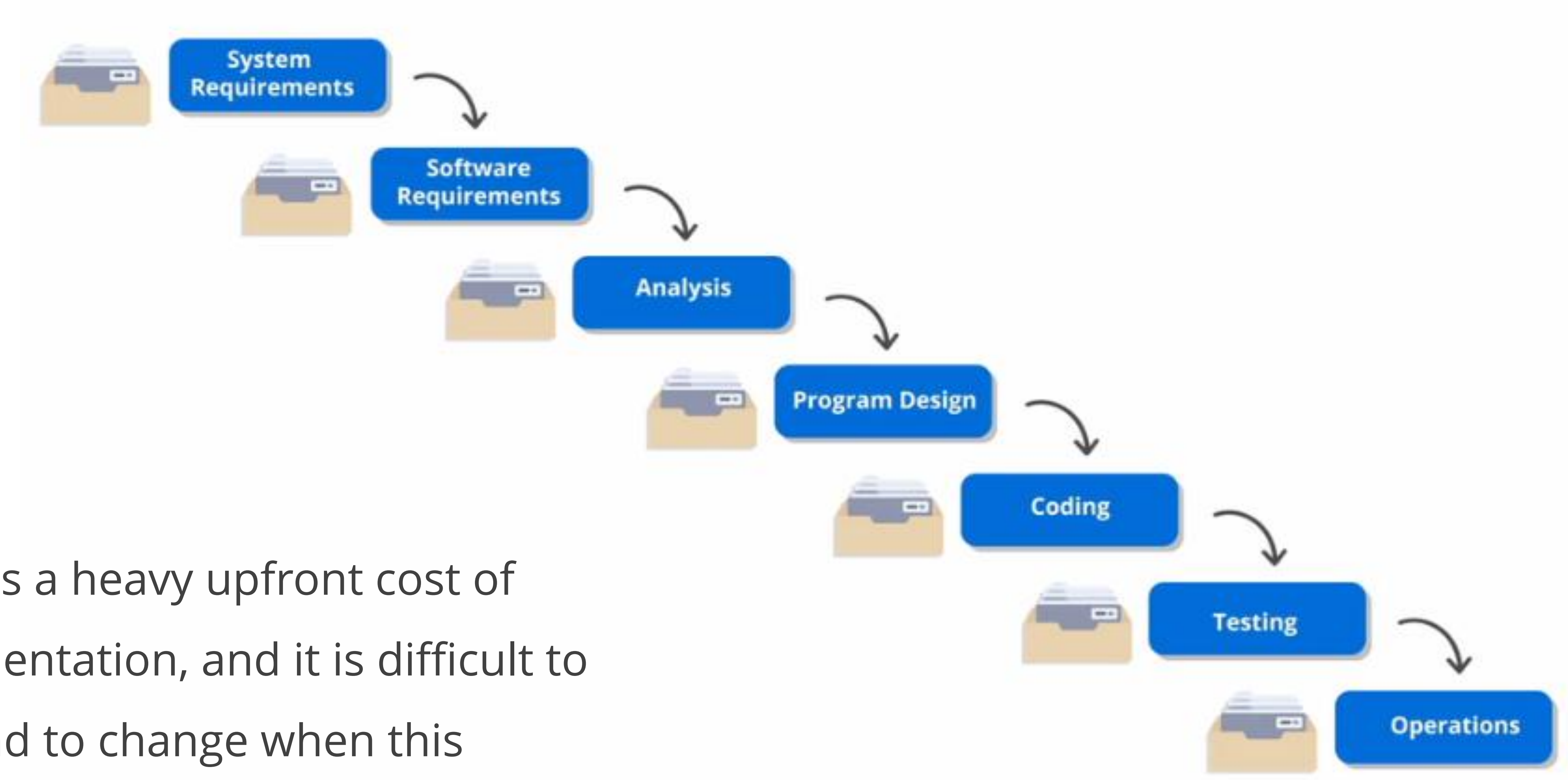
Note: It is recommended that you memorize these four lines before taking the Agile Scrum Foundation certification exam.

Agile Manifesto for Software Development (Contd.)

Individuals and interactions	over	Process and tools	Individuals and their interactions deliver better results when the emphasis is NOT on the processes and the tools.
Working software	over	Comprehensive documentation	Emphasis on working software can deliver real progress or value.
Customer collaboration	over	Contract negotiation	Organizations need to be accommodating rather than following detailed definitions listed in contracts.
Responding to change	over	Following a plan	Customer suggestions must be incorporated to deliver the value the customers want.

The Waterfall Approach

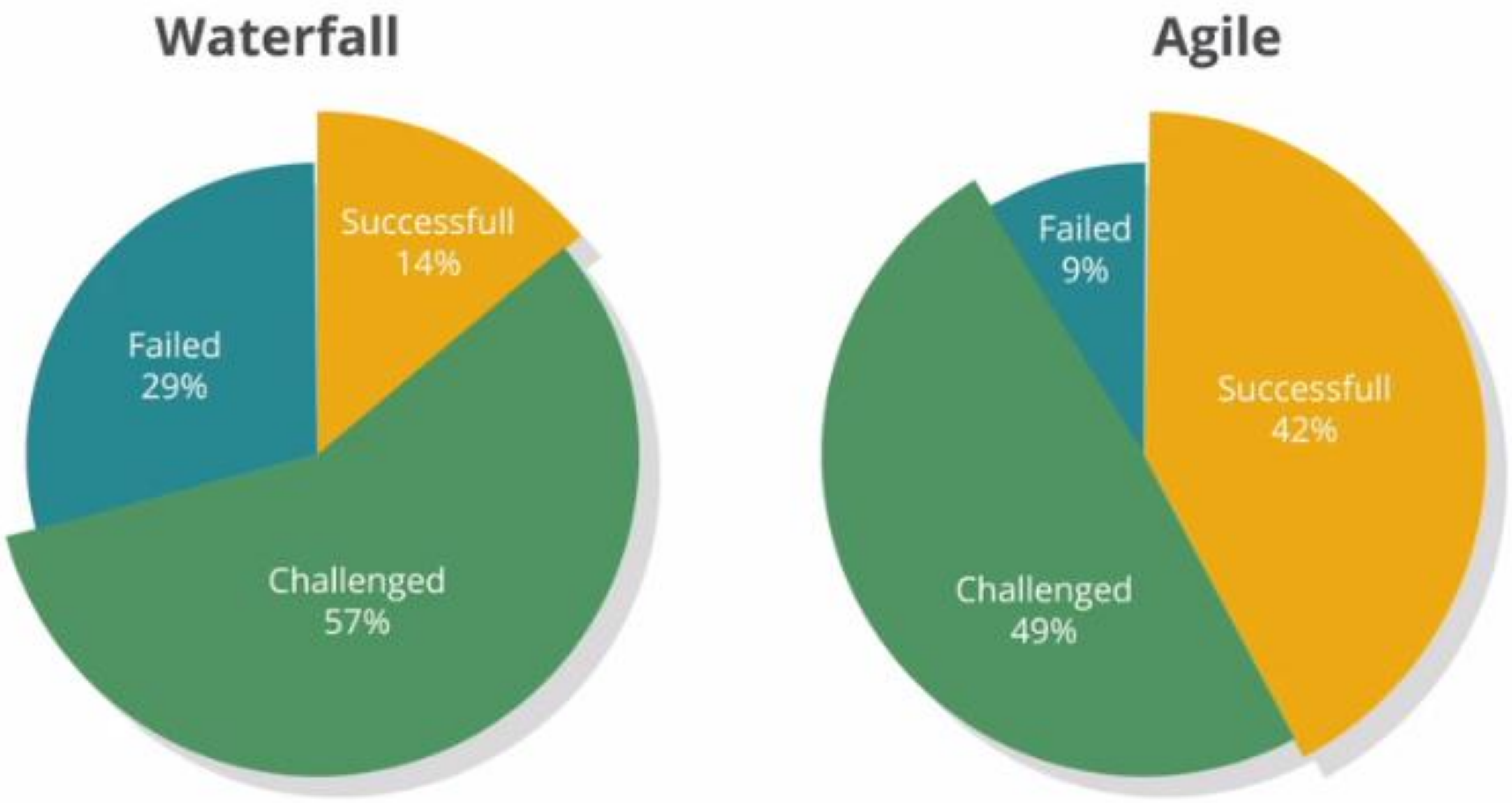
Traditionally used to plan and deliver software development projects.



There is a heavy upfront cost of documentation, and it is difficult to respond to change when this approach is used.

Waterfall vs. Agile Approach

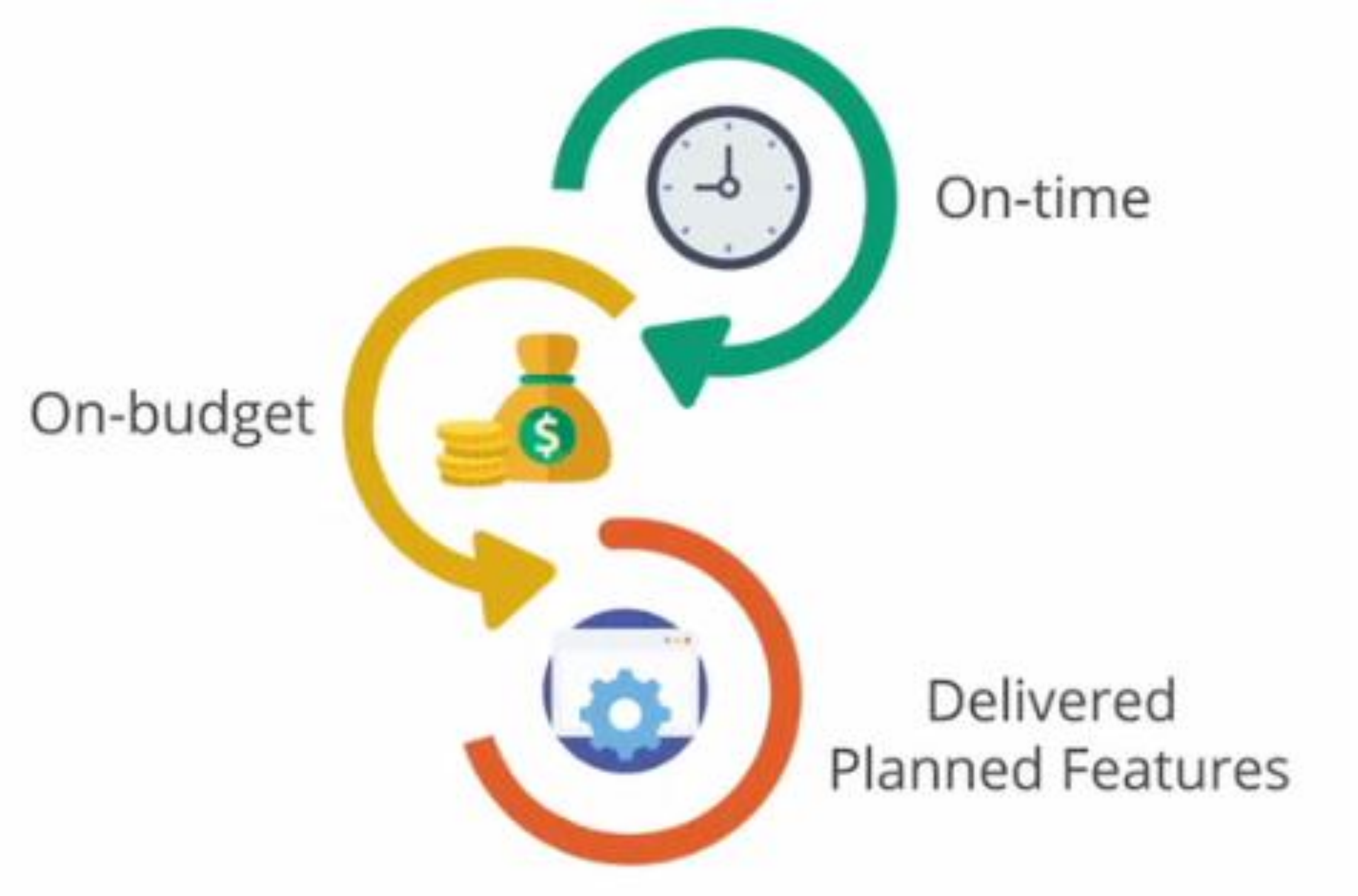
The Standish Group Study—CHAOS Manifesto showed that software development projects succeed three times more often when the agile approach is used.



CHAOS Project Database – 2002 to 2010

Waterfall vs. Agile Approach (Contd.)

The Standish Group Study—CHAOS Manifesto defined project success using the following parameters:



Waterfall vs. Agile Approach (Contd.)

The Standish Group Study—CHAOS Manifesto

THE STANDISH GROUP

“The Agile process is the universal remedy for software development project failure. Software applications developed through the Agile process have three times the success rate of the traditional Waterfall method and a much lower percentage of time and cost over-runs.”

Agile—Key Features

Focus on People

- The focus is on professionals and how to optimally use their skills

Customer Involvement

- The customer determines the scope of the project, prioritizes the work, and reviews the product

Multidisciplinary Teams

- The team includes different specialties that deliver the working software to the customer

Trust

- It is the basic characteristic each team member should have

Incremental

- Working software is delivered to the customer in short increments

Time Boxes

- The predetermined amount of time is never to be exceeded by the team

Agile is also used to address the quality of the software being delivered to the customer.

Metrics for Agile Projects

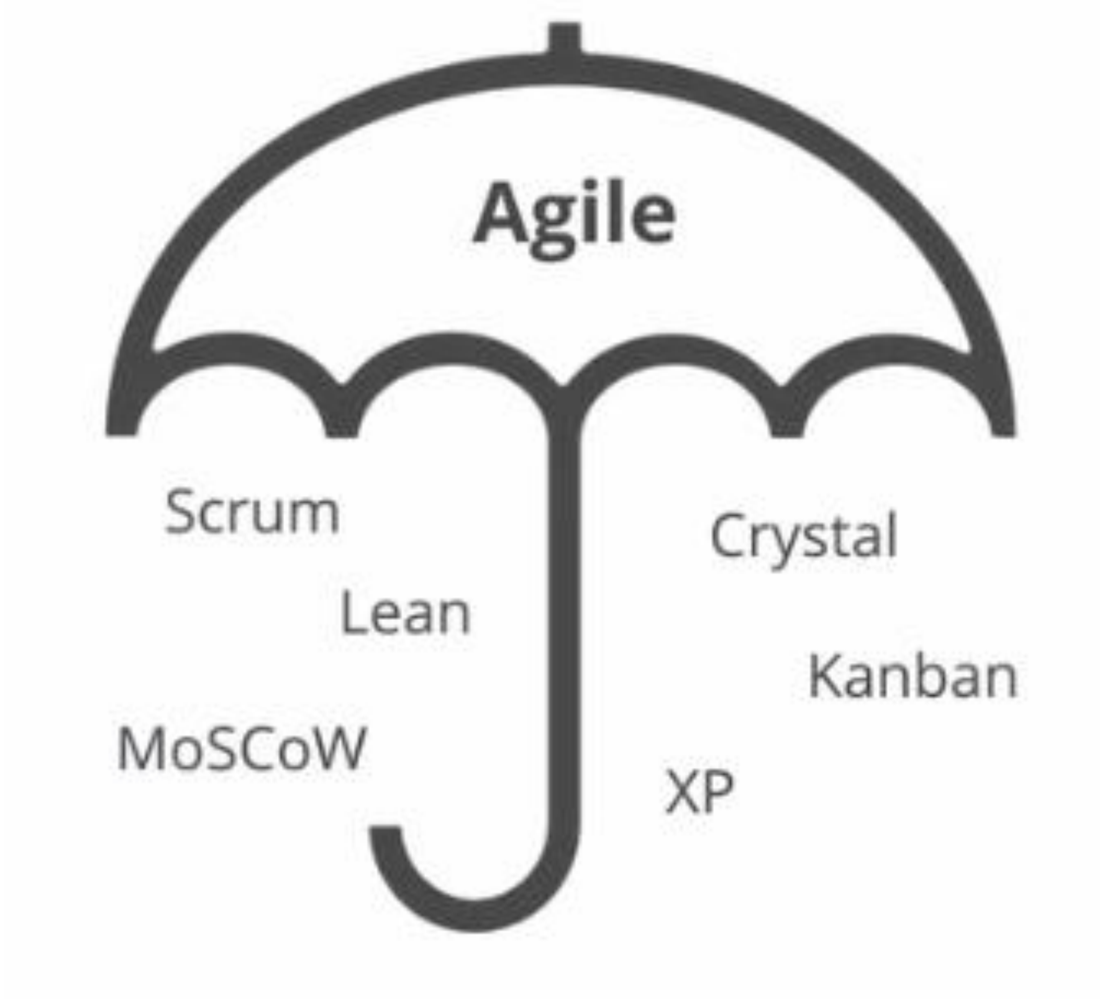
The Agile team determines how things will be measured during the course of the project.

One of the key metrics used in every Agile project is **Escaped Defects**.

This term refers to defects that were not found by, or ones that escaped from, the quality assurance team but were found by the customer.

Agile Methods

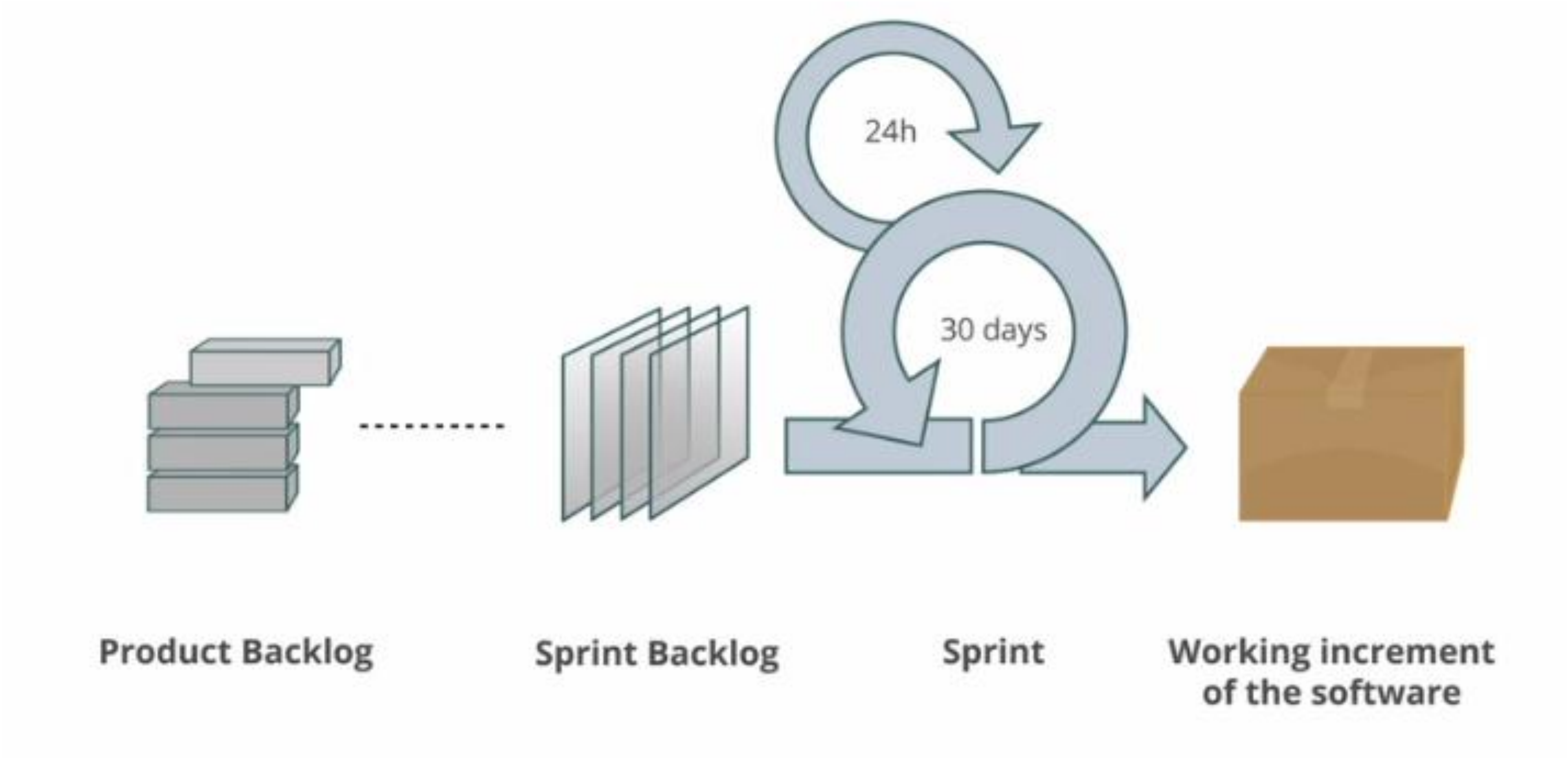
Agile is actually a family of methods. Some of the most common methods are:



All Agile methods use the Agile Manifesto as their fundamental guide.

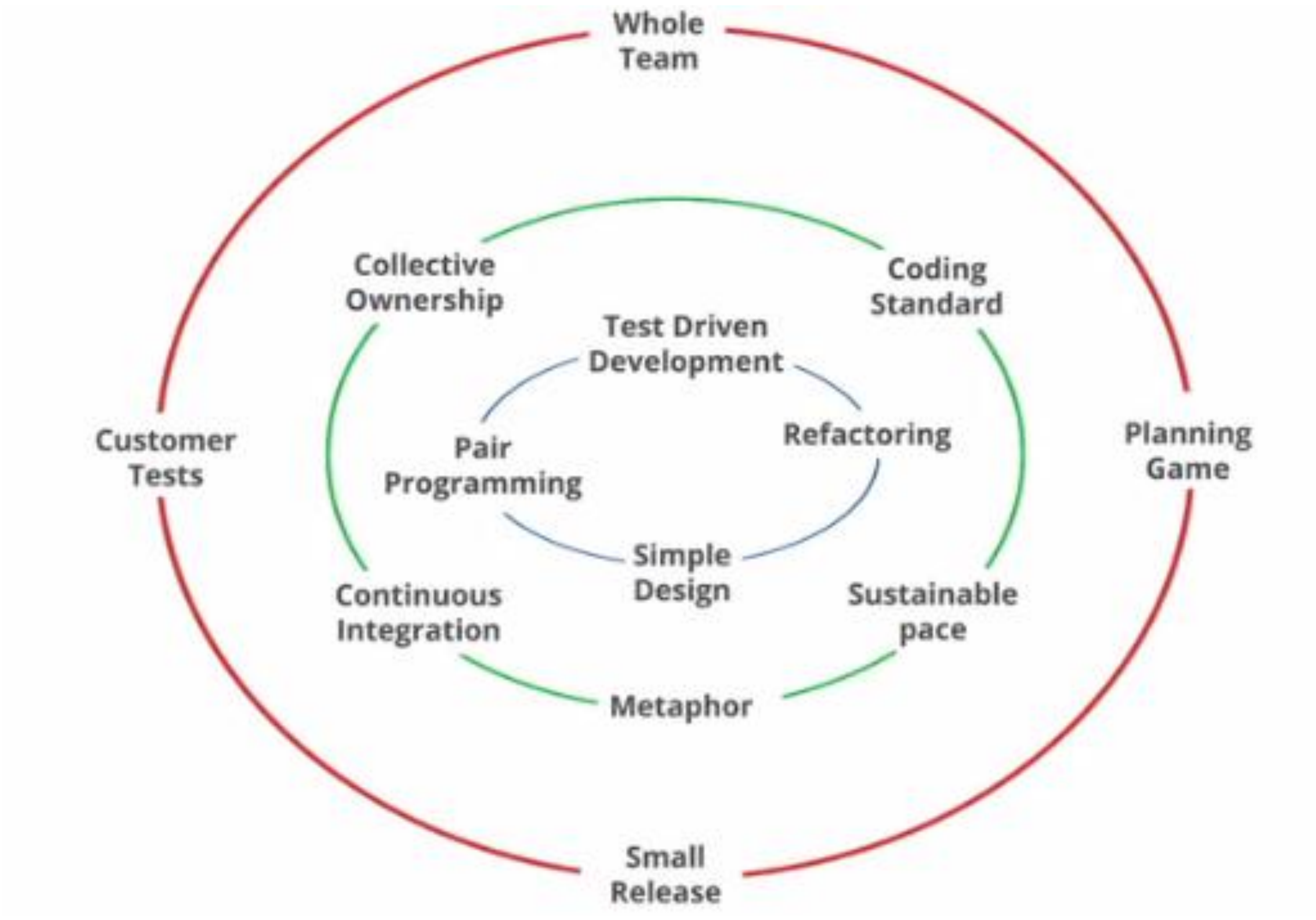
Agile Methods—Scrum

Scrum is a framework that uses tools and practices developed for other Agile methods.



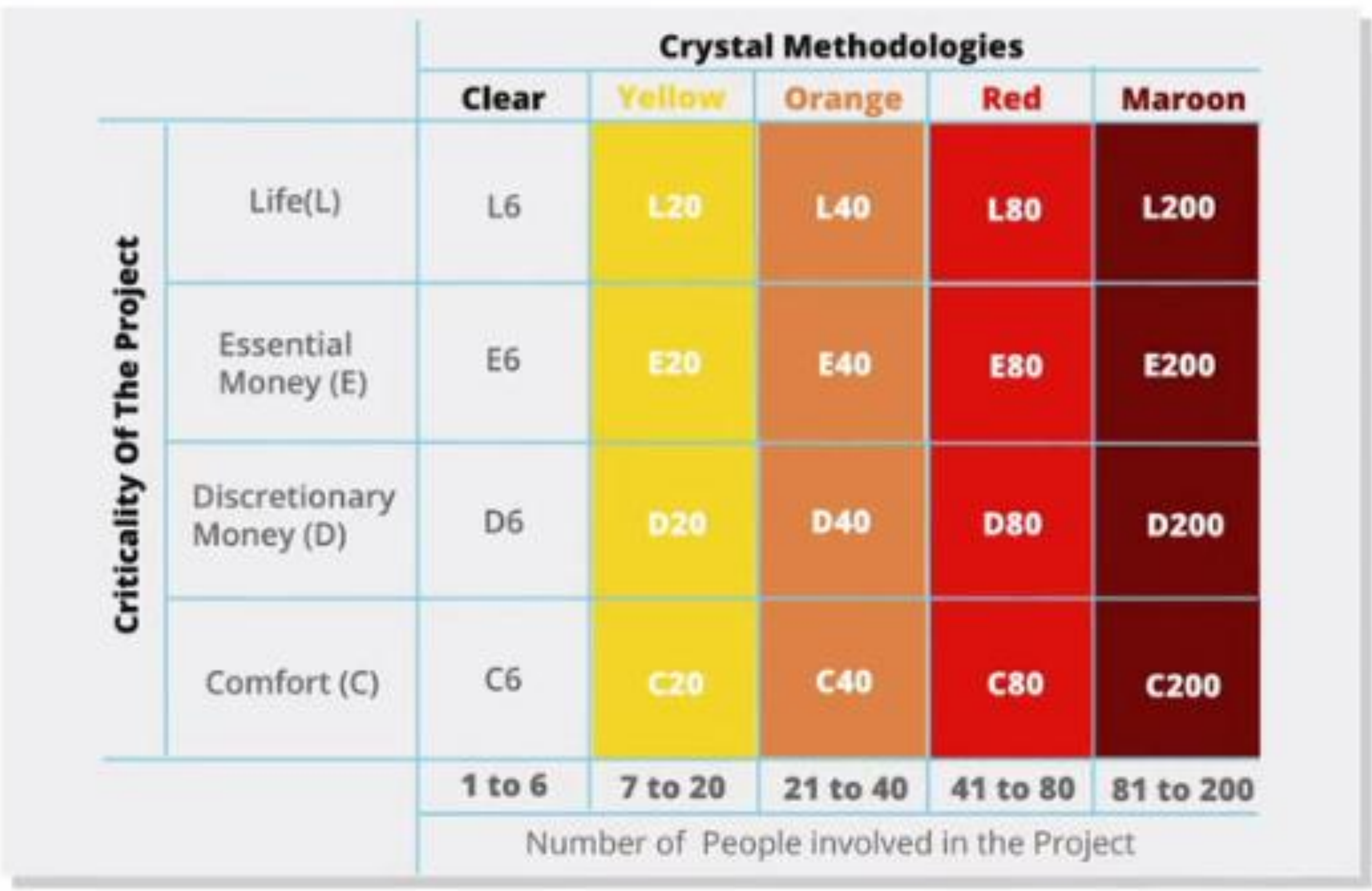
Agile Methods—eXtreme Programming (XP)

XP is the second most common Agile method; it developed the practices of continuous integration and pair programming.



Agile Methods—Crystal

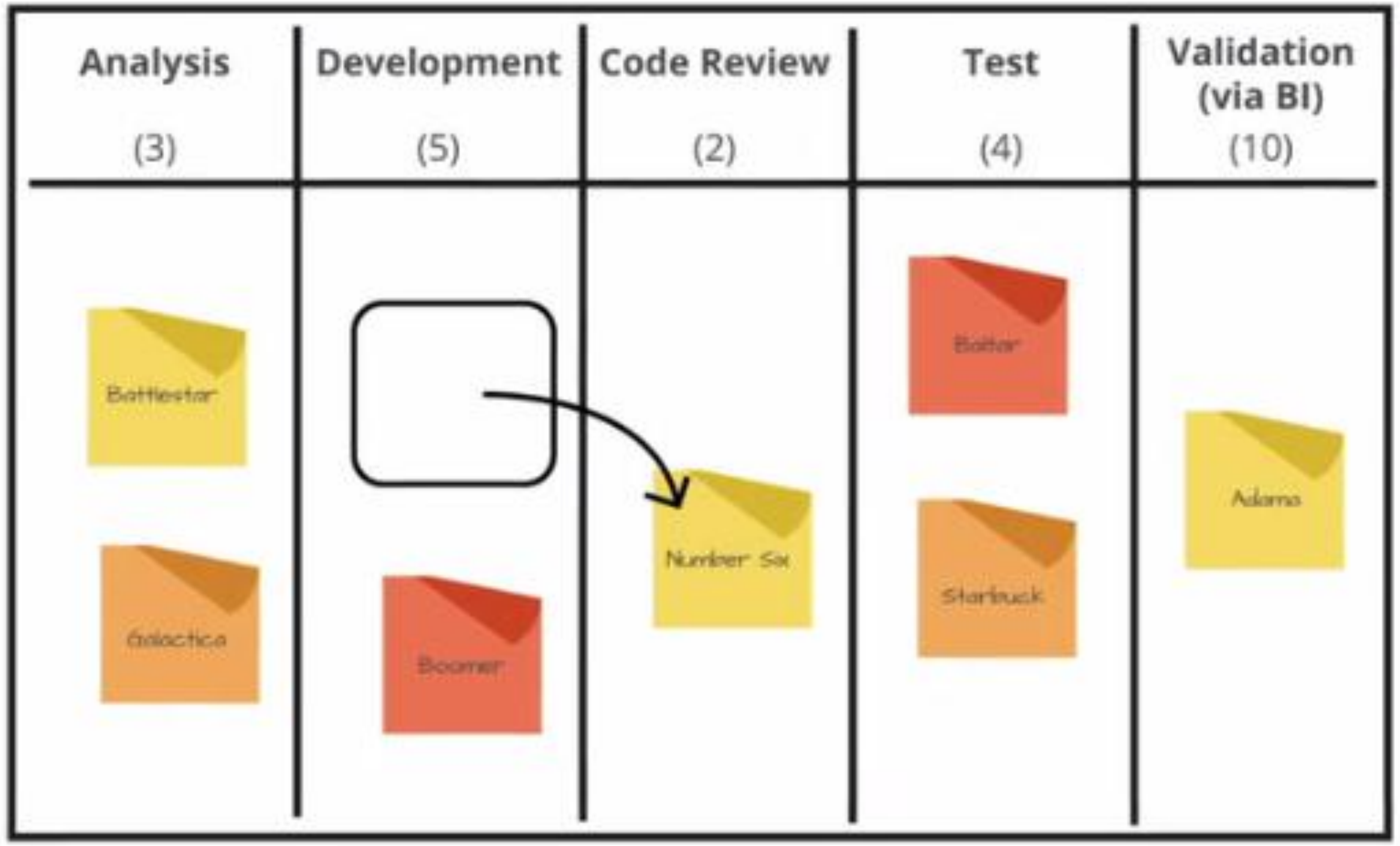
Crystal pioneered “Osmotic Communication”—Indirect information transfer through overhearing conversations around you.



		Crystal Methodologies				
		Clear	Yellow	Orange	Red	Maroon
Criticality Of The Project	Life(L)	L6	L20	L40	L80	L200
	Essential Money (E)	E6	E20	E40	E80	E200
	Discretionary Money (D)	D6	D20	D40	D80	D200
	Comfort (C)	C6	C20	C40	C80	C200
		1 to 6	7 to 20	21 to 40	41 to 80	81 to 200
		Number of People involved in the Project				

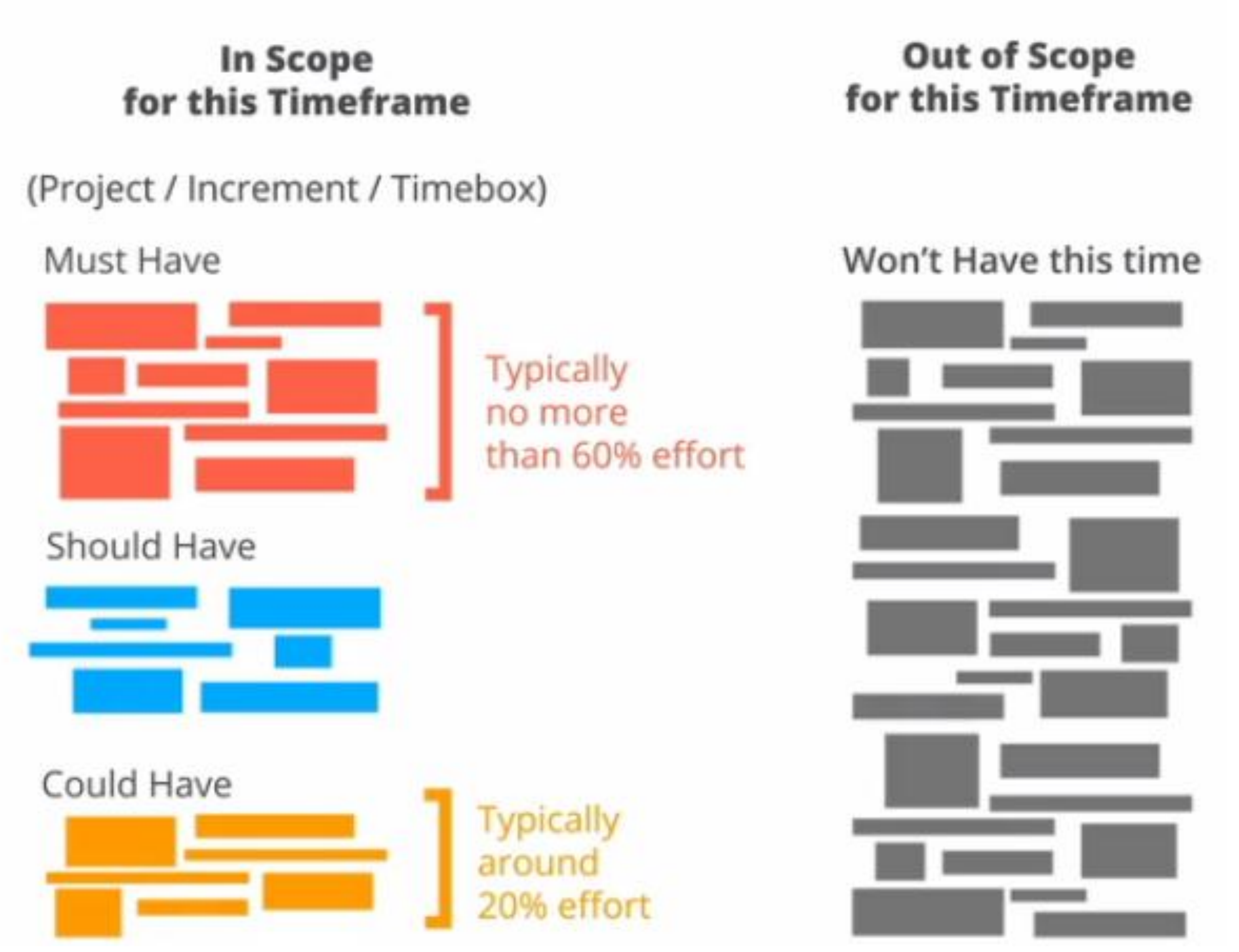
Agile Methods—Kanban

Kanban or task board is a work and workflow visualization tool that enables you to optimize the workflow.



Agile Methods—MoSCoW

MoSCoW is a prioritization tool used in Scrum; it was developed as part of Dynamic Systems Development Model (DSTM).

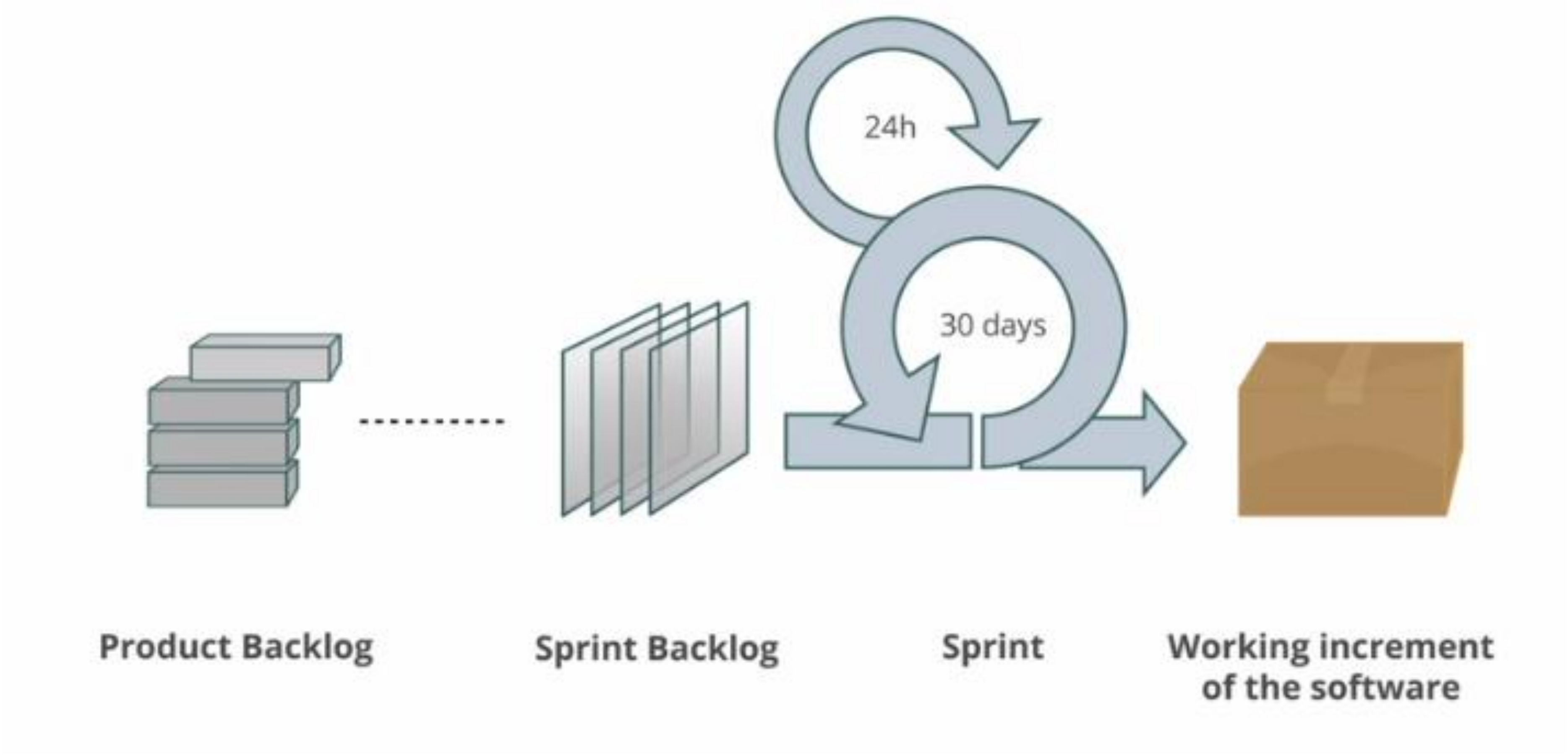


MoSCoW stands for Must, Should, Could, and Wont.

Scrum—Definition

Scrum is a lightweight framework designed to manage complex product development.

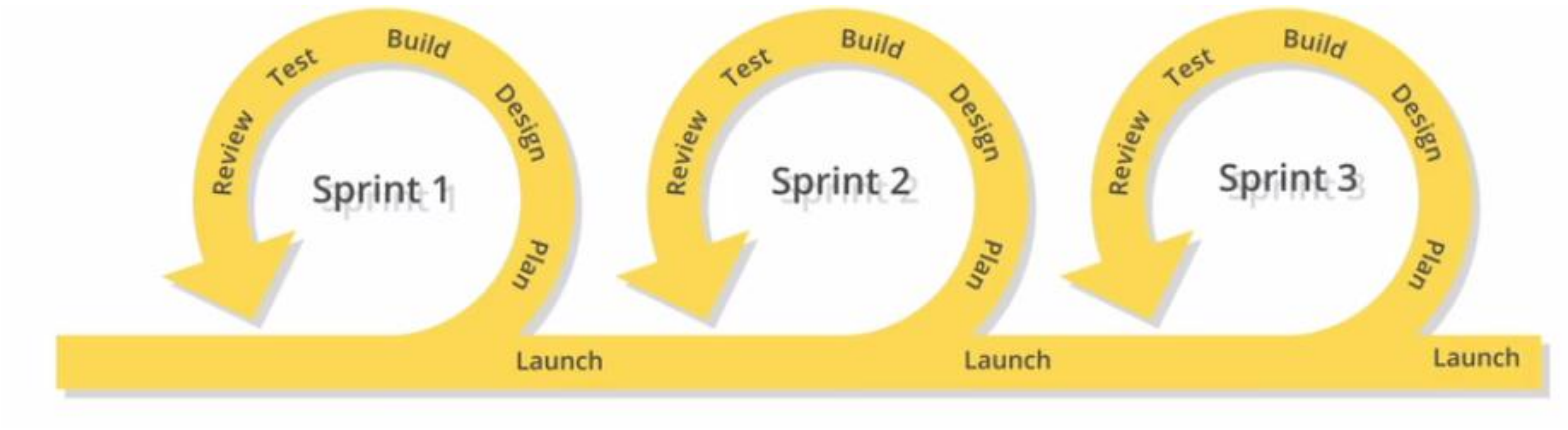
It is a framework within which various processes, techniques, and practices are employed.



It promotes developing products of the highest possible value in an iterative and incremental way.

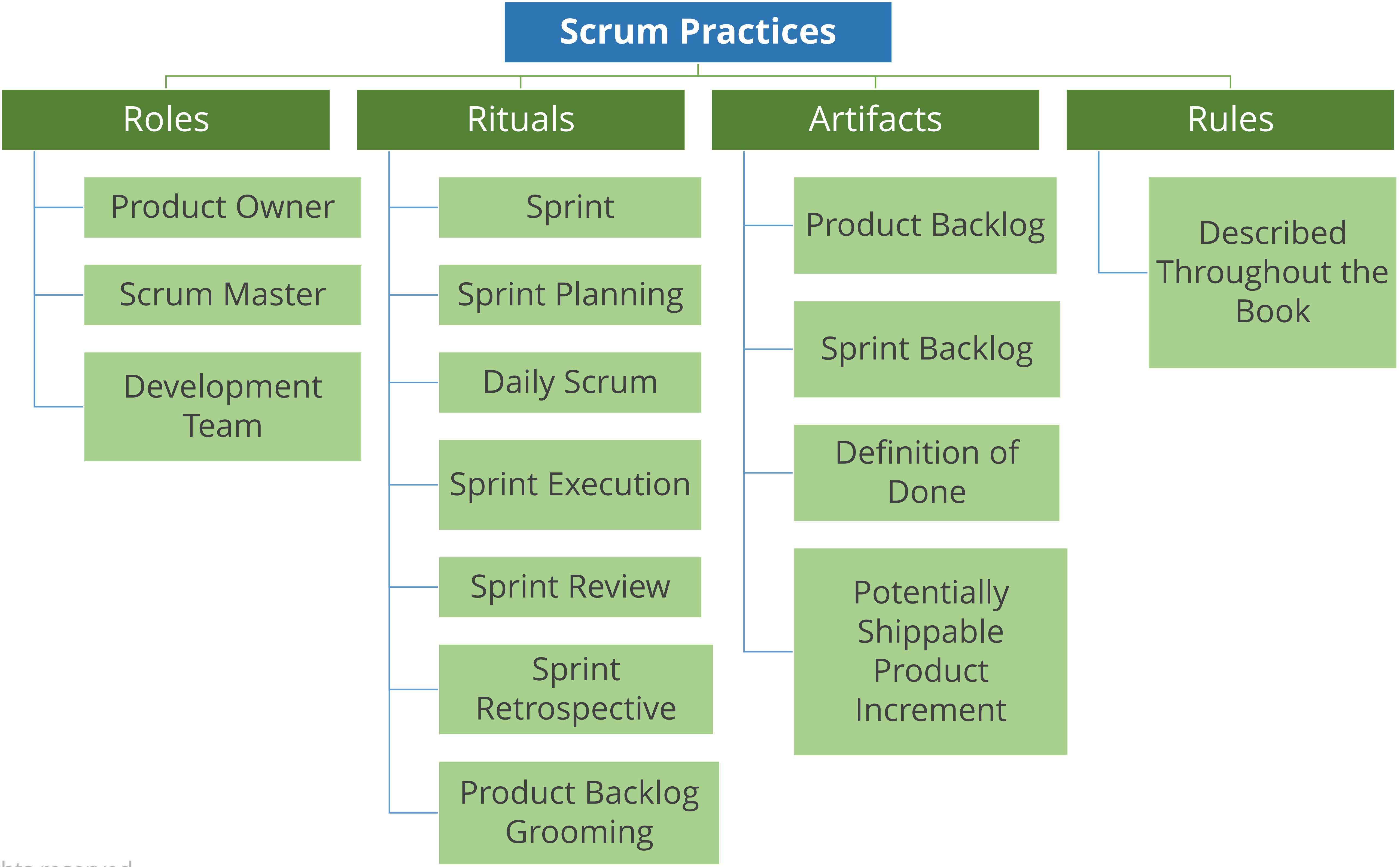
Scrum—Definition (Contd.)

In scrum, the iterations that deliver working software to the customer are called **sprints**.



In an iteration, each sprint results in potentially deliverable software.

Scrum—Overview


Scrum Practices

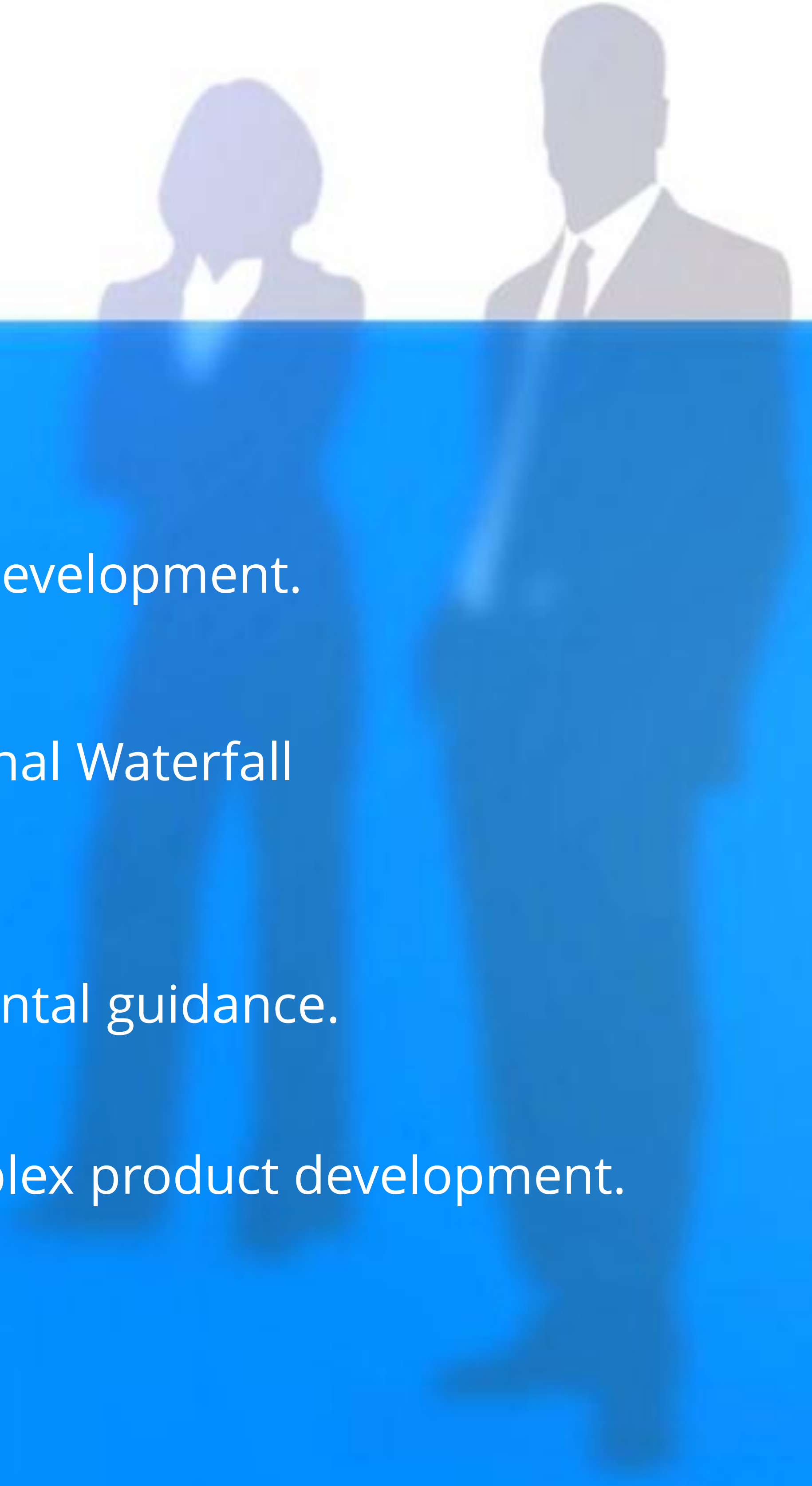
Roles
- Product Owner
- Scrum Master
- Development Team

Rituals
- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Execution
- Sprint Review
- Sprint Retrospective
- Product Backlog Grooming

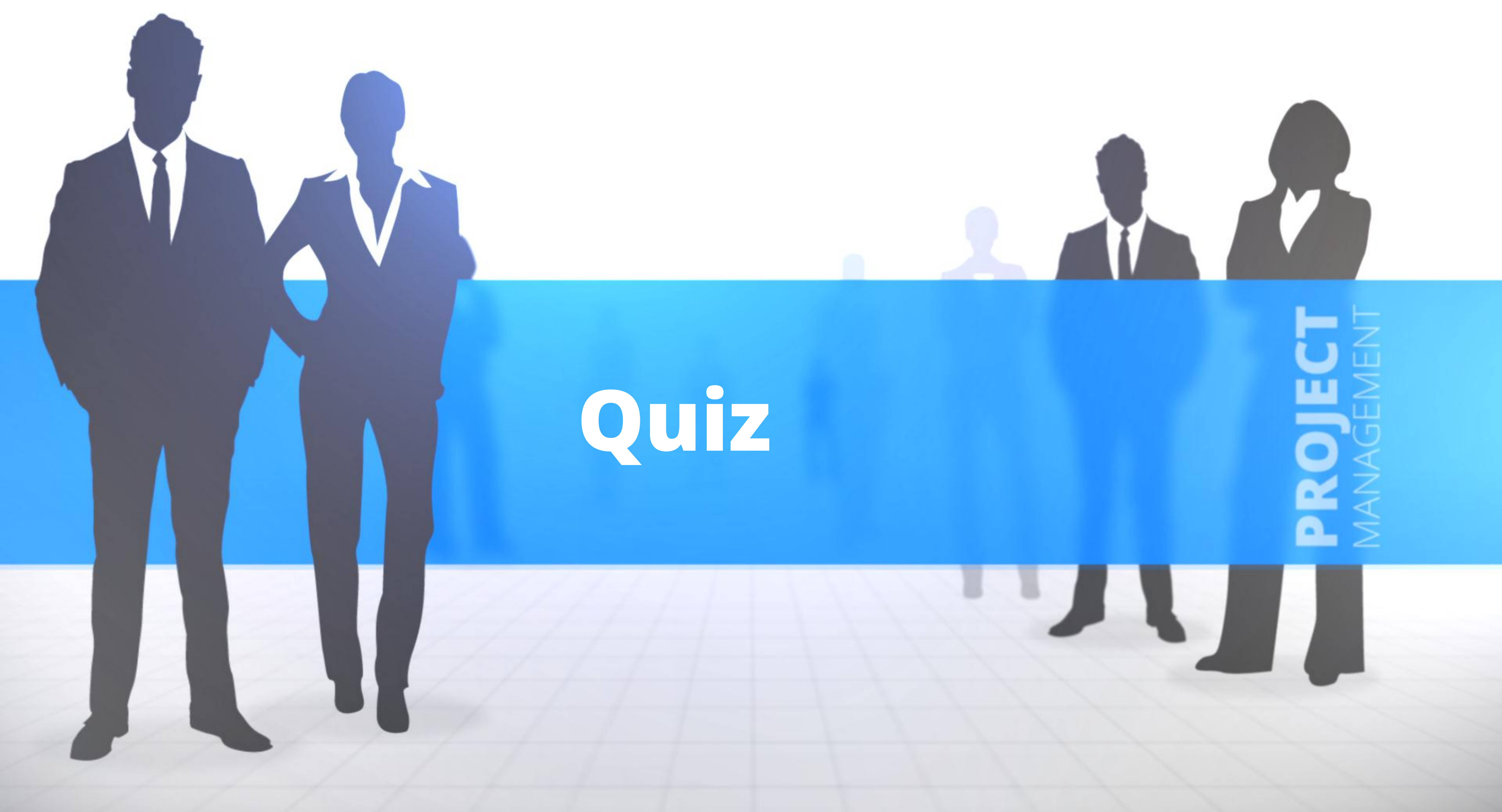
Artifacts
- Product Backlog
- Sprint Backlog
- Definition of Done
- Potentially Shippable Product Increment

Rules
- Described Throughout the Book

Topics Covered



- Memorize the Agile Manifesto—Statement of Value and its development.
- The two approaches of Project Management are the traditional Waterfall approach and Agile approach.
- The Agile methods use the Agile Manifesto as their fundamental guidance.
- Scrum is a lightweight framework designed to manage complex product development.



Quiz

QUIZ

1

Which of the following are assertions of the Agile Manifesto? (Choose Two.)

a. An Agile team must value following a plan over responding to change

b. An Agile team must value working software over comprehensive documentation

c. An Agile team must value customer collaboration over contract negotiation

d. An Agile team must value processes and tools over individuals and interactions

QUIZ

2

Which of the following would be considered most valuable by an Agile team?

a. Following a plan

b. Processes and tools

c. Comprehensive documentation

d. Working software

QUIZ

3

Which methodology is known for frequent releases and osmotic communication?

a. XP

b. Scrum

c. Crystal

d. Kanban

QUIZ

4

Which of the following statements describes Scrum?

a. A sequential design process in which progress is seen as flowing steadily downwards

b. A methodology in which you practice only continuous integration and pair programming

c. A basic workflow visualization tool that enables you to optimize your work

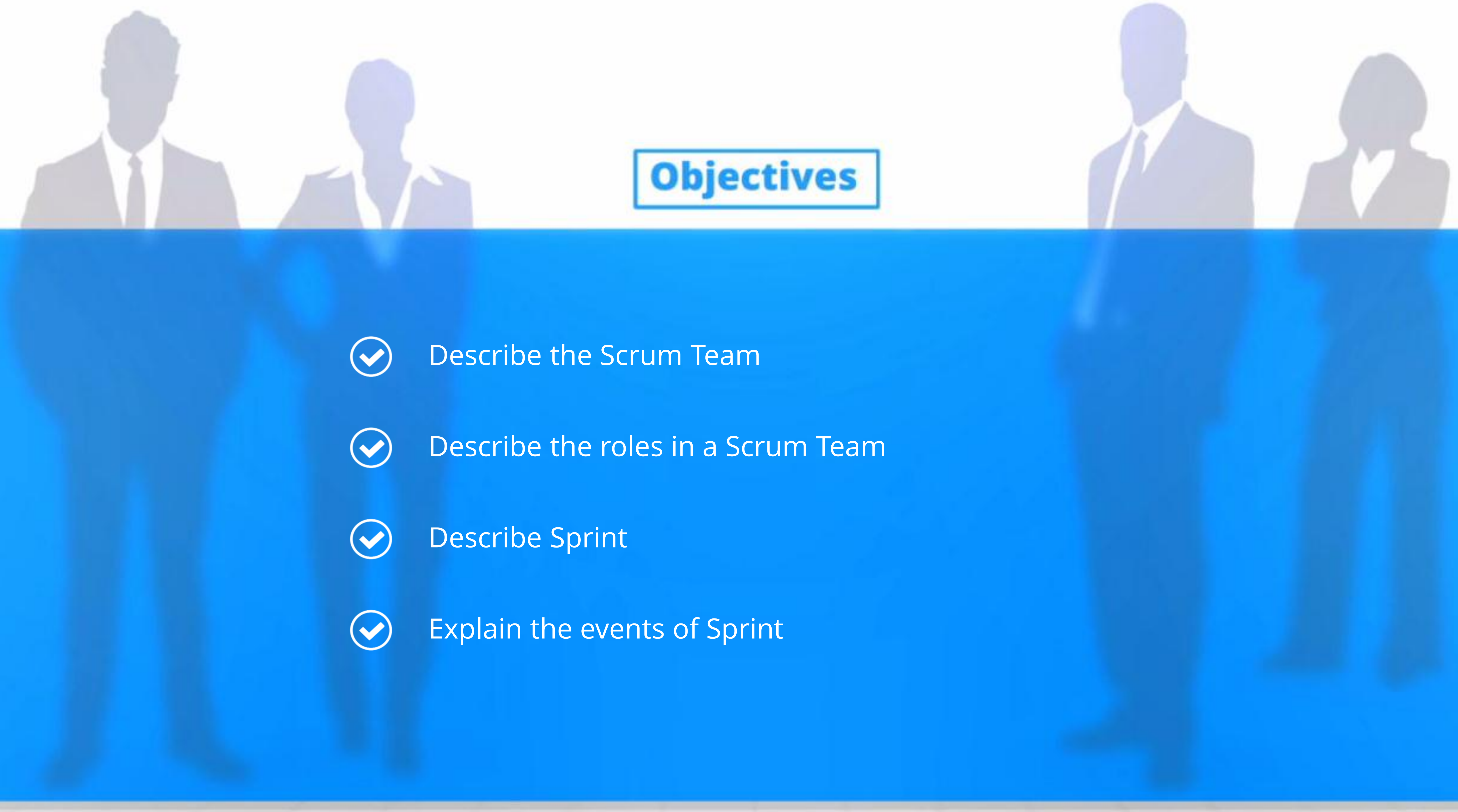
d. A lightweight framework designed to develop and manage complex products in an iterative and incremental way

This concludes “Agile Scrum Concepts.”

The Next Lesson is “Roles and Rituals.”

Agile Scrum Foundation

Lesson 2—Roles and Rituals



Objectives

- ✓ Describe the Scrum Team
- ✓ Describe the roles in a Scrum Team
- ✓ Describe Sprint
- ✓ Explain the events of Sprint

Scrum Roles

The three roles of Scrum are:

Product Owner

Is responsible for project success by defining a project’s vision, requirements, and priorities

Development Team

Comprises 6 ± 3 people with a mix of roles and skills; it is self-organizing and determines the best way to meet the goals of the Product owner

Scrum Master

Is a facilitator and servant leader who assists both the Product Owner and the Development Team to be successful in their respective roles

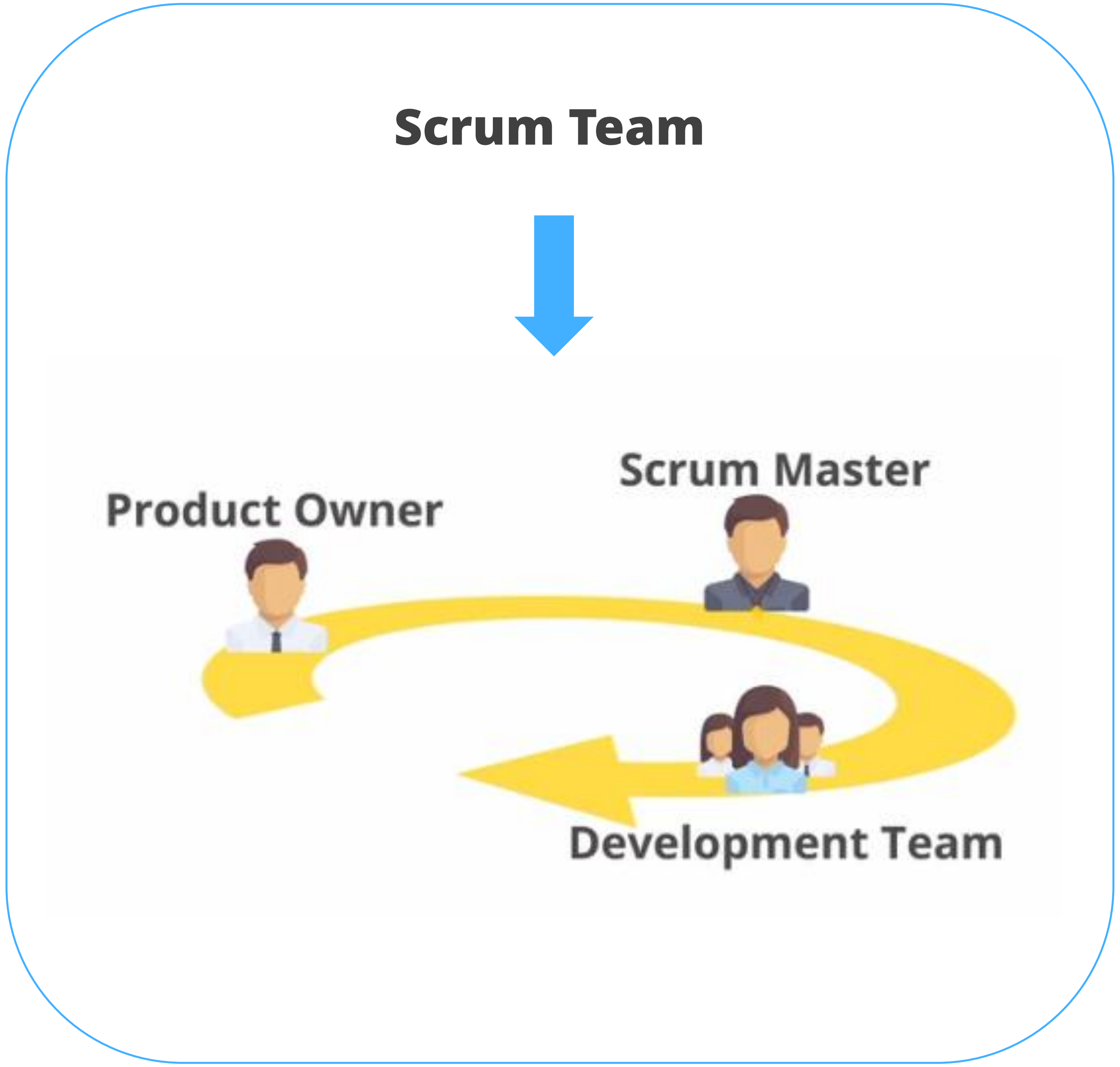
Scrum Team

Scrum Teams are self-organizing and cross-functional.

- Self-organizing teams choose the best way to accomplish their work.

Scrum Teams deliver project results in an iterative and incremental manner.

- Incremental deliveries ensure that a potentially useful, shippable version of the product is available.



Scrum Team—Product Owner

The Product Owner is responsible for:

- Maximizing the value of the product and the work completed by the Development Team
- Creating and managing the product backlog, which includes:
    - Creating and clearly communicating product backlog items
    - Prioritizing the product backlog items in order to optimize the value of the work being done by the Development Team
- Defining the scope of the project

Product Owner

Scrum Team—Scrum Master

The Scrum Master :

- Is responsible for ensuring that Scrum is understood and enacted
  - This is done by ensuring that the Scrum Team adheres to Scrum theory, practices, and rules.
- Is a servant leader and helps those who are not part of the Scrum Team to interact effectively with the Scrum Team
- Is a facilitator, mentor, and coach and is never a manager

Scrum Master

Scrum Team—Scrum Master (Contd.)

The Scrum Master serves the Product Owner by:

- Finding techniques for effective product backlog management
- Facilitating the creation of product backlog items
- Facilitating Scrum events
- Maximizing Return On Investment (ROI) for the product

The Scrum Master serves the Development Team by:

- Coaching and mentoring the team in self-organization and cross-functionality
- Removing obstacles to the team’s progress
- Facilitating Scrum events
- Coaching the team on the best Scrum practices

Scrum Team—Development Team

The Development Team:

- Comprises people who deliver a potentially releasable piece of the product at the end of each Sprint
- Is structured and empowered to organize and manage its work
- The size of the team should be 6 ± 3.
    - Having more than nine members in a team generates too much complexity and need for coordination. In other words, team members are no longer Agile.

Development Team

Scrum Team—Development Team (Contd.)

Characteristics of a Development Team are:

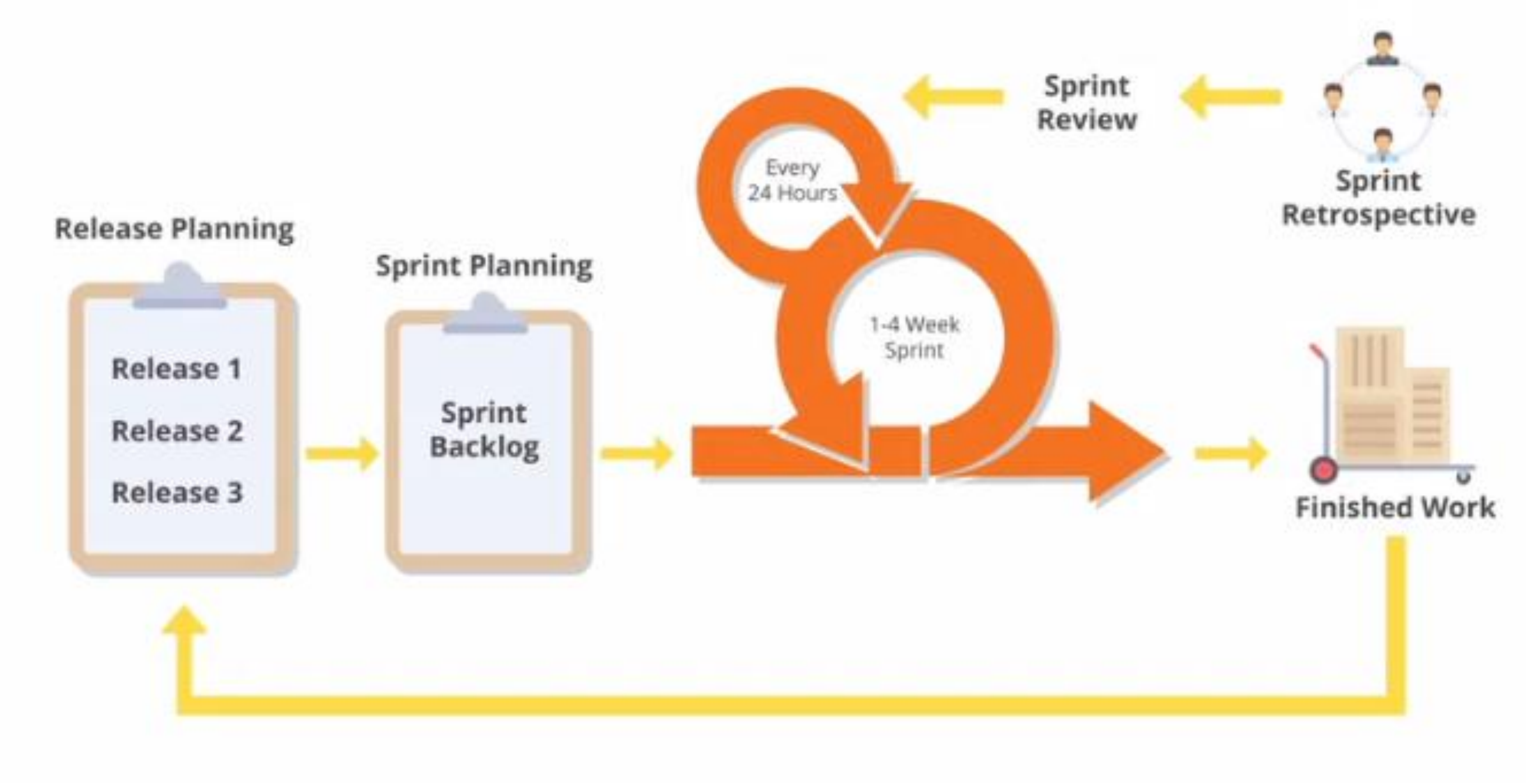
- It is self-organizing. No one tells the team how to do its work.
- It is cross-functional and possesses all of the skills necessary for the project.
- There are no sub-teams. There may be team members with specialized skills.
- It takes accountability.

Development Team

Scrum Events / Scrum Rituals

The main Scrum Events include:

- Sprint
- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

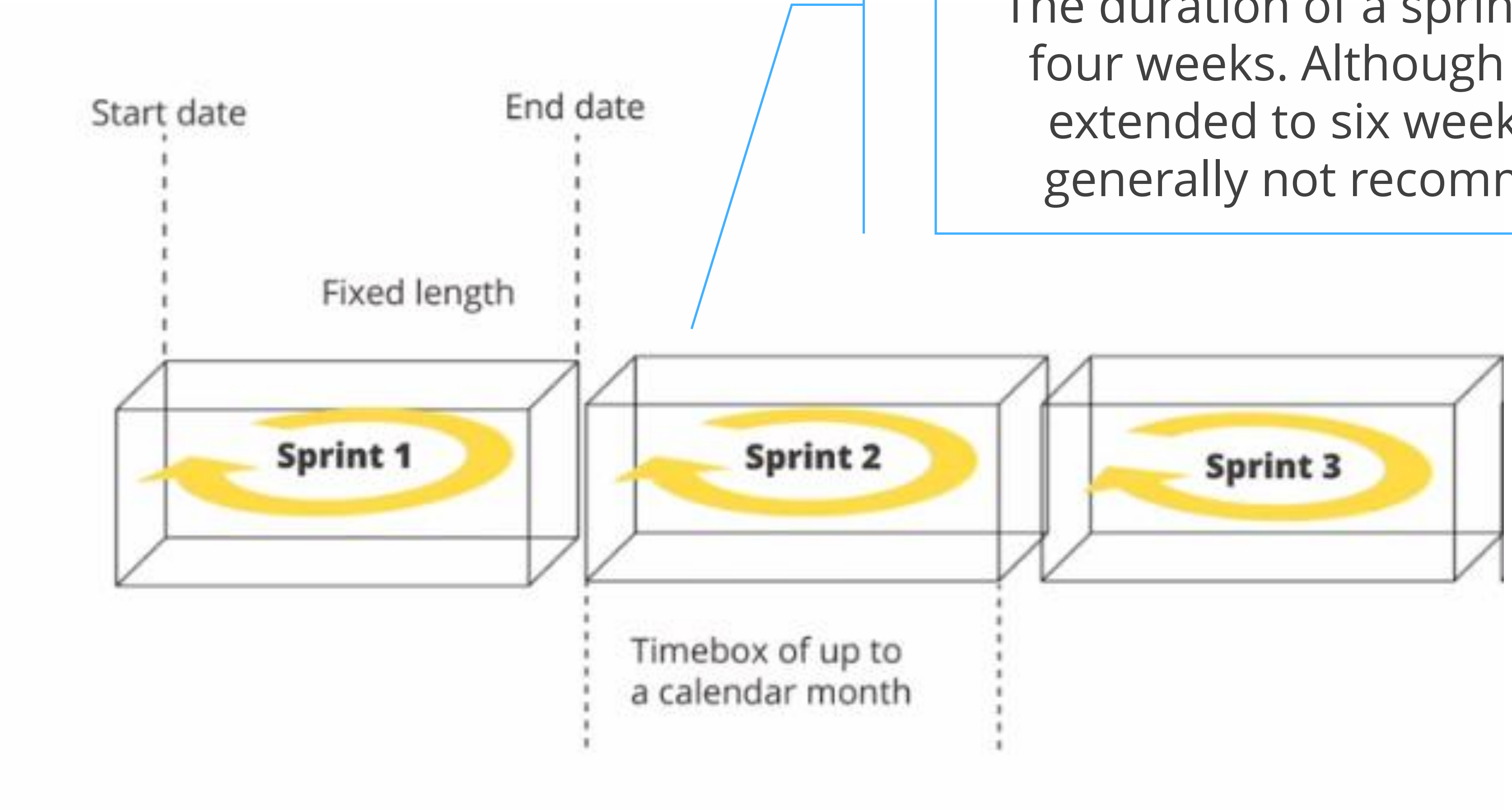


Scrum Rituals—Sprint

A sprint is a period of time during which the development team creates potentially releasable product.

A sprint is performed in this sequence:

- Sprint Planning Meeting
- Daily Scrums
- Sprint Review
- Sprint Retrospective



The duration of a sprint is two to four weeks. Although it can be extended to six weeks, that's generally not recommended.

During the sprint, no changes are made that can danger the sprint goal. Quality goals do not change, and scope may be clarified and renegotiated with the Product Owner as more is learned.

Scrum Rituals—Sprint Planning Meeting

A Sprint Planning Meeting is a timeboxed event scheduled to last two hours for each week of the sprint duration.

Backlog
Sprint Planning Meeting
Sprint Backlog
TASK
TASK
TASK
TASK
TASK
TASK
TASK
TASK
TASK
TASK
TASK
TASK

Sprint planning is used to determine the work that is going to be performed during the sprint. The Scrum team attends the sprint planning meeting.

Scrum Rituals—Sprint Planning Meeting (Contd.)

If you are performing a two week sprint, the sprint planning meeting would be for four hours.

The first half of the meeting is to select the user stories and to define tasks.

The second half of the meeting is to determine the ways to execute tasks.

At the end of the meeting, the Development Team should be able to describe goal accomplishment.

Note: Once a sprint begins, no additional product backlog items are ever added to or removed from the sprint.

Scrum Rituals—Daily Scrum

Daily Scrum is a timeboxed meeting lasting no longer than 15 minutes. It is used by the Development Team to synchronize its activity and to create a plan for the next 24 hours.

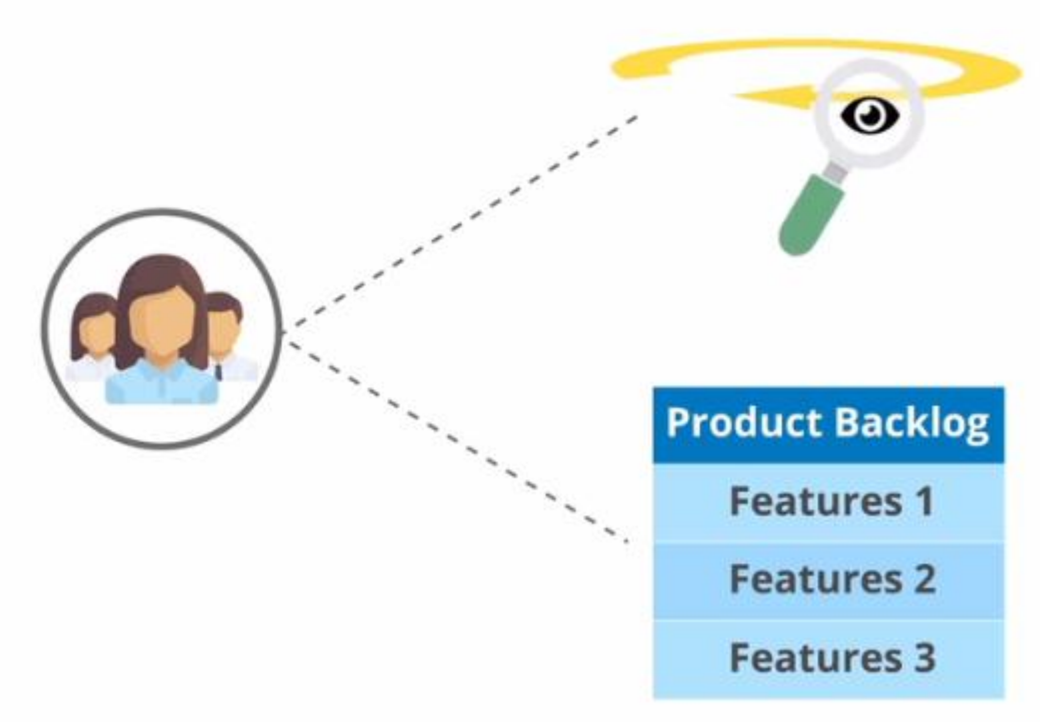
It is accomplished by asking three questions:

- What did I complete yesterday?
- What is my plan for today?
- What are the impediments to my work?

Daily Scrum should ideally take place at the same time and place everyday (for example, at the beginning of the workday).

Scrum Rituals—Sprint Review

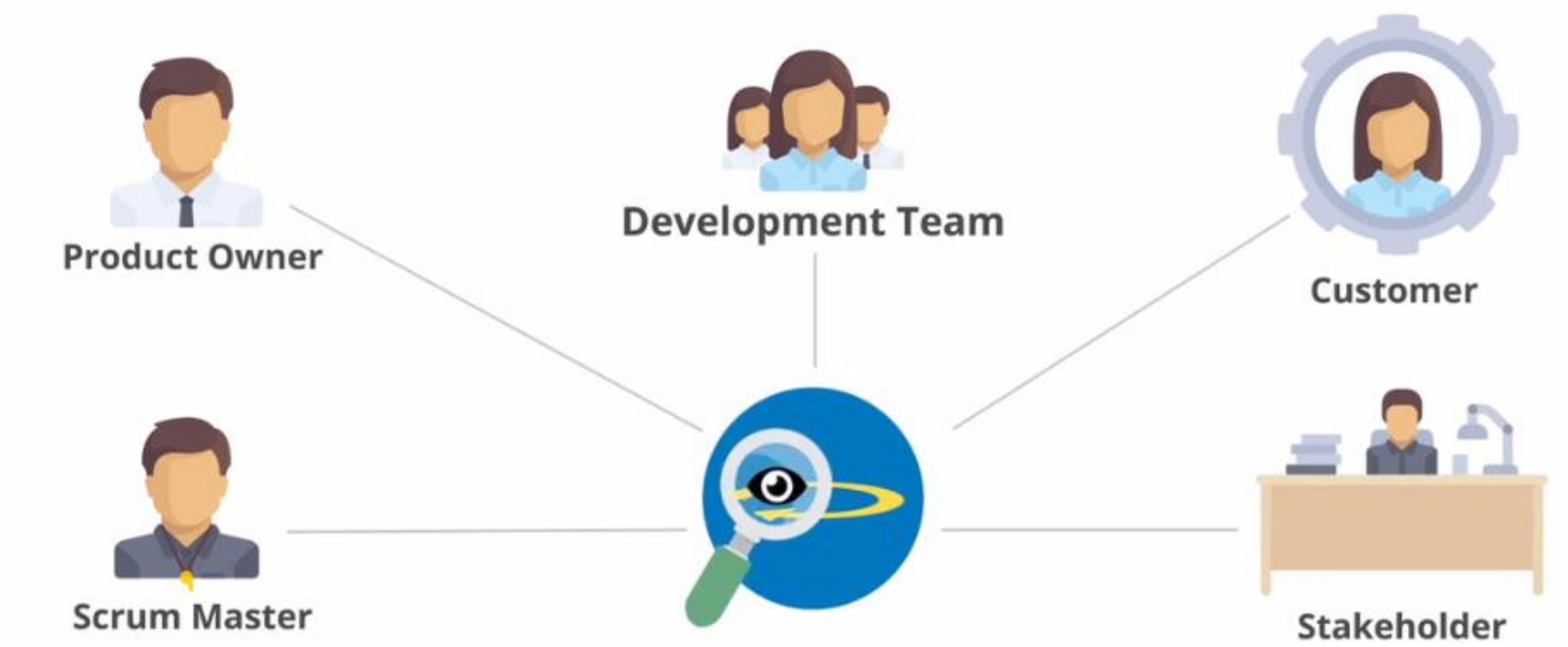
Sprint reviews are held at the end to inspect the results of the sprint and to potentially make changes to the product backlog.



The main purpose of the sprint review is to obtain feedback and faster collaboration. It is a timeboxed meeting that lasts an hour and takes places once a week during the sprint.

Scrum Rituals—Sprint Review (Contd.)

It is advisable to have customers and stakeholders along with the Scrum Team.



- The Development Team demonstrates the work completed during the sprint.
- The Product Owner accepts or rejects the work.
- The Product Backlog is revised based on the feedback obtained.

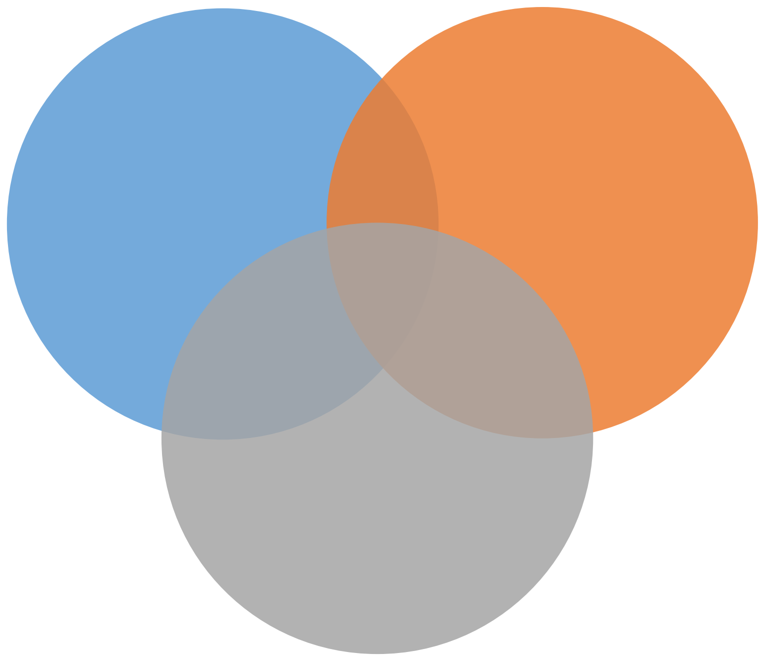
Scrum Rituals—Sprint Retrospective

The Sprint Retrospective is an oppurtunity for the scrum team to inspect itself and to determine how best to implement improvements for the next sprints.

- It is a timeboxed event with 45 minutes alloted to each week of a sprint.
- It focuses on the project and never devolves into blaming individual team members.

Scrum Rituals—Sprint Retrospective (Contd.)

Sprint retrospective focuses on three main questions.



What went well?

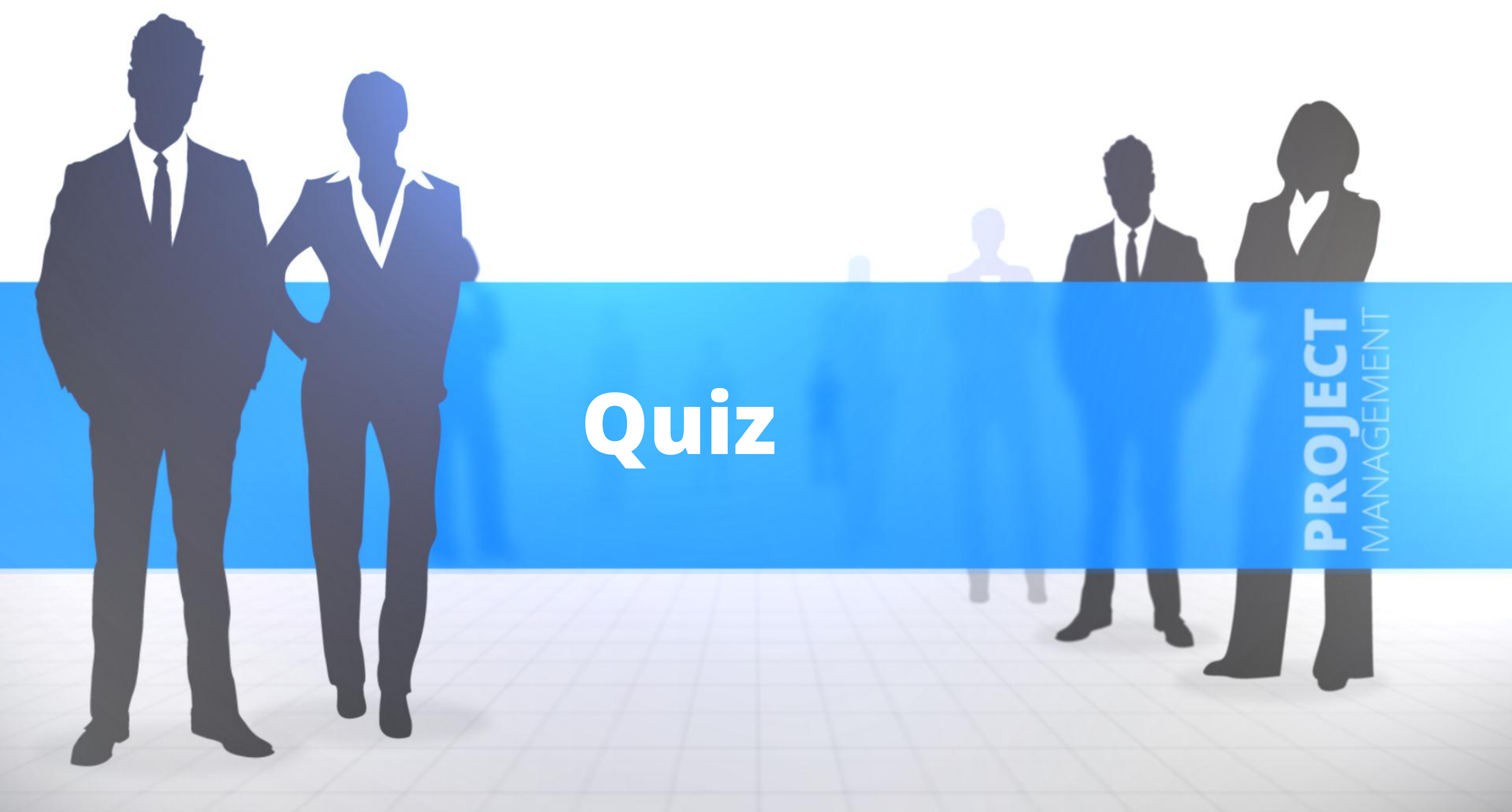
What did not go well?

What should be done differently next time?

This is a critical event in Scrum and continuous achievement will not be possible without it.

Topics Covered

- ✓ A Scrum Team includes a Product Owner, Scrum Master, and a Development Team.
- ✓ The Scrum Master acts as a servant leader and contributes to the success of the Product Owner and the Development Team.
- ✓ The Product Owner and the Development Team have their own responsibilities.
- ✓ Sprint events include Sprint Planning, Daily Scrums, Sprint Reviews, and Sprint Retrospectives



Quiz

QUIZ

1

The stakeholders of a company decide to transition the software development team to Scrum. They already have Project coordinator who facilitates interactions, removes impediments, and acts as the coach of the Team. What should this role be called after the transition?

a. Project Owner

b. Project Manager

c. Scrum Master

d. Scrum Project Manager

QUIZ

2

Why is diversity a necessary attribute of a Scrum Team?

a. To build cultural sensitivity

b. To enable proper succession planning for each role

c. To comply with Federal Regulations

d. To encourage different viewpoints and healthy debate

QUIZ

3

Which of the following are the questions that are asked during a Daily Scrum? (Choose Two)

a. What are the impediments to your work?

b. What will be accomplished before the next meeting?

c. Who should take care of the next task?

d. What did you do with the customer feedback?

QUIZ

4

Toward the end of the Sprint, the Development Team realizes that it will not be able to complete the stories it has committed to. What is the ideal course of action for the Development Team?

a. Decide on stories that can be delayed until the next Sprint in consensus with the Scrum Master

b. Ask the Product Owner to decide which stories can be delayed until the next Sprint

c. Develop a new Definition of Done for the Sprint Backlog Items

d. Ask for more team members to meet the goals of the current Sprint

This concludes “Roles and Rituals.”

The Next Lesson is “Scrum Practices.”

Agile Scrum Foundation

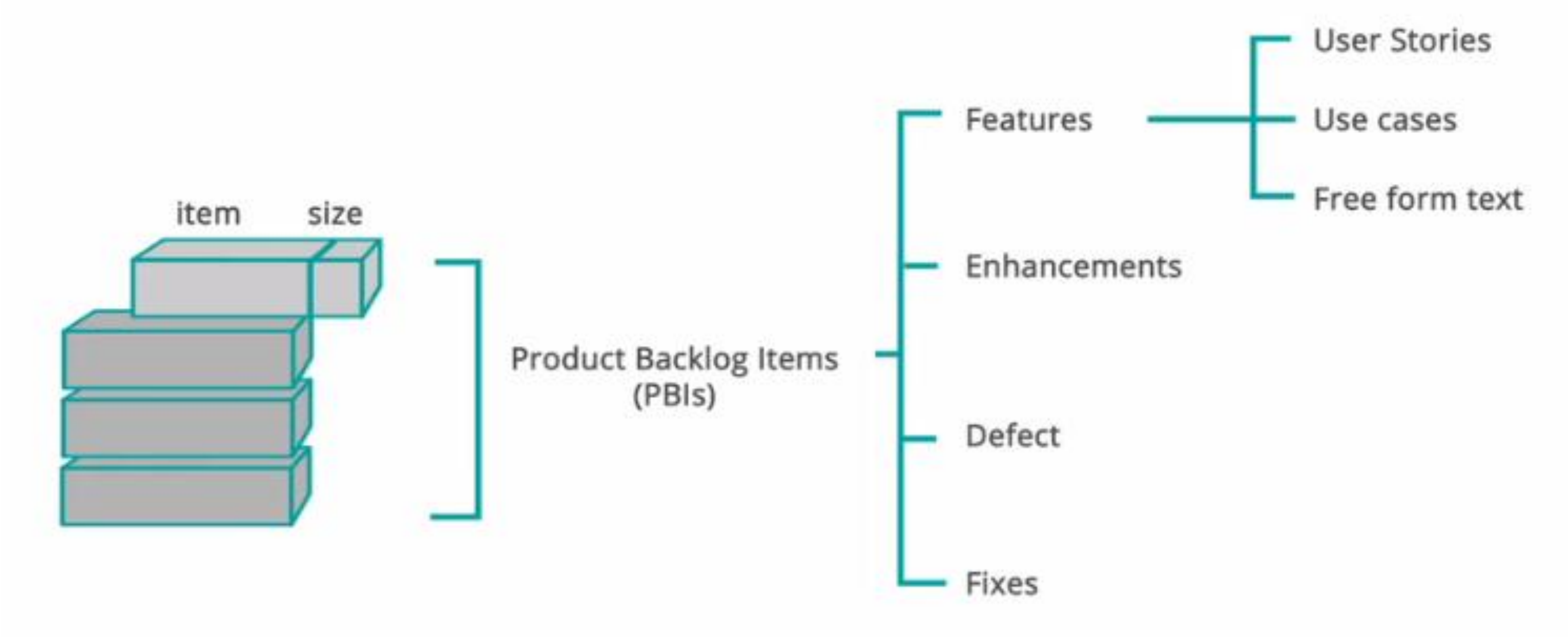
Lesson 3—Scrum Practices

Objectives

- ✓ Describe Product Backlog
- ✓ Explain what makes a Good User Story
- ✓ Describe Sprint Backlog
- ✓ Explain Time-boxing and its importance
- ✓ Describe Extreme Programming and its practices
- ✓ Describe Test Driven Development
- ✓ Explain the “Definition of Done”

Product Backlog

It is a prioritized list of "requirements" that is created and maintained by the Product Owner of a product.



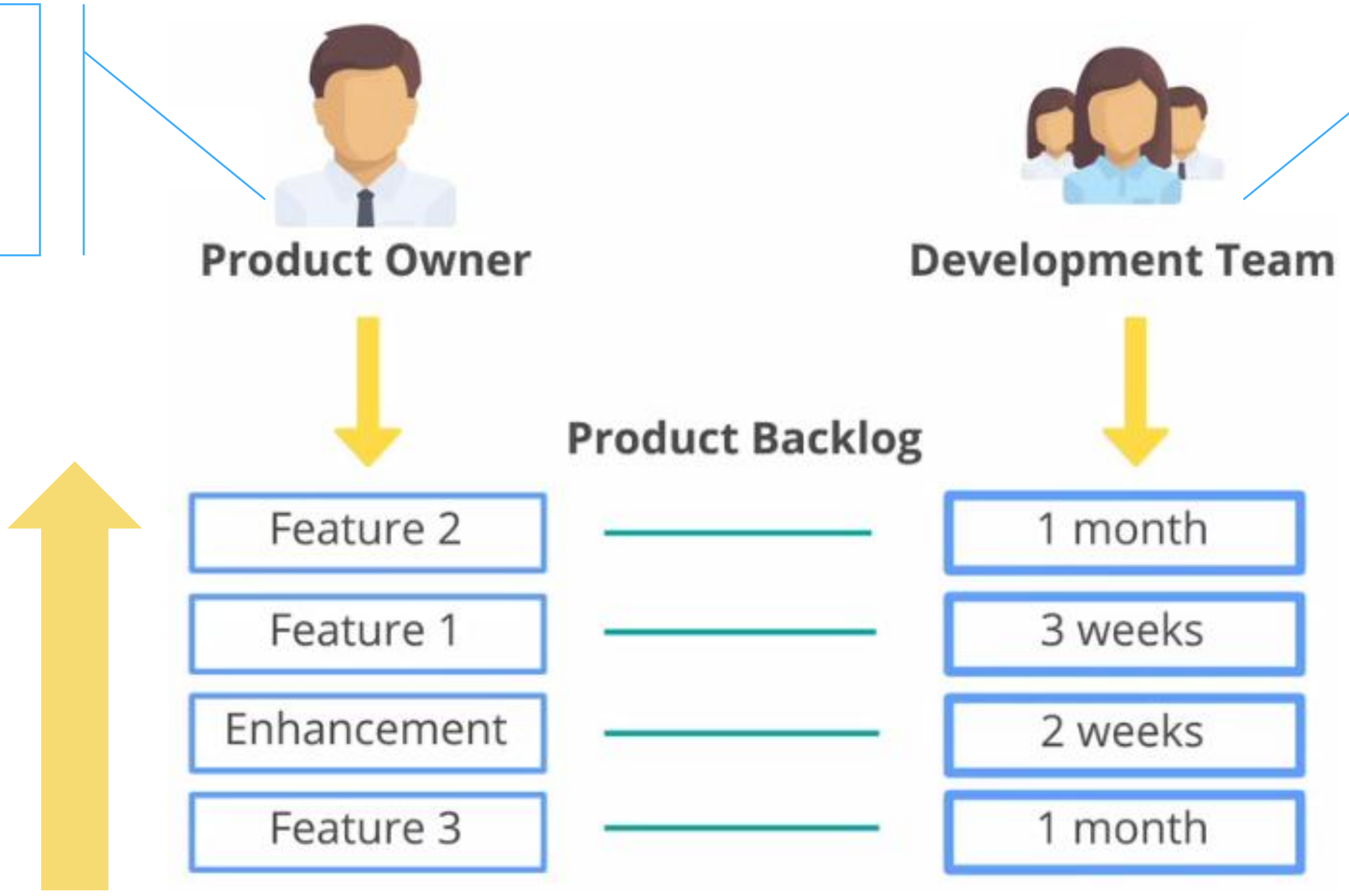
The product backlog evolves in order to be appropriate, competitive, and useful.

Product Backlog (Contd.)

The Development Team contributes to the product backlog by identifying non-functional items and risk items of the project.

Selects items based on business value

Estimates items either in story points or estimated hours



Product Owner

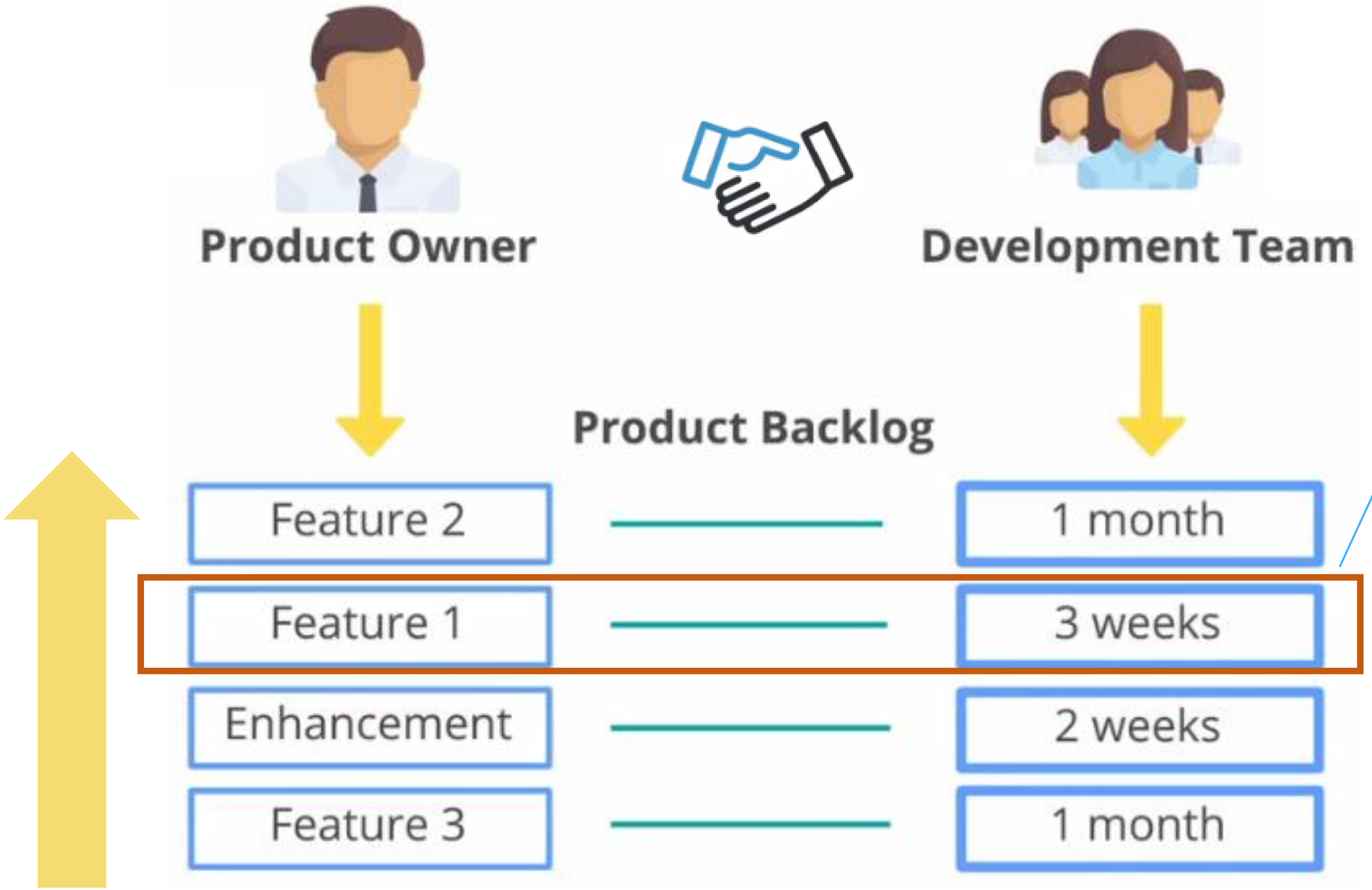
Development Team

Product Backlog

Feature 2	1 month
Feature 1	3 weeks
Enhancement	2 weeks
Feature 3	1 month

Product Backlog (Contd.)

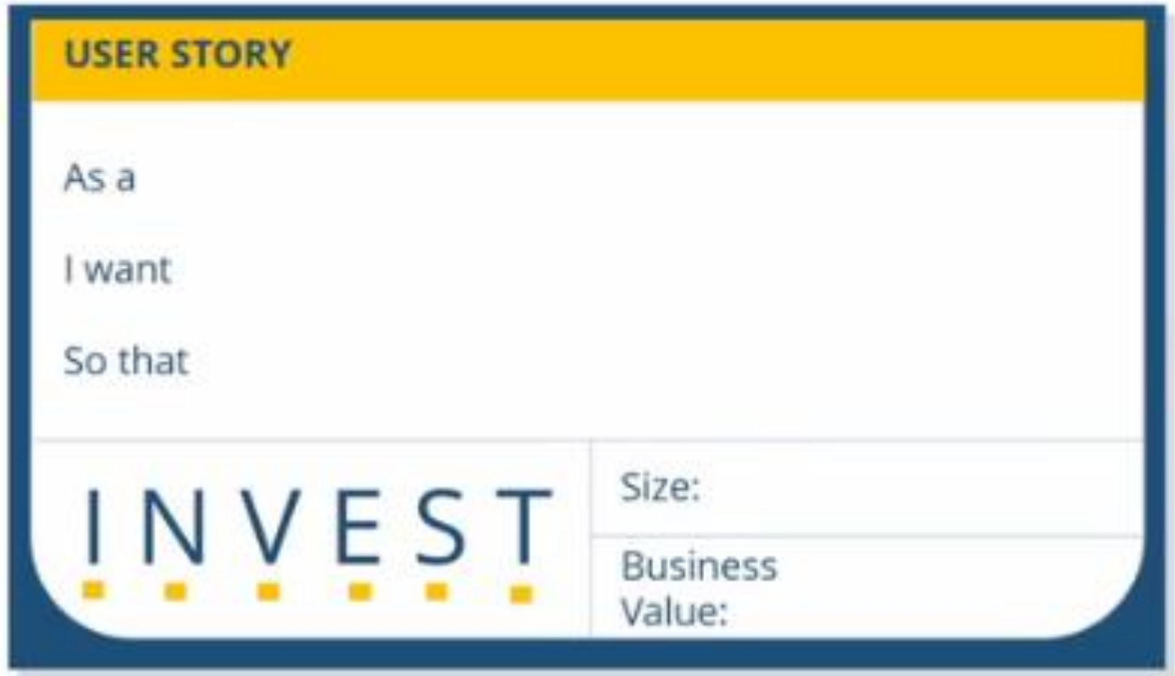
A product backlog item, often called a user story, is a lightweight mechanism to quickly capture requirements.



- A user story acts as an agreement regarding the features and the requirements that need to be developed.
- It helps to shift the focus from writing about requirements to talking about them.

Product Backlog (Contd.)

- User stories typically follow a simple template.
- They are written on index cards and are arranged on walls or tables; this facilitates further planning and discussion.



USER STORY

As a

I want

So that

INVEST

Size:

Business Value:

Example

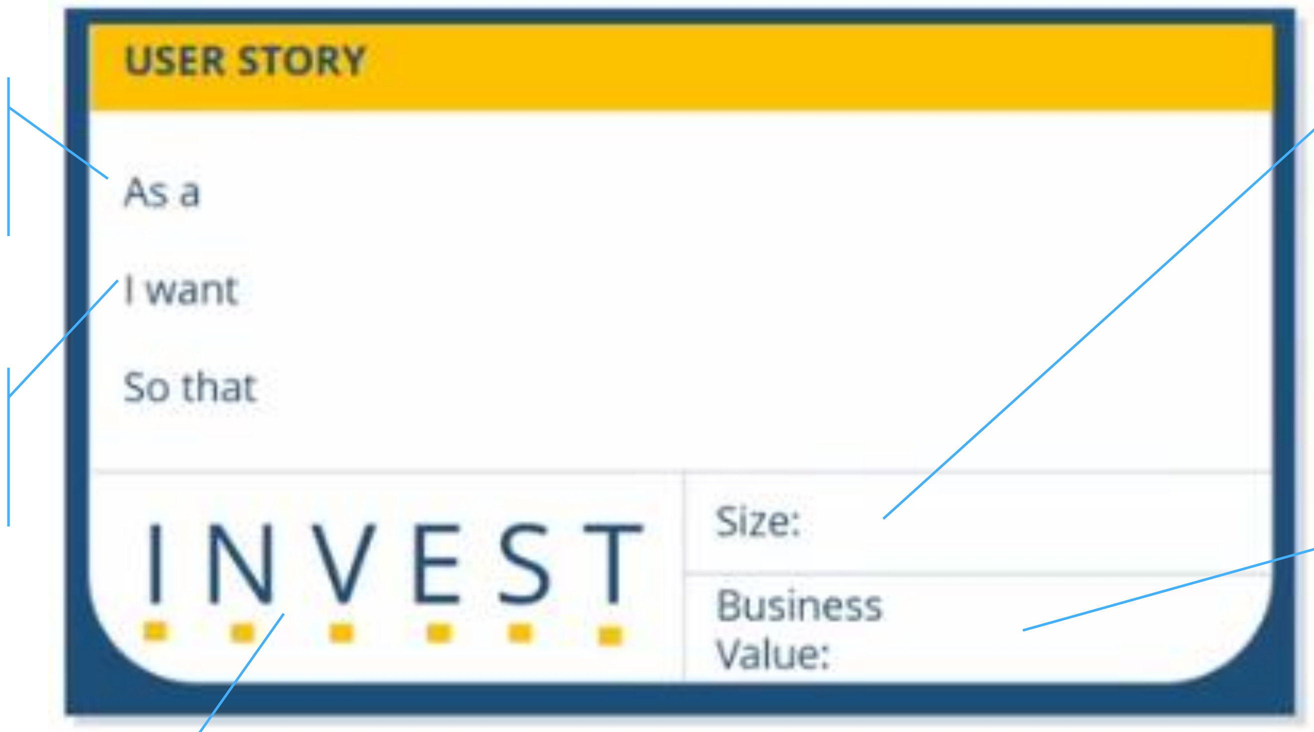
#	Backlog Item (User Story)	Story Point
1	As a teller, I want to be able to find clients by last name so that I can find their profile faster.	4
2	As a system admin, I want to be able to configure user settings so that I can control access	2

Product Backlog (Contd.)

A user story card will typically contain:

An item identifier and name

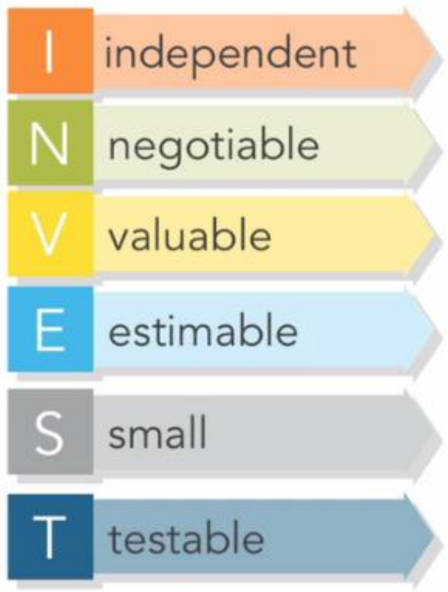
Description



The estimated effort required from the team’s perspective, along with the associated risks, dependencies, and acceptance tests.

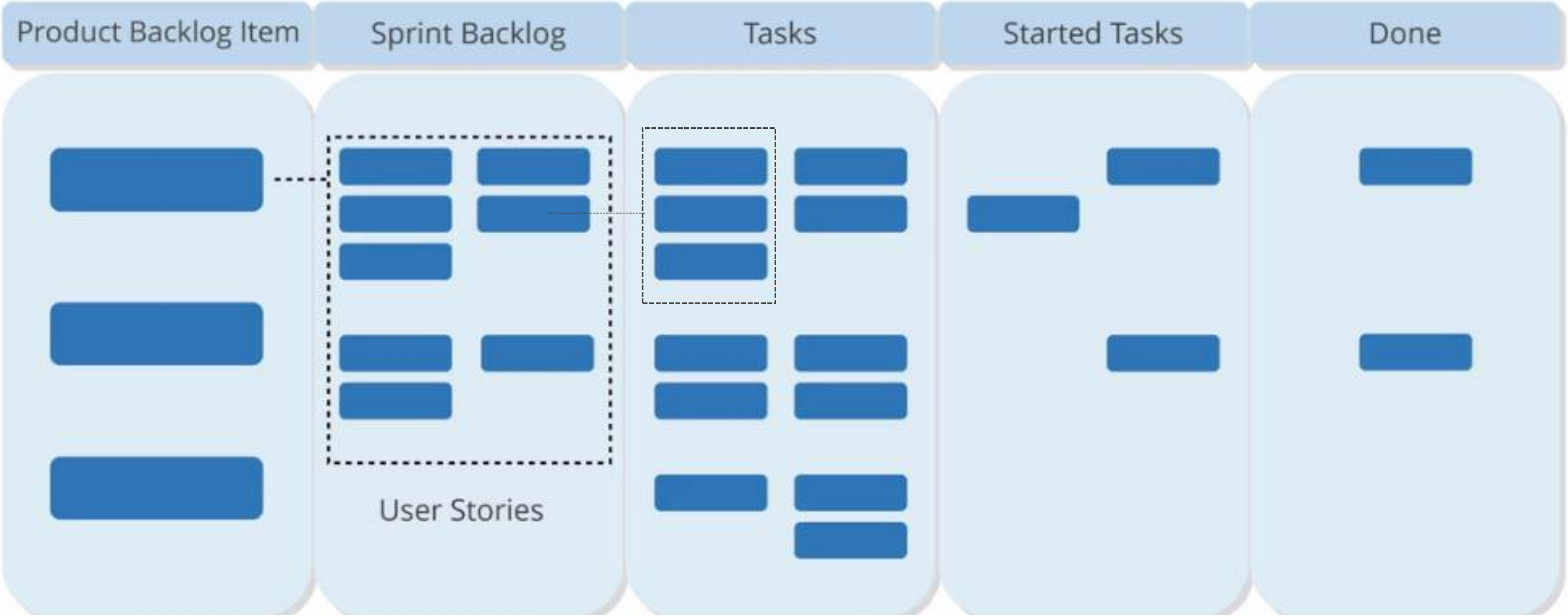
An estimated value from the product owner’s perspective

The acronym INVEST is used to communicate the attributes of a good user story.



Sprint Backlog

Sprint Backlog is a subset of Product Backlog. It emerges during Sprint Planning and does not change during the course of the Sprint.



Sprint Backlog (Contd.)

During Sprint Planning, the Development Team determines tasks and estimates the effort to complete each task.

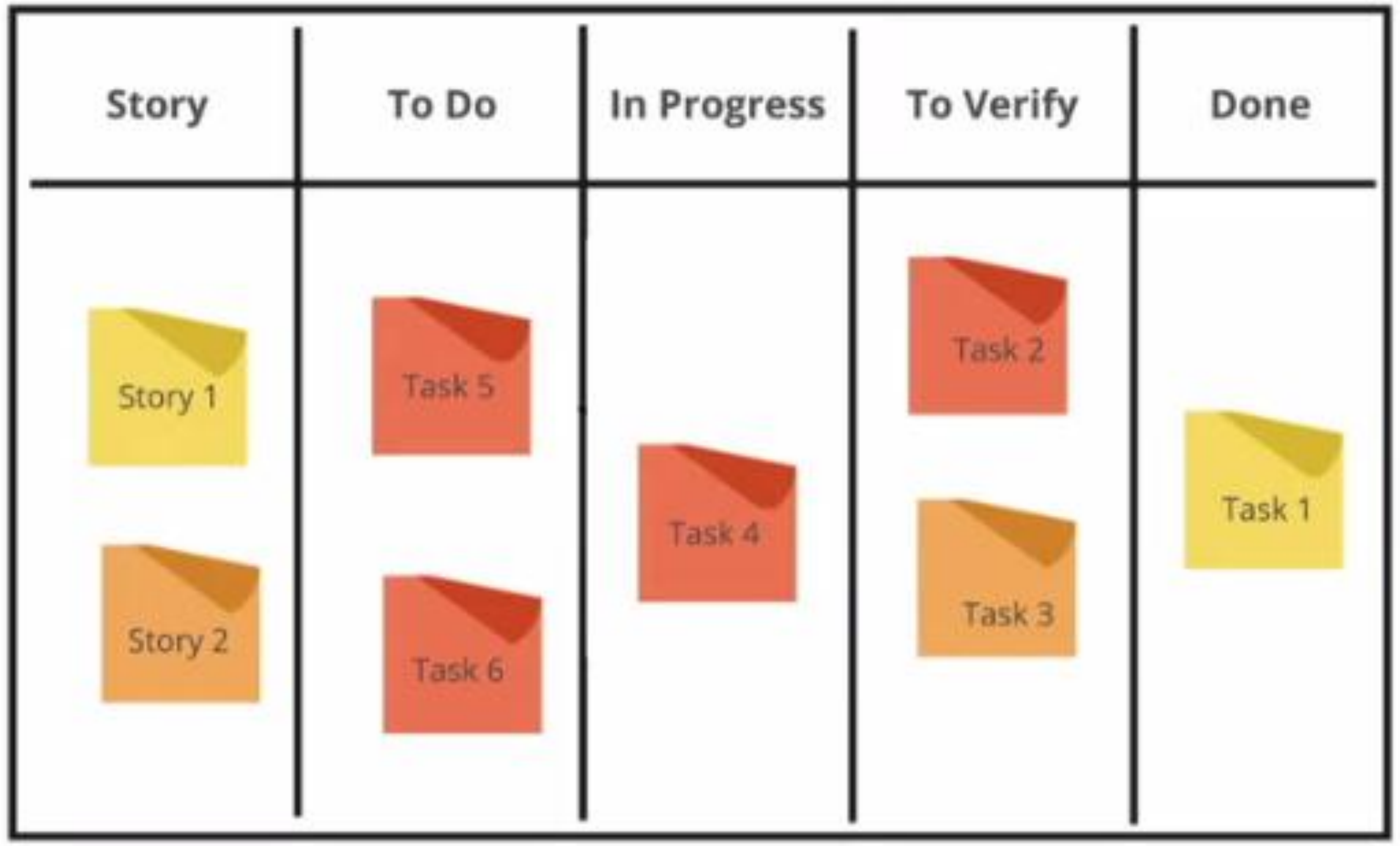
Development Team

The team also considers risks, dependencies, and constraints associated with each of the user stories.

Task Board

Tasks are tracked on a task board or Kanban board.

It is a work visualization tool that provides high visibility of the progress and status of the Sprint.



Story	To Do	In Progress	To Verify	Done
Story 1 Story 2	Task 5 Task 6	Task 4	Task 2 Task 3	Task 1

The Sprint Backlog is owned and maintained by the Development Team.

Time-boxing

All events are time-boxed. If any task cannot be completed in the designated period of time, it is referred to the next period.

In scrum, time-boxing is critical for:

- Continous improvement
- Determining the team's velocity
- Improving colloboration

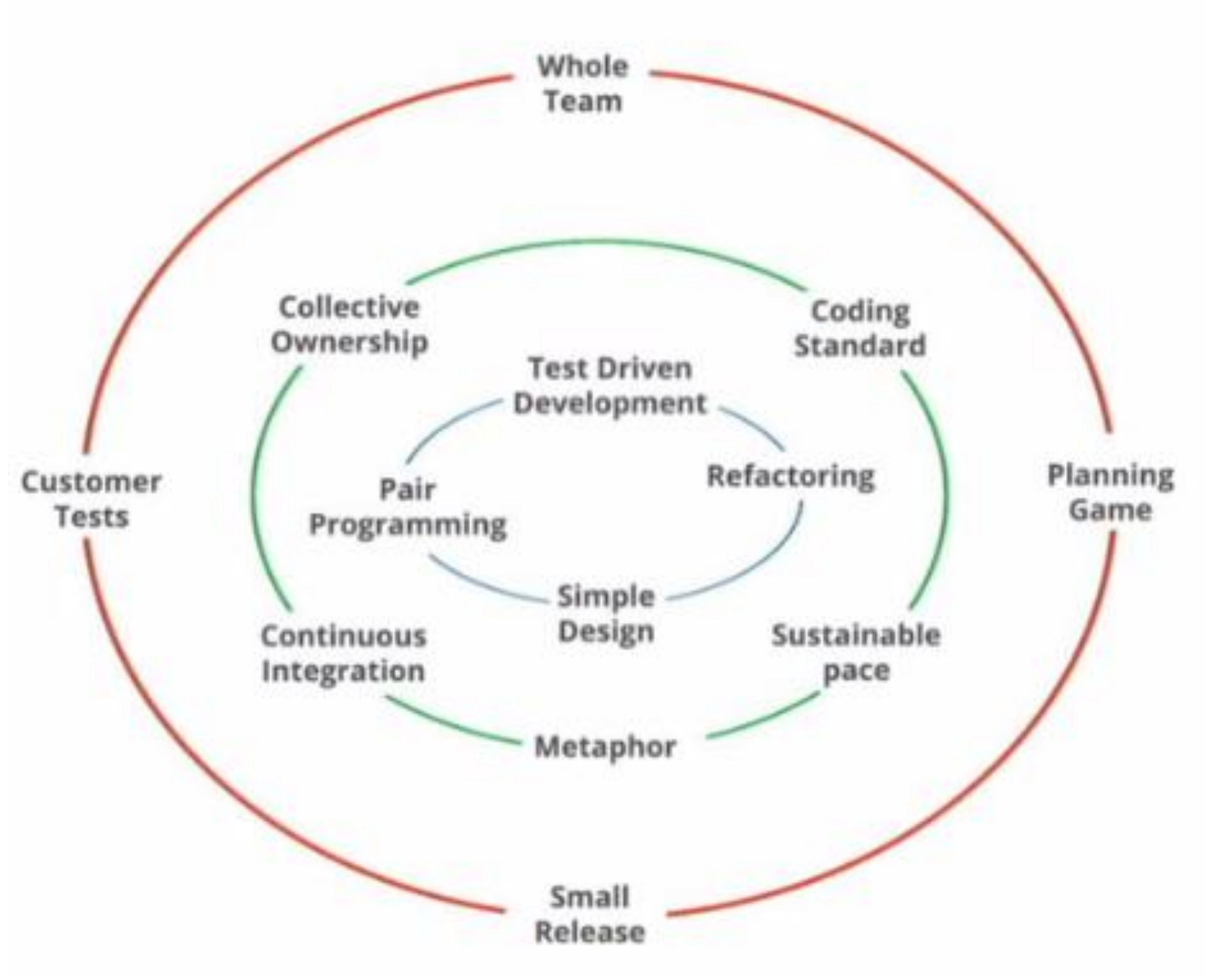
Maximum Duration:	
Sprint	2 to 4 weeks
Sprint Planning	2 hours for each week of the Sprint
Daily Scrums	15 minutes
Sprint Reviews	1 hour for each week of the Sprint
Sprint Retrospective	45 minutes for each week of the Sprint

eXtreme Programming (XP)

Extreme Programming, referred as XP, was developed in the 1990s by Kent Beck and Ward Cunningham.

Their main purposes in developing XP were to:

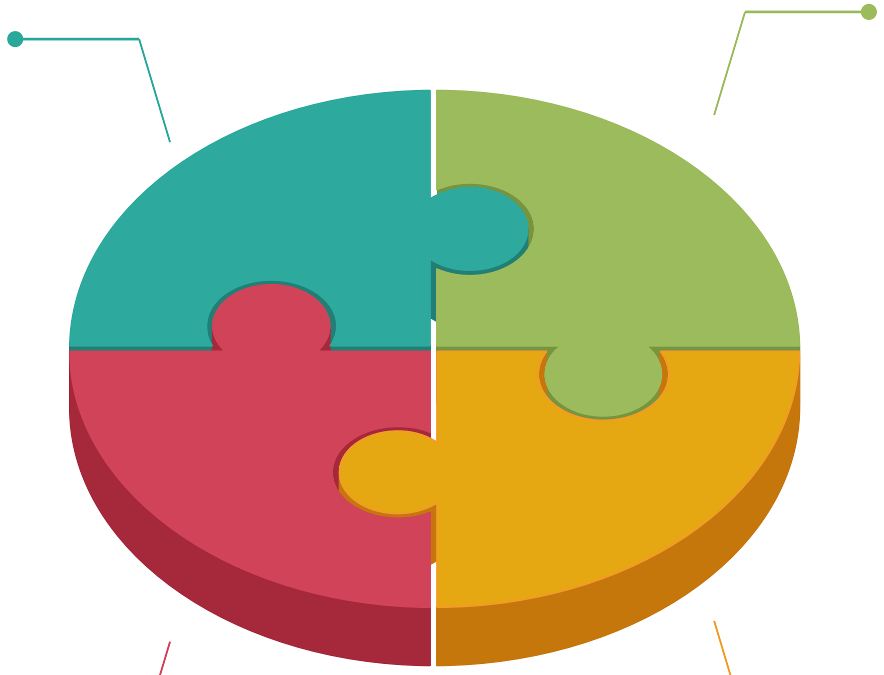
- Respond to the high cost of the changing requirements
- Establish strong engineering practices in order to improve software quality



eXtreme Programming (XP) (Contd.)

XP introduced many revolutionary concepts to software development that are now standard practices.

XP Practices used in Scrum include:



Test Driven Development (TDD)

Continuous Integration

User Stories

Iteration (Sprint)

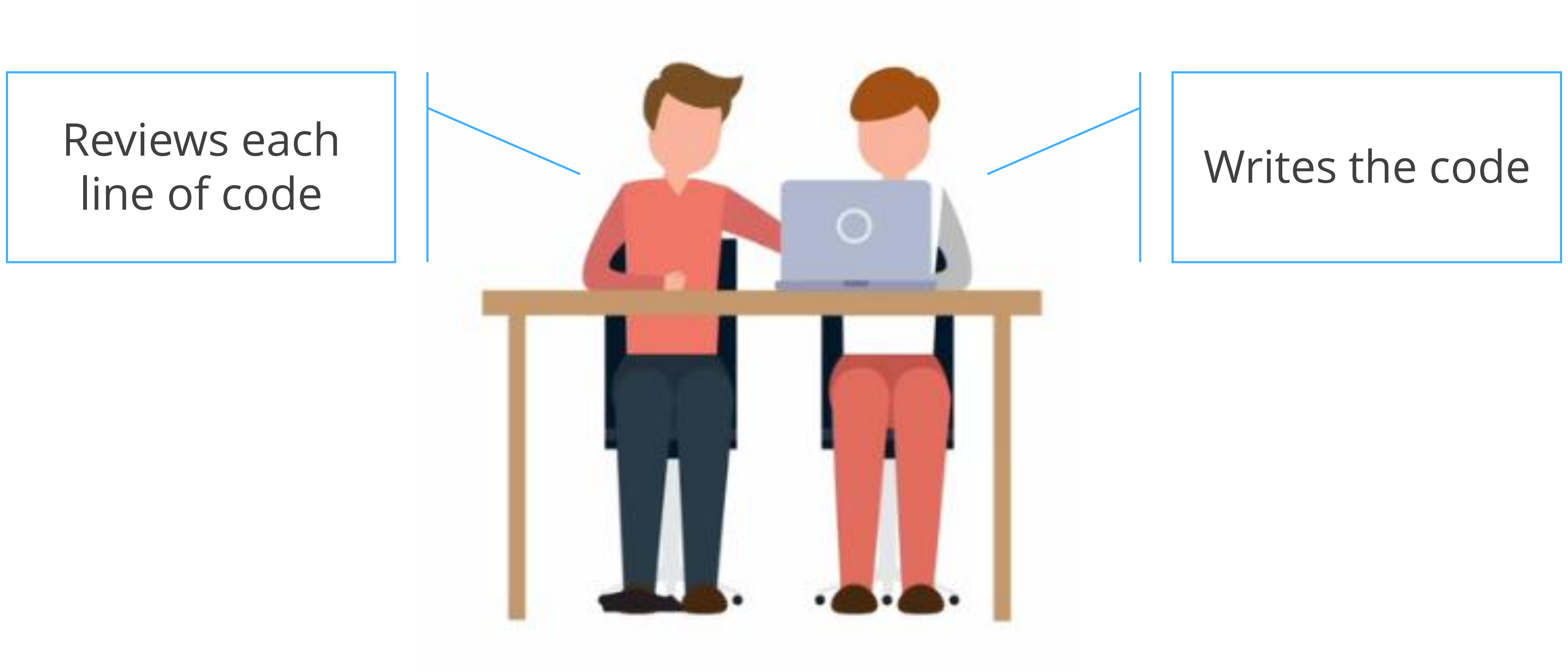
XP—Code Refactoring

Code Refactoring is a process of restructuring existing computer code—changing the factoring—without changing its external behavior.

It Improves the nonfunctional attributes of the software and simplifies code, making future coding on the project much easier.

XP—Pair Programming

Pair Programming is a technique in which two programmers work together at one workstation.



Reviews each line of code

Writes the code

It has an element of constant reflection (Real-time Reviewing) and reduction of noise and distraction.

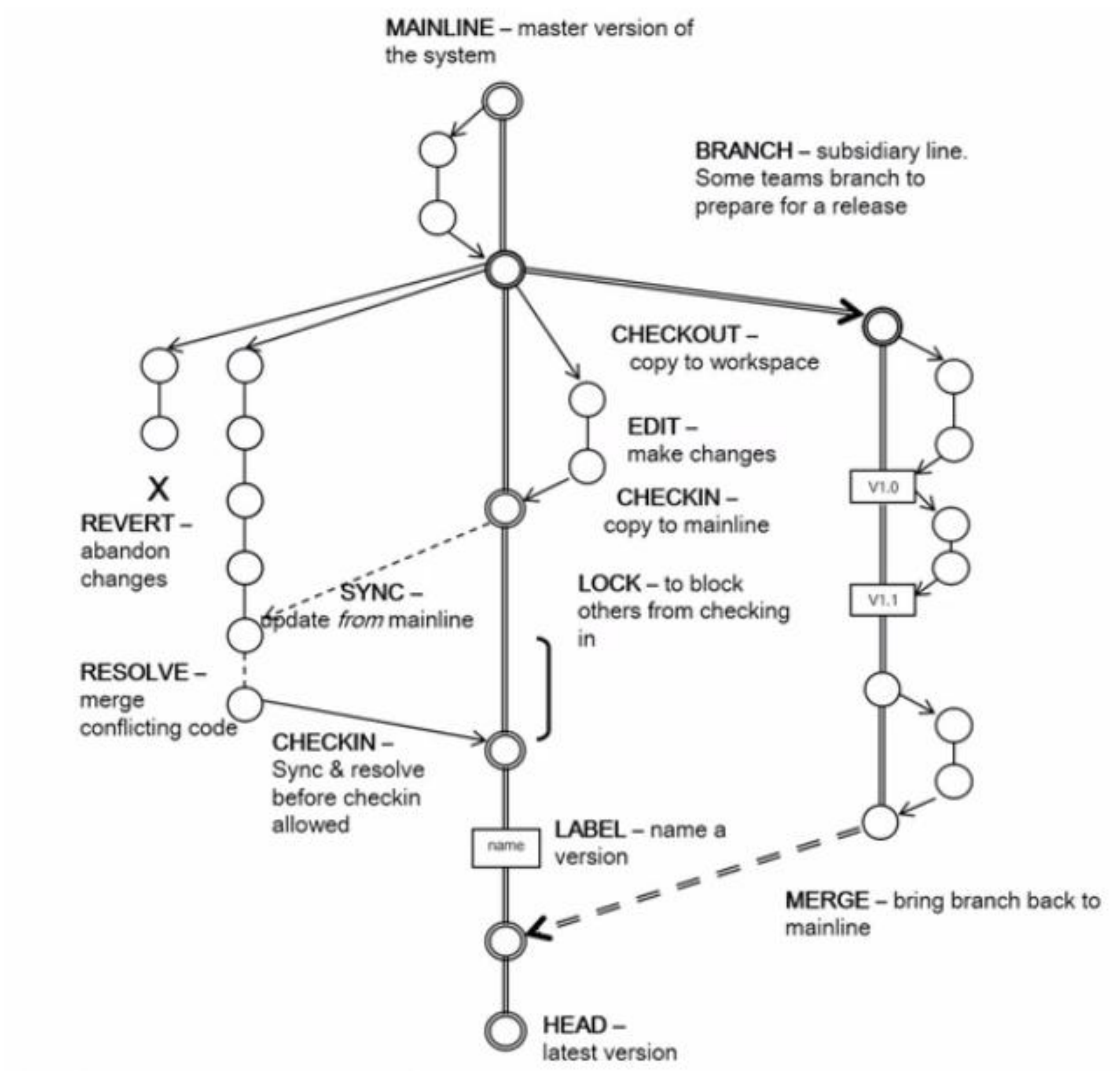
XP—Software Configuration Management (SCM)

SCM is the task of tracking and controlling changes to the software.

It is used to bring control to the software development process and ensures development of high-quality software.

XP—Software Configuration Management (SCM) (Contd.)

SCM is accomplished by identifying the configuration of software and implementing a change control process so that changes can be tracked and their implementation can be monitored.



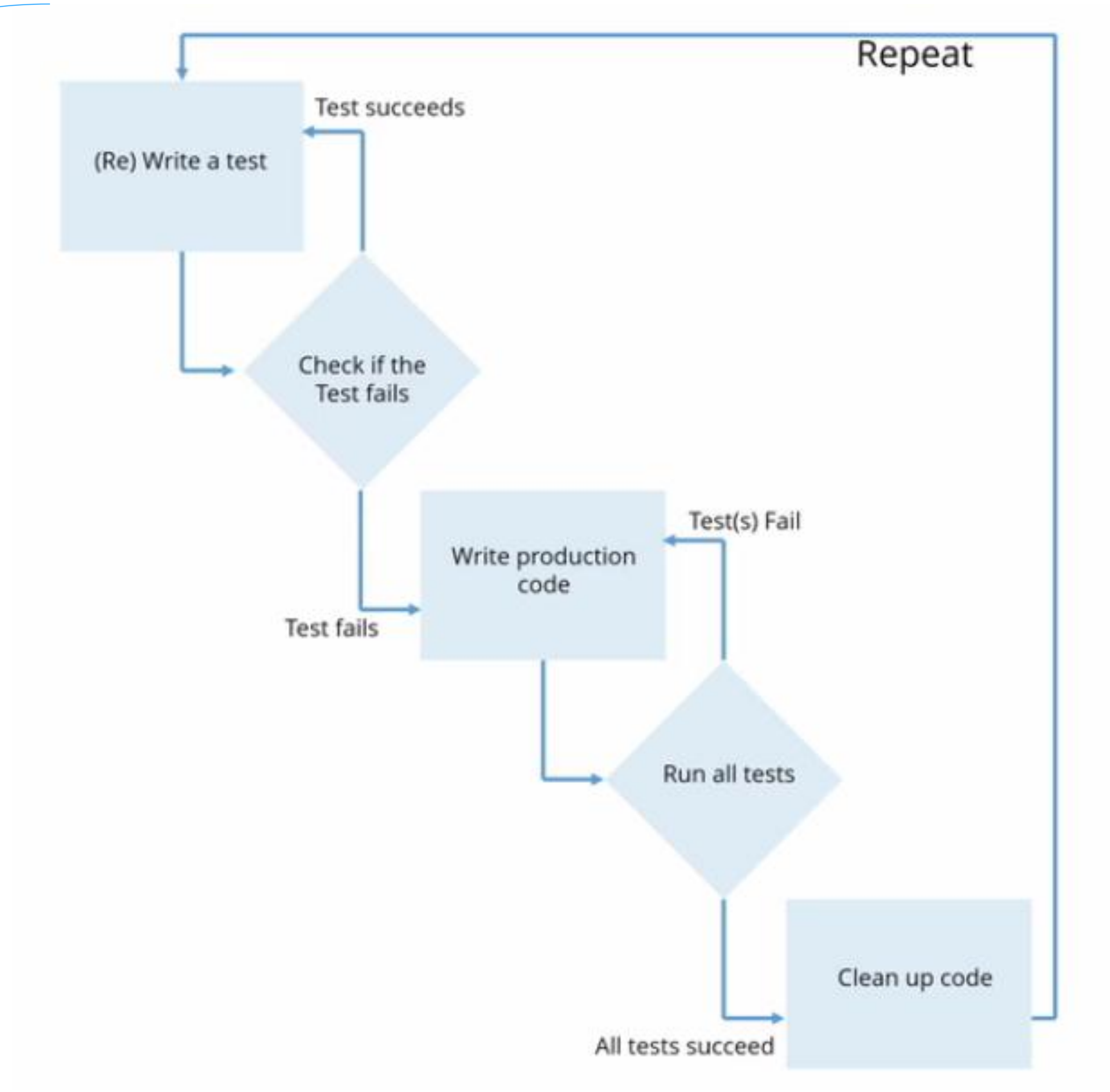
MAINLINE – master version of the system
BRANCH – subsidiary line. Some teams branch to prepare for a release
CHECKOUT – copy to workspace
EDIT – make changes
CHECKIN – copy to mainline
X
REVERT – abandon changes
SYNC – update *from* mainline
LOCK – to block others from checking in
V1.0
V1.1
RESOLVE – merge conflicting code
CHECKIN – Sync & resolve before checkin allowed
name
LABEL – name a version
MERGE – bring branch back to mainline
HEAD – latest version

SCM can determine what was changed and who changed it and can implement an intelligent resolution.

XP—Test-Driven Development (TDD)

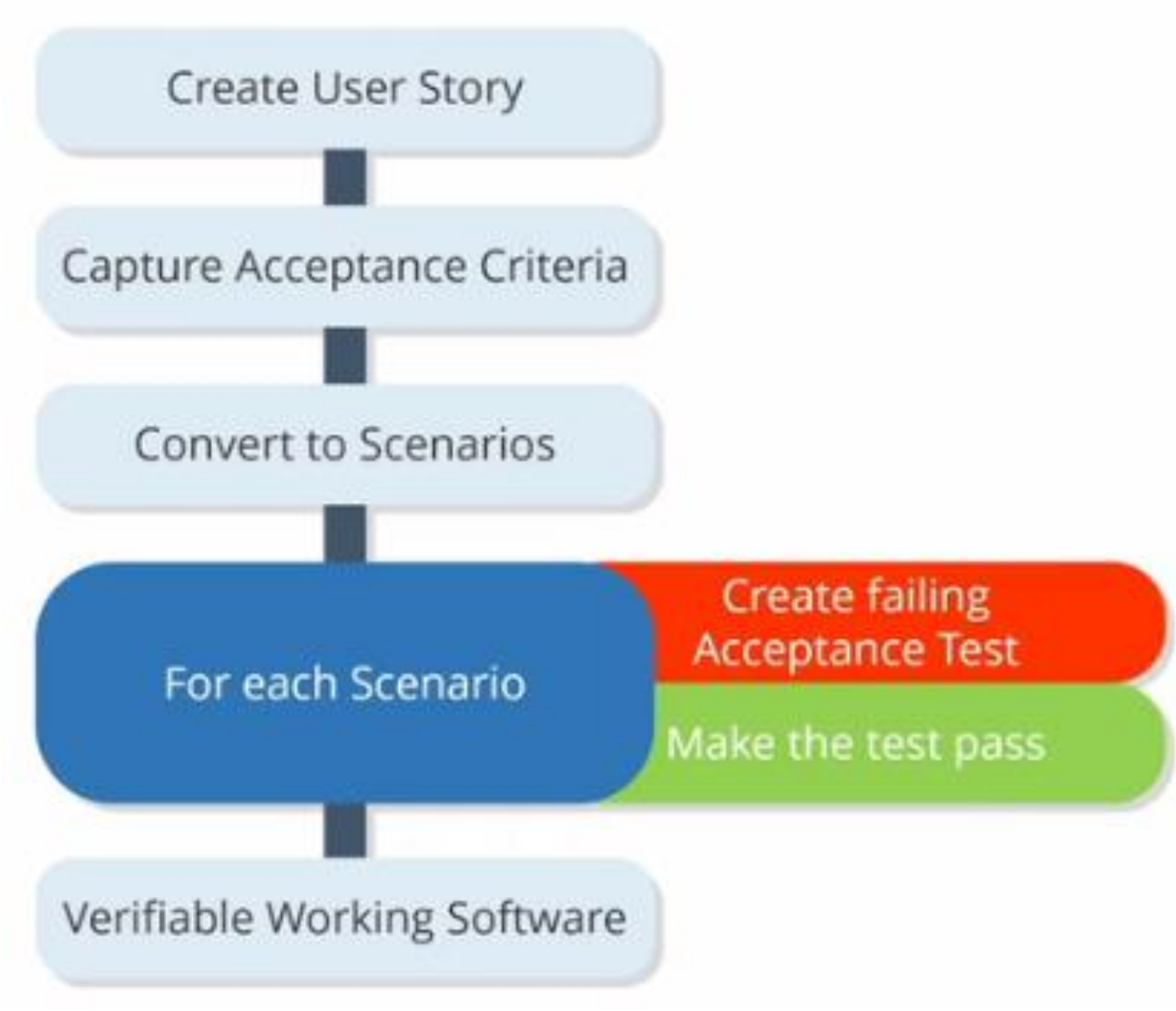
It is an evolutionary approach to software development where developers first write the test and then write the code; this satisfies the conditions of the test.

The flow of TDD



Acceptance Test-Driven Development (ATDD)

ATDD is similar to TDD but is different in that it is used to define a user story's acceptance criteria before and after the development.


Create User Story
Capture Acceptance Criteria
Convert to Scenarios
For each Scenario
Create failing Acceptance Test
Make the test pass
Verifiable Working Software

Definition of Done

The artifact ‘Definition of Done’ is a checklist of things that must be verified before an item or a story is marked done.

‘Definition of Done’ can be applied to a:

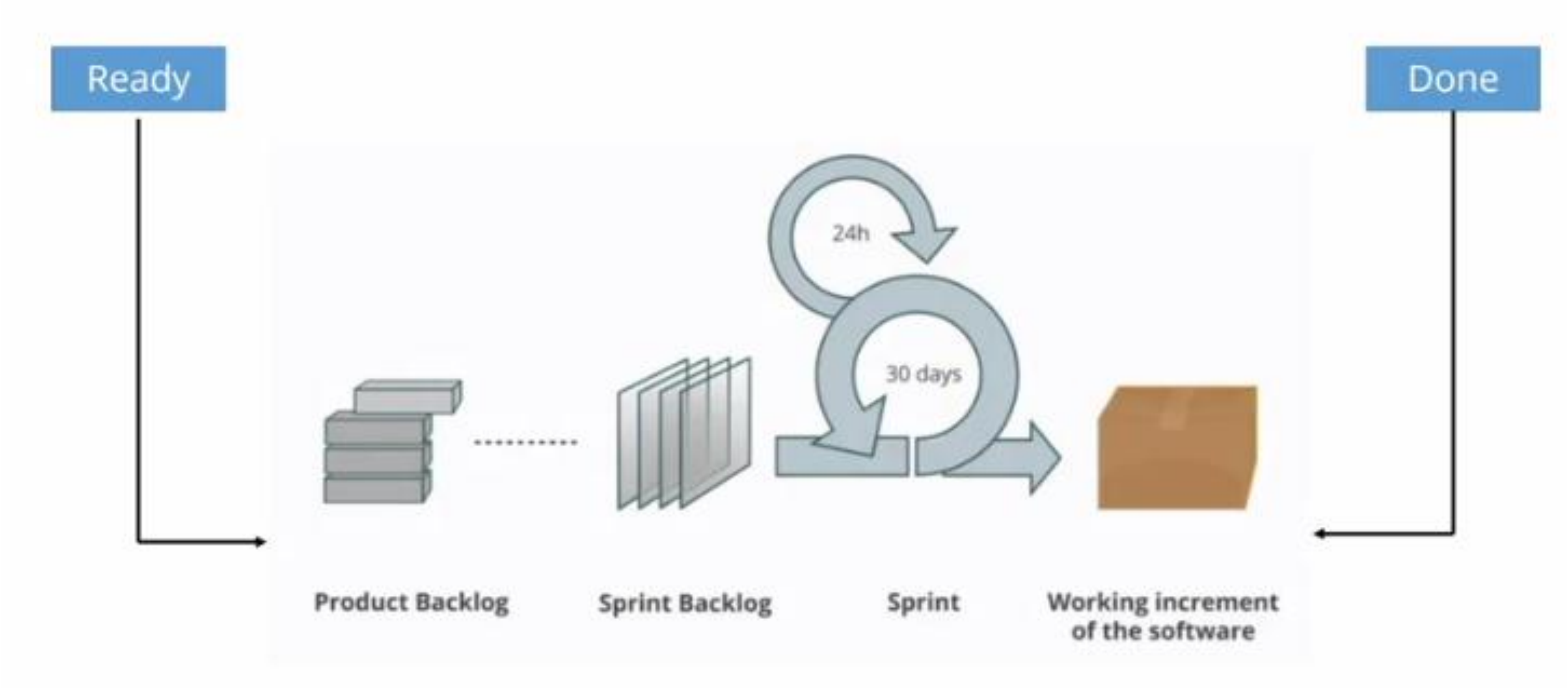
- User story
- Sprint
- Release
- project

Example	
✓	The “billed” is ready for release and is available for download.
✓	Documentation is complete.
✓	Testing is complete.
✓	The source code is committed to the server.
✓	Code has been reviewed, and the product owner has given approval.

The Scrum Team collaboratively develops and agrees to all of the stipulations of the definition.

Definition of Done (Contd.)

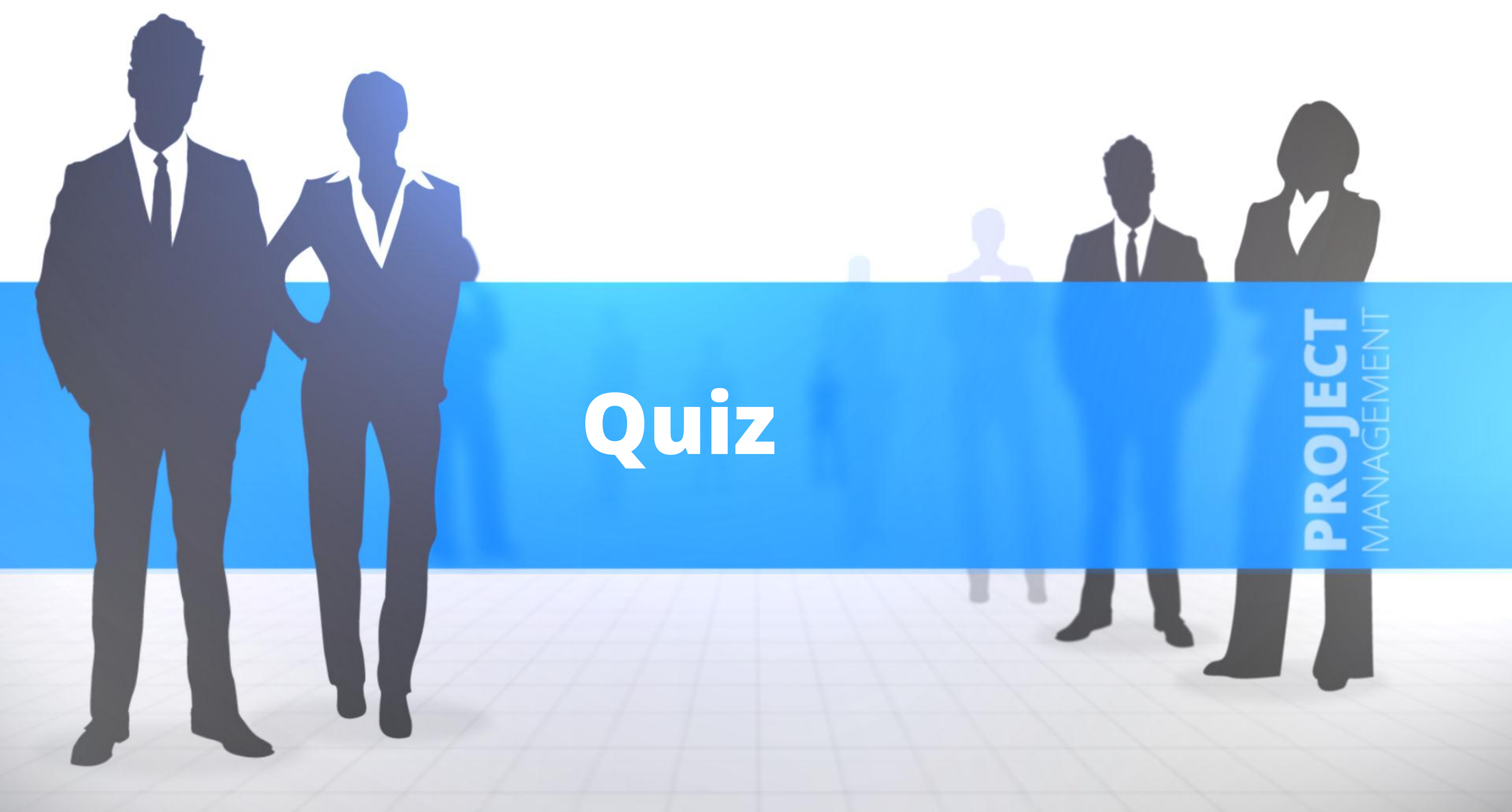
It is expected that the Definition of Done will evolve as the Scrum team matures.



This graphic illustrates various Definition of Done elements, from Ready to Done.

Topics Covered

- Product Backlog and Sprint Backlog
- User Stories and how to invest in them
- Importance of Time-boxing
- Extreme Programming and its practices
- The Definition of 'Done'



Quiz

QUIZ

1

A Product Owner wants a story to be completed in four days. The Development Team working on the Story reckons it will take seven days. The Scrum Master feels it should take five days. A subject matter expert, who has worked on similar Stories in the past, thinks it should take a maximum of three days. Whose estimate should be used for planning?

a. The Development Team’s

b. The Scrum Master's

c. The Subject Matter Expert’s

d. The Product Owner's

QUIZ

2

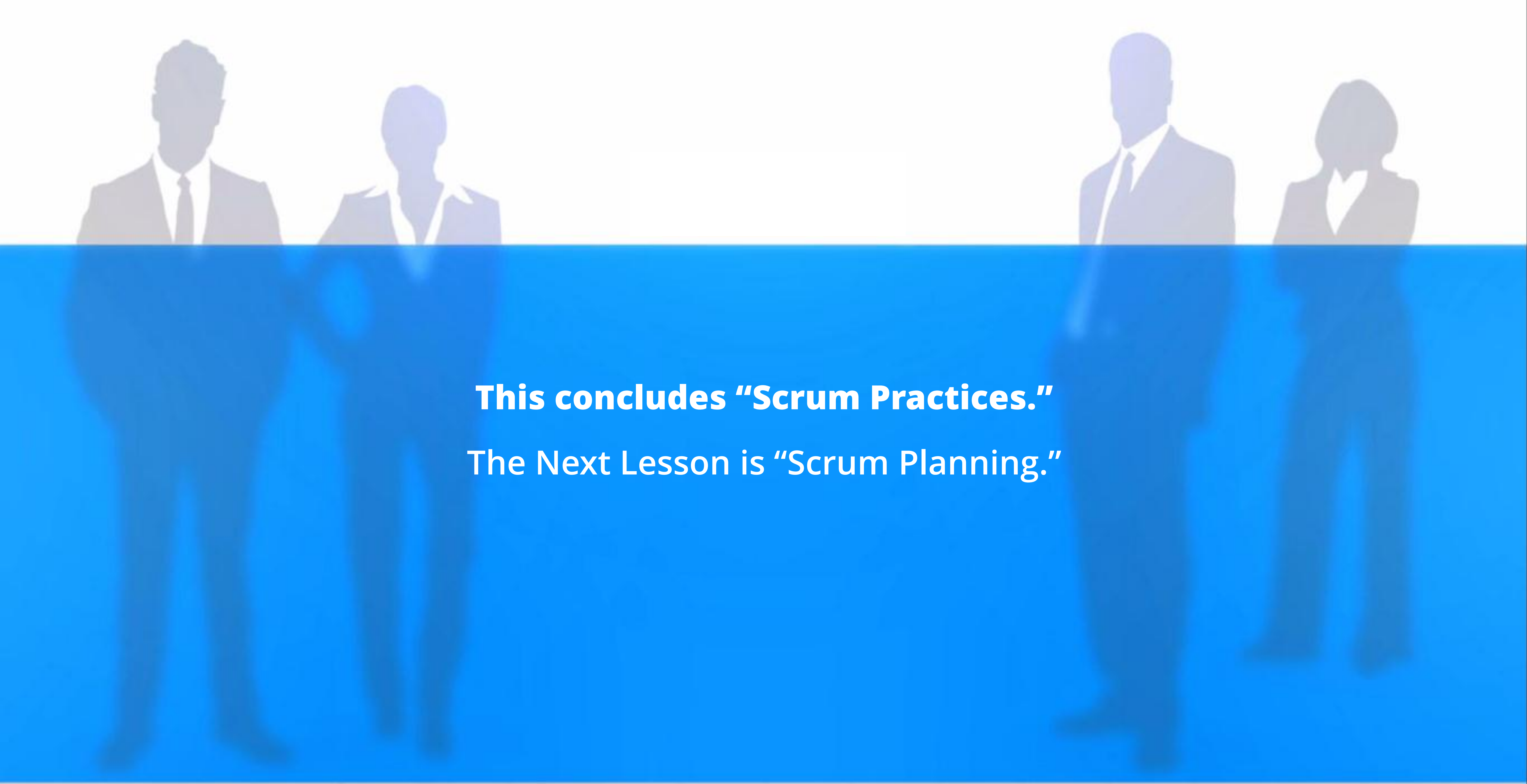
Which items on the product backlog should be small?

a. The items at the bottom of the Product Backlog

b. The items at the top of the Product Backlog

c. Only items on the Sprint Backlog must be small

d. All items in the Product Backlog

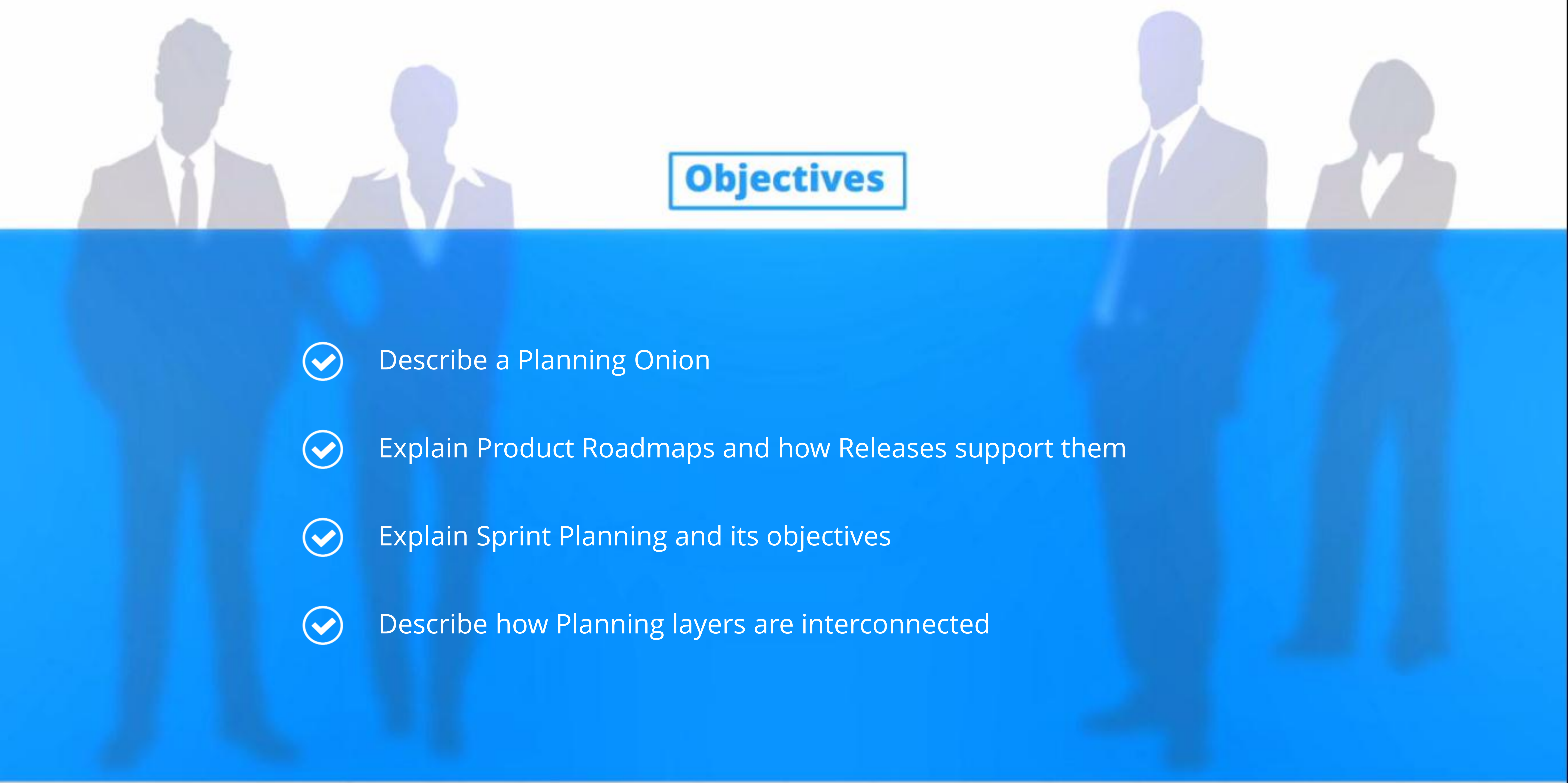


This concludes “Scrum Practices.”

The Next Lesson is “Scrum Planning.”

Agile Scrum Foundation

Lesson 4—Scrum Planning

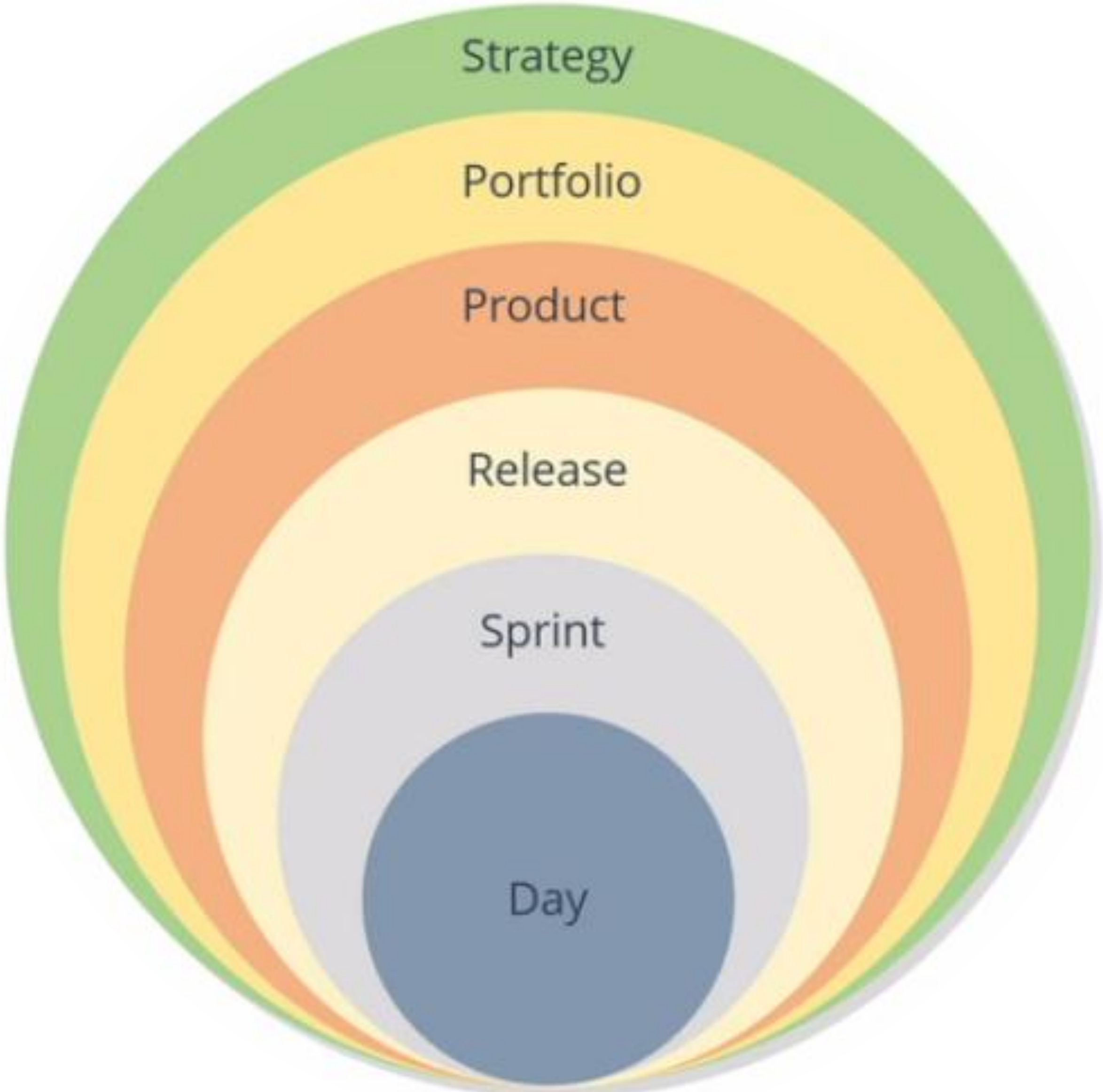


Objectives

- ✓ Describe a Planning Onion
- ✓ Explain Product Roadmaps and how Releases support them
- ✓ Explain Sprint Planning and its objectives
- ✓ Describe how Planning layers are interconnected

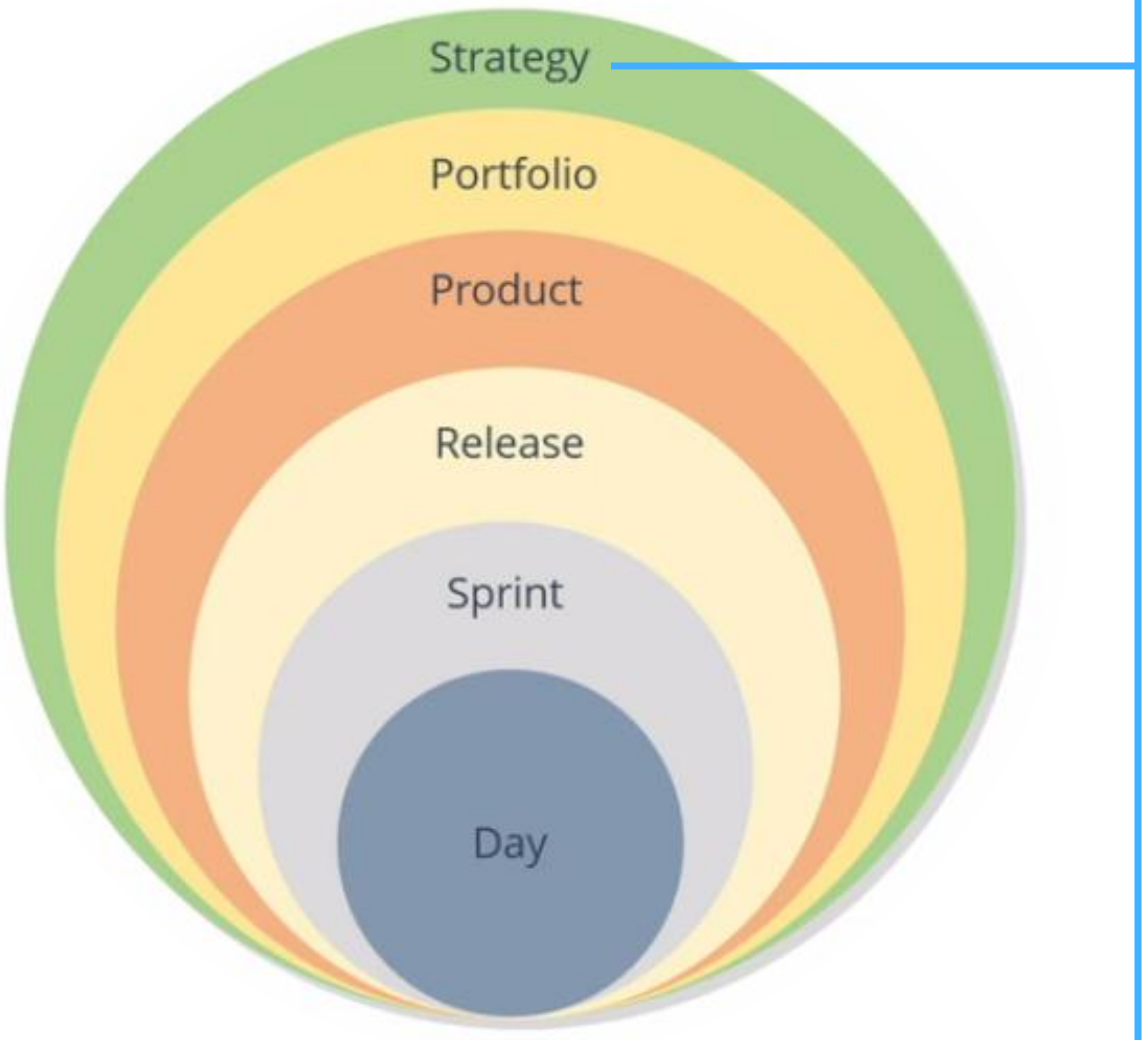
The Planning Onion

Scrum projects are planned at multiple levels, and this is represented through a planning onion.



Each layer drives the planning of the layers below.

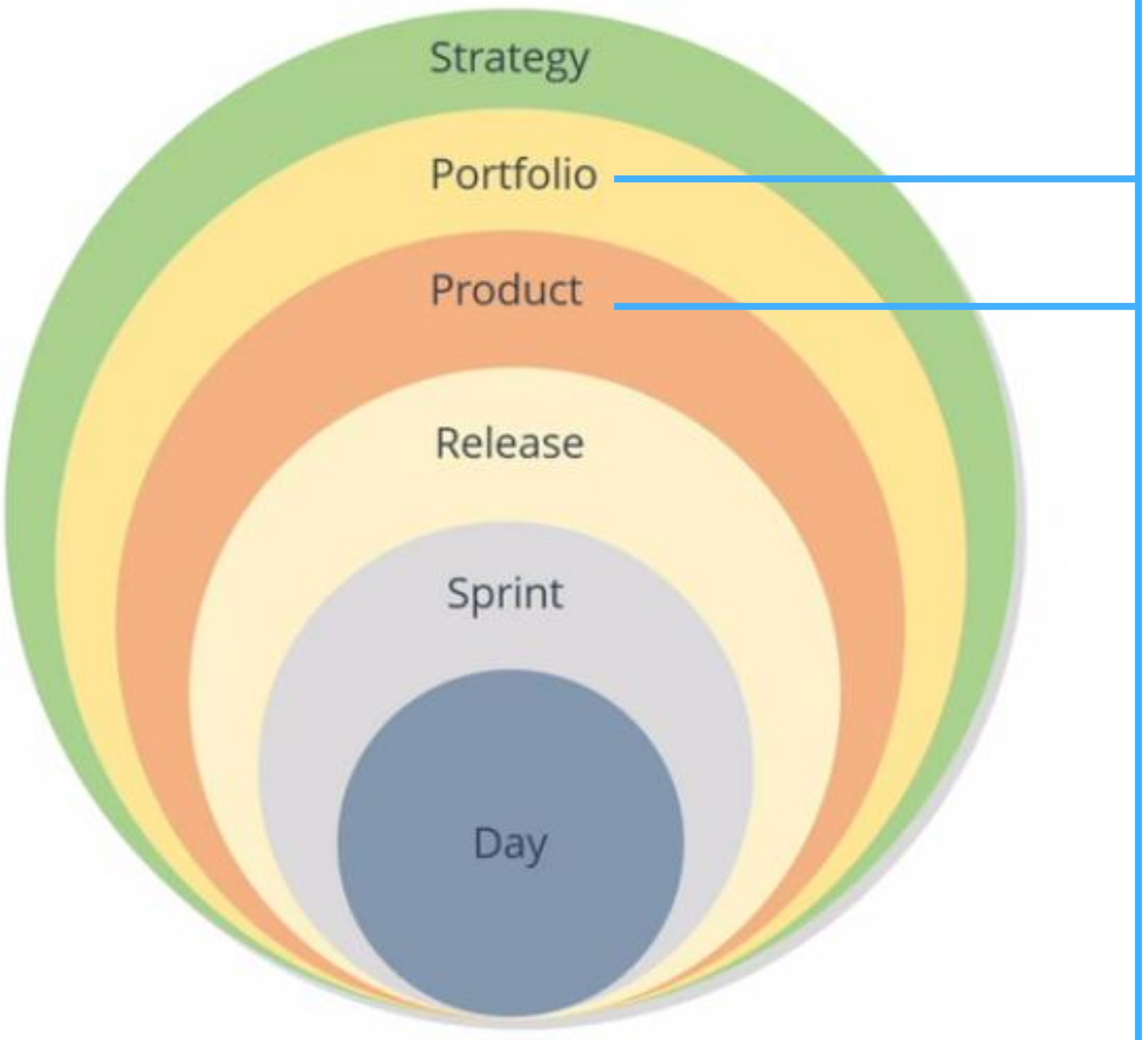
The Planning Onion (Contd.)



Strategic Level

- The executive leadership of the company defines and governs the execution of the strategic goals.
- Companies provide a plan for 5 years or longer and share their strategic vision and objectives with the key management; these are then passed down the chain.

The Planning Onion (Contd.)



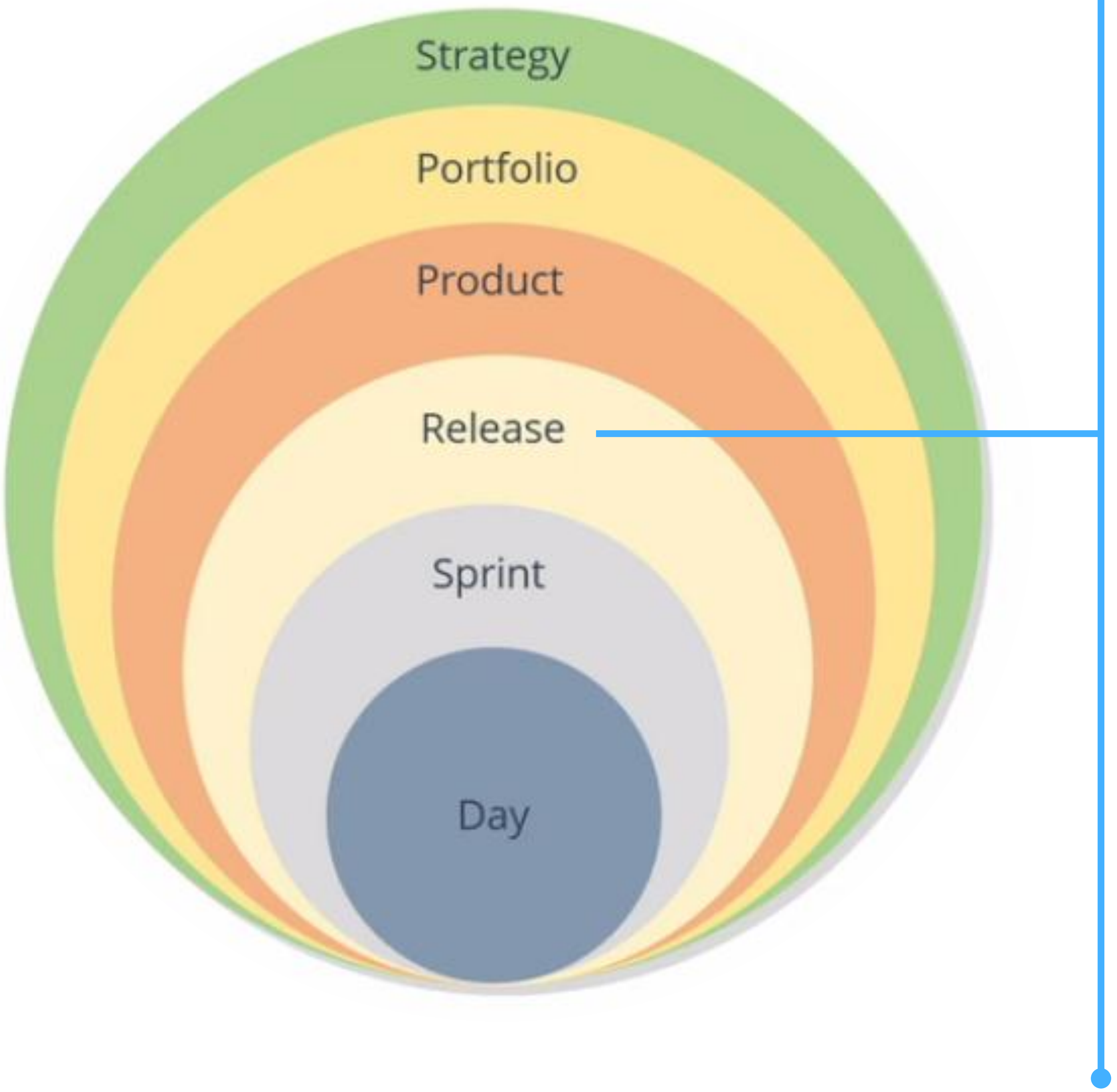
Portfolio Level

- The overall product offerings that will best implement the vision established at the strategic level

Product Level

- Each Scrum team sets a product vision and outlines product roadmaps for their various projects.
- The planning horizon is typically about 12 months.

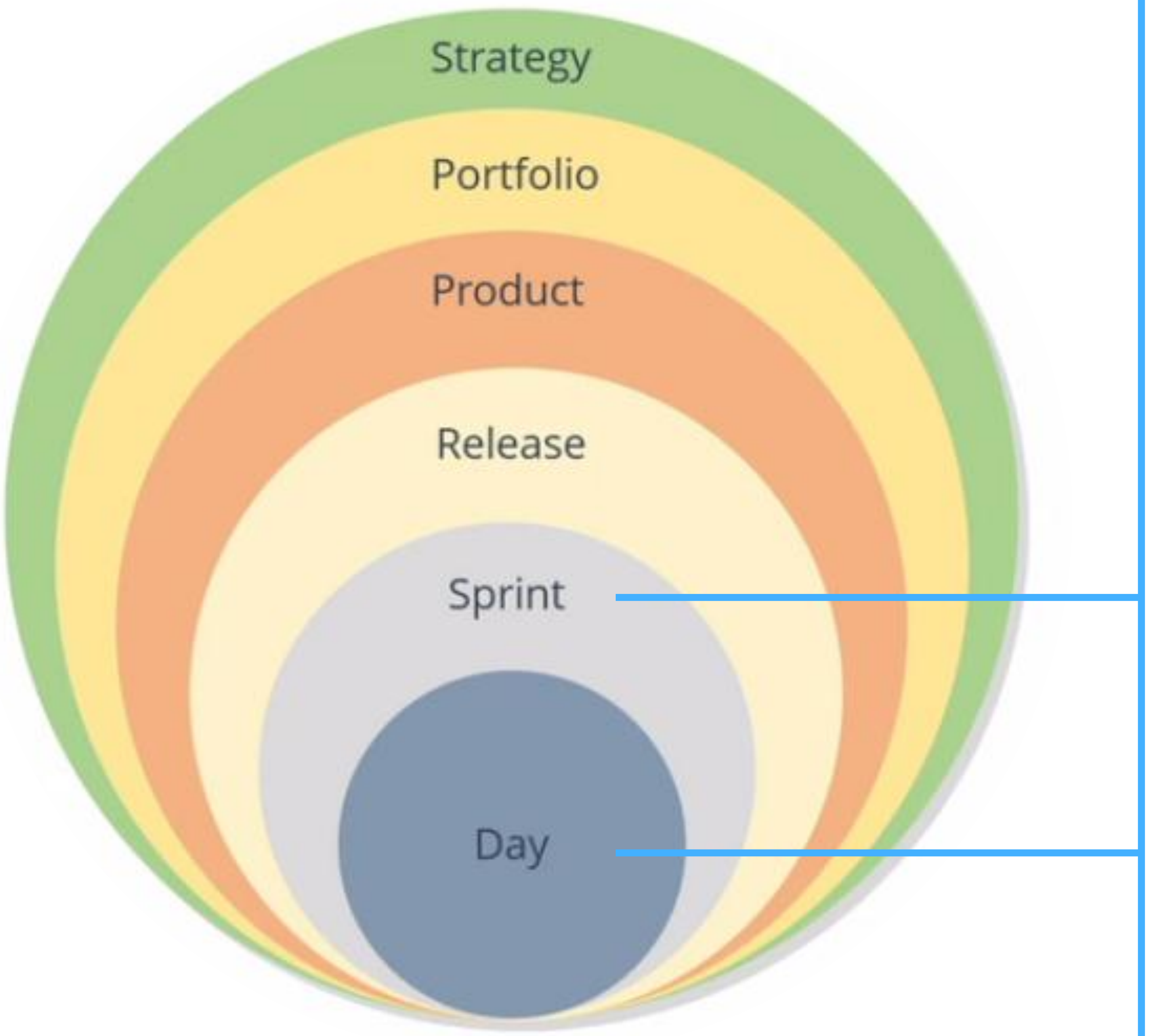
The Planning Onion (Contd.)



Release Level

- The Scrum team groups Product Backlog items into smaller Releases that drive toward the product vision.
- A time-boxed period for a Release is typically three to six months.

The Planning Onion (Contd.)



Sprint Level

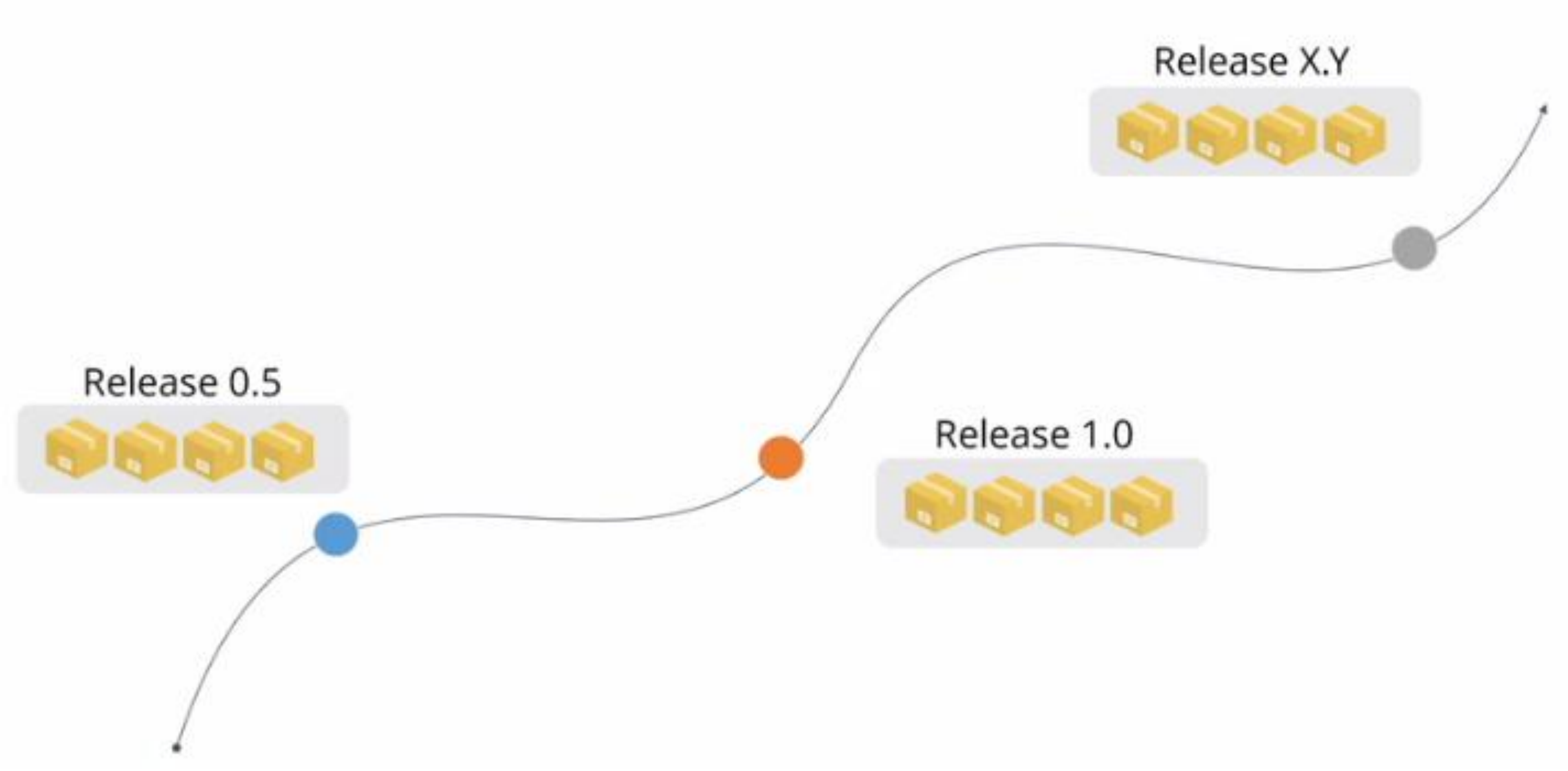
- The Scrum team determines the user stories that can be completed during the Sprint.
- A time-boxed period for a Sprint is typically two to four weeks.

Day Level

- The Scrum Team meets every day for a status update and makes a plan of action for the next 24 hours.

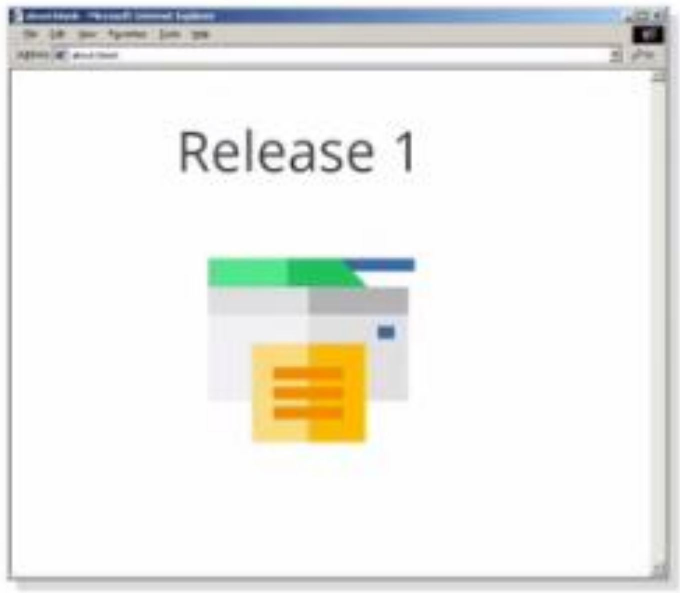
Release Level of Planning

Most Scrum projects are broken down into releases that match the Product Roadmap.

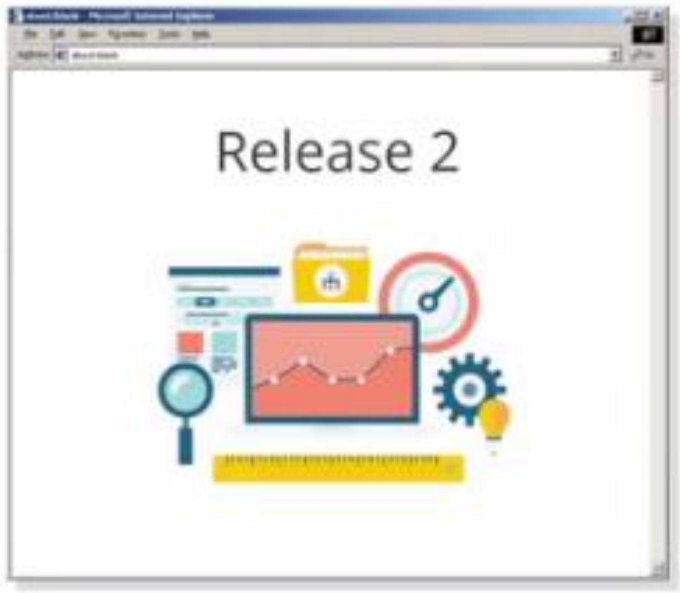


Release Level of Planning—Example

Consider a real life project that has three separate versions of a website.



Free version that is available to everybody

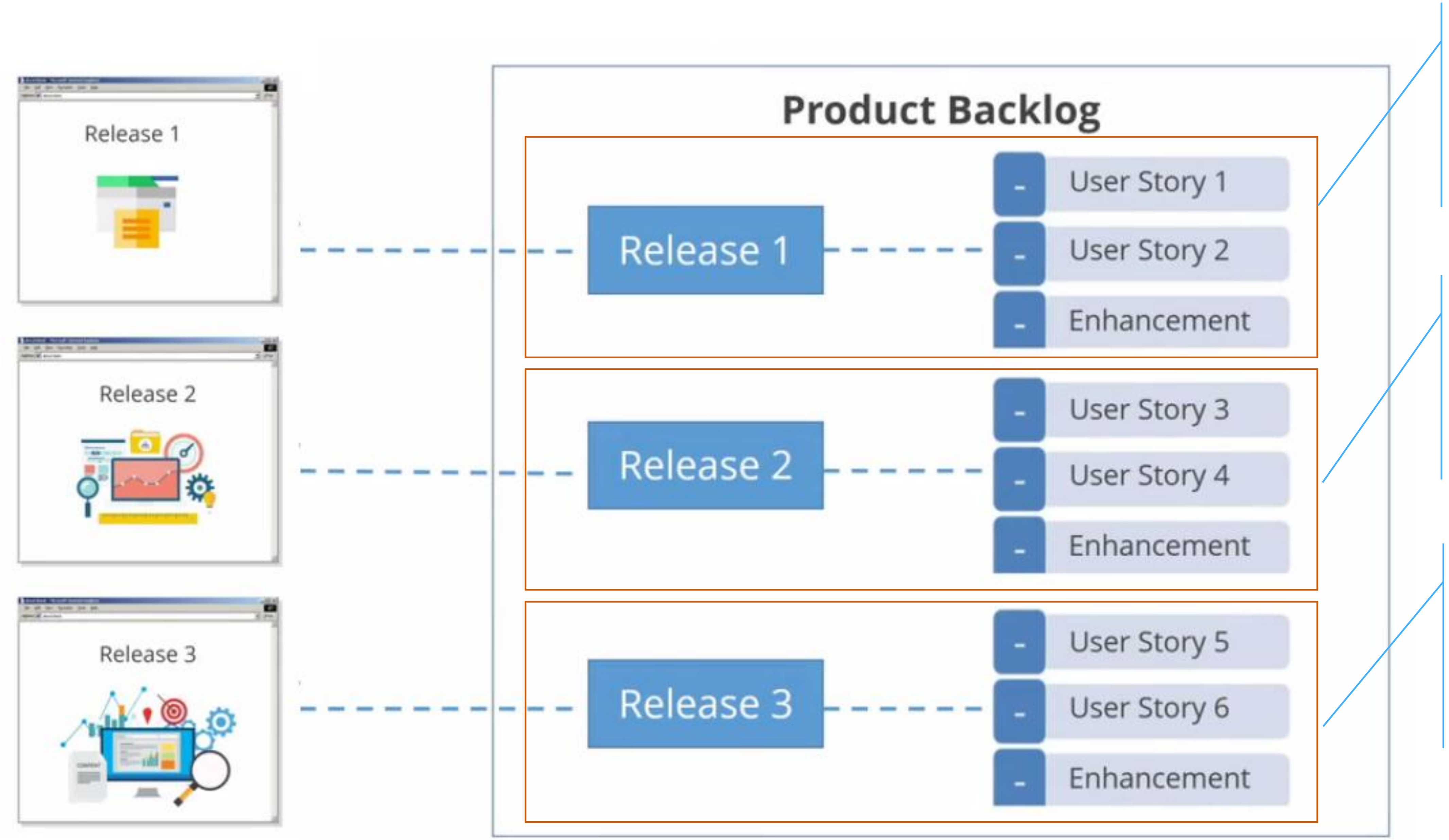


Version available to members only

Version available at a premium level of membership

Release Level of Planning—Example (Contd.)

The product backlog for this project is grouped into three separate releases.



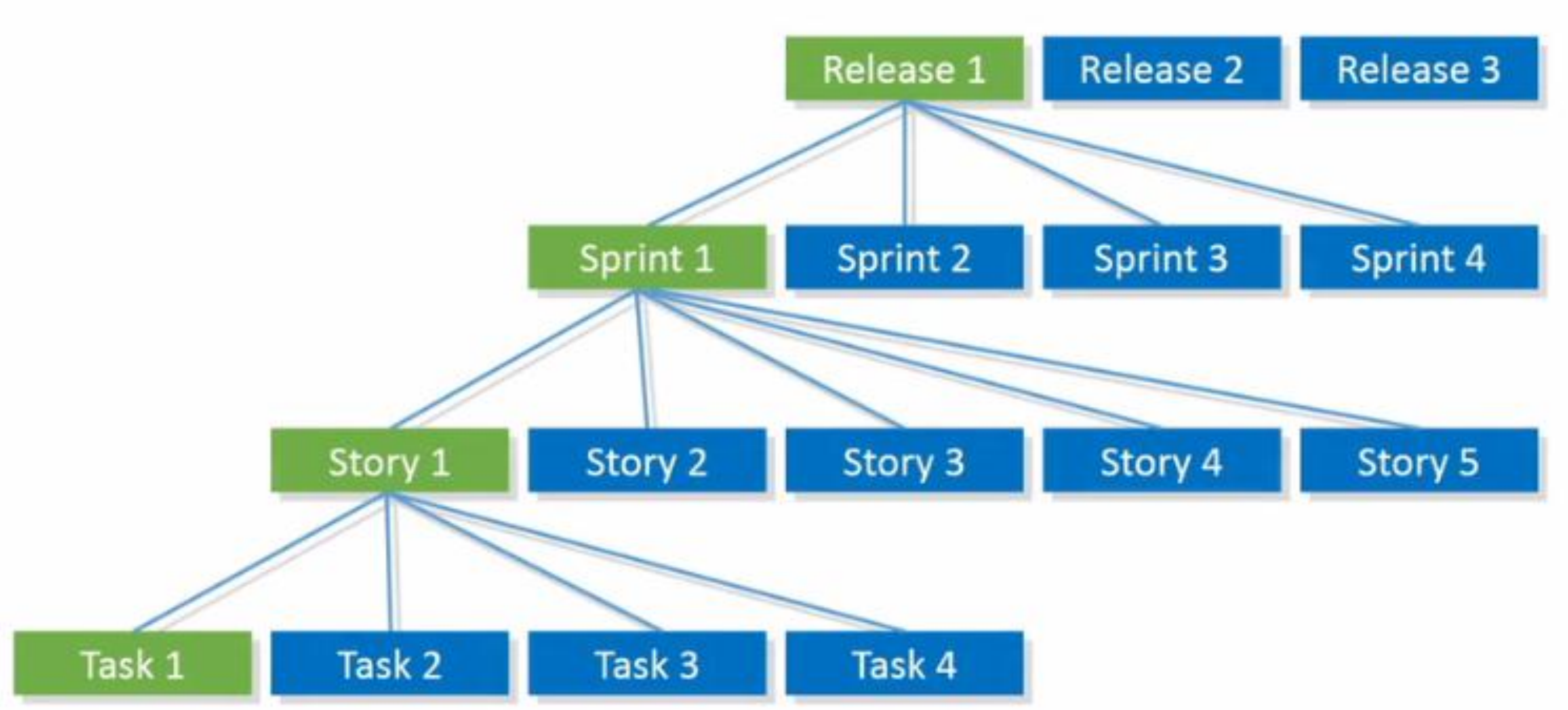
Includes all of the user stories necessary for version 1 to be released.

Includes all of the user stories necessary for version 2 to be released.

It is implemented by completing all user stories that are included in release 3.

Release Level of Planning—Example (Contd.)

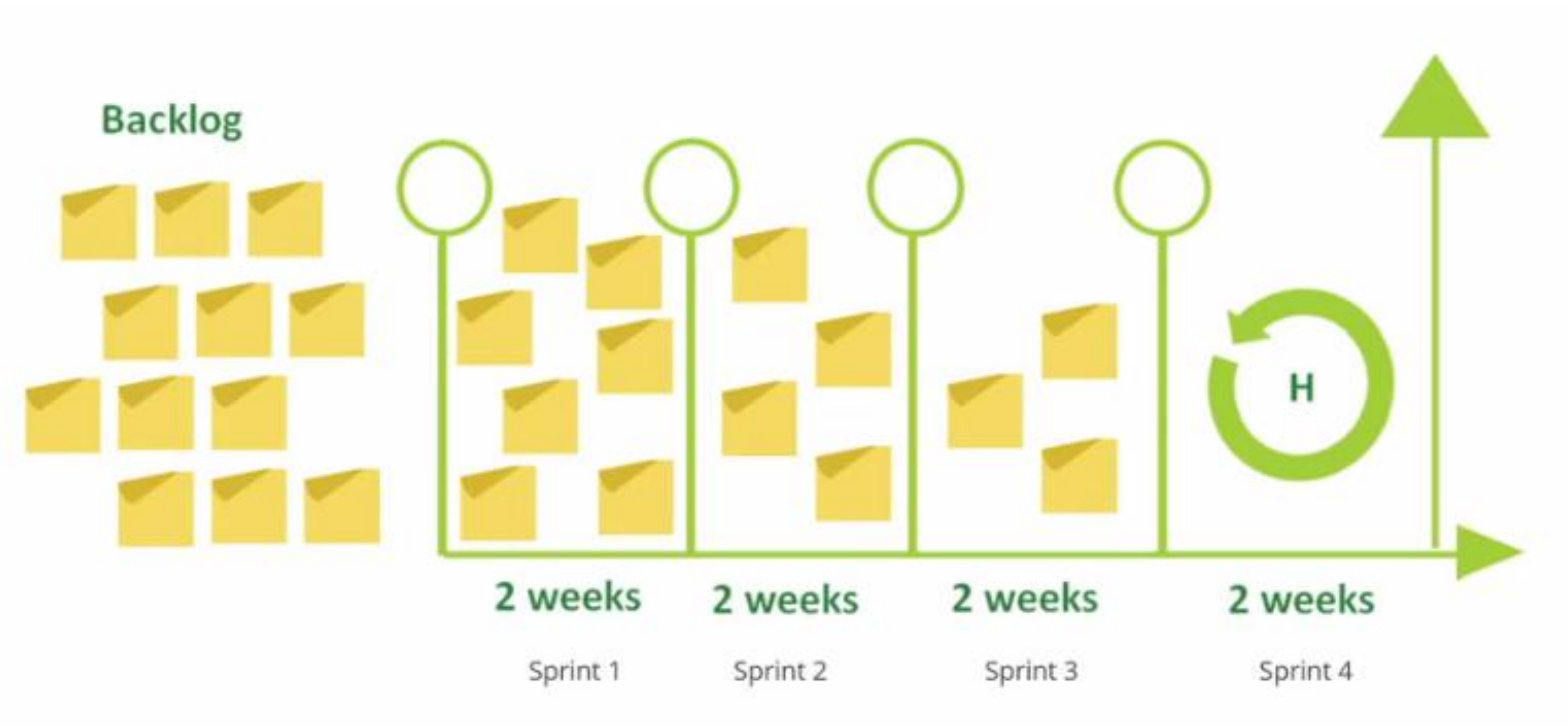
The graphic illustrates the Product Roadmap.



The duration of releases can vary depending on the complexity of the project and the granularity of the product roadmap.

Release Plan

Release planning can have a horizon of up to nine months, but most commonly, it lasts about three months.



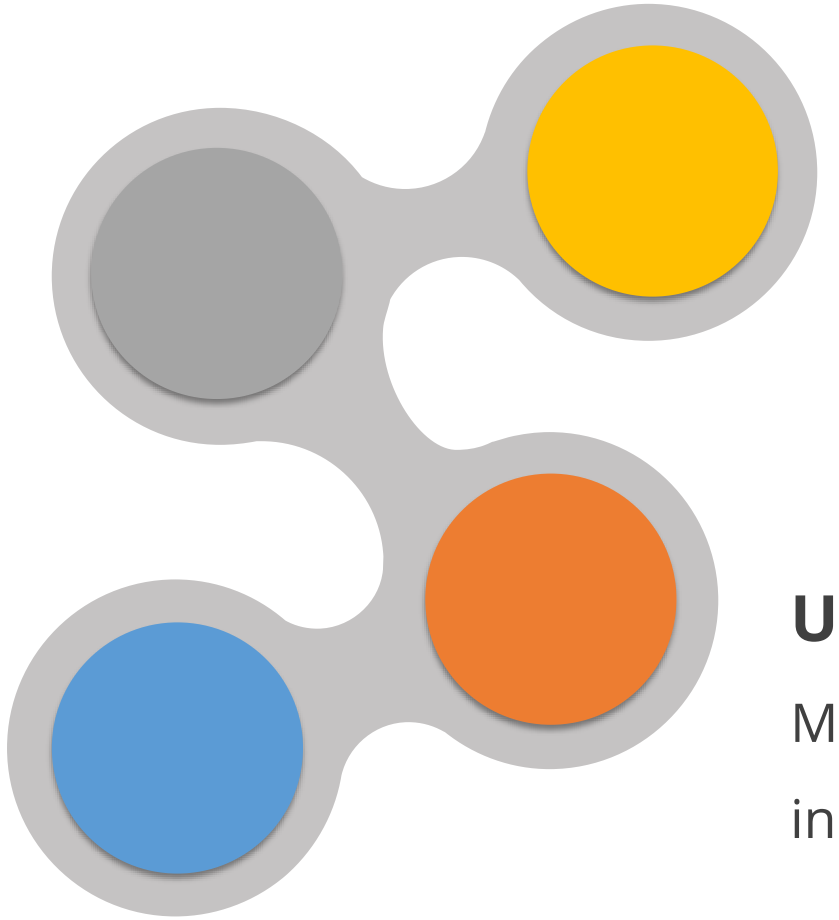
A release plan consists of:

- Goal
- Target date
- Prioritized list of user stories

The graphic illustrates how release planning is done.

Scrum Projects

Scrum projects support the vision and goals established at the Portfolio level and can extend for years. They are accomplished through:



Releases

Used to support product roadmaps

Themes and Epics

Used to support long-term visions and portfolio

Sprints

Support releases

User Stories

Make up the content of an individual sprint

Scrum Projects (Contd.)

Multi-Level Planning

Small Projects	Large Projects
Three to six Sprints	Greater than six Sprints
Six to twelve weeks	More than six months
Single team	Several teams
Story-level planning • Release • Sprint	Plan at several levels: • Business area • Theme/Epic • Features

Note: If the project is enterprise-sized, each portion of the release can be done by separate scrum teams. Each of them could be running sprints to complete their portion of the release, and their work is coordinated with scrum-of-scrums.

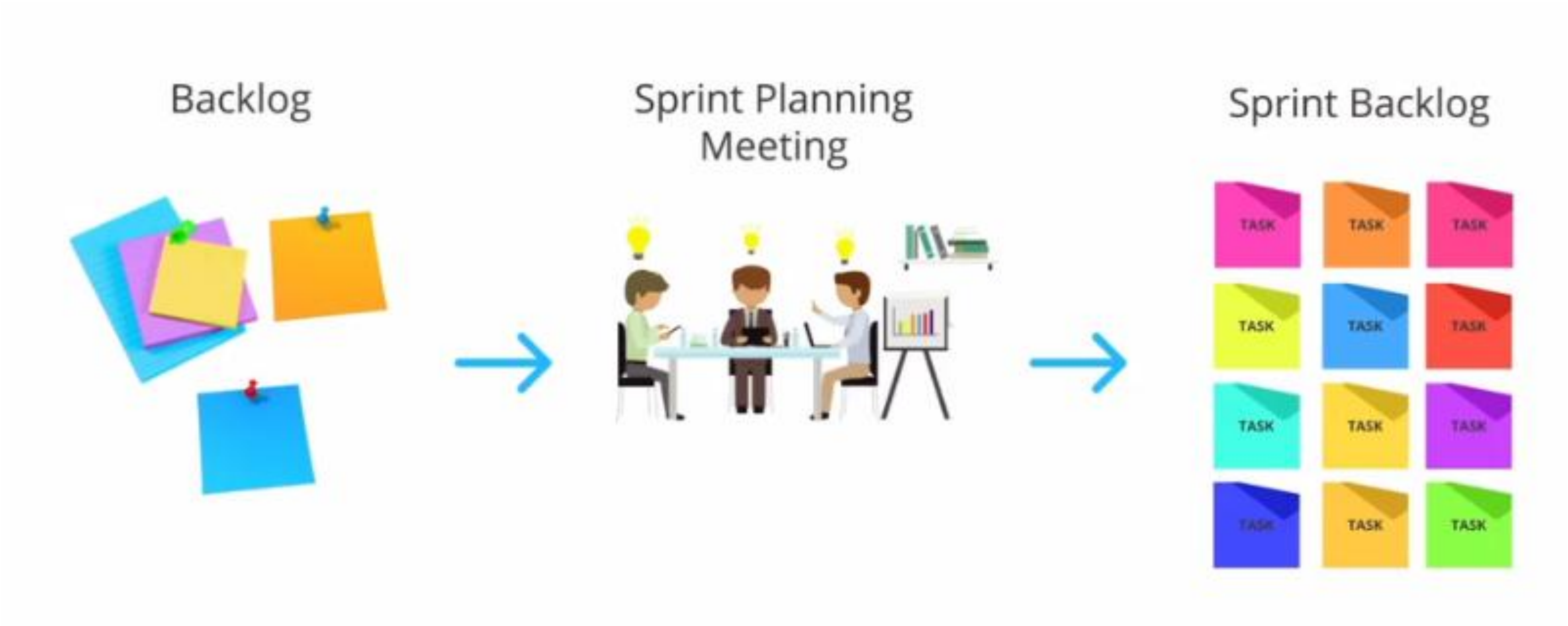
Scrum Projects (Contd.)

Scrum-of-Scrums

One or two representatives from a team participate in a daily meeting with the representatives from other teams working on the Release.

Sprint Planning Meeting

It determines the work that is going to be performed during the Sprint and involves the Scrum Team and, sometimes, key stakeholders.

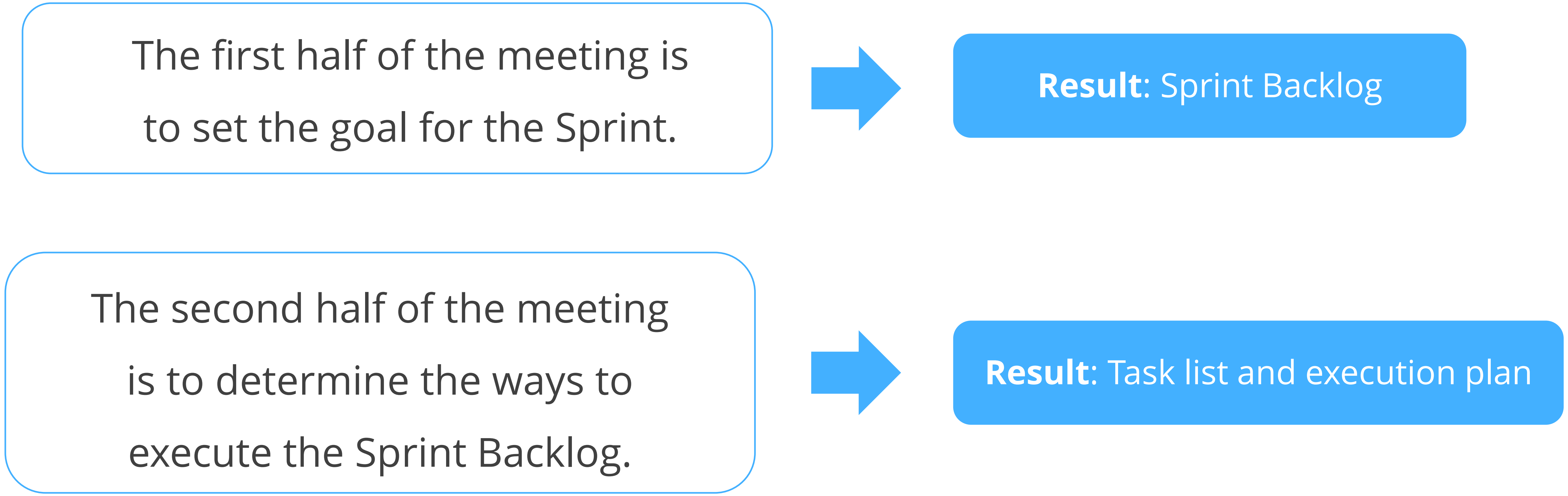


It is time-boxed event scheduled to last two hours for each week of the Sprint’s duration.

Sprint Planning Meeting (Contd.)

If you are performing a four week sprint, the sprint planning meeting would be for eight hours.

The sprint planning meeting is divided into two separate segments.



The first half of the meeting is to set the goal for the Sprint.

Result: Sprint Backlog

The second half of the meeting is to determine the ways to execute the Sprint Backlog.

Result: Task list and execution plan

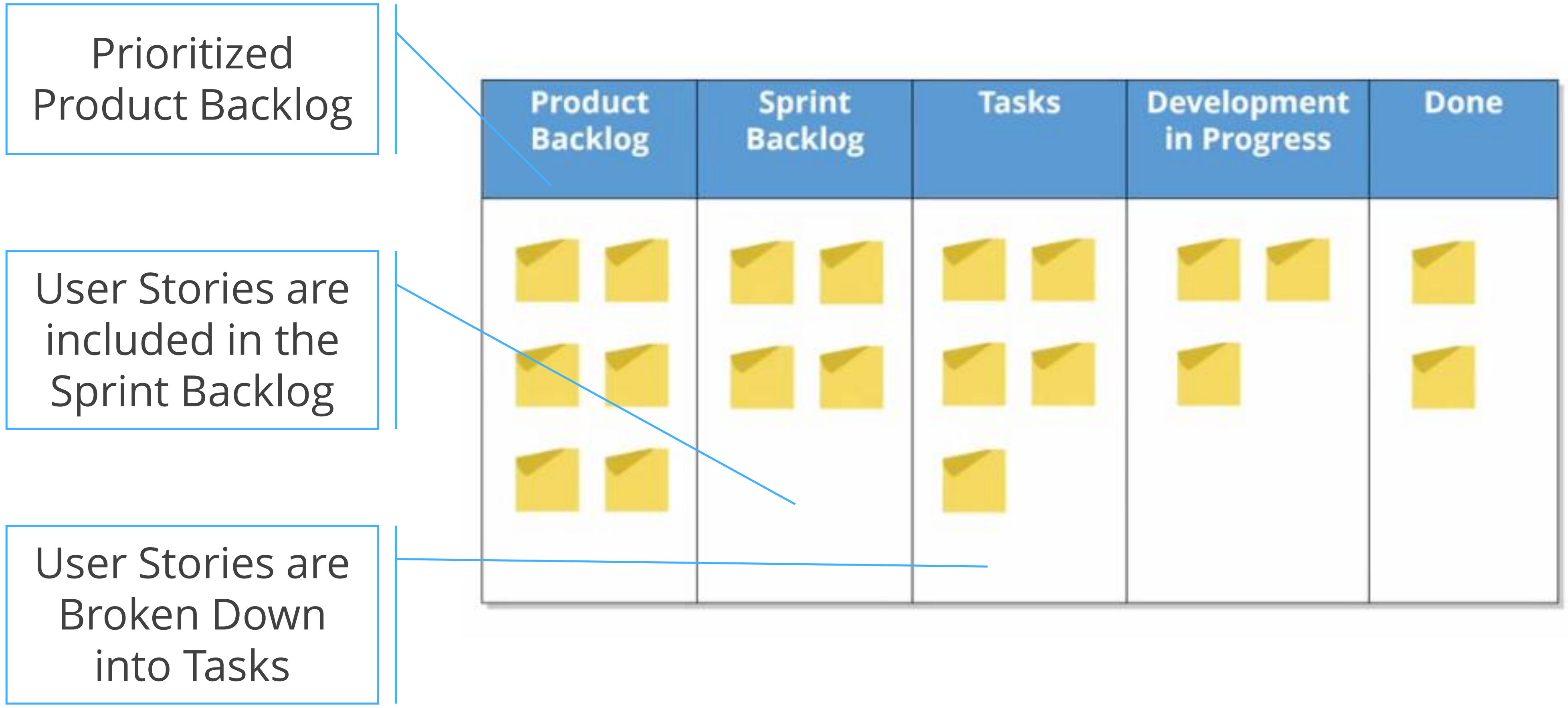
Sprint Planning Meeting (Contd.)

Before performing the sprint planning meeting, you need to look at:

- Product Backlog
- Team Capacity based on Past Performance
- Business Conditions
- Technology Stability
- Most Recent Product Increment
- Current Status of the Project

Sprint Planning Meeting (Contd.)

The graphic illustrates a simple Kanban board.



Prioritized Product Backlog

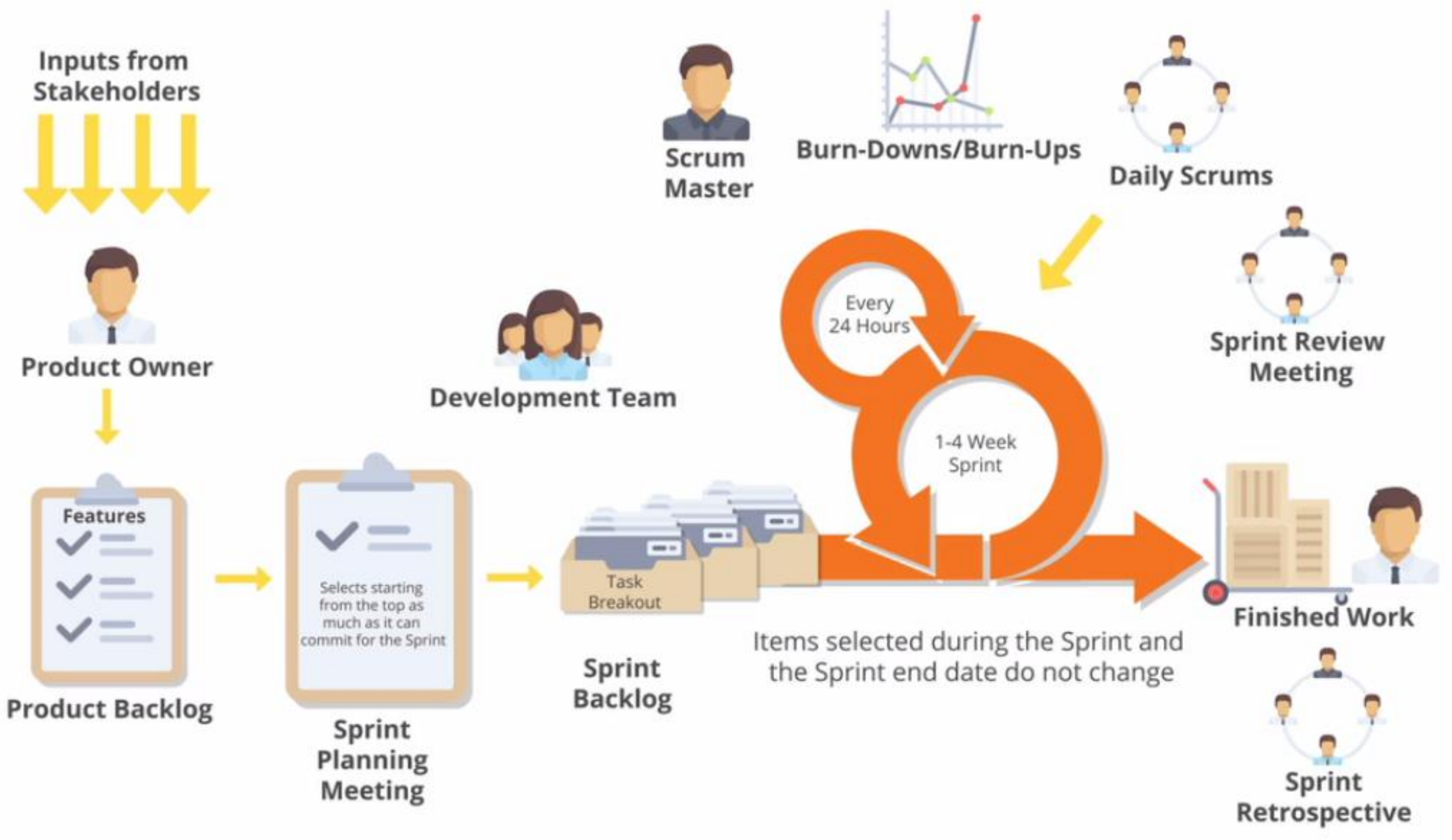
User Stories are included in the Sprint Backlog

User Stories are Broken Down into Tasks

- The product backlog is the master list of user stories of the project
- The sprint backlog includes the user stories to be completed during the current sprint.
- Tasks are the individual user stories
- Development in progress refers to the tasks that have to be completed
- Done refers to the tasks that are completed.

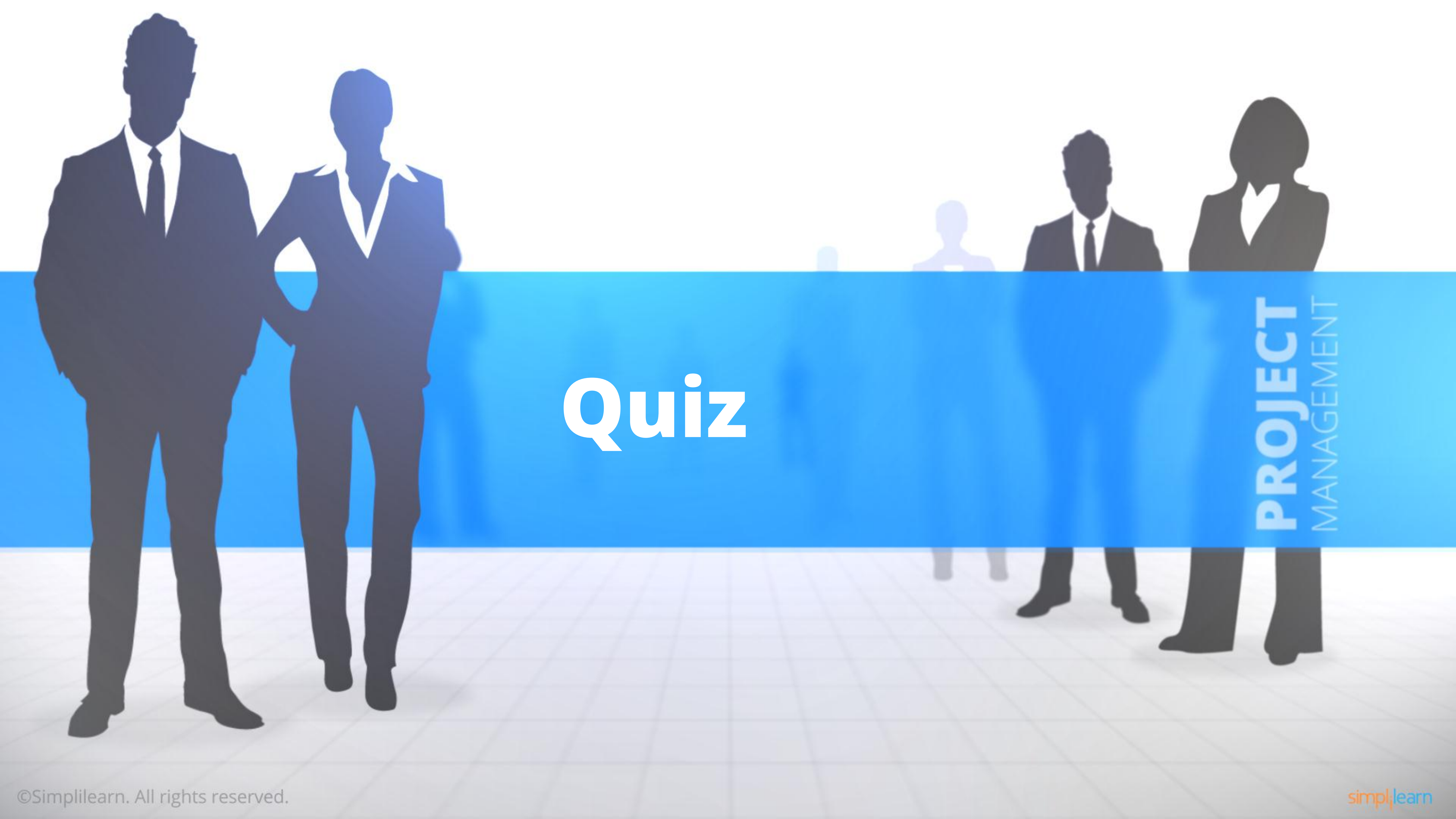
Note: This is a simple illustration. The Kanban board can be much more detailed.

Sprint Planning Meeting (Contd.)



Topics Covered

- Planning Layers of the Planning Onion and their functions
- How releases support Product Roadmaps
- How sprint supports Releases
- Sprint Planning and Sprint Backlog
- User Stories and tasks
- Sprint tracking using a Kanban Board



Quiz

QUIZ

1

Agile planning happens at multiple levels. Which of the following terms best describes the multi-level planning?

a. Planning Onion

b. Sprint Planning layers

c. Strategic Level Planning

d. Release Planning layers

QUIZ

2

During a Release planning, the Product Backlog items are grouped into smaller releases that drive toward the product roadmap. Who plans a Release?

a. Product Owner

b. Stakeholders

c. Product Owner and Scrum Master

d. Scrum Team

QUIZ

3

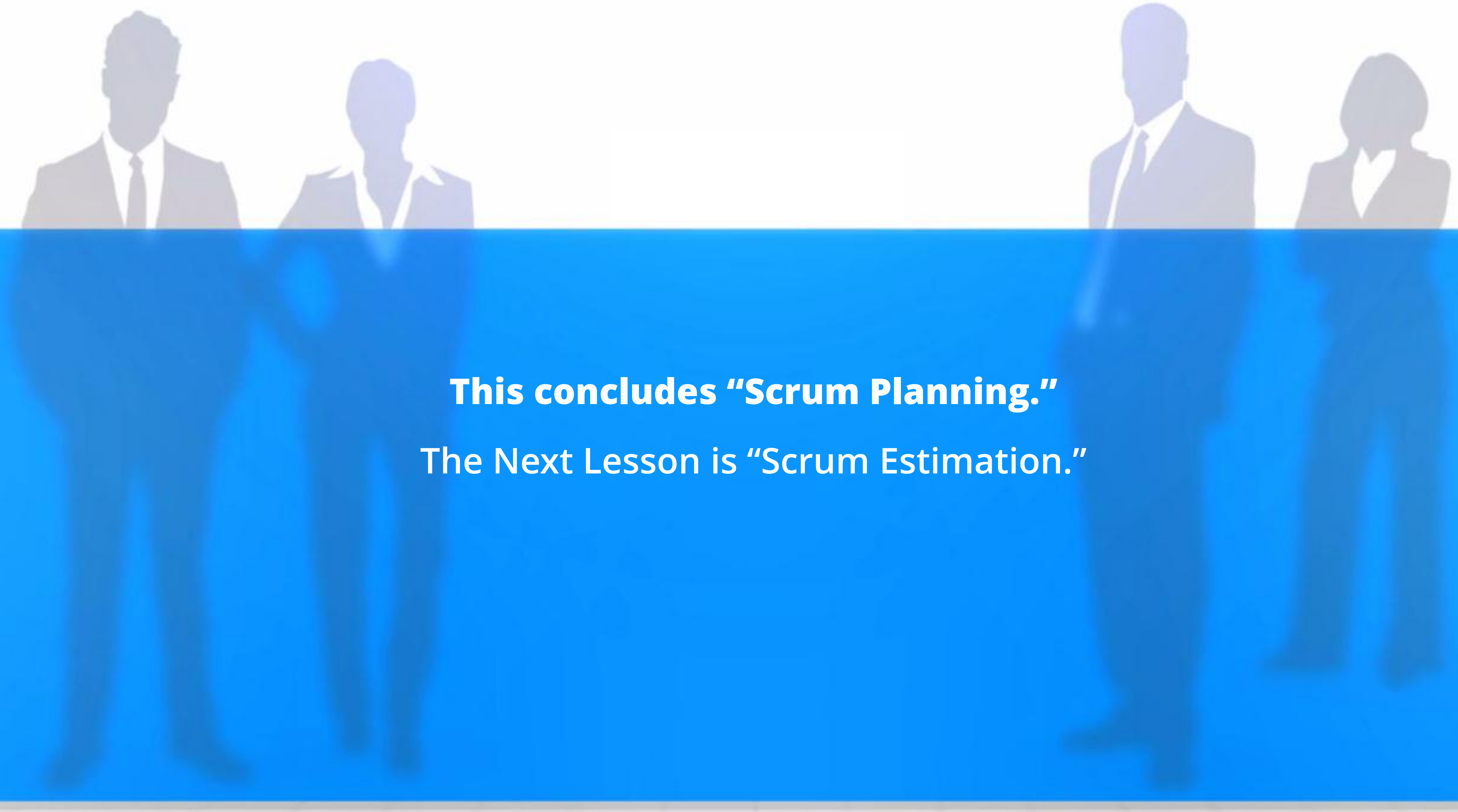
What is a typical time-boxed period for Release Planning?

a. As long as it takes

b. 2 to 4 months

c. 2 to 4 weeks

d. 3 to 9 months



This concludes “Scrum Planning.”

The Next Lesson is “Scrum Estimation.”

Agile Scrum Foundation

Lesson 5—Scrum Estimation

Objectives

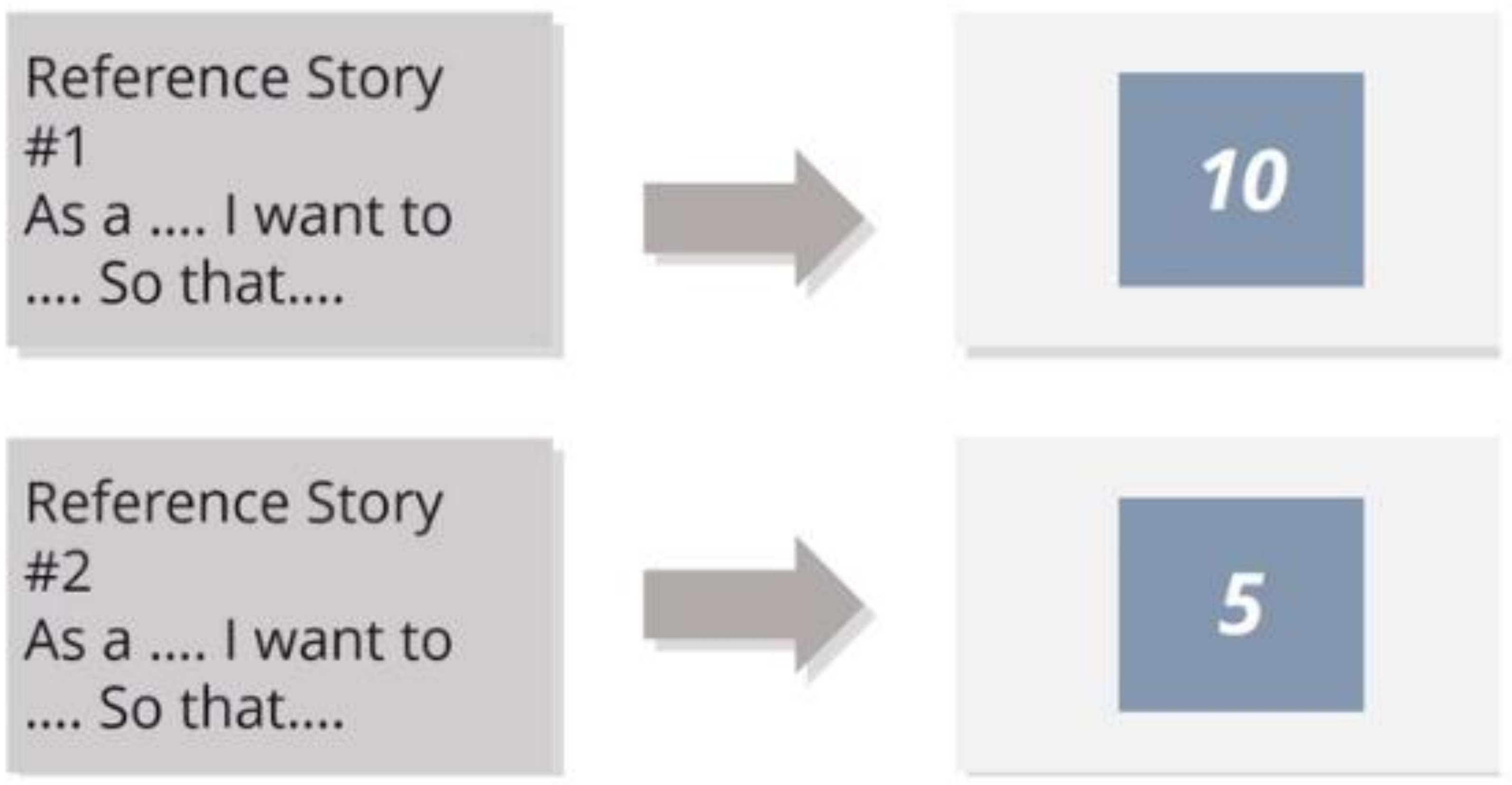
- Describe the Velocity of a Team
- Explain Agile Estimation
- Describe Planning Poker
- Distinguish between Story Points and Ideal Time

Story Points

In Agile software development, most estimations are not done in terms of lapse time.

User stories are estimated by size, which is referred to as **Story Points**.

Example



Reference Story #1
As a …. I want to …. So that….

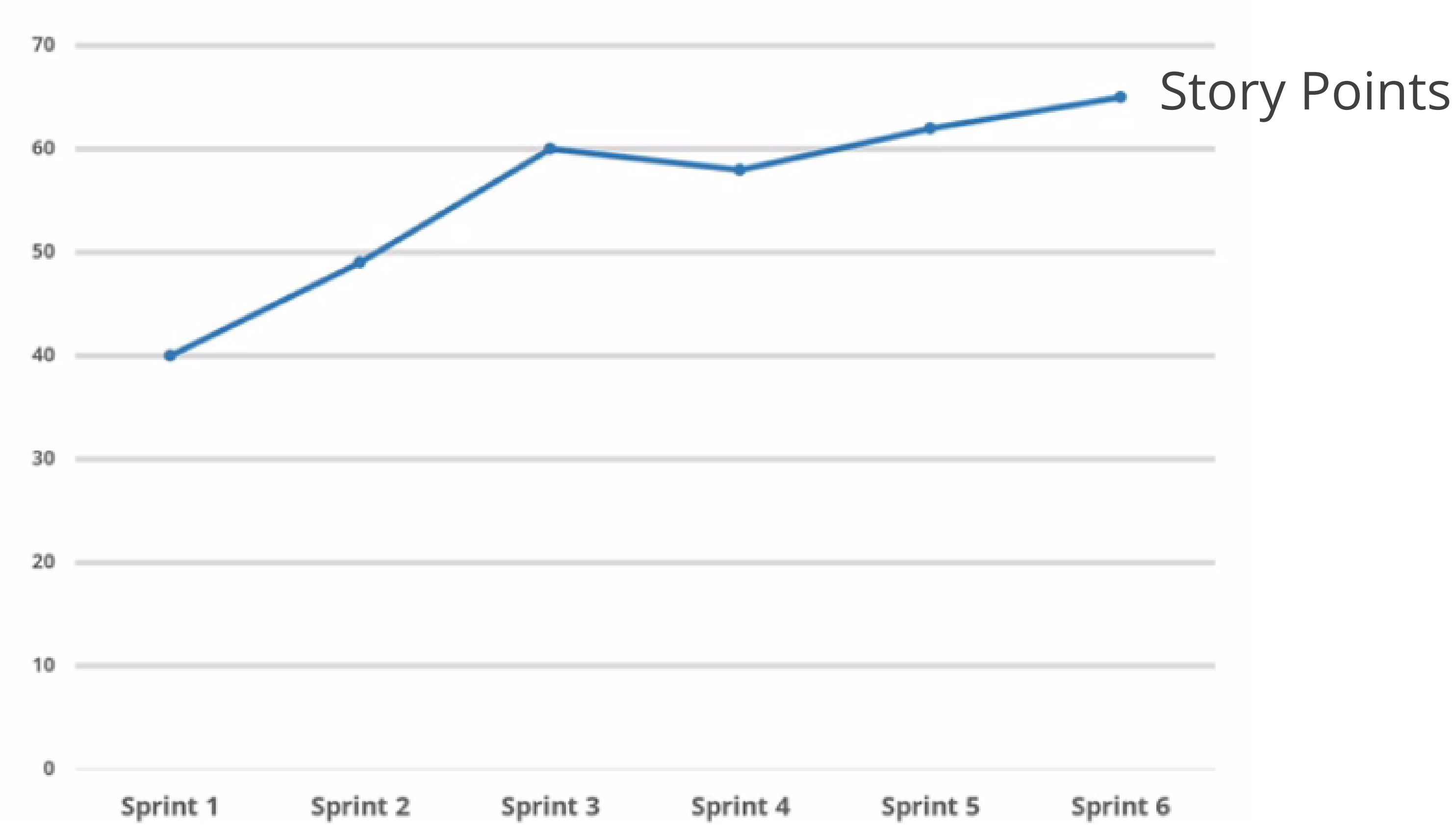
10

Reference Story #2
As a …. I want to …. So that….

5

Velocity of the Team

Velocity refers to the Development Team's ability to complete story points in a Sprint.



Velocity is not a prediction of how much a team can do during a sprint.

It is an observation based on historical data of the team.

Velocity of the Team—Example

The development Team has completed 136 Story Points in the past 12 months.

Velocity of the Team = 136 Story Points/12 Months

= 11.33 point per Sprint, rounded off to 12

Agile Estimation Techniques

There are two basic ways of estimation in Agile.

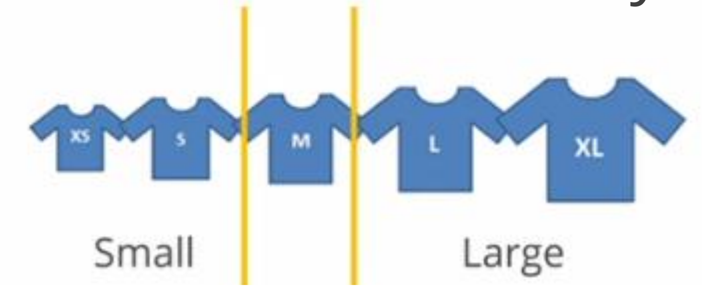
Blind Estimation

It requies background work and involves the following steps:

- Estimate Product Backlog
- Decompose Reference Story
- Identify Team Capacity
- Estimate Team velocity

Affinity Estimation

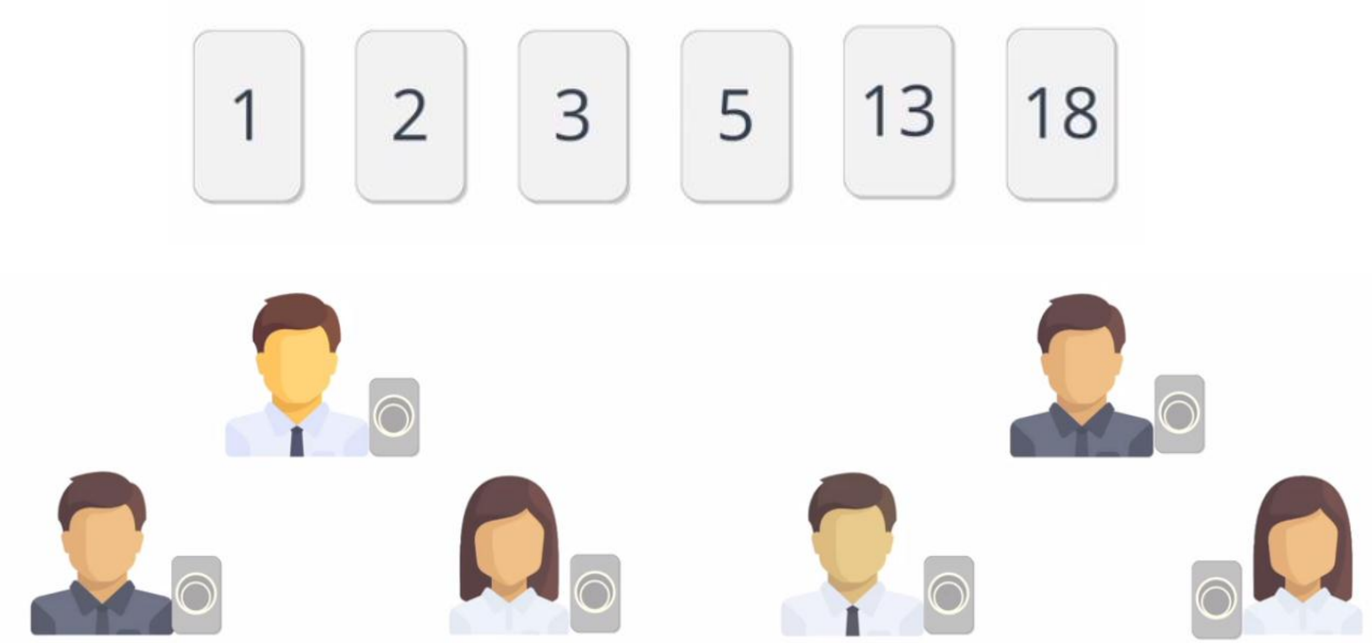
User stories are grouped into relative sizes. This is done by:



- Quickly categorizing user stories
- Applying estimates to categories
- Recategorizing user stories

Planning Poker

Planning Poker is a technique in which the entire team collaboratively estimates the effort involved in completing a user story.



Each member of a team receives a deck of cards numbered 1, 2, 3, 5,13, and 18, which is a modified Fibonacci series.

Planning Poker—Process

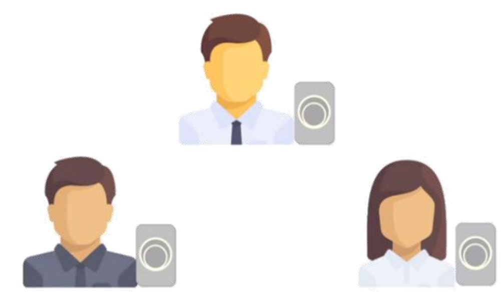
Product Owner

The product owner reads the story card and clarifies any queries the team may have.

Scrum Master

The scrum master requests the team members to display their cards.

Each team member will disclose the number he or she allotted for the story.



Team members who numbered the story extremely low or high will be asked for an explanation.

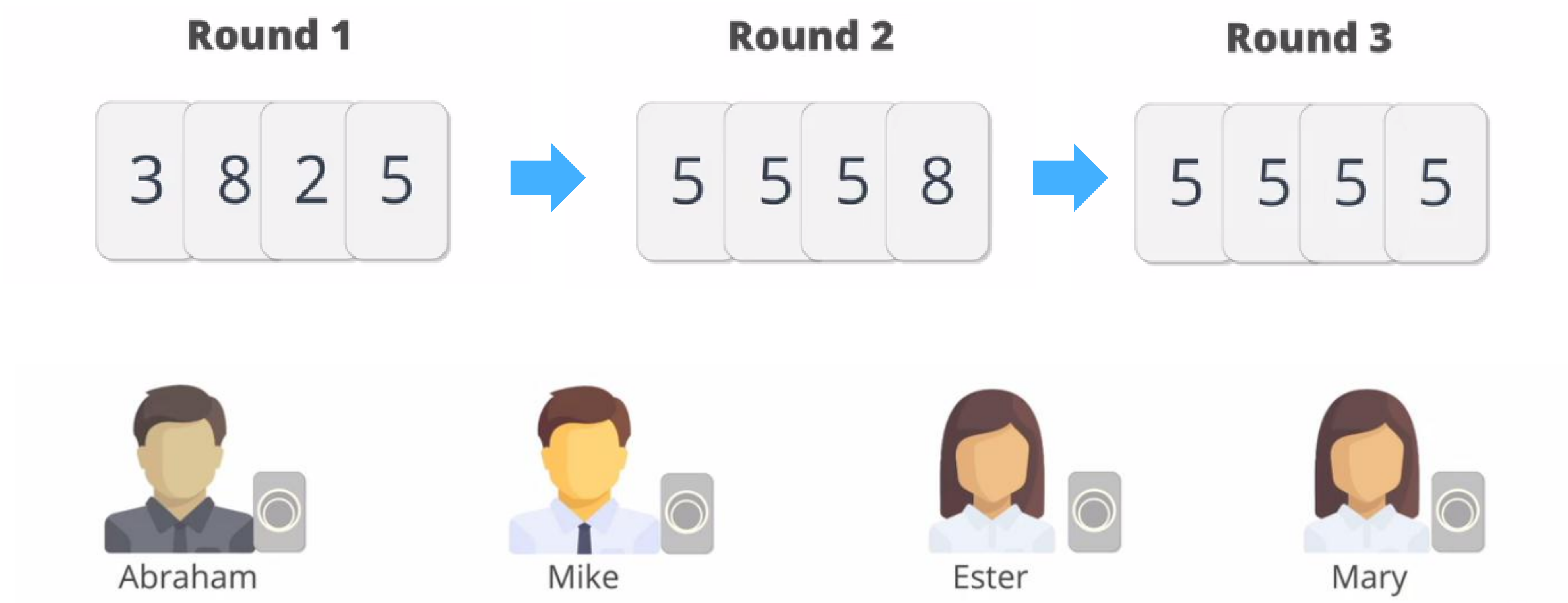
This process continues until the team reaches the consensus.

Planning Poker—Example

Product Owner

"As an insurance agent,
I want to create a quote
so that I can share with customers"

Scrum Master



Round 1

3 8 2 5

Round 2

5 5 5 8

Round 3

5 5 5 5

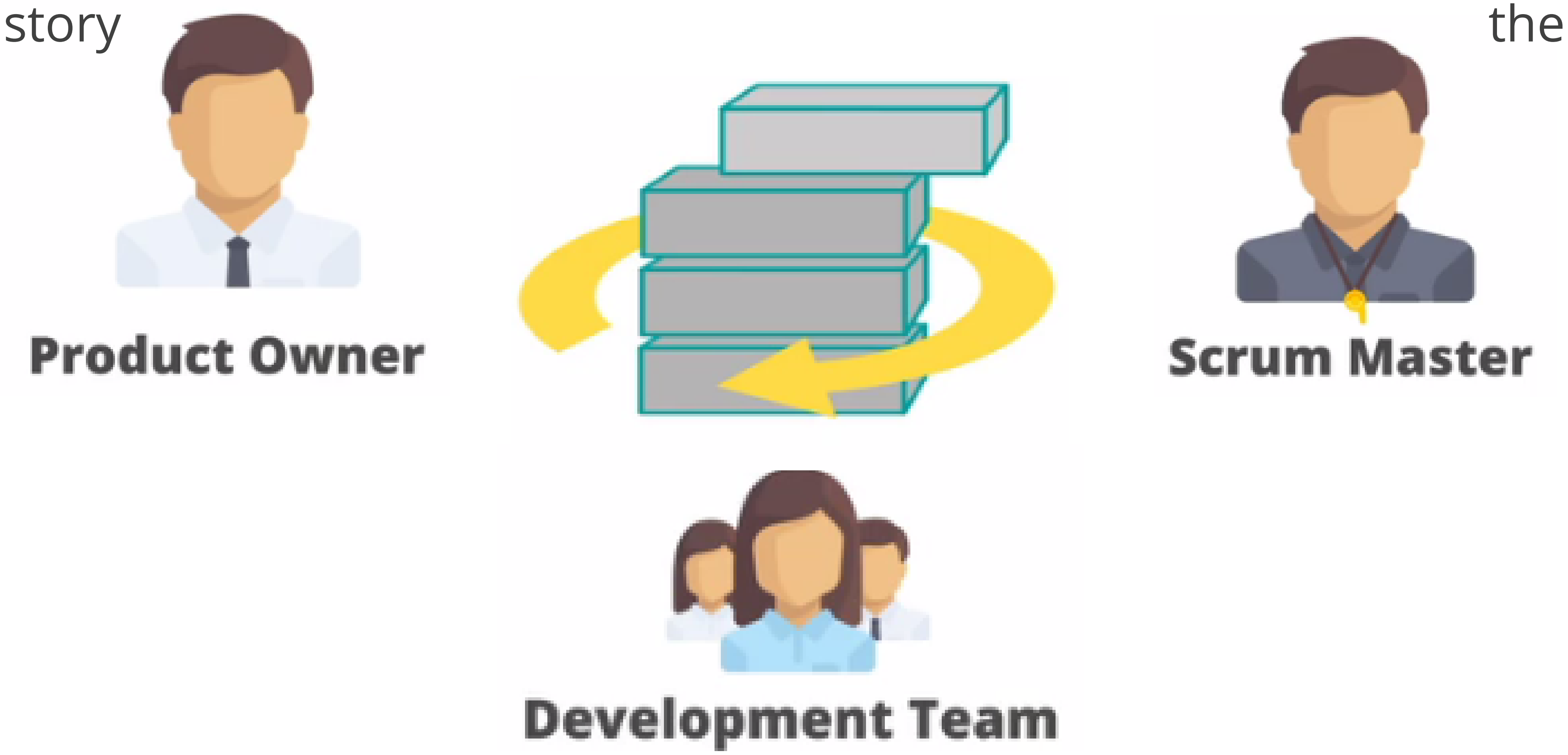
Abraham

Mike

Ester

Mary

Planning Poker—Example (Contd.)



Clarifies and describes the user story

Coaches and facilitates the user story

Estimates collaboratively

Note: The team should periodically revisit their estimates and revise them as the project progresses.

Story Point Vs. Ideal Time

Estimating using Story points is a common and preferred method. Another approach that can be used for Agile estimating is Ideal Time.

Story Points

- They help drive cross-functional behavior.
- Estimates do not decay.
- They are a pure measure of size.
- Time required for story point estimation is low.

Vs.

Ideal Time

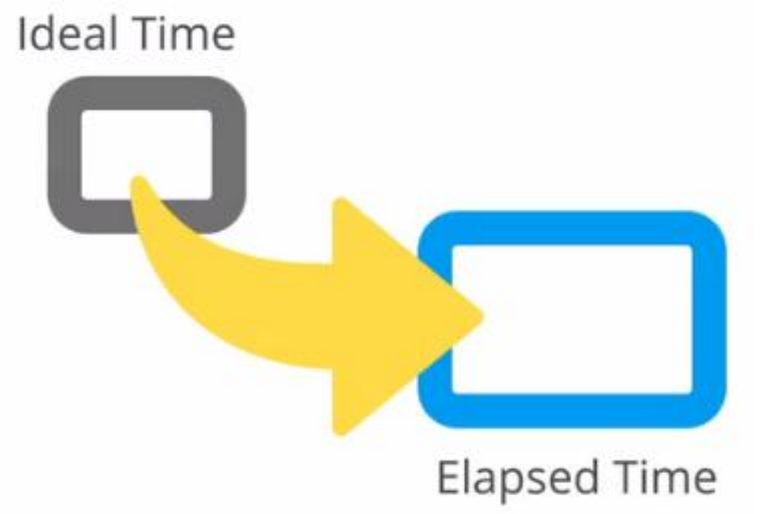
- This may differ between members of even the same team.
- It is easier to explain to people outside the team as story points are more abstract.
- It is easier to estimate, but takes longer.
- It can compel companies to confront time wasting activities.

Ideal Time

For Ideal Time estimation, answer the question

"How long would it take to implement a story, given that:

- Focus is on the task at hand without any interruptions
- Everything needed is available."



Eventually, every Ideal Time estimation will have to be converted to elapsed time in order to account for the normal interruptions that occur during the day, such as phone calls, meetings, and so on.

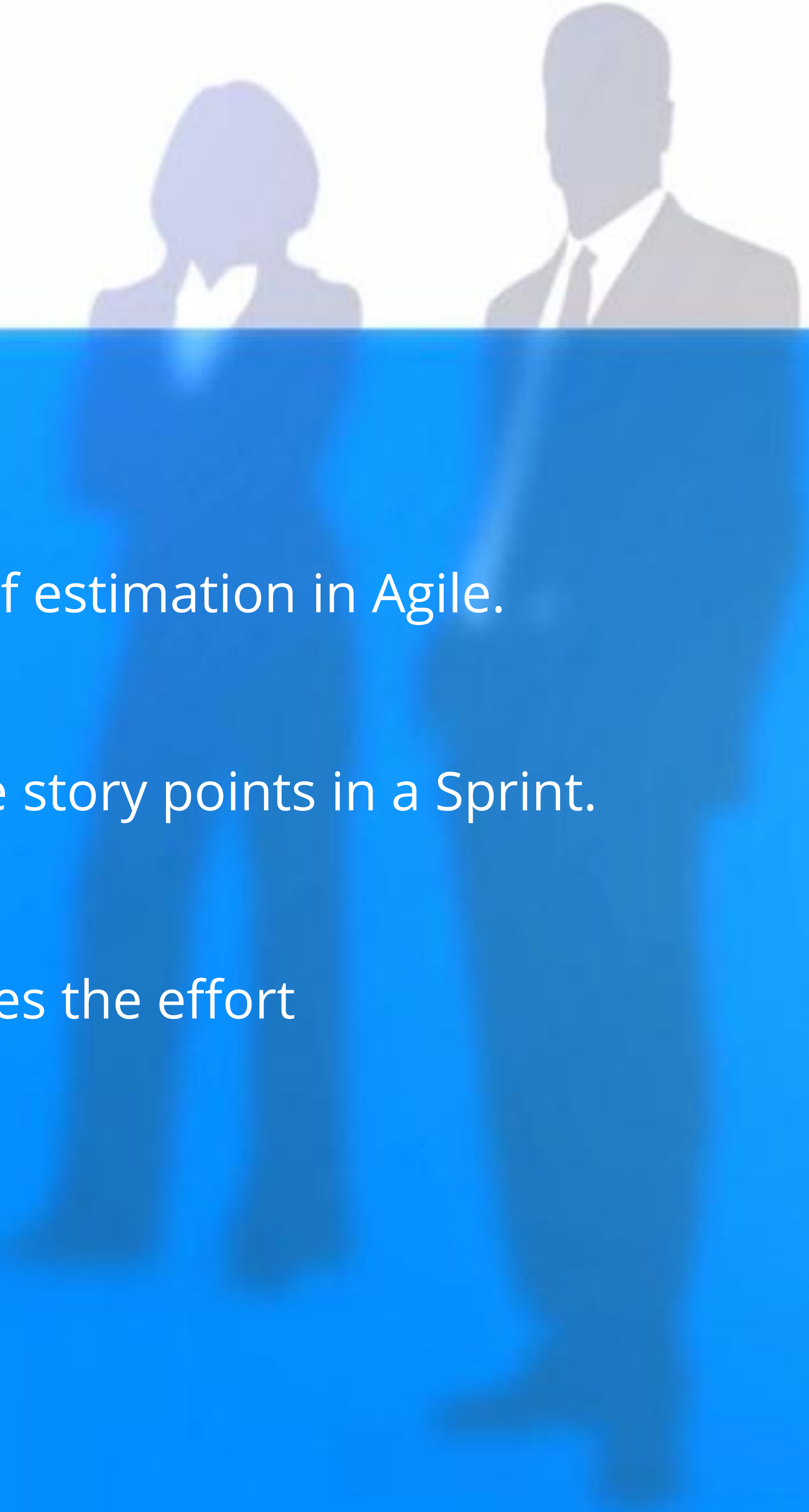
Agile Estimating

Most teams will estimate user stories in the product backlog using Story Points.

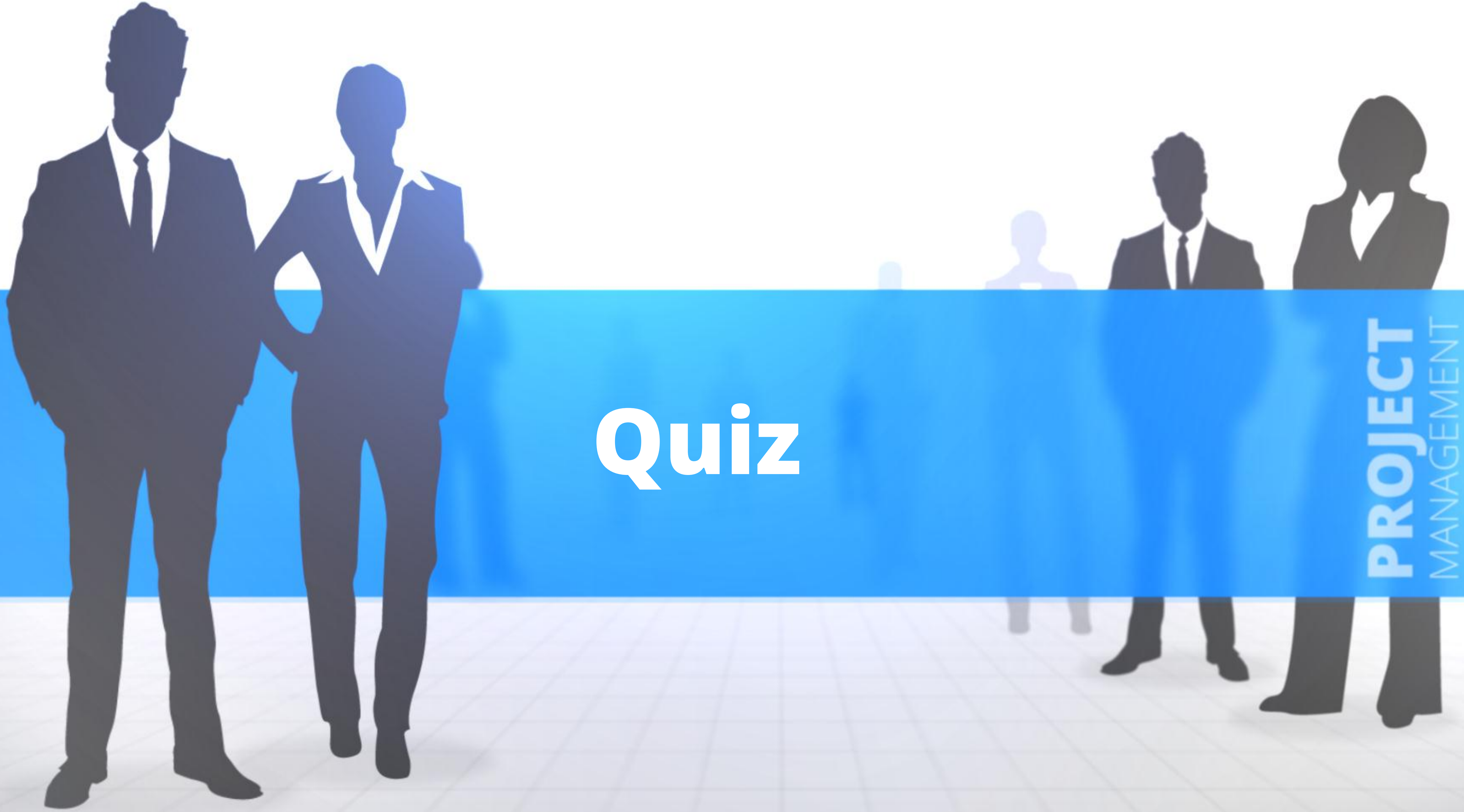
To review Agile estimating:

- Create the Definition of ‘done’
- Determine the team’s velocity
- Estimate the user stories using Story Points
- Decompose features into tasks
- Estimate the Ideal Time for each task
- Convert into elapsed time.

Topics Covered



- Blind Estimation and Affinity Estimation are two basic ways of estimation in Agile.
- Velocity refers to the Development Team’s ability to complete story points in a Sprint.
- Planning Poker is a technique where the entire team estimates the effort needed to complete a user story.
- Ideal Time is an approach used in Agile estimating.



Quiz

PROJECT MANAGEMENT

QUIZ

1

In the past 12 Sprints, the Scrum Team has completed 123 story points. Based on the data, what is the velocity of the team?

a. 10.5 story points

b. 11 story points

c. 10.25 story points

d. 10 story points

QUIZ

2

In the Planning Poker technique, each team member estimates the effort required for a given story point and then each individual's estimate is discussed with the team.

a. True

b. False

QUIZ

3

Ideal time is actually elapsed time estimate.

a. True

b. False

This concludes “Scrum Estimations.”

The Next Lesson is “Scrum Monitoring.”

Agile Scrum Foundation

Lesson 6—Scrum Monitoring

Objectives

- Identify KPIs for monitoring Agile projects
- Describe information radiators and their uses
- Explain burn down charts and Niko-niko calendars
- Explain the importance of a good team space

Key KPIs in Monitoring Agile Projects

During a scrum project, the team needs to monitor the performance of the project, detect problems, and implement resolutions.

Common KPIs include:	
Sprint Goal Success Rate	Working product that meets the Sprint Goal
Defects	Number of errors that make it past the testing process
Burn Down Rate	Speed of delivering value
Velocity	Capacity of the team to complete work
Team Output	Actual output of each team
Satisfaction	Delivering value to customer
Team Member Turnover	Generally low owing to high morale

Information Radiator

The concept of information radiator was invented by Alistair Cockburn, one of the initiators of the Agile movement.

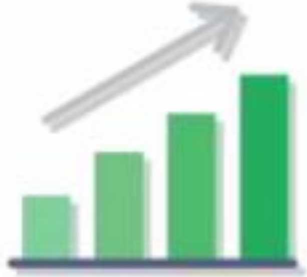
Definition

> "An information radiator displays information in a place where passers-by can see it. They don't need to ask any questions; the information simply hits them as they pass. Information radiators enable team members and other stakeholders to view the current state of the project and its progress."
>
> —Alistair Cockburn

Information Radiator—Examples

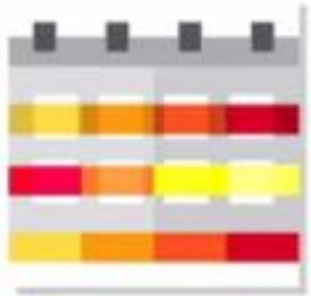
Task board

Burn down chart



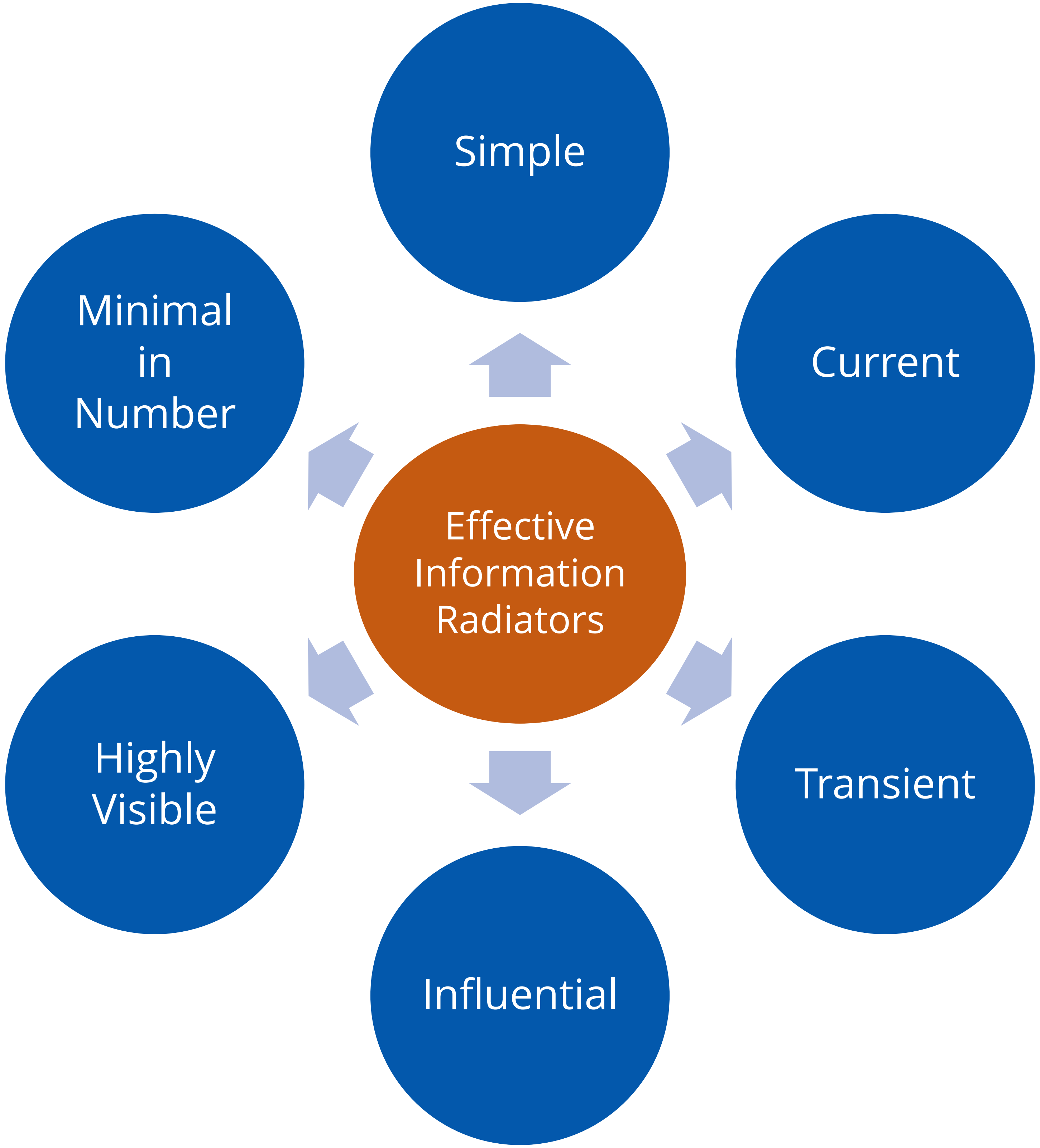
Burn up chart

Niko-niko calendars



Continuous build health indicator

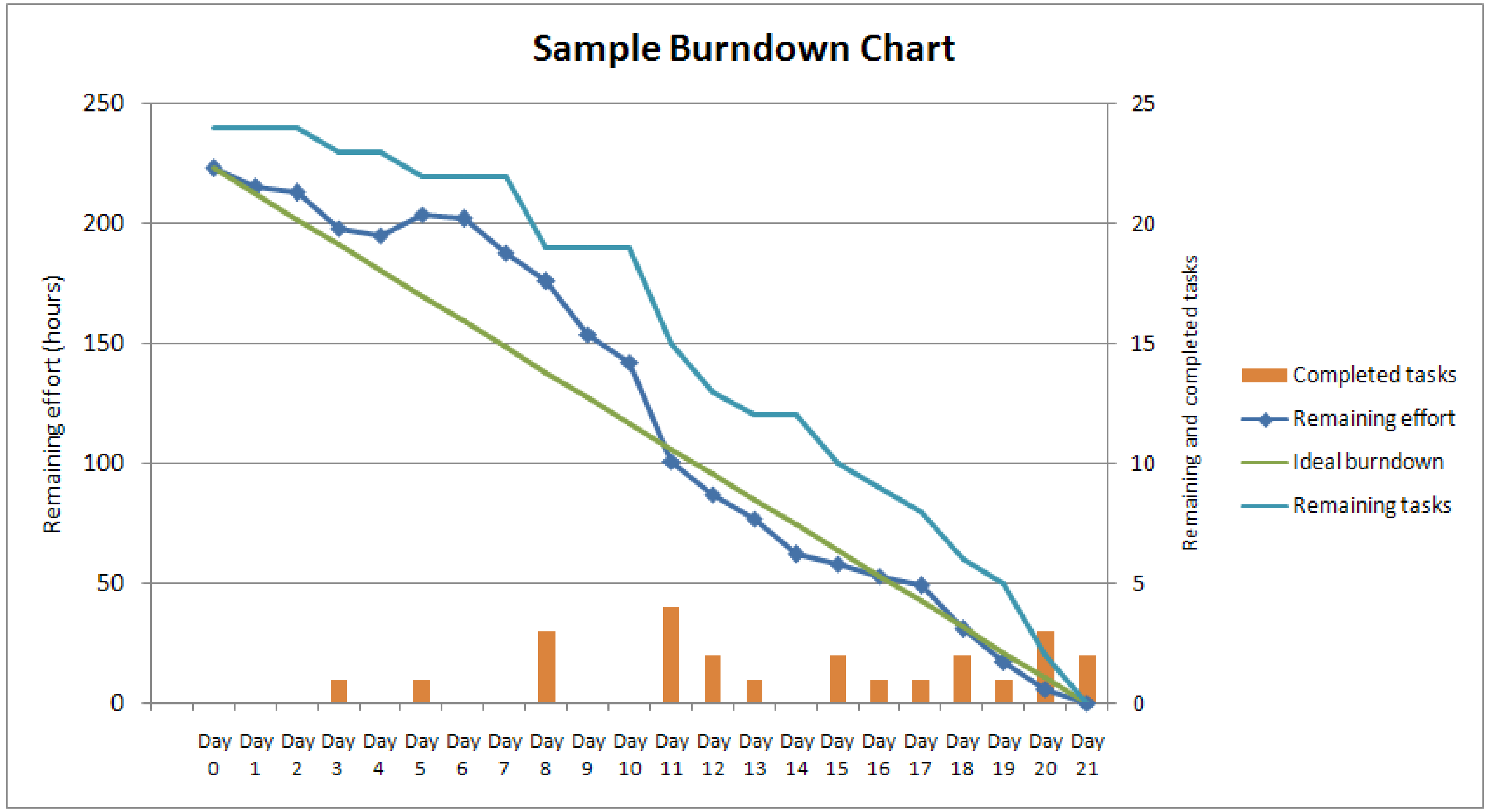
Information Radiator—Characteristics



Information Radiator—Burn Down Chart

A Sprint Burn Down Chart is an example of an information radiator.

- It is also called a Big Visible Chart
- It is updated every day and provides a simple view of the Sprint's progress
- It provides a quick visualization for stakeholders
- It can also be in the form of a release burn down chart, which shows the work remaining and release



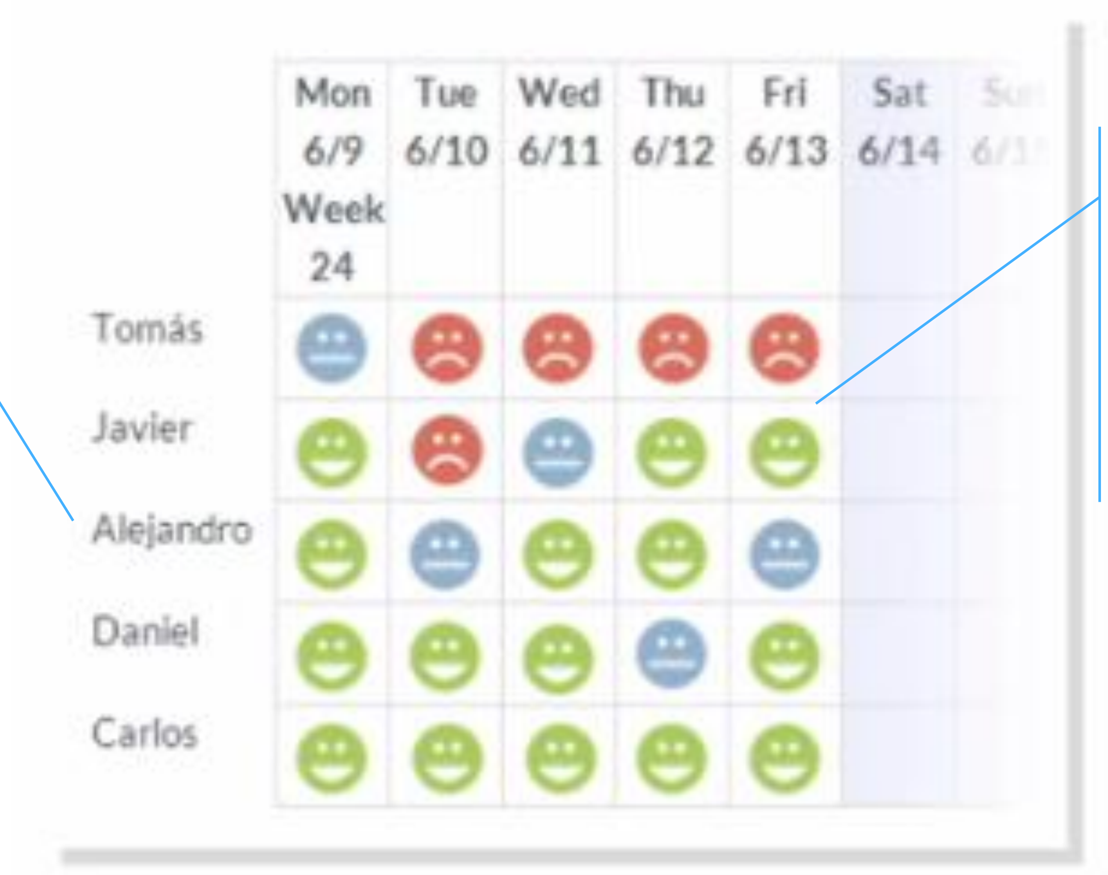
"SampleBurndownChart" by Pablo Straub - Own work. Licensed under Public domain via Wikimedia Commons

Information Radiator—Niko-niko Calendar

In Japanese, Niko-niko is an ideophone for smiling.

Ideophones are words that evoke an idea in a sound.

The team members are listed on the left side of the calendar with dates in the other columns.



	Mon 6/9 Week 24	Tue 6/10	Wed 6/11	Thu 6/12	Fri 6/13	Sat 6/14	Sun 6/15
Tomás	😐	☹	☹	☹	☹		
Javier	😊	☹	😐	😊	😊		
Alejandro	😊	😐	😊	😊	😐		
Daniel	😊	😊	😊	😐	😊		
Carlos	😊	😊	😊	😊	😊		

At the end of each day, each team members puts either a smiley face, a neutral face, or a frown face on the chart.

The idea here is the feelings provide the fastest feedback possible.

By glancing at the Niko-niko calendar, one can get a quick idea of how the project is progressing.

Team Space

The work space in which the team does its daily work is called team space; it has an enormous impact on the success of Agile projects.

Should enable team members to seek assistance

Should promote face-to-face and osmotic communication

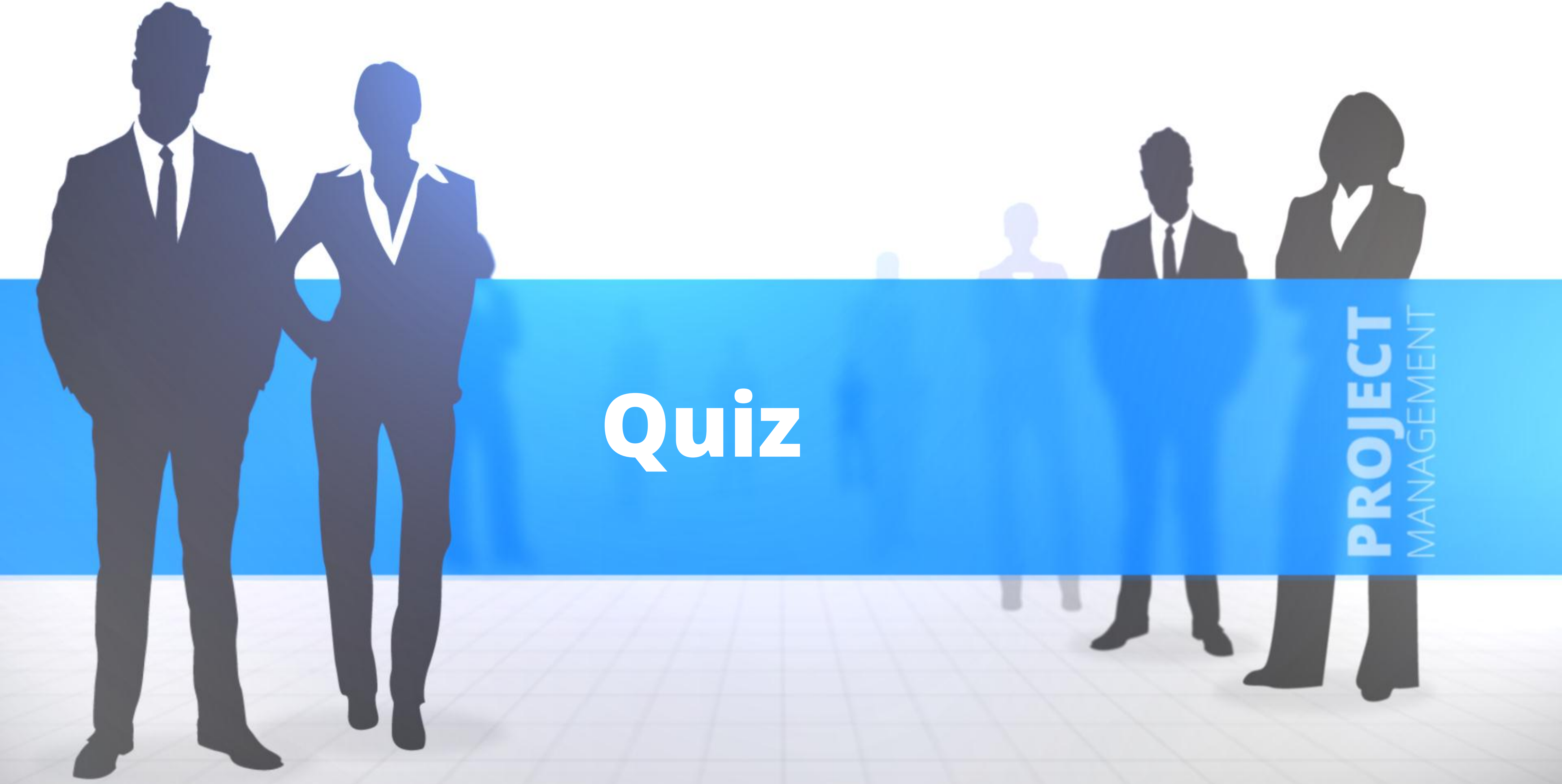
Information radiators should be displayed on the walls of team rooms

Should facilitate simple and faster communication

Agile flourishes when scrum team members work closely together in an environment that supports the process.

Topics Covered

- ✓ There are key metrics to monitor Agile projects
- ✓ Information radiators are used for lightweight documentation
- ✓ Burn down chart and niko-niko calendars are examples of information radiators
- ✓ Team space has an enormous impact on the success of Agile projects



Quiz

QUIZ

1

Which of the following is the highest priority for an Agile project?

a. Escaped defects

b. Sprint Burn-down rate

c. Customer satisfaction

d. Velocity

QUIZ

2

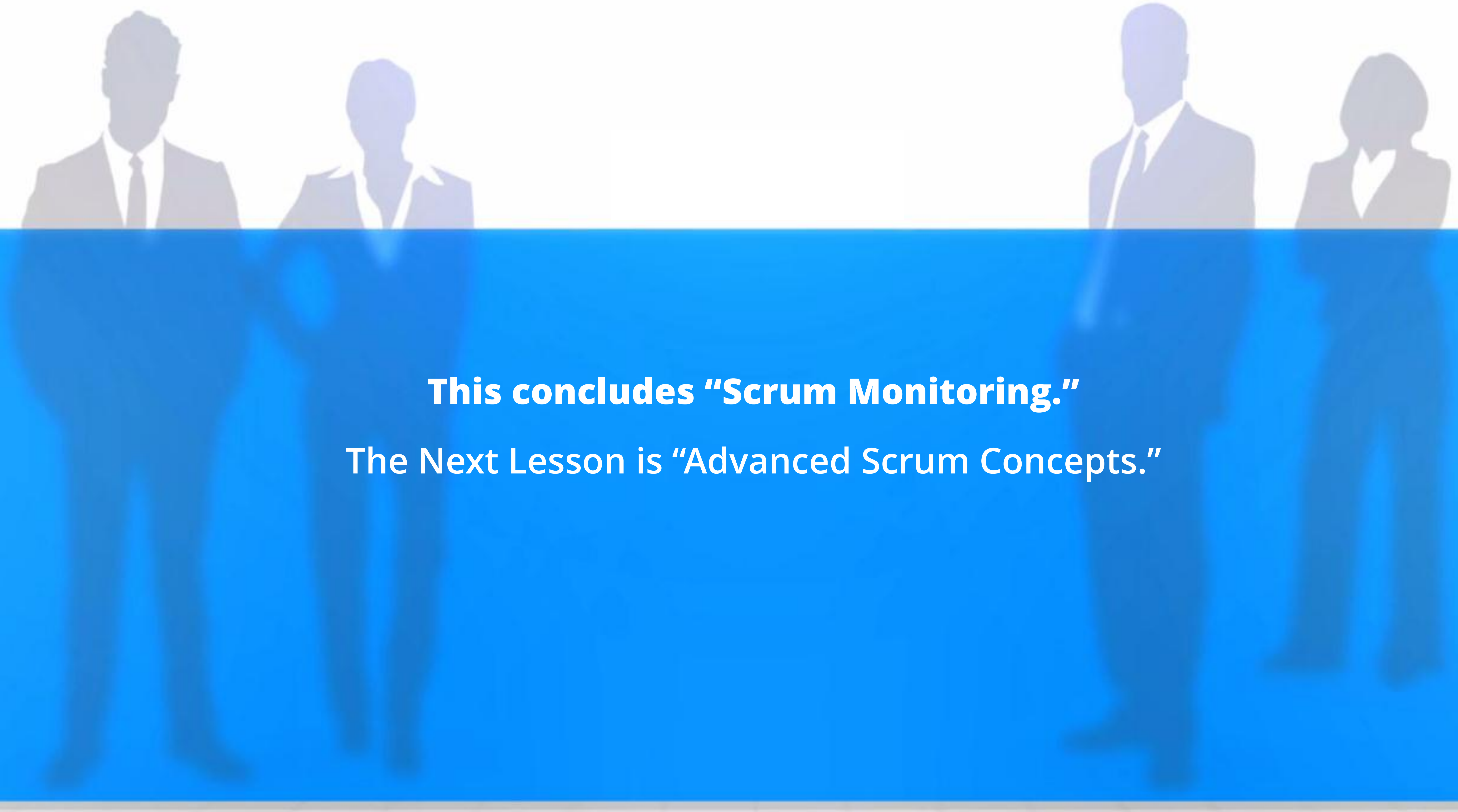
Which of the following types of information radiators provides a daily measure of work remaining in an iteration?

a. Task board

b. Burn-down chart

c. Niko-niko calendar

d. Build health indicator



This concludes “Scrum Monitoring.”

The Next Lesson is “Advanced Scrum Concepts.”

Agile Scrum Foundation

Lesson 7—Advanced Scrum Concepts

PROJECT MANAGEMENT

Objectives

- Explain how Agile and Scrum can be used for large projects
- Describe the Scrum of Scrums
- Identify the role of support and maintenance teams in Agile
- Explain Agile contracting
- Describe best practices in transitioning to Agile

Agile for Large and Complex Projects

Initially, Agile was meant for small teams working on small projects.

However, improvements in technology have made it possible to scale up Agile to be used effectively for large and complex projects.

Multiple Scrum teams need to work in parallel on these larger projects.

The product backlog will be larger and requires multiple product owners to manage it.

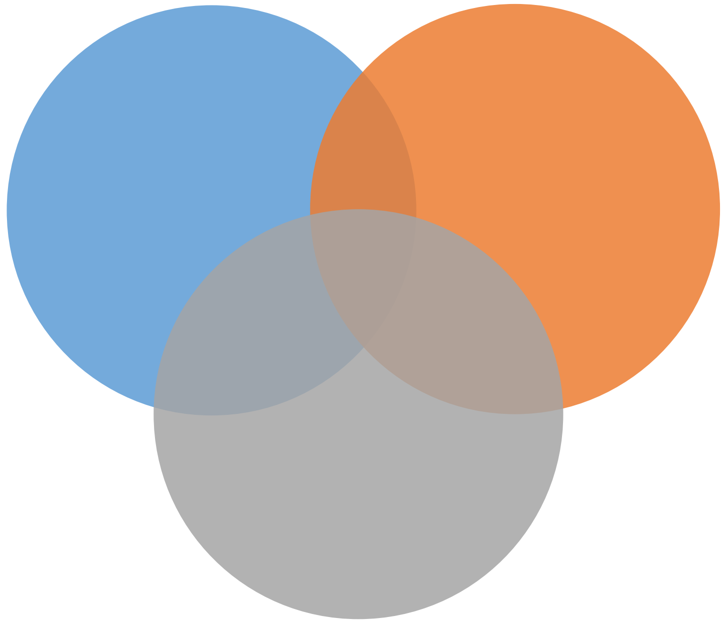
Agile for Large and Complex Projects (Contd.)

The user stories, themes, and epics require work from multiple teams; it is quite difficult to integrate the work of multiple teams.

Scaling up Agile projects involves:

More specialization

- This leverages the agility of small teams by having multiple small teams



Coordination between multiple small teams

- This is facilitated by the Scrum of Scrums

Development Team size should be 6 ± 3

Agile for Large and Complex Projects (Contd.)

Each of the teams continue to have their Daily Scrums at the beginning of each day.

The Scrum of scrums then coordinates the work of multiple teams and solves problems related to:

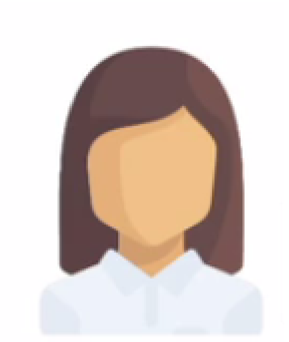
Dependencies

Technical issues

Scheduling

Agile for Large and Complex Projects (Contd.)

The Scrum of Scrums meeting should be attended by a representative from each of the participating teams. The representative can be chosen by the team and could be the scrum master.



Team Member

Scrum Master

However, the representatives must technical knowledge and insights of the team’s work to discuss issues.

Agile for Large and Complex Projects (Contd.)

The Scrum of Scrums meeting could be done daily, two or three times a week, or once in a week depending on the criticality of the interdependencies and issues being discussed.

The purpose of the meeting is to solve problems that require the collaborative effort of multiple teams.

Scrum of Scrums—Agenda

The agenda for Scrum of Scrums includes questions like:

- What has each team done since the last meeting that might affect other teams?
- What will the teams do until the next meeting that might affect other teams?
- What assistance does a team need from other teams?
- What are the issues with the product backlog?

Changing Role of Support and Maintenance

The support and maintenance teams are generally new to the concepts of product backlog and time-boxed sprint.

They work with the product owner to identify bugs, which become user stories for the Scrum team.

They are coached by the Scrum Master on the need to be involved throughout the project.

Changing Role of Support and Maintenance (Contd.)

Beyond product backlog and time-boxing, other changes to the traditional approach that most support and maintenance teams should be familiar with include:

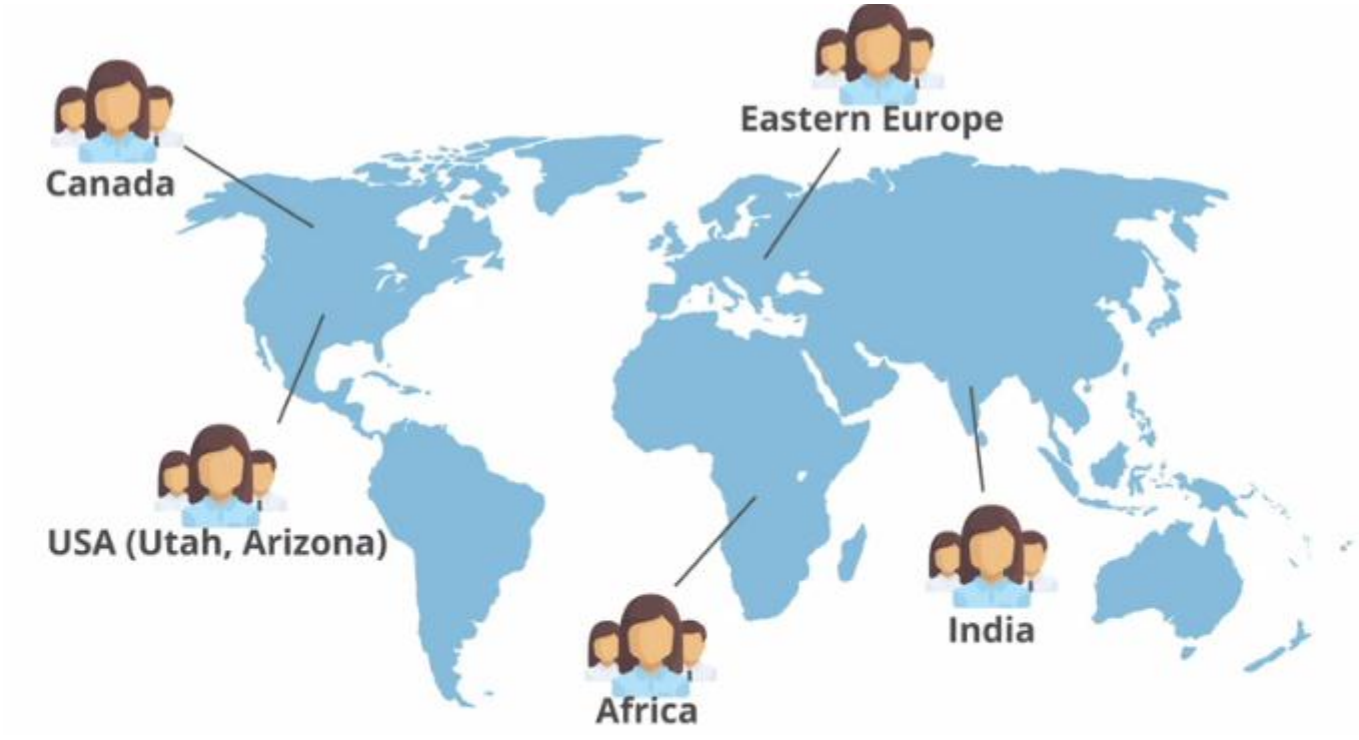
- Aligning SLAs (Service Level Agreements) with product backlog
- Reviewing all bugs and issues
- Creating user stories based on the review of bugs and then prioritizing them in the product backlog
- The empowered, self-organizing nature of the Development team
- The increased visibility and involvement of the management.

Distributed Agile Teams

Collocation is the traditional way of conducting a meeting, where everyone in the Agile team work together in the same room.

Distributed Agile Teams (Contd.)

With the advent of Internet and other technological advancements, it is common to have distributed Agile teams.



Distribute Teams

However, working with virtual Agile teams requires some changes to the traditional collocated agile team; technology must be leveraged to replicate collocation as much as possible.

Distributed Agile Teams—Considerations

A few considerations for managing distributed teams:

Schedule face-to-face time if budget permits

- For kick-off meeting or for project or release planning

Use collaboration tools that best suit the team environment

- Simple and basic
- Robust and complex

Establish working hours and core hours (together)

Schedule meetings around time zones

Agile Contracting

Unlike the Waterfall approach, requirements are unknown at the beginning of agile project, but it needs a contract.

Consider the pros and cons before deciding what type of contract is best for your project.

Agile Contracting (Contd.)

Common contract types:

Fixed Price

- Easy when team has proven velocity
- Profit decreases as the project progresses
- Considered when there is contingency built

Time and Materials

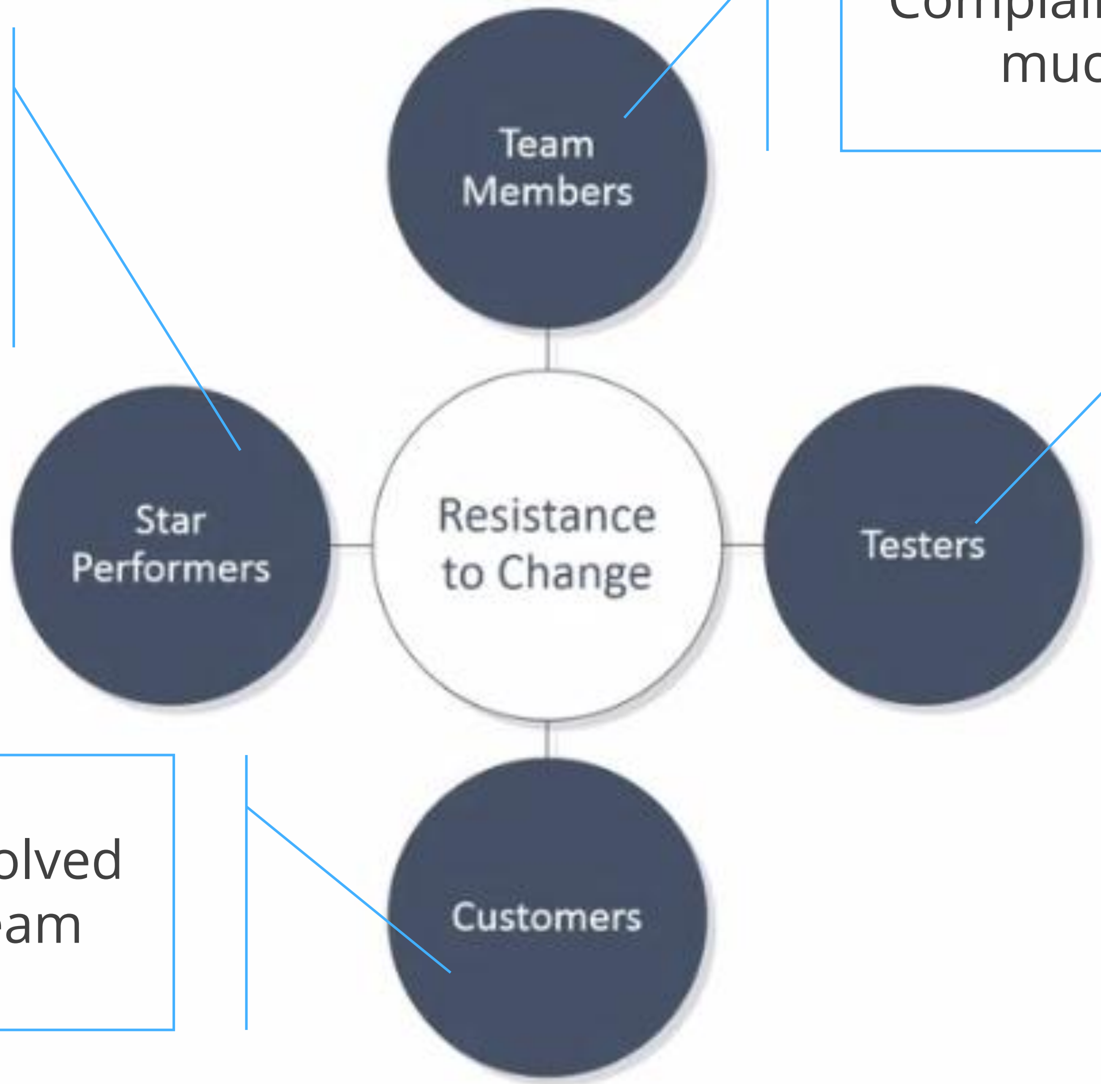
- Commonly used for Agile projects
- Usually includes cost or time cap
  - When the cap is reached, the project is terminated
- The team is paid only for the work accomplished

Because Agile contracts are open-ended, there’s no need to define requirements upfront.

Agile contracting is a fascinating concept that speaks more to real world scenarios.

Transitioning to Agile

Transitioning from a traditional Waterfall approach to an Agile can have many pitfalls.



Might be bothered by the fact that everything is measured based on team rather than being ranked individually.

Complain they are spending too much time in meetings

Complain about having to test everything even though it is not being shipped yet

Complain about being involved full time as part of the team

Transitioning to Agile (Contd.)

These are some suggestions that improve the likelihood of a successful Transitioning to Agile.

Identify an influential sponsor in the organization who can act as a product owner for transition.

Create a transition team or a committee comprised of evangelists who will initiate change, assist teams, and engage with stakeholders.

Create a product backlog that can be used for tracking the progress of the Agile adoption effort

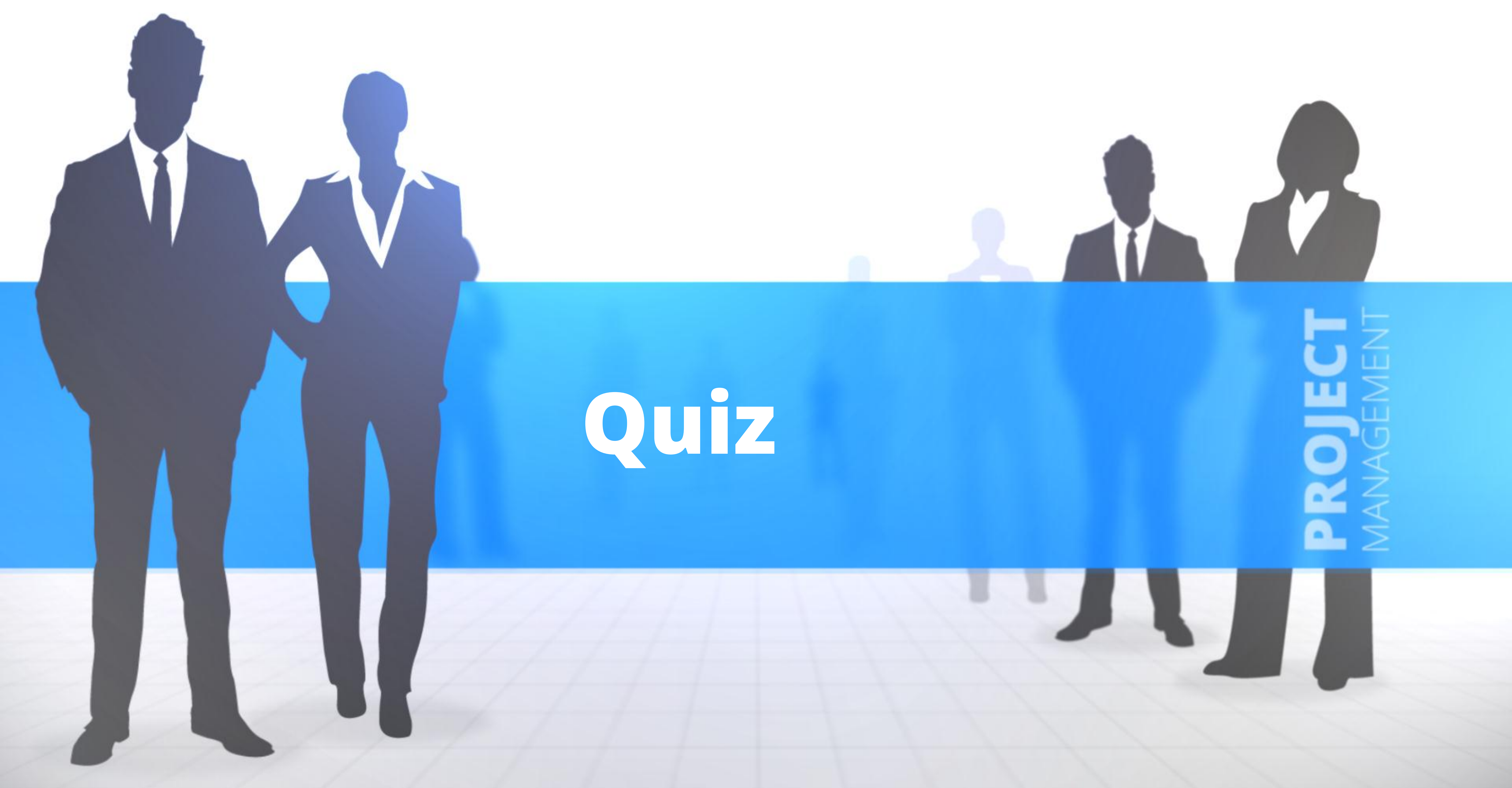
Select a suitable pilot project that should:

- Be an important project but not a critical one
- Have a medium duration
- Have a strong business sponsorship but not be high profile

Be prepared to address complaints that you might encounter and be prepared to overcome individual resistance

Topics Covered

- ✓ Agile can be scaled for large and complex projects
- ✓ Scrum of Scrums is used to coordinate multiple teams
- ✓ Changing role of Support and Maintenance
- ✓ Leveraging technology for distributed Agile teams
- ✓ Planning a successful transition to Agile



Quiz

QUIZ

1

A Scrum of Scrums is a meeting between ________.

a. Only the Scrum Masters of all the Scrum teams

b. All the Scrum teams

c. A couple of members from each Scrum team

d. A meeting within each Scrum team

QUIZ

2

In adopting Agile, Support and Maintenance teams have to primarily become used to _________.

a. SLAs

b. Reviewing bugs

c. Working with a backlog

d. Higher visibility

QUIZ

3

A Fixed Price contract is best suited for an Agile project.

a. True

b. False

QUIZ
4

What is the best way of transitioning from a traditional approach to Agile?

a. Ensure that you get a buy-in from all stakeholders

b. Treat the transition like an Agile project

c. Give people a choice: adjust or leave

d. Ask star performers to campaign for Agile

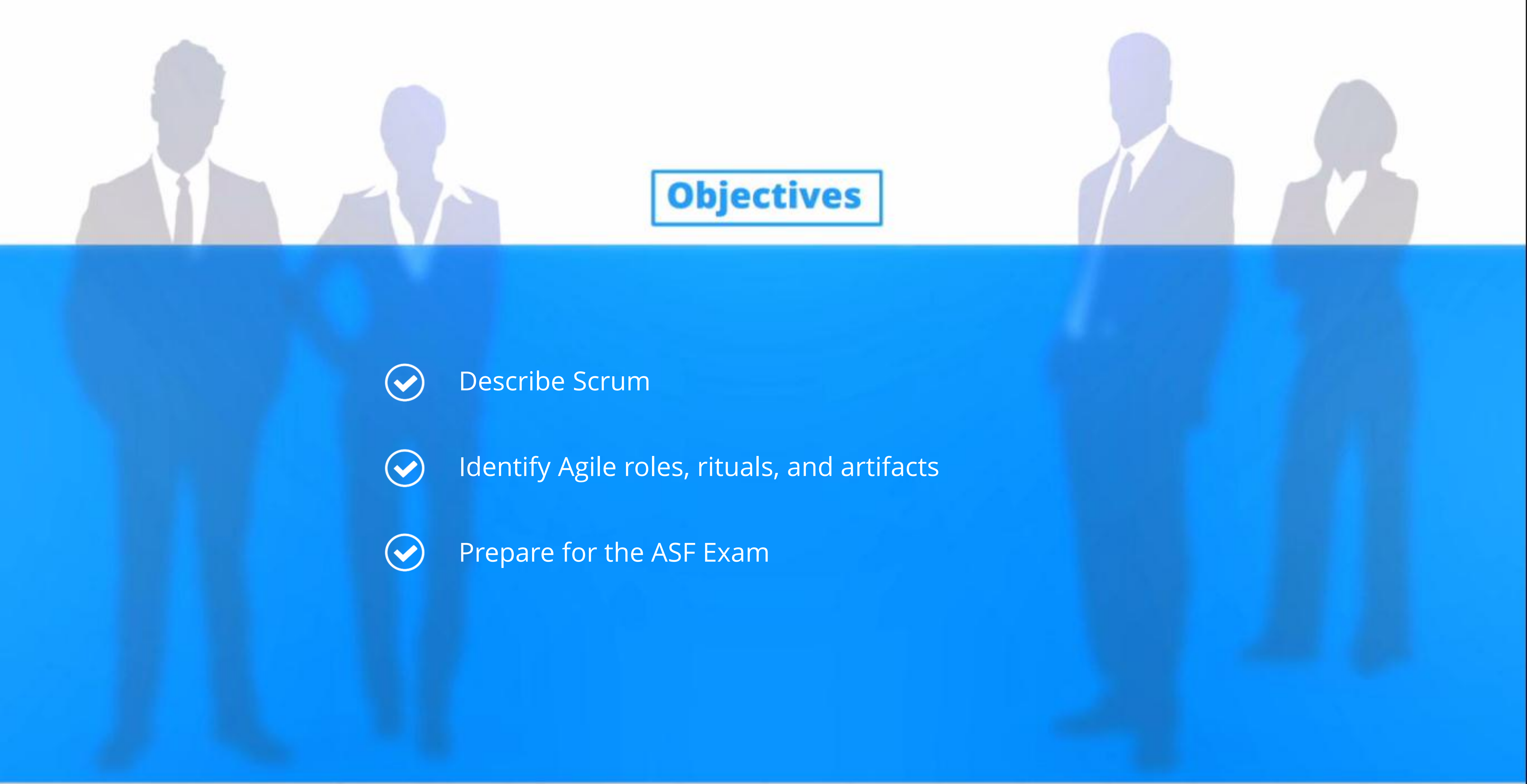
This concludes “Advanced Scrum Concepts.”

The Next Lesson is “Scrum Overview.”

Agile Scrum Foundation

Lesson 8—Scrum Overview

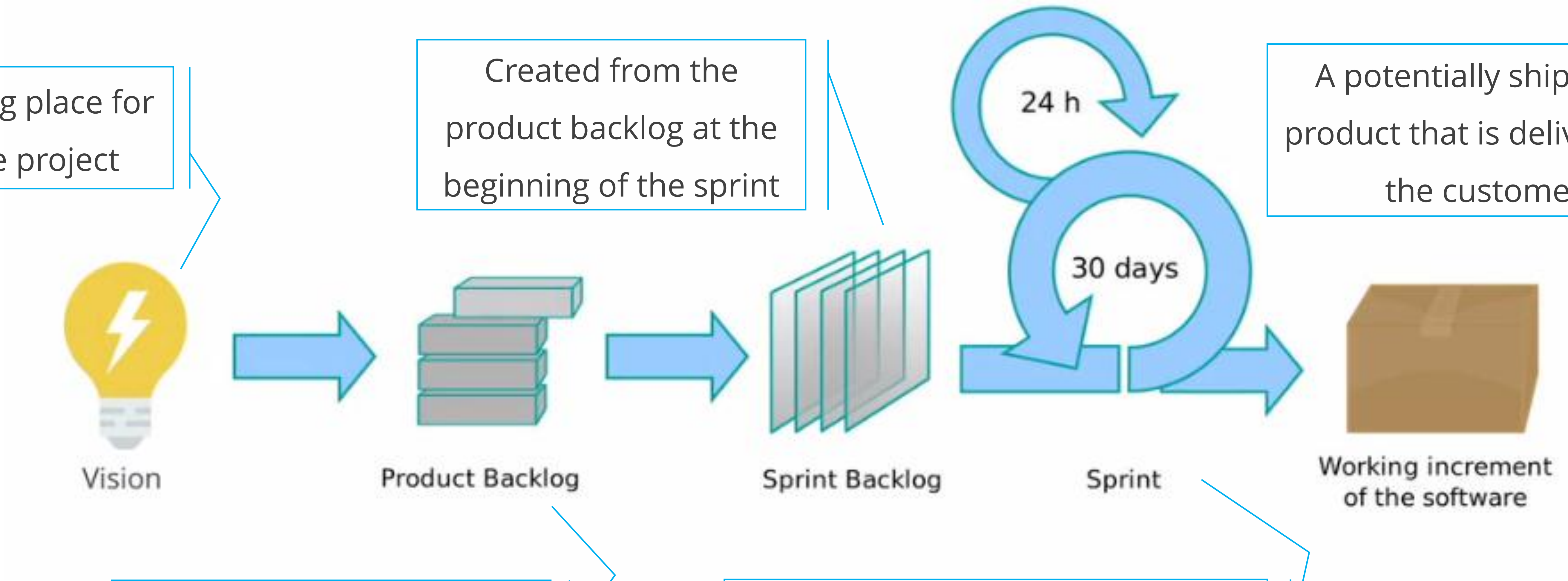
PROJECT MANAGEMENT



Objectives

- Describe Scrum
- Identify Agile roles, rituals, and artifacts
- Prepare for the ASF Exam

Scrum Overview



Starting place for Agile project

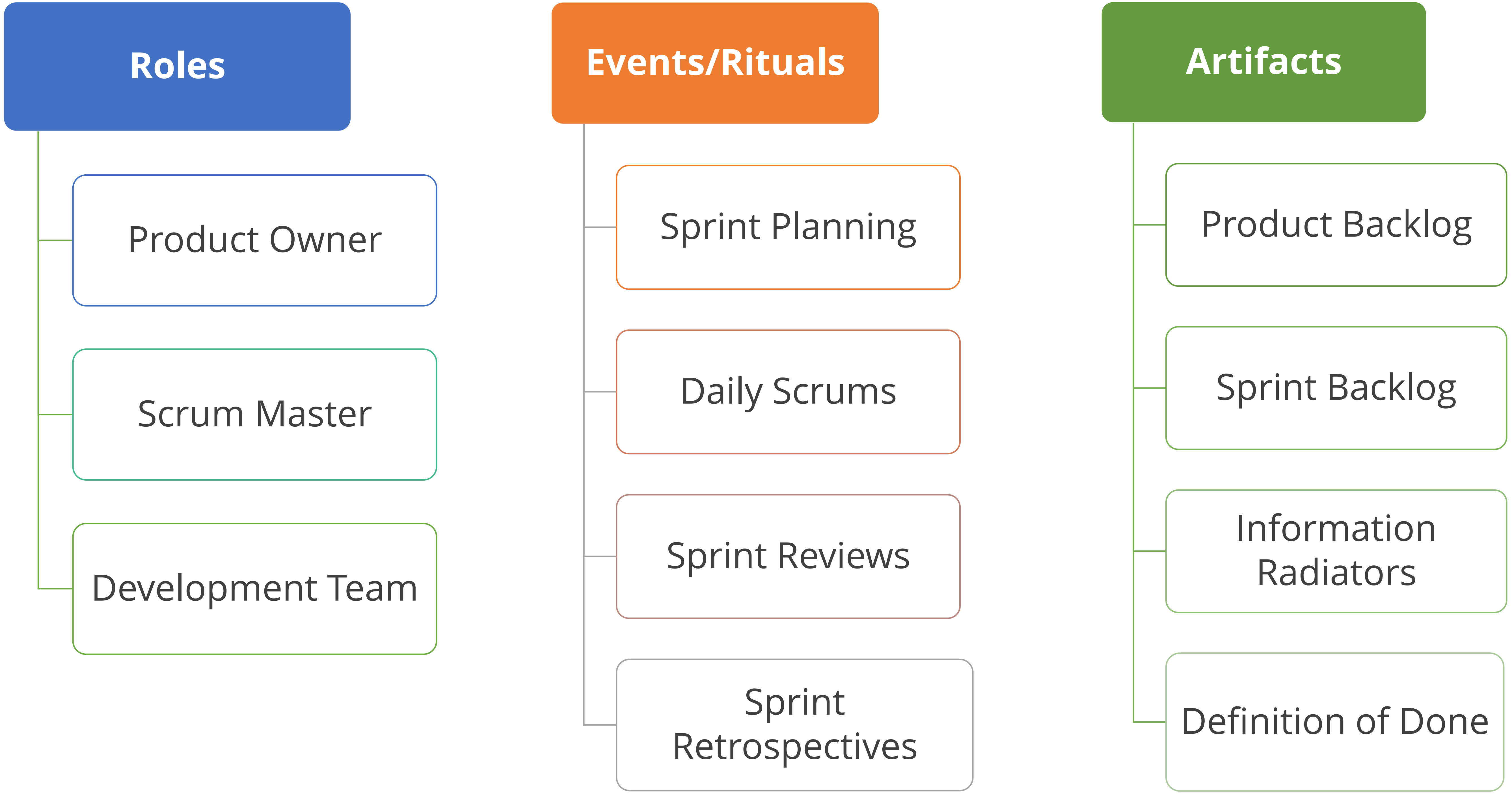
Created from the product backlog at the beginning of the sprint

A potentially shippable product that is delivered to the customer

Contains the user stories necessary to transform the user vision to reality

Lasts for two to four weeks. All the development work is done when the daily scrums are conducted

Basics of Scrum



Roles

- Product Owner
- Scrum Master
- Development Team

Events/Rituals

- Sprint Planning
- Daily Scrums
- Sprint Reviews
- Sprint Retrospectives

Artifacts

- Product Backlog
- Sprint Backlog
- Information Radiators
- Definition of Done

Preparing for the ASF Certification Exam

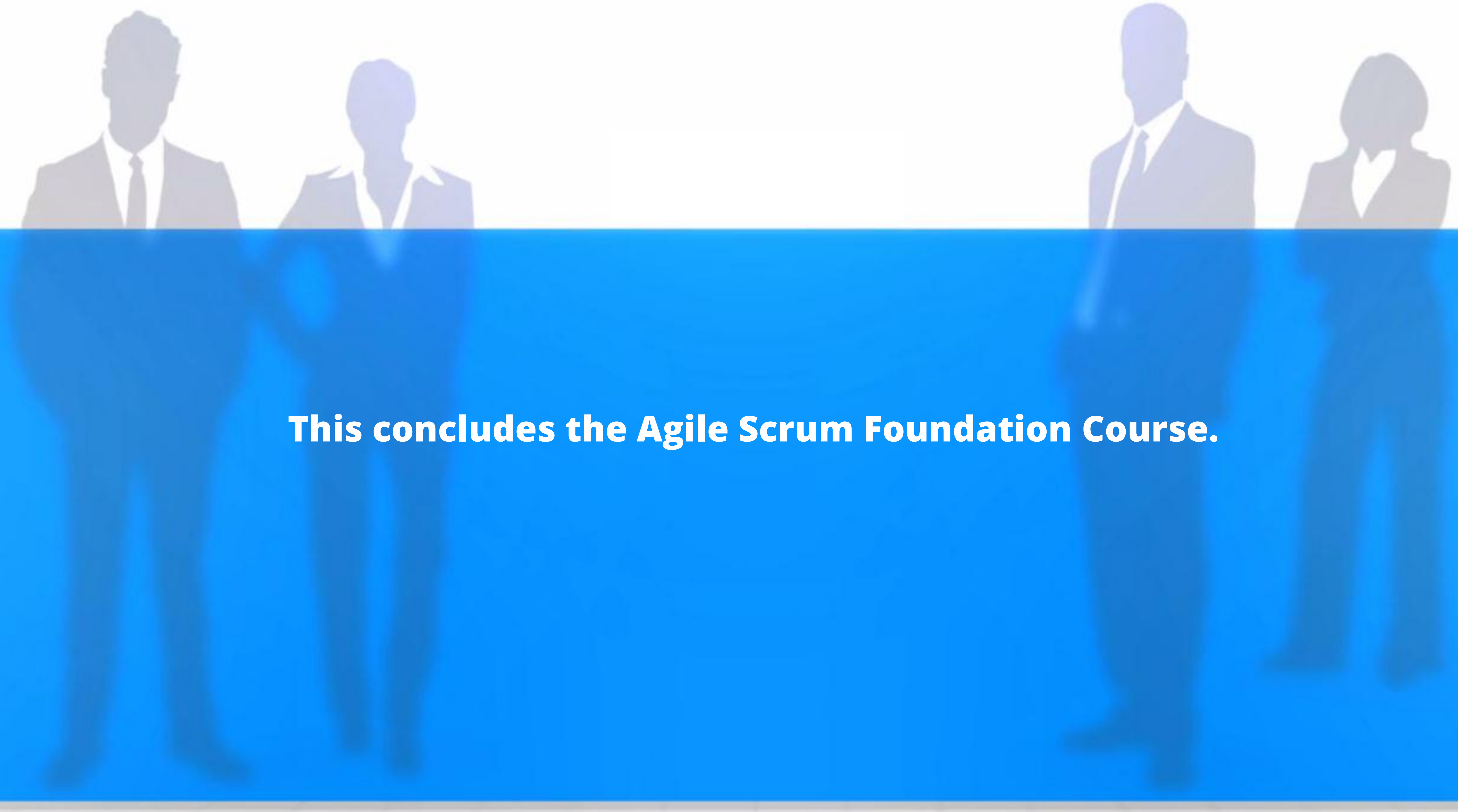
Lesson-end Quizzes

Help you remember the concepts and prepare for the certification exam

Practice Tests

Aim to score 80% in the practice tests, though the pass rate is 65%

Agile world is fascinating and getting better at it takes you one step ahead in your career.



This concludes the Agile Scrum Foundation Course.

Thank You