Project Management Institute.

EMpower
شركة الحلول الإداريـة المتقدمة المحدودة

PMI® Authorized Certified Associate in Project Management (CAPM)® Exam Prep Course

Adaptive Approaches

Flexible Approaches

In This Session:

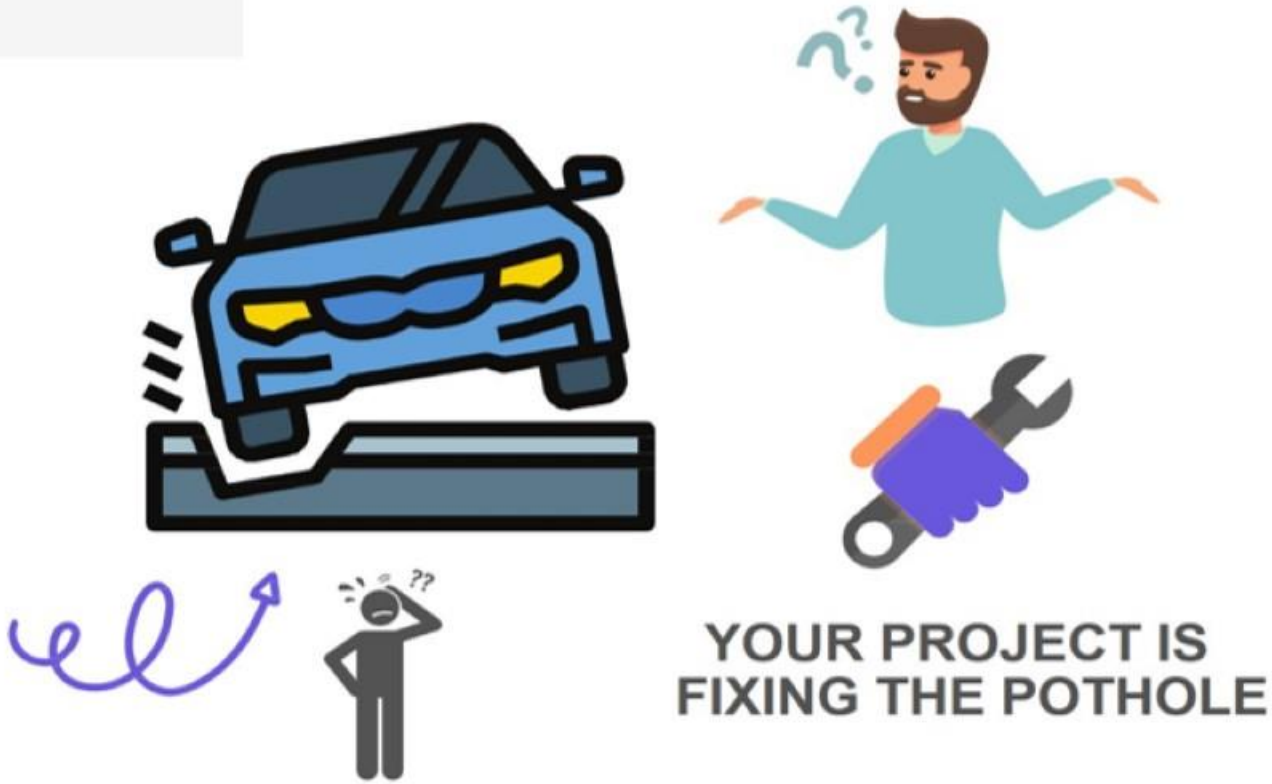
- When to use an adaptive approach
- Team structure in adaptive projects
- Requirements for the adaptive project environment
- Structure and culture of adaptive teams
- Steps in an adaptive project
- Agile life cycles

Simple Project

A Simple project

As the Project Manager, your assigned project is to oversee the repair of the pothole.

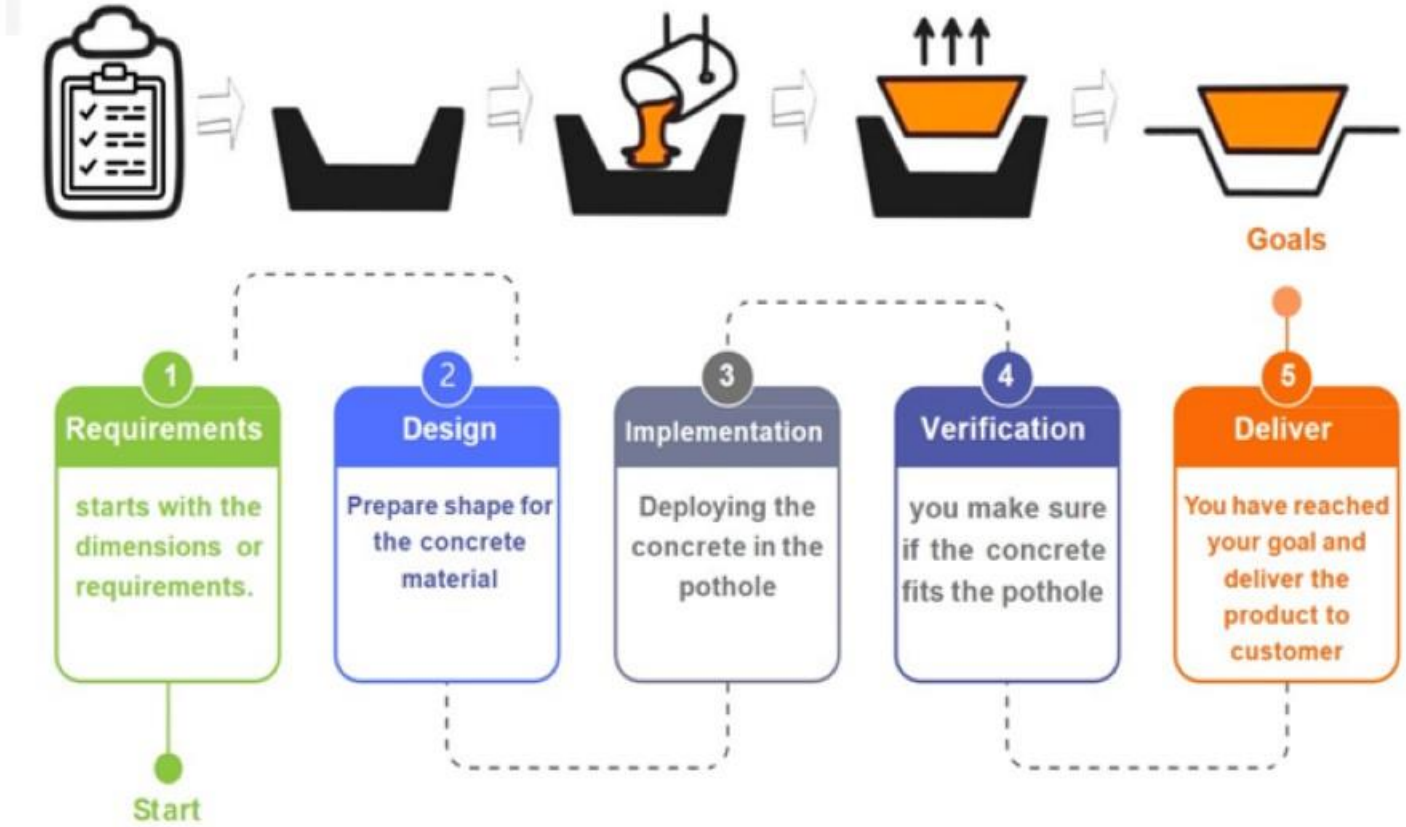
As the Project Manager, you are responsible for addressing the pothole repair by employing predictive, iterative, incremental, and agile methodologies.



Predictive Approach

- You know the dimensions of the pothole.
- You know the requirements of your project.

Waterfall- Sequential Relationship



Predictive/ Waterfall

Sequential relationship.

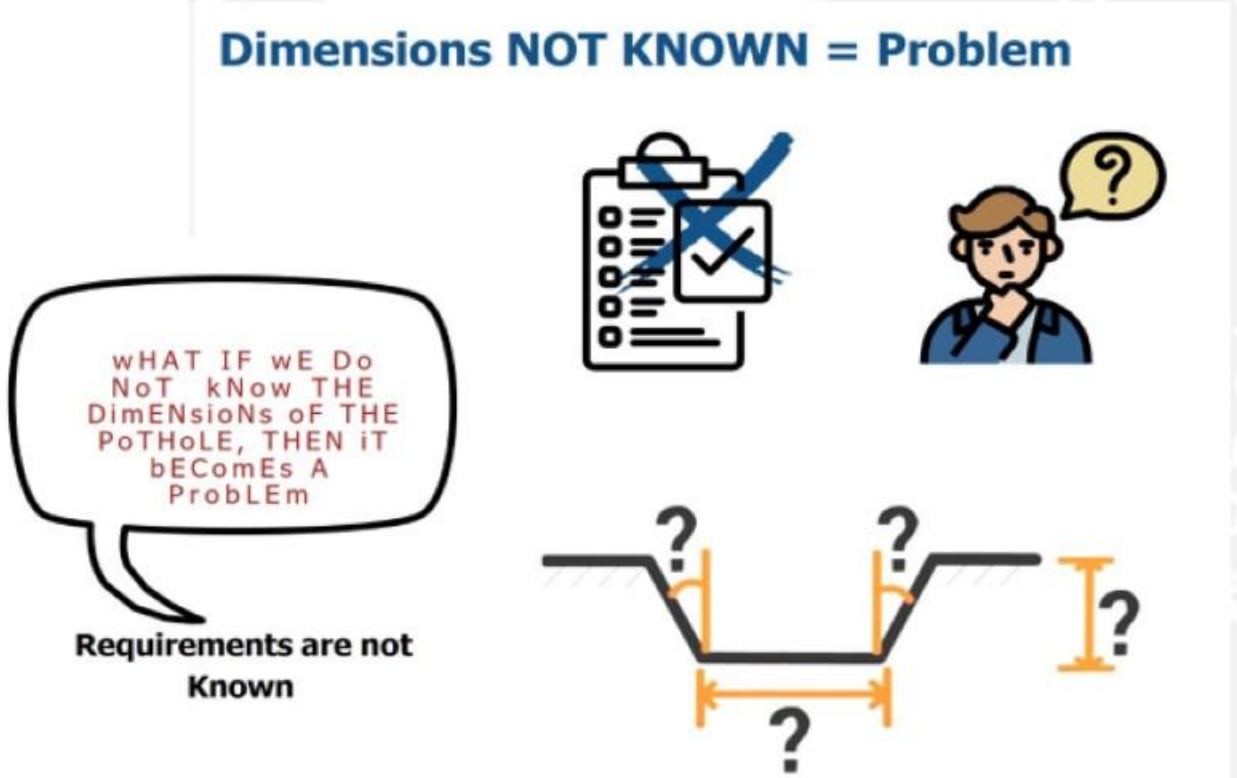
Predictive means planning everything in detail from the start.

In a Predictive approach, also known as the Waterfall model, you plan all the project details upfront.

Changes are hard to make once the project starts.

Dimensions/ Requirements are not known

The problem arises when the dimensions or requirements are not clearly specified.

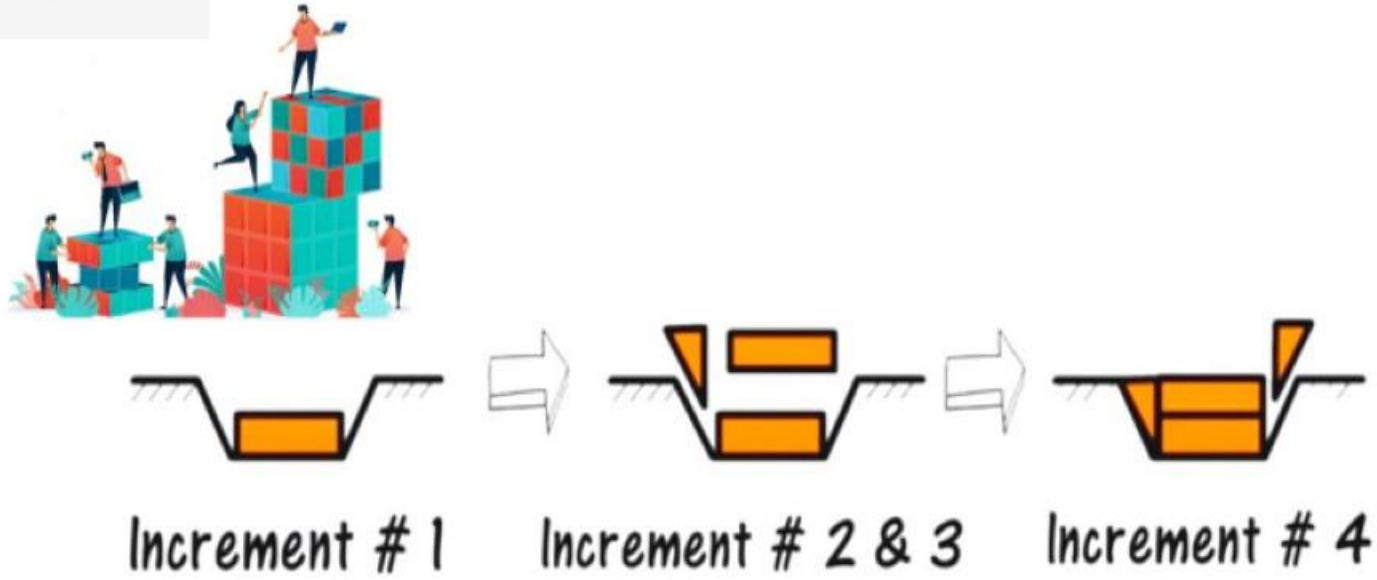


Iterative Approach

- Iterative means doing something over and over, making improvements each time.
- The goal is to learn from each iteration and make improvements in the next one. This aligns with the PMI's focus on continuous improvement.

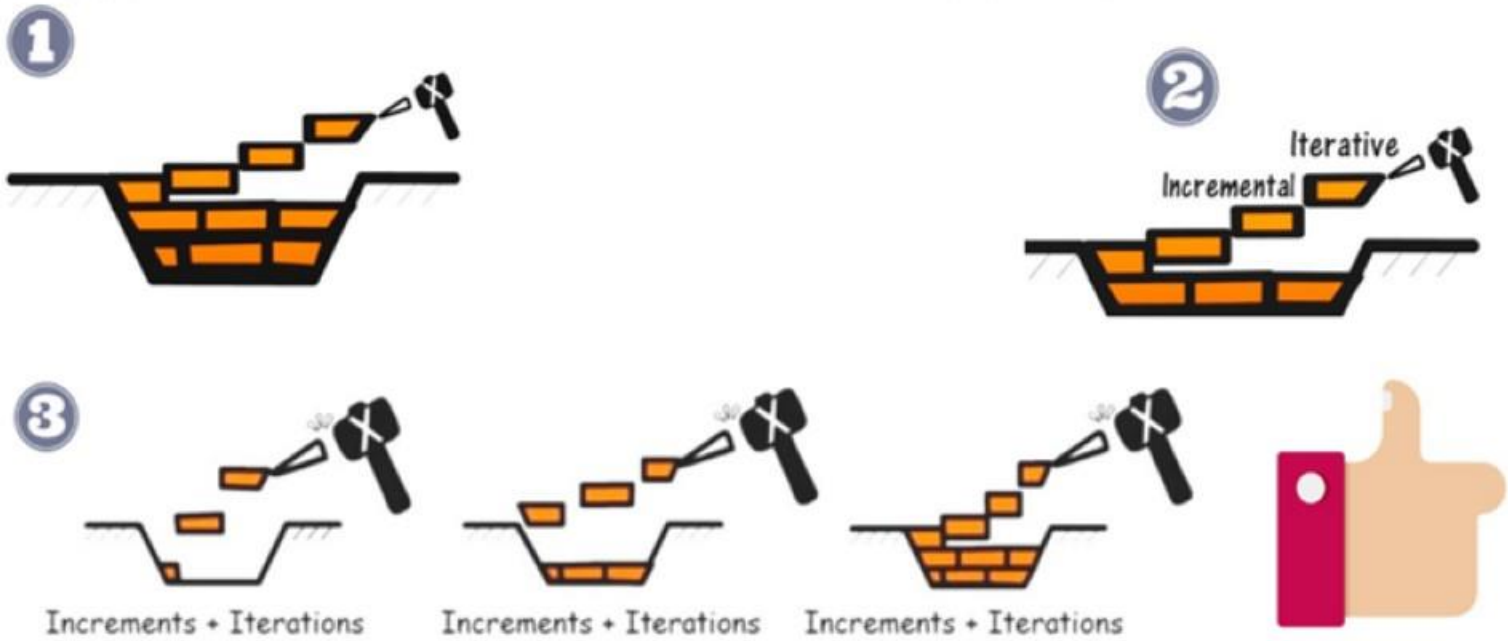
Incremental Approach

- Incremental means adding new features or components step-by-step.
- In an Incremental approach, the project is divided into smaller parts, and each part adds value to the overall project.
- You build upon what you've already done, adding new features or components in each step.

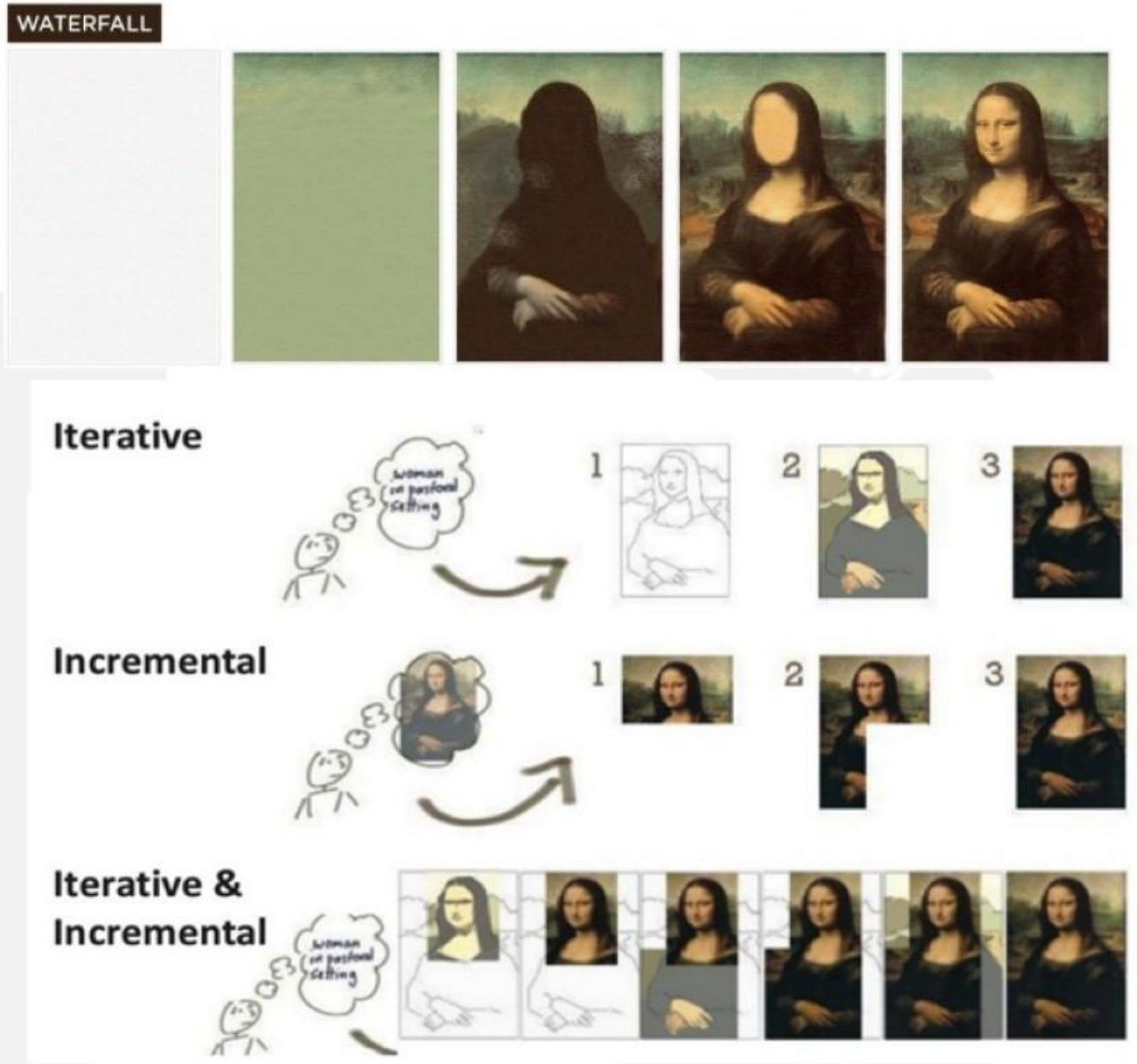


Agile Approach

- Agile methodology incorporates both Iterative and Incremental approaches.
- It breaks the project into smaller cycles (iterations) where planning, execution, and review happen.
- At the same time, each iteration delivers a usable product or feature, adding value incrementally.



Agile Development And The Mona Lisa

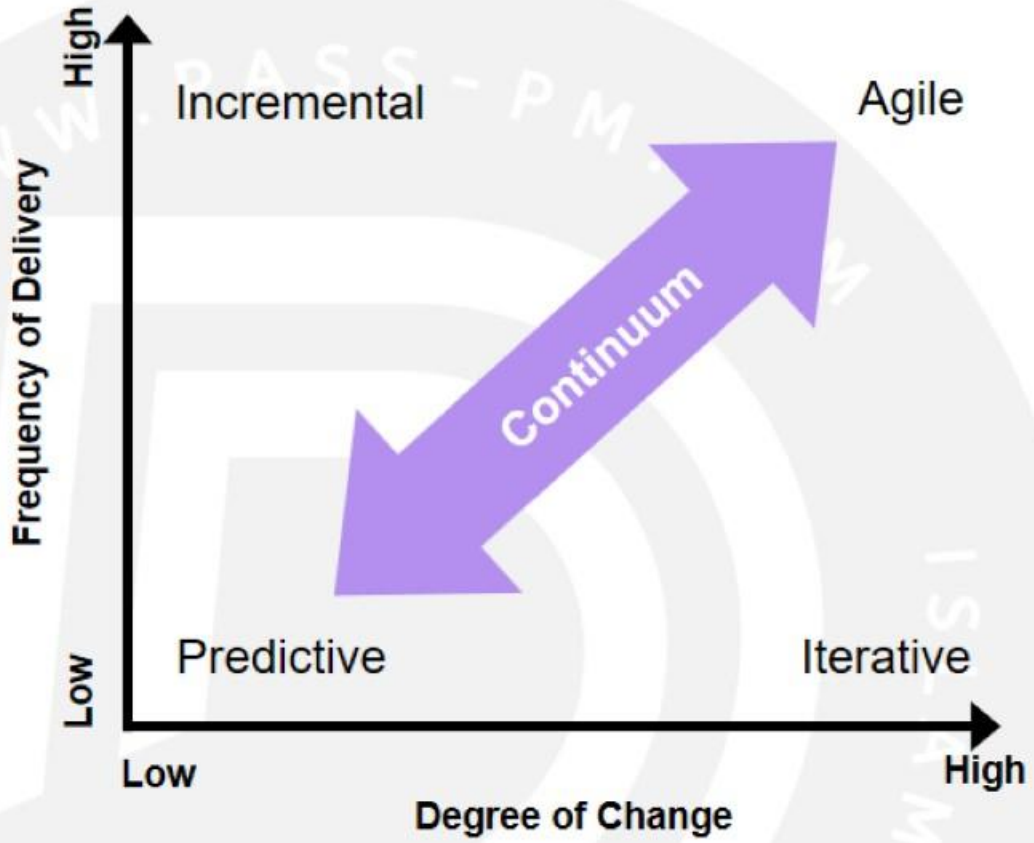


Agile Life Cycles

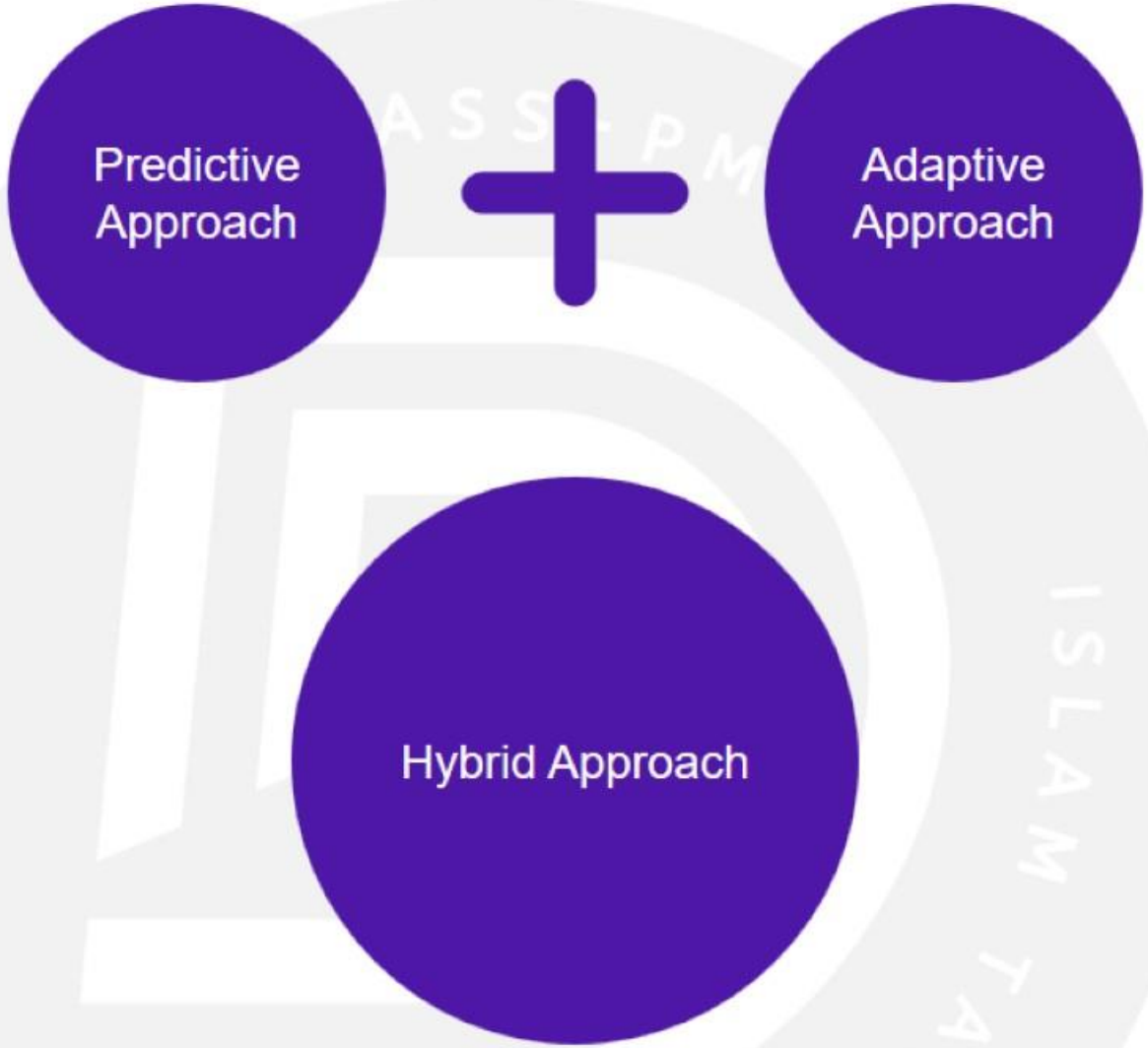
Agile Life Cycles

Agile life cycles are a special case of adaptive life cycles. Basically, if an adaptive life cycle demonstrates characteristics of both iterative and incremental delivery, it is considered an agile life cycle.

A goal of agile life cycles is to uncover hidden or misunderstood requirements. In an agile environment, the team expects requirements to change.



Hybrid Approaches



The Agile Manifesto

Why Look at the Agile Manifesto?

Four Guiding Principles/ Values

We value **individuals** and **interactions**

over process and tools.

We value **working software**

over comprehensive documentation.

We value **customer collaboration**

over contract negotiation.

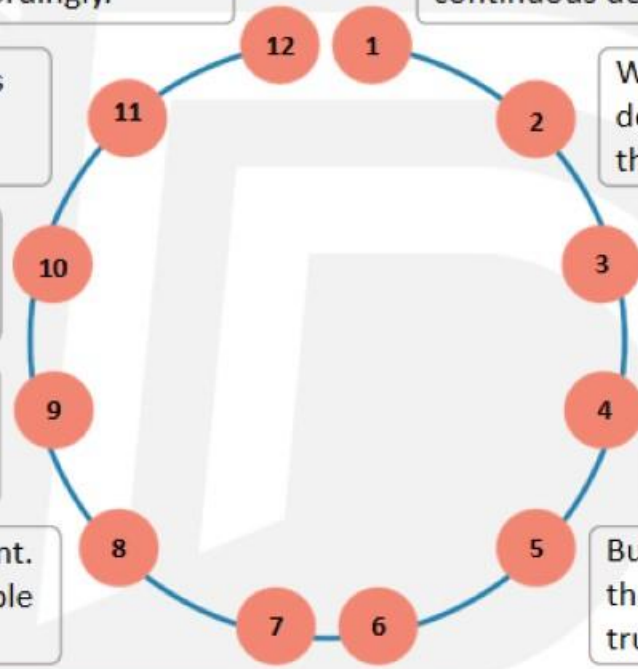
We value **responding to change**

over following a plan.

www.agilemanifesto.org

Principles of the Agile Manifesto

There are 12 principles given in the Agile Manifesto, such as:



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

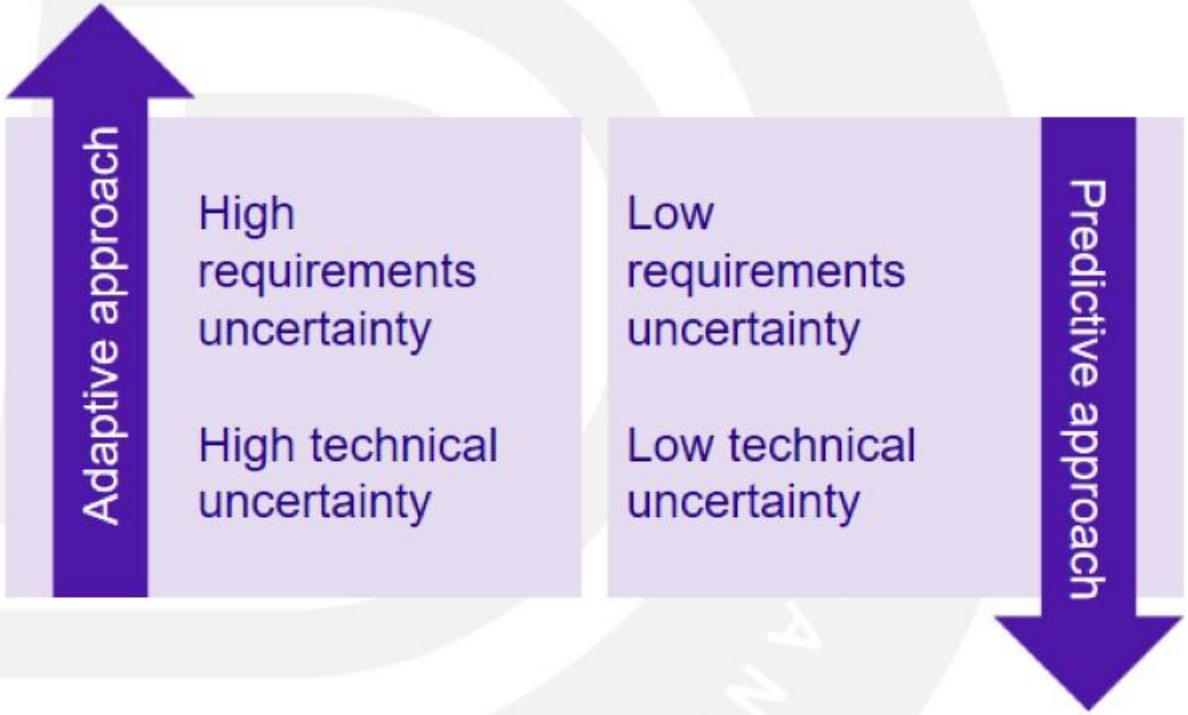
When to Use the Adaptive Approach

Product-, Service-, and Result-based Criteria

Attribute	Adaptive Approach	Predictive Approach
Degree of innovation	Highly innovative deliverables are less understood and are developed over time.	The project involves incremental innovation because it has a known project scope.
Requirements certainty	The full set of requirements is unknown in the initiation phase.	Requirements are known in the initiation phase.
Scope stability	The likelihood of scope change during project implementation is high.	Scope is relatively well known; major changes are unlikely.
Ease of change	Deliverables can easily be adapted.	The nature of the deliverables makes incorporating change in later stages difficult.
Delivery options	Multiple deliverables are possible.	There is a single point of delivery at the end of the project.
Risk	Modular design and development can mitigate high risks.	Known high risks require significant effort for initial planning.

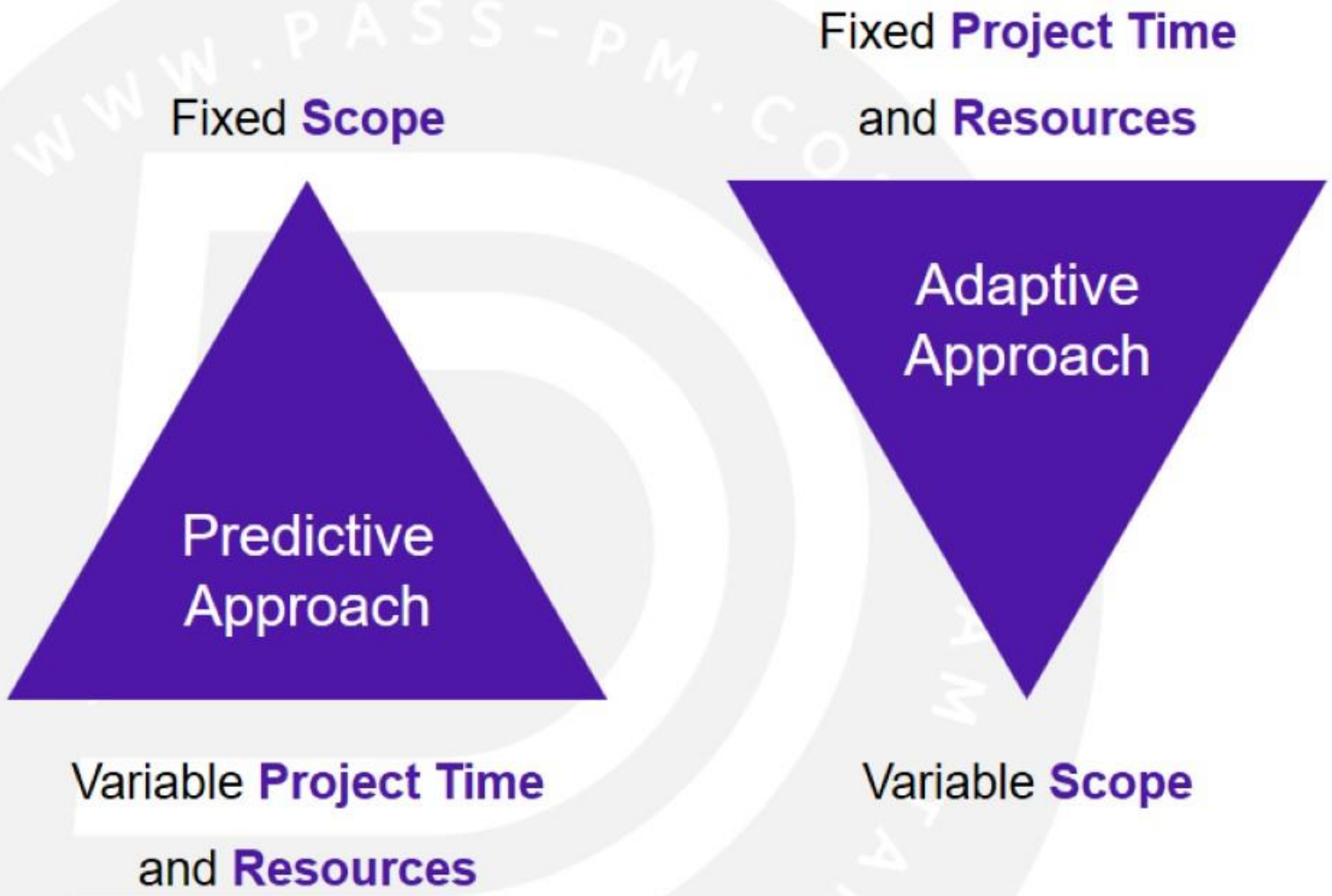
Dealing with Uncertainty

These factors are not **isolated**, but should be seen as **criteria** that, taken together, favor the use of either an adaptive or a predictive approach.

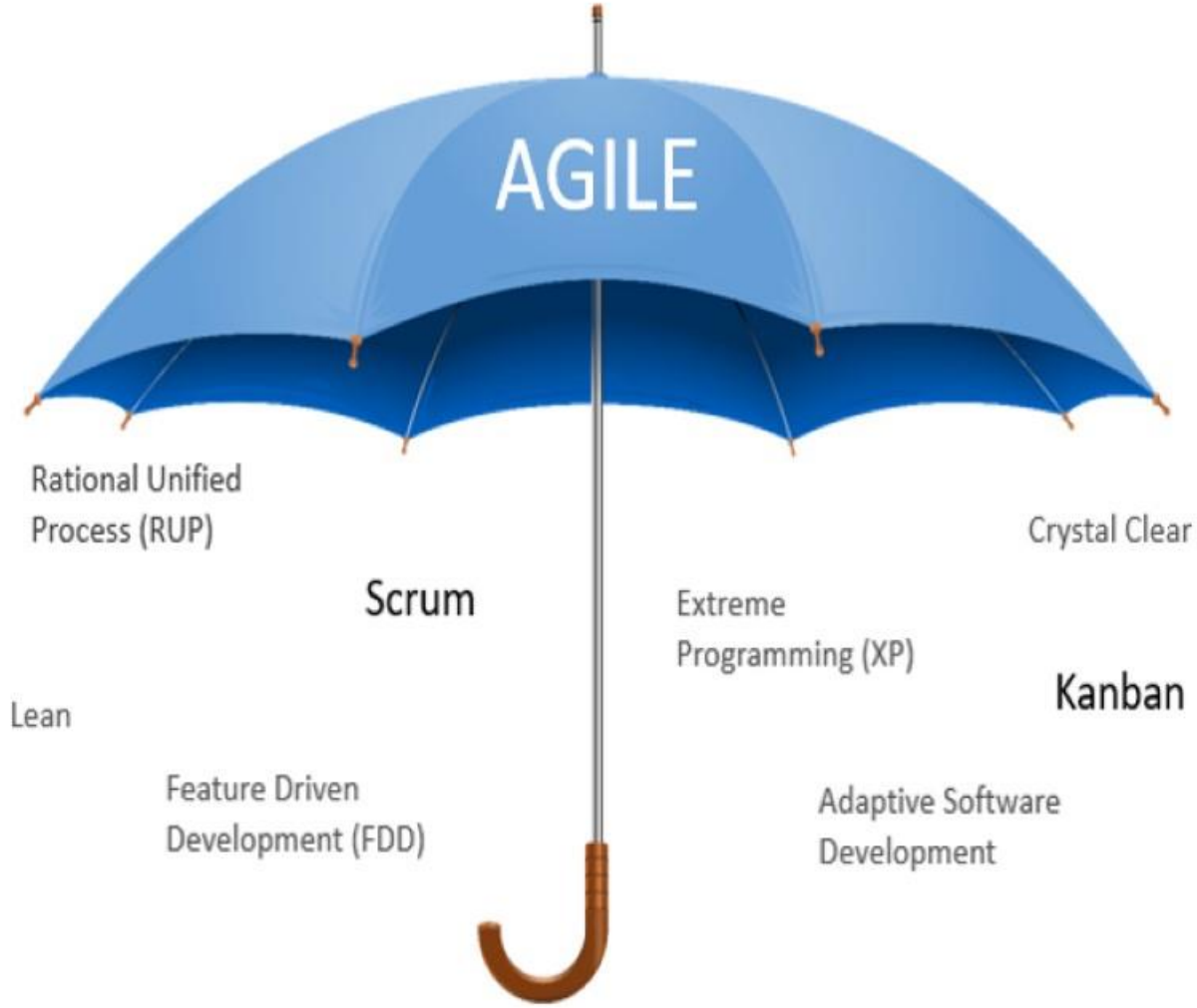


Constraints and Variables

- On predictive projects, scope is fixed, significant changes are carefully controlled, but time and resources can vary.
- In adaptive projects, on the other hand, time and resources become the fixed constraints, and the scope is variable.
- Adaptive approaches were developed to deal with a high likelihood that scope will change during the project.

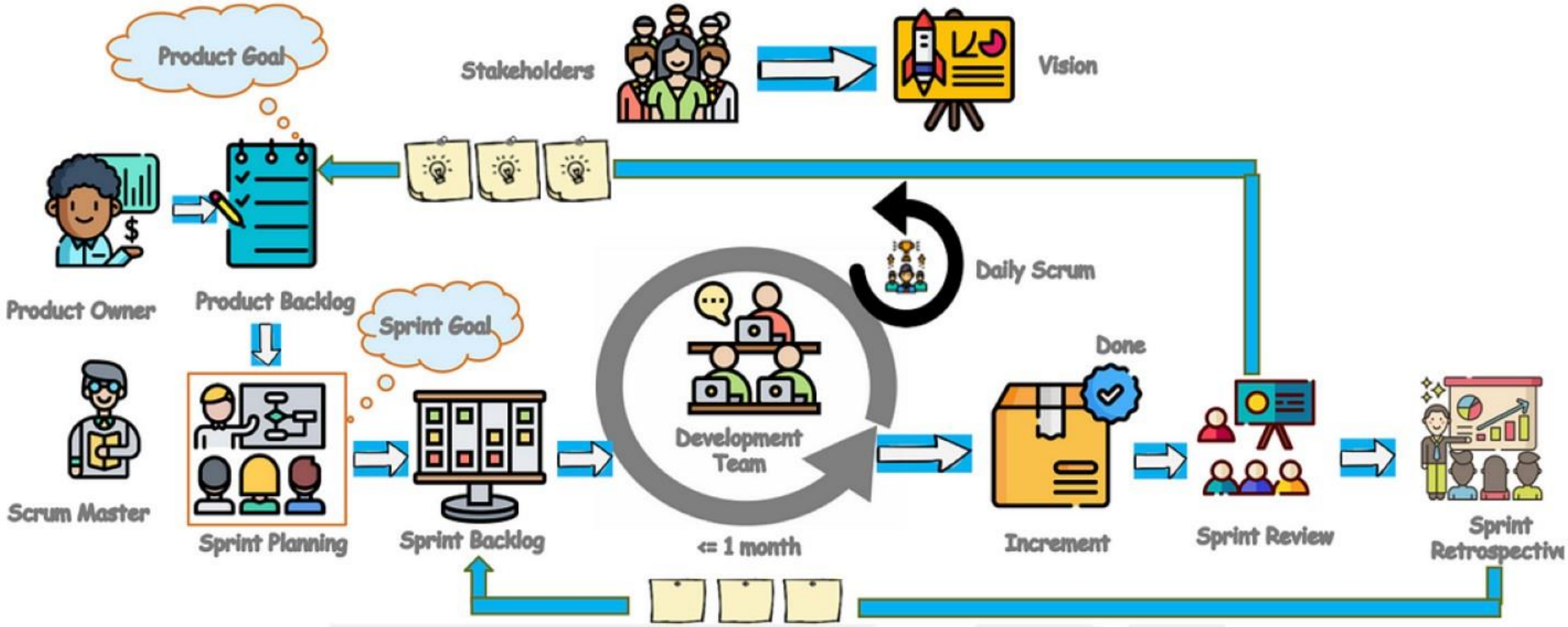


Agile Methodologies, Frameworks, and Processes

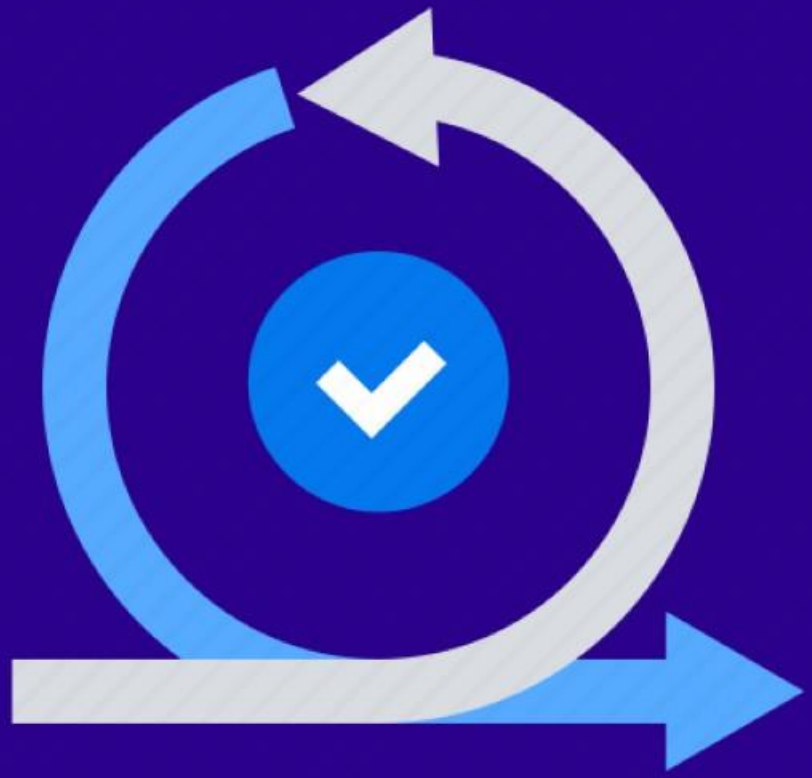


Scrum

Scrum Framework



Scrum Definition



The Scrum framework is a **lightweight**, agile framework for developing, **delivering, and sustaining complex products.**

Scrum is structured to help teams naturally **adapt to changing conditions** and user requirements, with **re-prioritization built into the process** and short time frames for commitment.

Components of Scrum

Accountabilities
(roles)

Events

Artifacts

Scrum Roles

Product Owner

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

Development Team

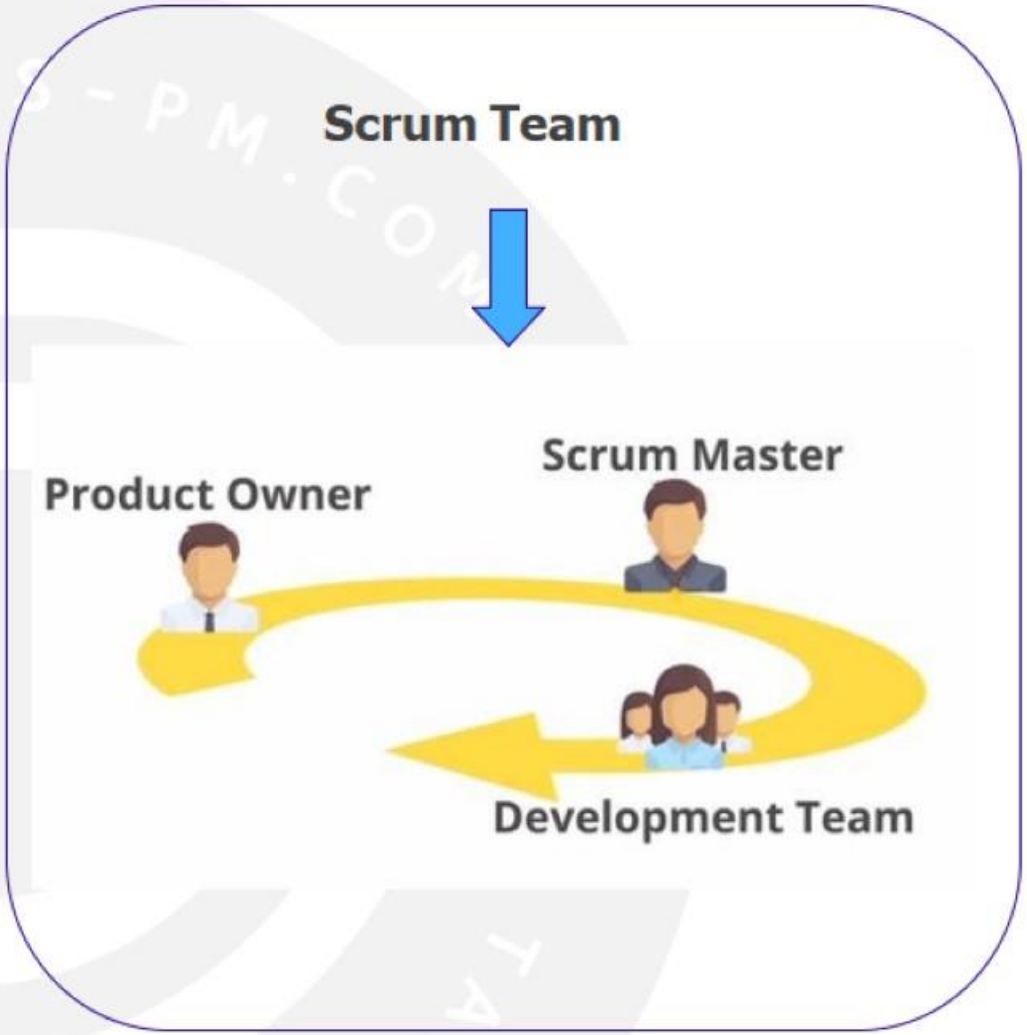
Comprises 7 ± 2 people with a mix of roles and skills; it is self-organizing and determines the best way to meet the goals of the Product owner

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 means that the **team manages its own work and doesn't rely on external management** to assign tasks or solve problems.
- Cross-functional means that the team **has all the skills and expertise needed to complete the project** from start to finish. This ensures that the team can handle all aspects of the project internally, **without relying on external departments or specialists.**



Product Owner

- The Product Owner is responsible for **defining the features of the product and prioritizing** the Product Backlog.
- Reviewing the **functionality at the end of each sprint**, and in between.
- Reviewing the backlog and **prioritizing features** to provide the **most value**.
- The product owner is the **key decision maker** representing the customer and the business.

Product Owner

Scrum Master

- The Scrum Master serves as a **facilitator and coach** for the Scrum Team.
- They help the team **understand Scrum practices** and principles,
- Remove impediments that the team faces
- Work to **improve the team's performance.**
- The Scrum Master also **protects the team from outside** interruptions and distractions.

Scrum Master

Development Team

- The Development Team consists of **professionals who do the actual work of designing, developing, and testing** the product.
- They are **self-organizing and cross-functional**, meaning they have all the skills necessary to create the product without relying on others not part of the team.
- The size of the team should be 7 ± 2.

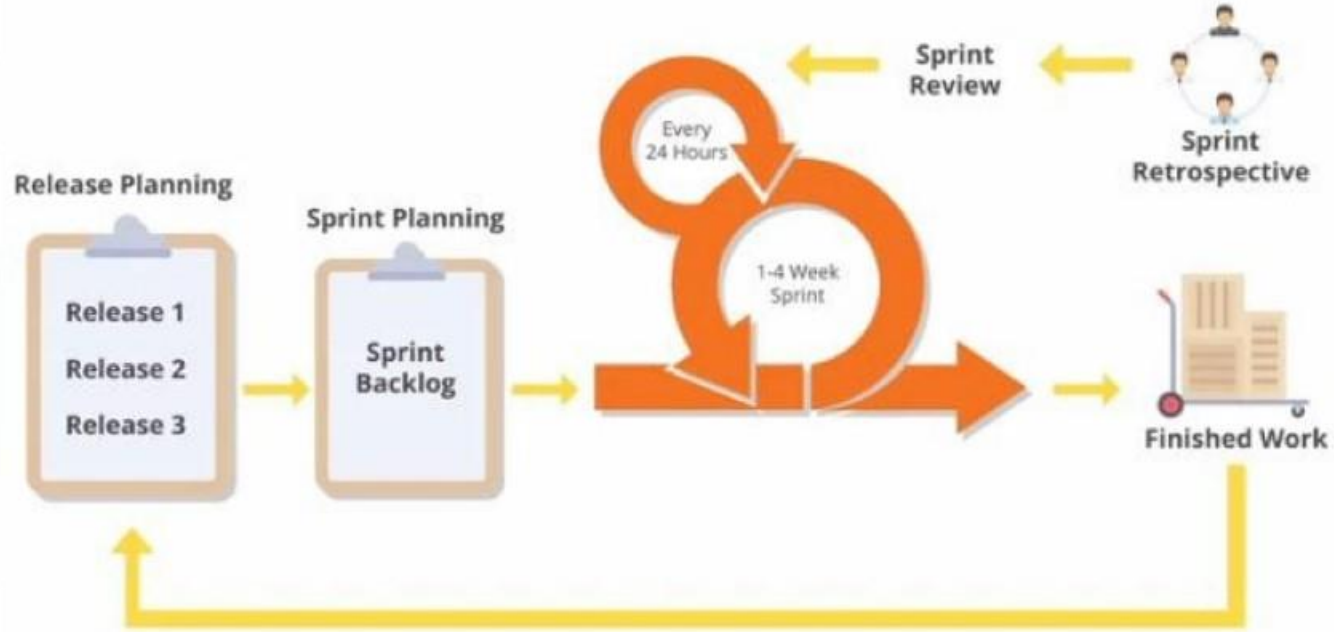
Scrum Events and Artifacts

Events	Artifacts
Sprint Sprint planning Daily scrum Sprint review Sprint retrospective	Product backlog: product goal Sprint backlog: Sprint goal Increments: Definition of done

Scrum Events

The main Scrum Events include:

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

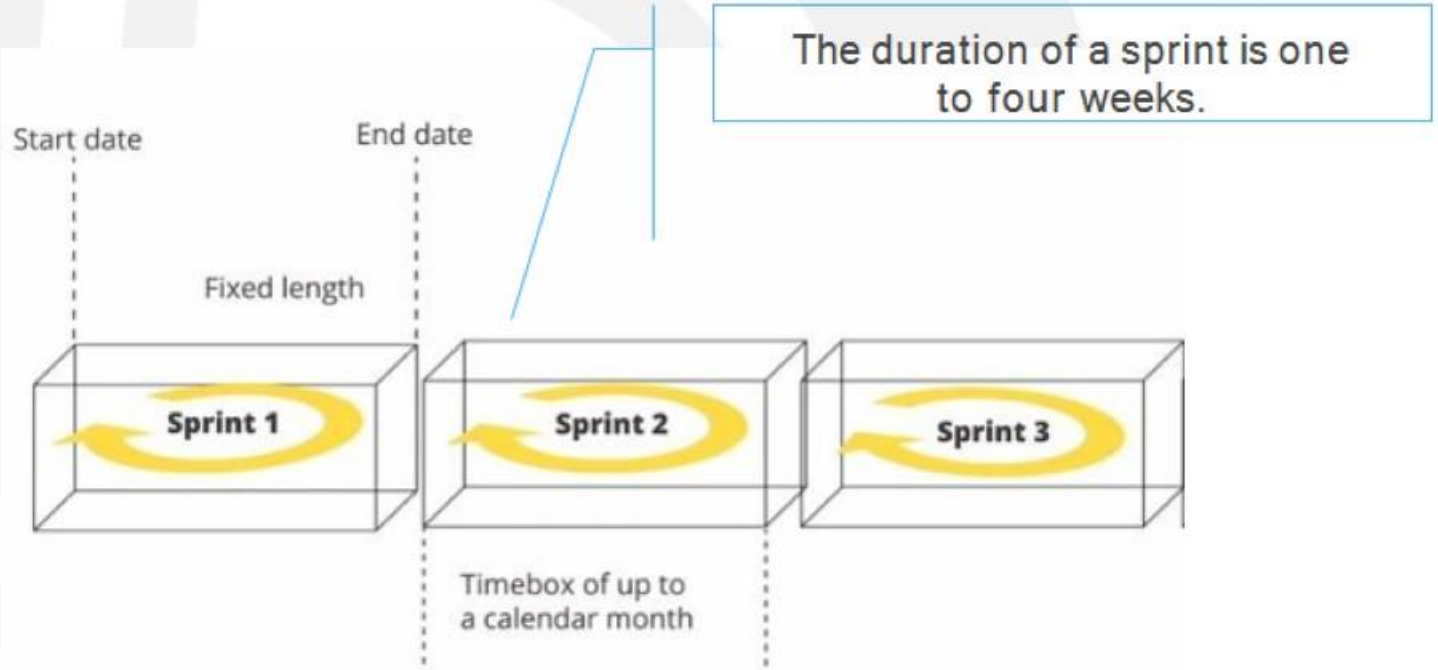


Sprint/ iteration

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



Sprint Planning Meeting

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.

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

Daily Scrum/ Daily Standup

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?

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.
- 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.

Sprint Retrospective

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

Sprint retrospective focuses on three main questions.

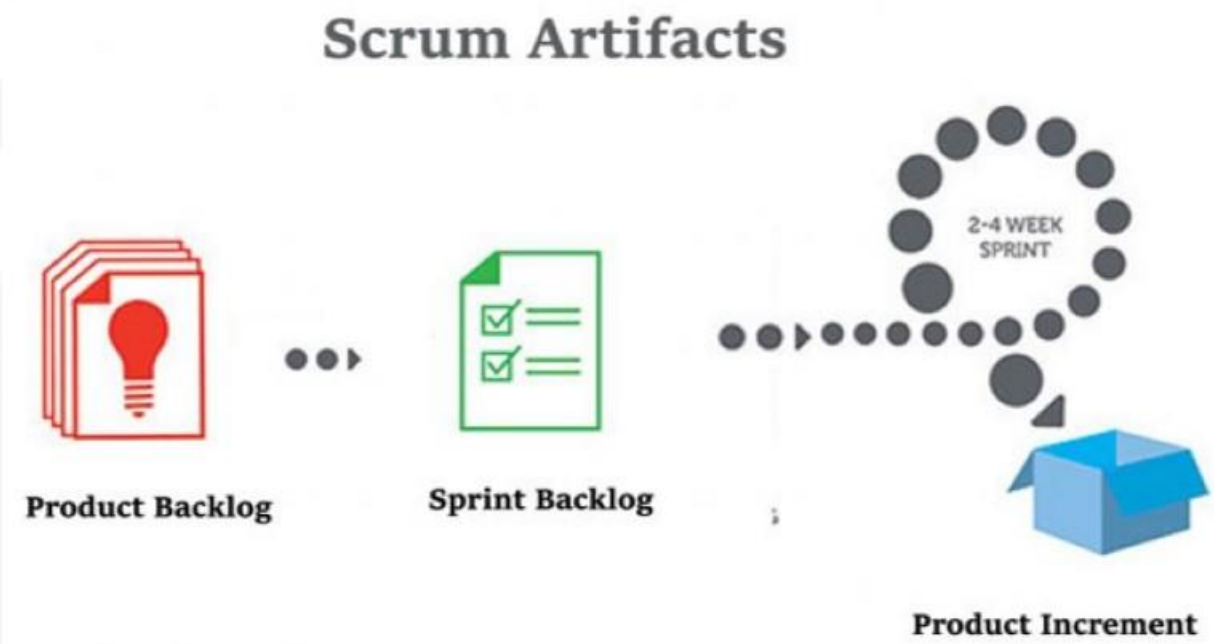
What went well?

What did not go well?

What should be done differently next time?

Scrum Artifacts

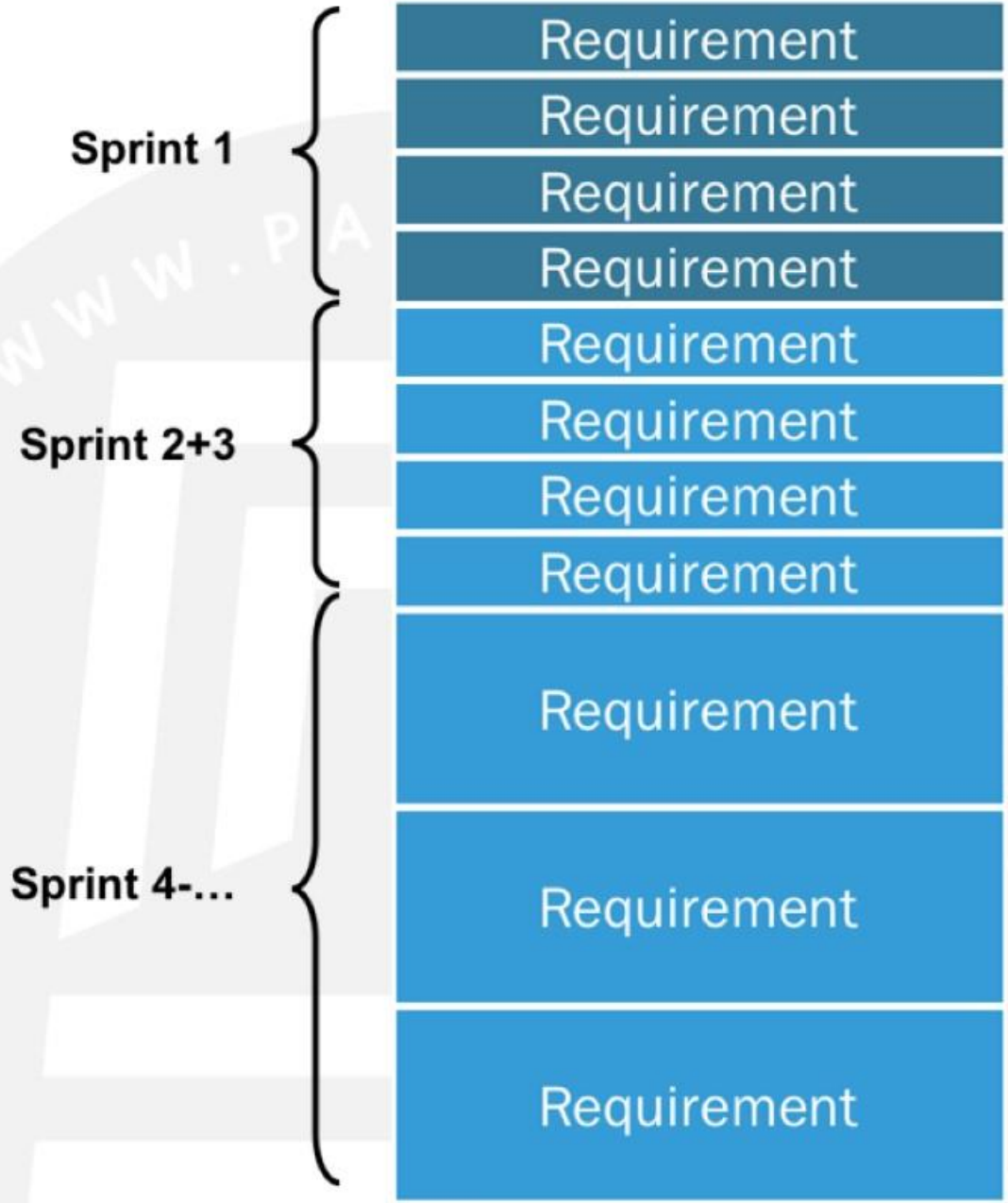
- Scrum Artifacts are specific information repositories designed to maximize transparency and foster understanding among Scrum Team members and stakeholders.
- The primary Scrum Artifacts are the Product Backlog, the Sprint Backlog, and the Increment.



Product Backlog

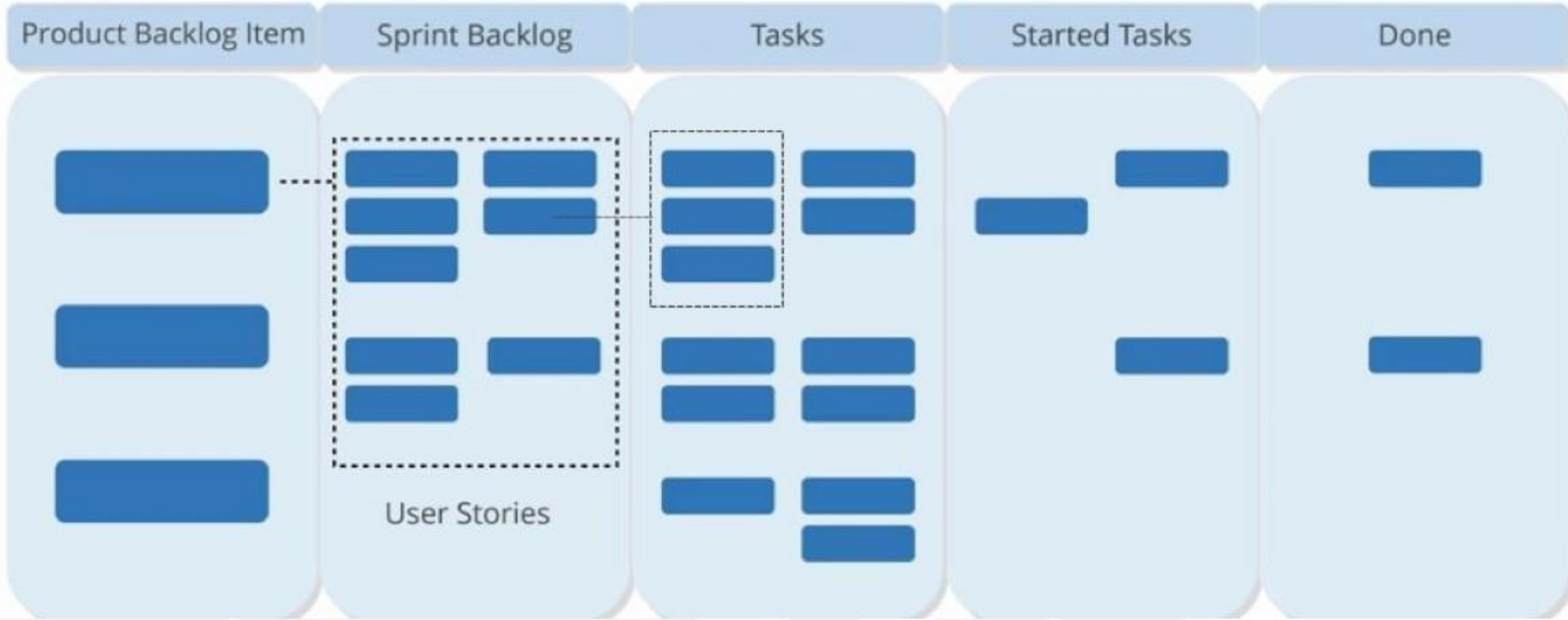
It is a **prioritized** list of **"requirements"** that is created and maintained by the **Product Owner** of a product.

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



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.



Timeboxing

Planning

8 hours for a 1-month sprint; 2 hours or less for a 1-week sprint

Sprint

Cadence of 1 to 4 weeks

Daily Scrum

15 minutes

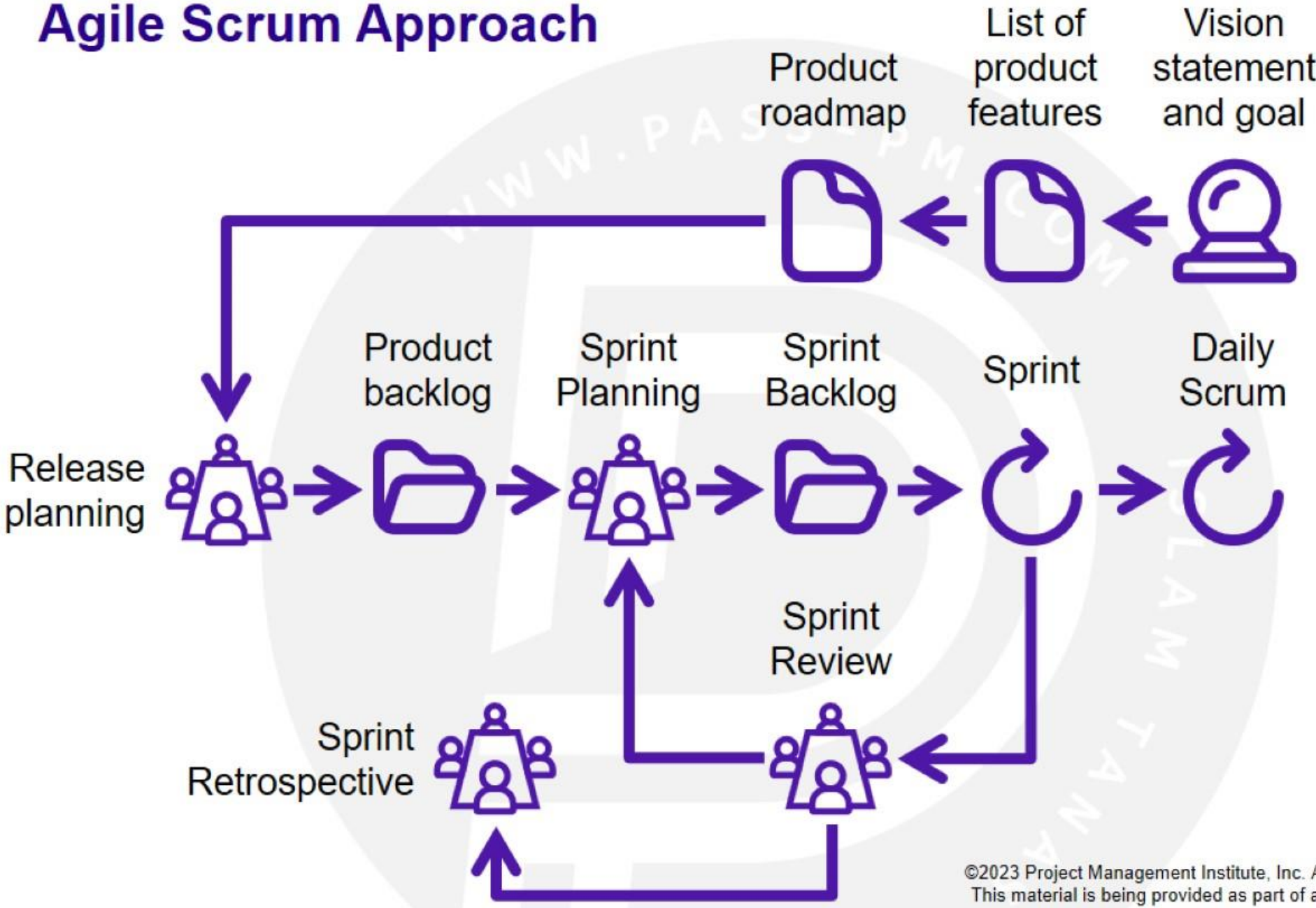
Sprint Review

1 hour for every week of sprint length

Sprint Retrospective

45 minutes for every week of sprint length

Agile Scrum Approach



Adaptive or Predictive?

A company wants to create a website to sells concert tickets.

Would you use a plan-based **predictive** **or** **adaptive** approach?

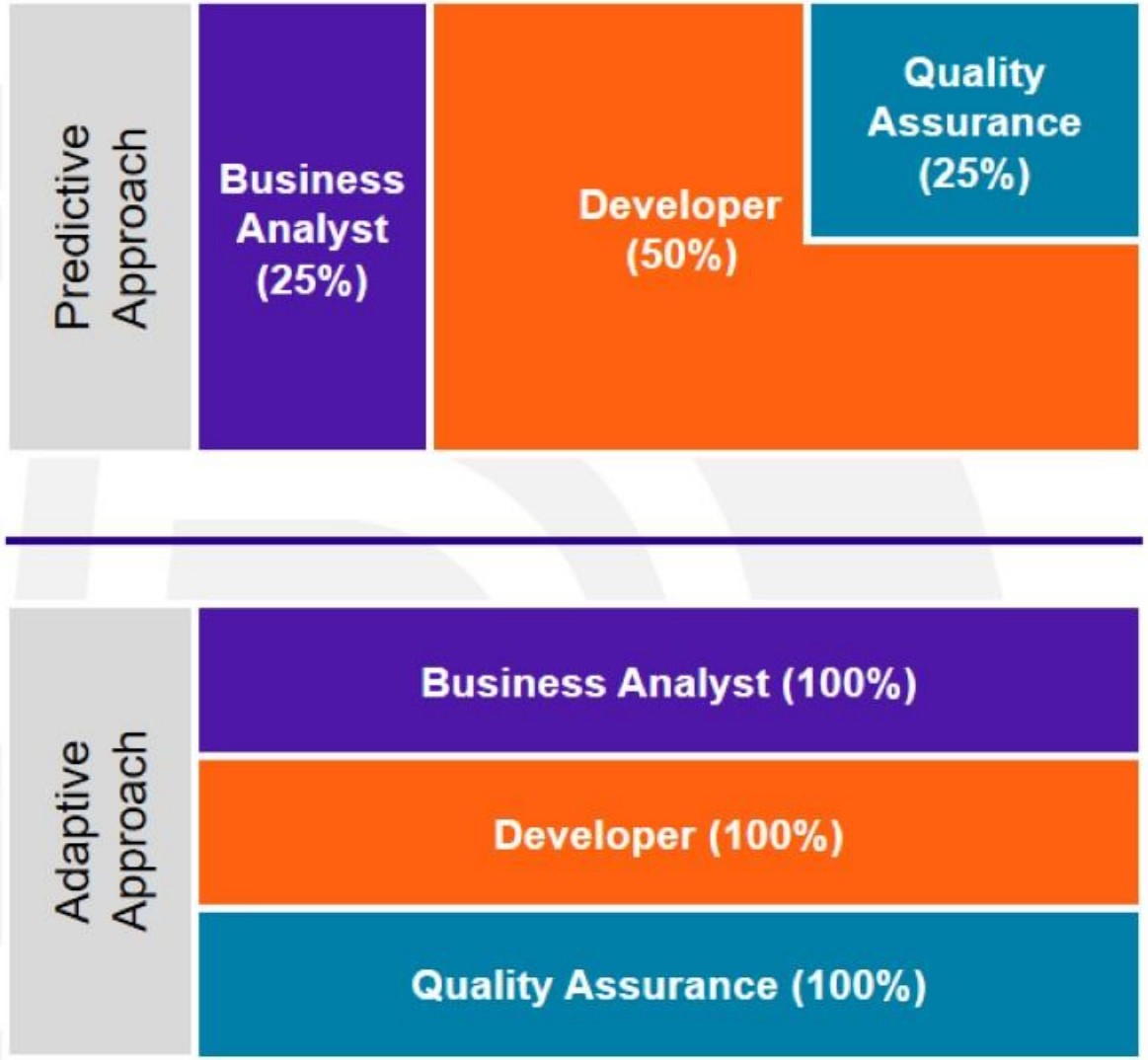
Team Structure in Adaptive Projects

T-Shaped Skills

- T-shaped employees are highly skilled in their main area.
- They also have knowledge and experience in other areas.
- And they're willing to learn and work with experts to improve their skill set.

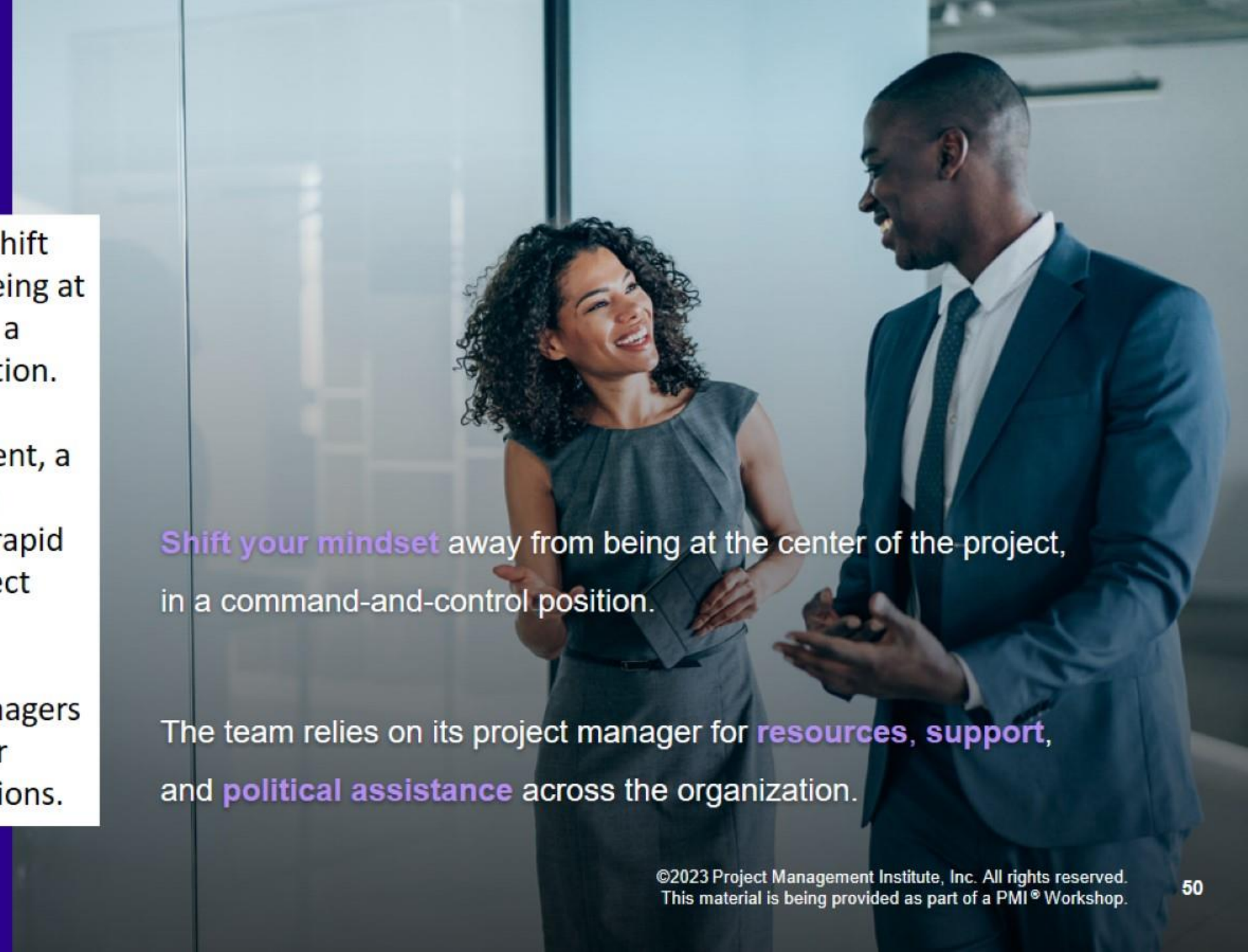
Team Roles in Predictive and Adaptive Approaches

- In the predictive approach, clear roles are assigned to team members.
- Adaptive projects emphasis on cross-functional team roles, with the team sharing responsibility for tasks.



Servant Leadership

- Project managers need to shift their mindset away from being at the center of the project in a control and command position.
- In a high-change environment, a development team is often better positioned to make rapid decisions and achieve project goals successfully.
- Otherwise, the project managers can become a bottleneck or constraint for making decisions.



Shift your mindset away from being at the center of the project, in a command-and-control position.

The team relies on its project manager for **resources**, **support**, and **political assistance** across the organization.

Servant Leader Behaviors

Obstacle removal

Diversion shield

Encouragement and development

The Structure and Culture of Adaptive Teams

Predictive vs. Adaptive Culture

Predictive

Hierarchical, centralized management approach

Management team manages; team members do the work

Adaptive

Distributed management approach

May self organize; management role may shift among team members

Factors that Contribute to High-Performing Teams

- Open communication
- Shared understanding
- Shared ownership
- Trust
- Collaboration
- Adaptability
- Resilience
- Empowerment
- Recognition
- Colocation
- Limited team size
- Experienced members

Advantages of Experienced Team Members

- Collaborate
- Define more stable project goals
- Make more reliable predictions and then honor due dates
- Finish valuable work more efficiently
- Apply previous lessons learned to avoid common pitfalls
- Communicate more efficiently and effectively within the team and with other stakeholders
- Identify and evaluate central project risks

Value-Driven Delivery

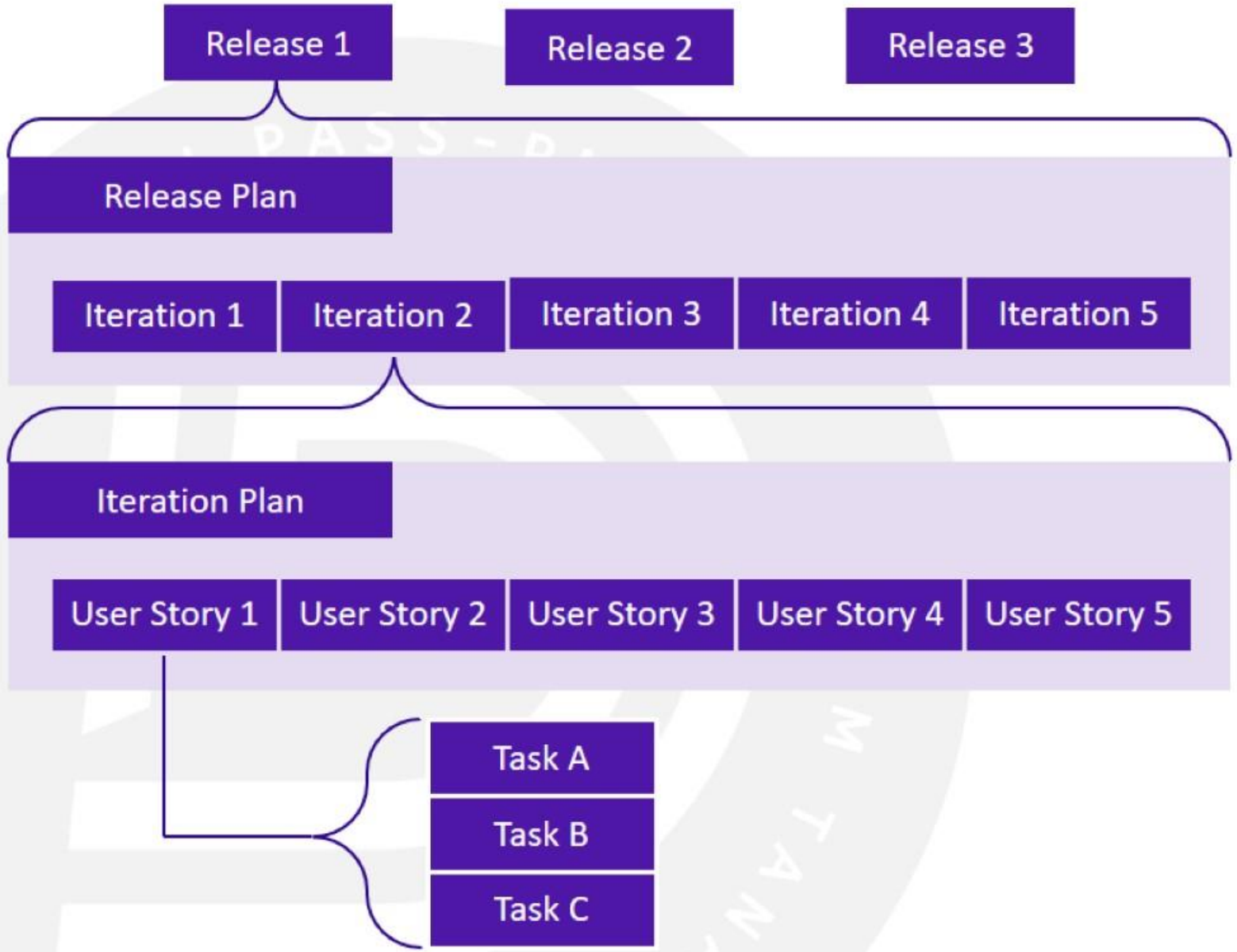
If an organization supports **value-driven delivery** and can embrace the core principles of the *Agile Manifesto*, it can successfully use an adaptive approach.

Core Principles That Support Value-Driven Approaches

Core Principles	Value-Driven Approaches
Value-based prioritization	• Customer-valued prioritization is important. • Identify what should and should not be done in the limited time.
Delivery cadence	• Time is viewed as a limiting constraint. • Timeboxing provides a cadence for all stakeholders to work and contribute.
Iterative and incremental delivery	• An opportunity must exist to verify and validate requirements. • The customer might not be able to identify all the requirements at the start of the project.
Self-organization	• Adaptive teams are empowered. • In a change-driven and timeboxed project environment, teams need to react quickly to opportunities and challenges.

Adaptive Planning

Collect and Decompose Requirements

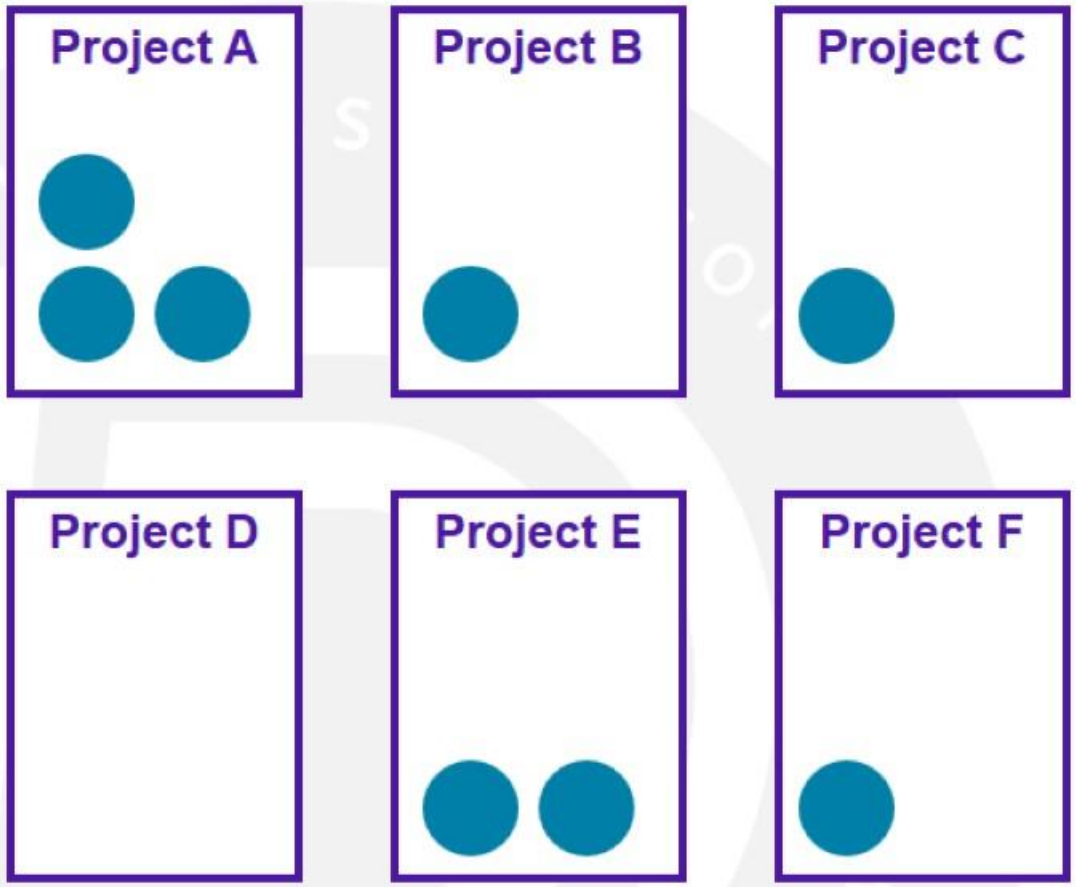


Prioritization Techniques

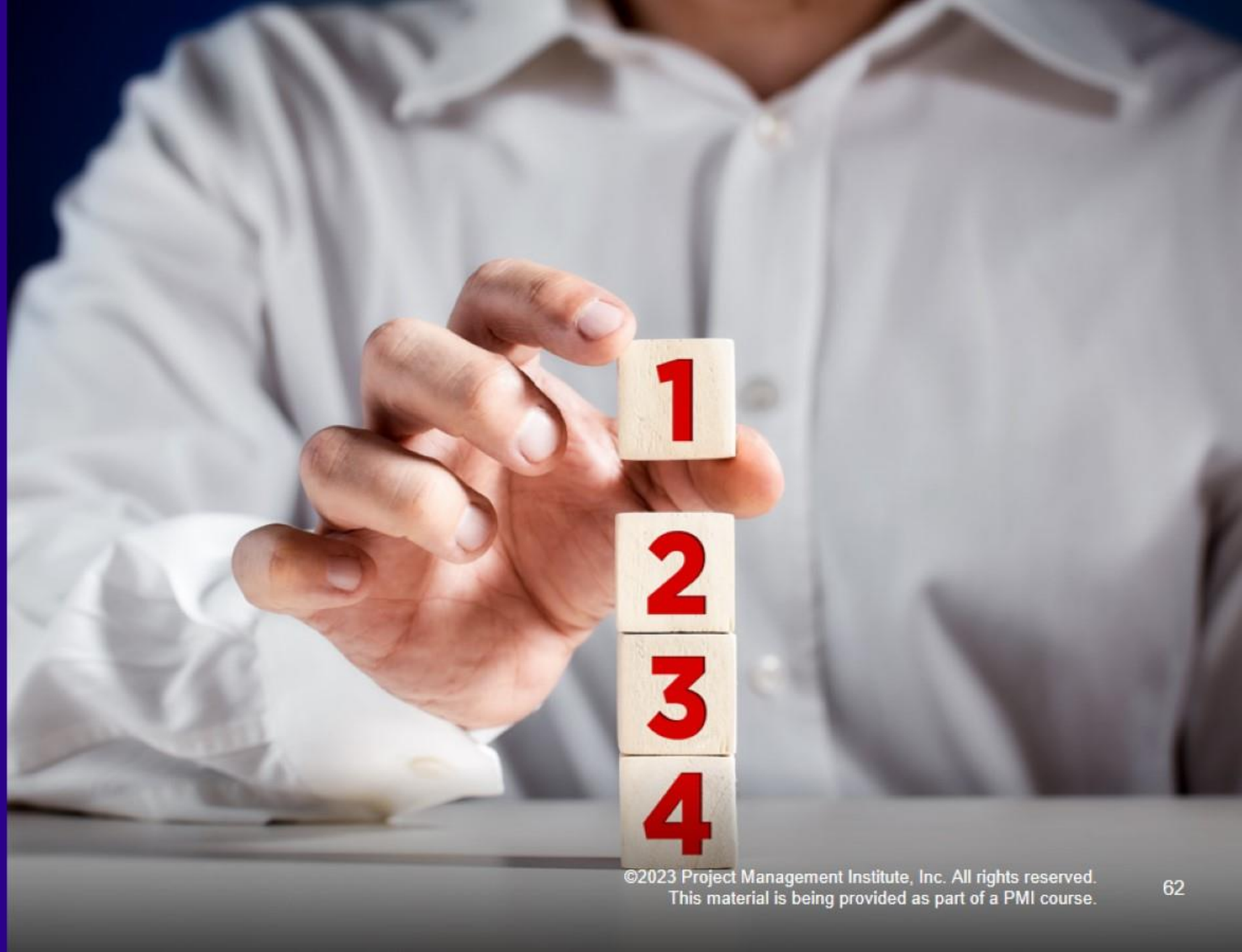
The team must know what its **priorities** are—and have a hand in setting those priorities.

Dot Voting or Multi-Voting Schemes

Each team member is given a small number of dots—often using sticky notes—and uses them as votes to indicate the size and importance of an item.



What Gets Prioritized?



Project Artifacts That Need to Be Prioritized

Product backlog

Release backlog

Sprint backlog

Scrum tasks

Develop a Release Plan

- The entire team's focus is to identify and deliver the **most valuable features** at all times.

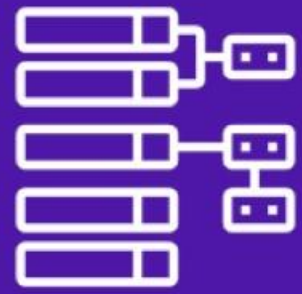
- This is known as delivering a **minimum viable product, or "MVP,"** which essentially identifies the fewest number of features or requirements that are both functional and usable.

Estimate Effort with Story Points

Absolute estimates

Generate explicit actual quantities

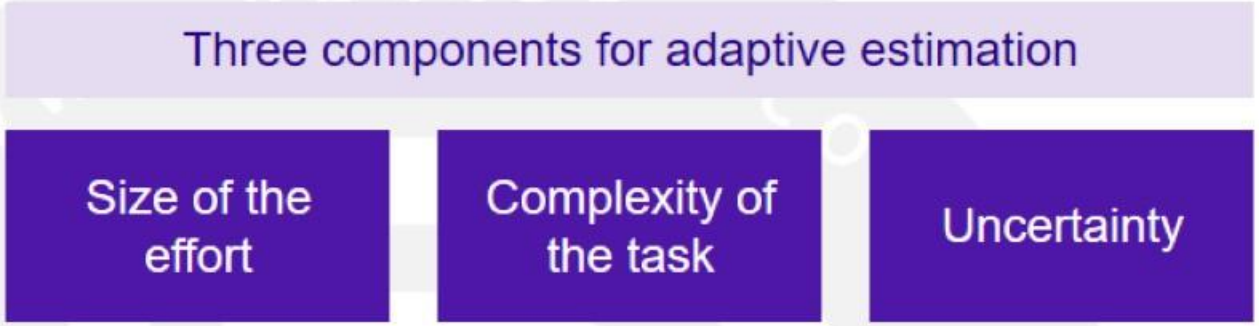
Relative estimates



Shown in comparison to other estimates and have meaning only within a given context

Estimate Effort with Story Points

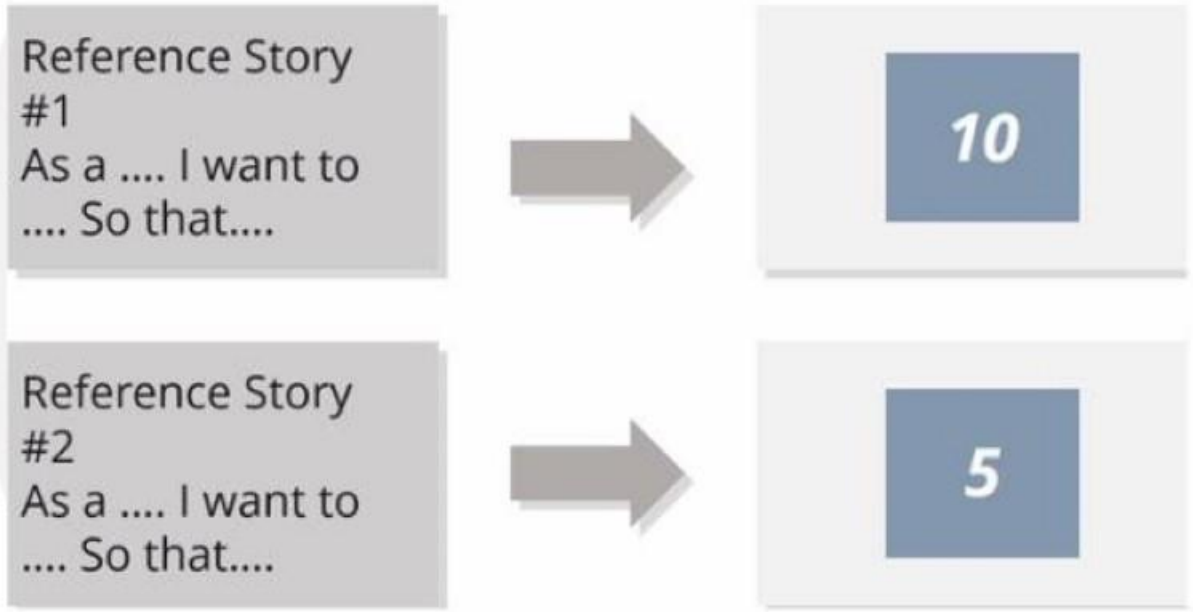
- Adaptive estimation begins with a list of user stories and team members participating in the estimation process.
- Everyone, including the product owner and external experts, is invited to review all the user stories and rank-order them, keeping three attributes in mind: size, complexity, and uncertainty.



Story Points

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

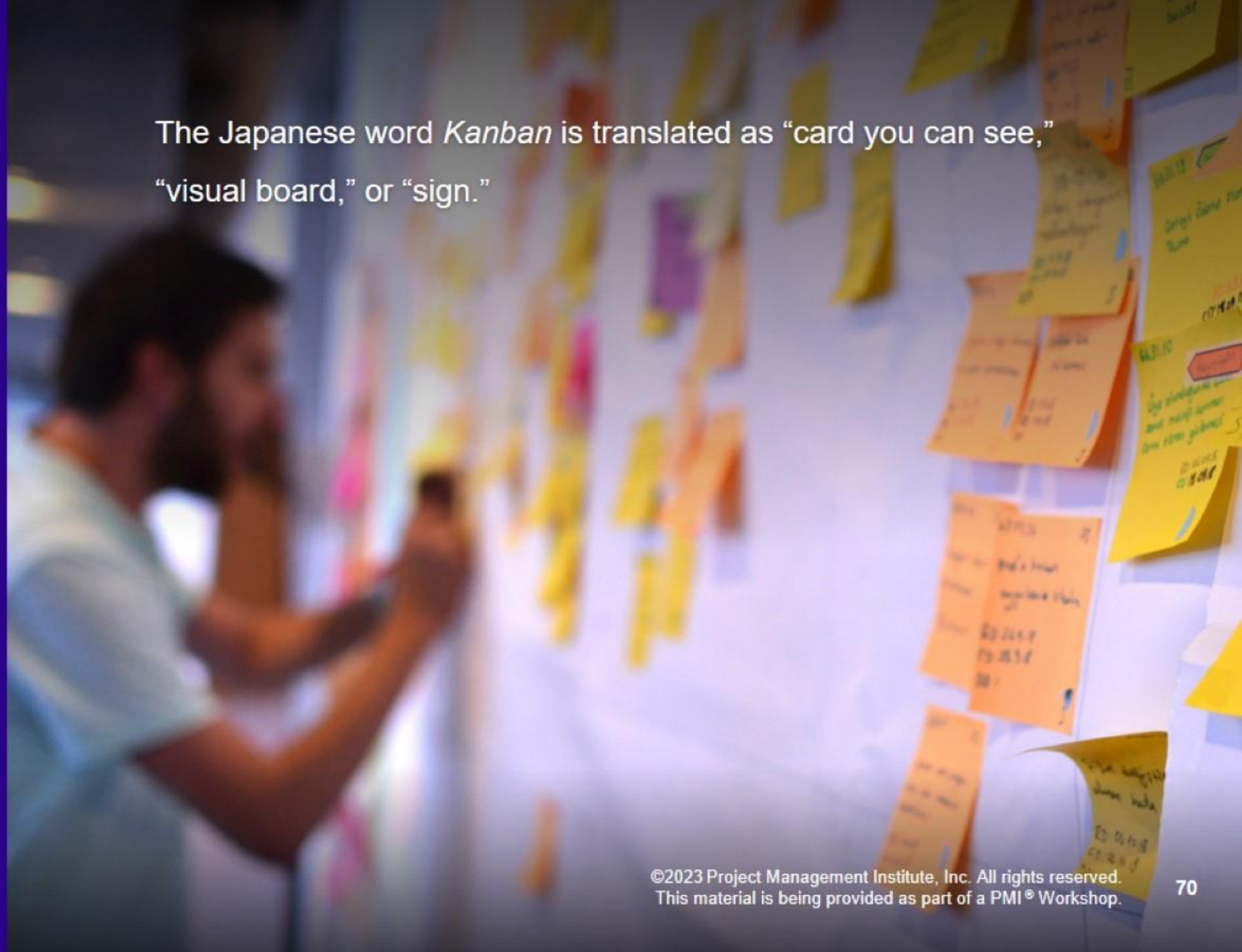
Example



Overview of Other Adaptive Frameworks

Kanban

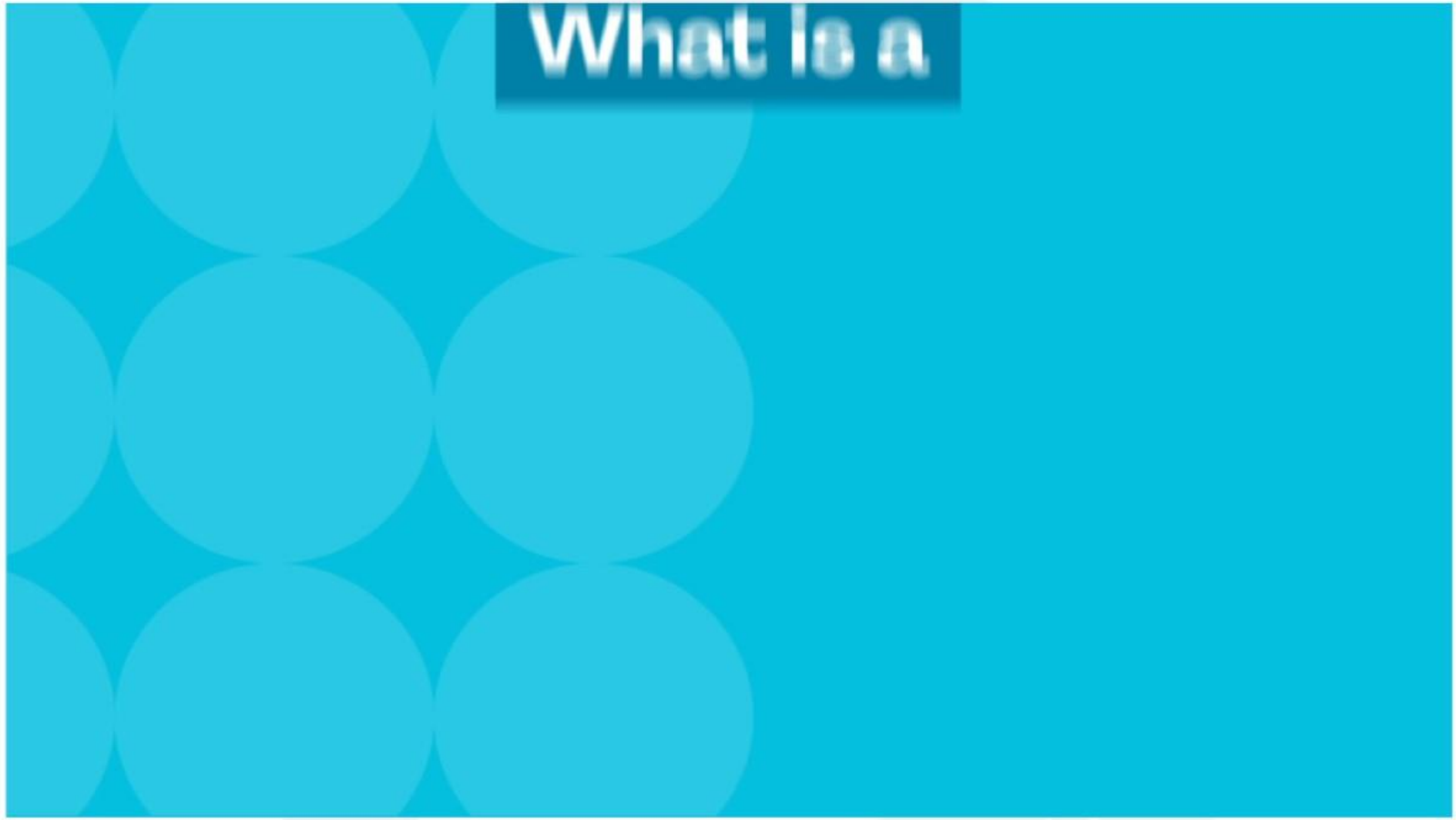
Kanban (看板)



The Japanese word *Kanban* is translated as “card you can see,” “visual board,” or “sign.”

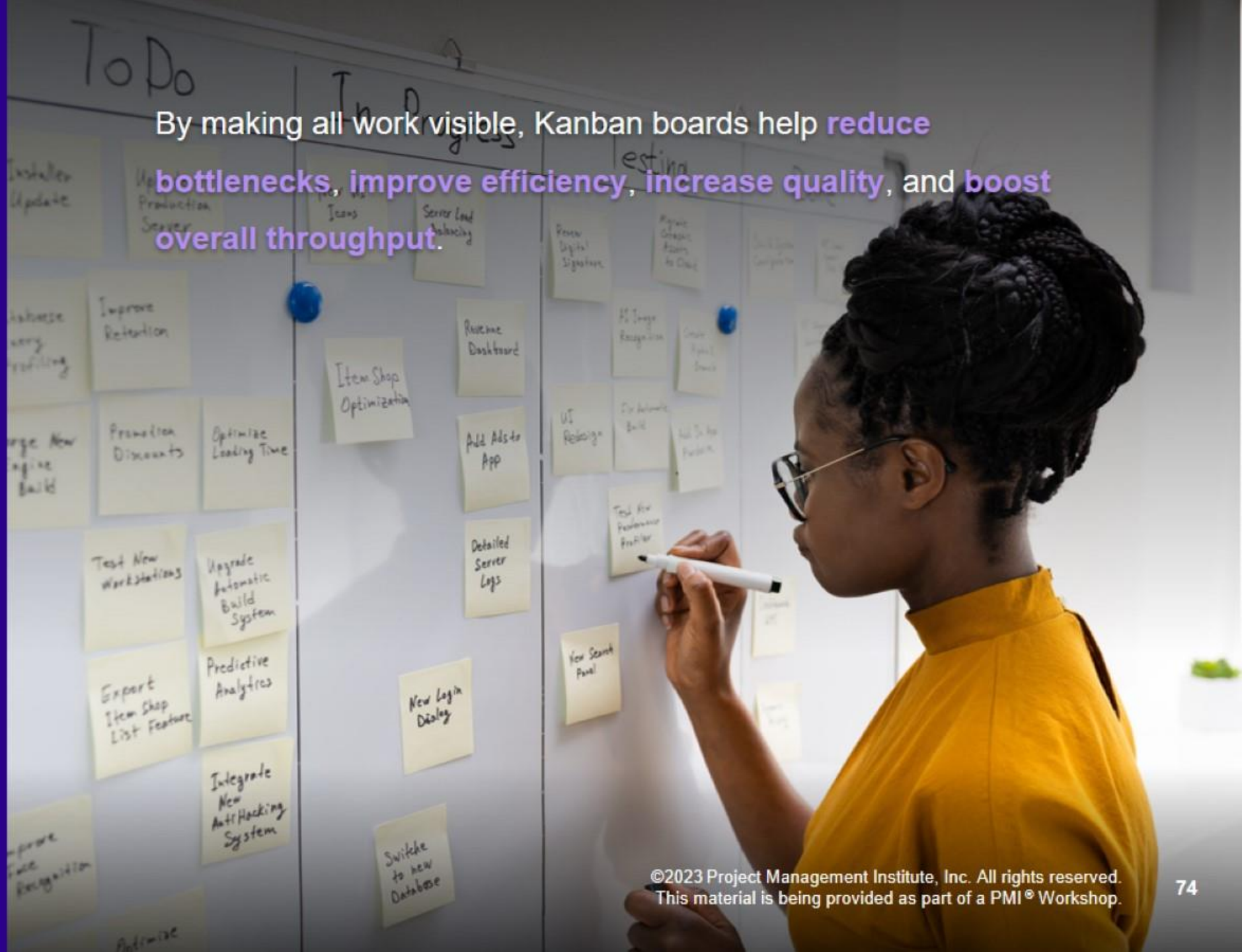
Kanban Board

What Is a Kanban Board?



Workflow Focus

By making all work visible, Kanban boards help **reduce bottlenecks**, **improve efficiency**, **increase quality**, and **boost overall throughput**.



Comparing Kanban and Scrum

Factor	Kanban	Scrum
Team management	Within existing hierarchical structures	Self-management
Process improvement	Process changes can be made anytime, and immediate modifications are actively encouraged	Process improvements are generally made after the sprint retrospective
Productivity measurement	Uses cycle time, lead time, and work in progress to assess productivity	Uses velocity and burndown rates to assess productivity
Life cycle	Flow-based agile	Iteration-based agile
Time focus	The primary focus is on cycle time and lead time rather than on meeting due dates; it is flow-oriented on continuous deliveries.	Sprints are typically 1 to 4 weeks in length, and a product increment or a version of the product is delivered at the end of each sprint.
Commonalities	• Both encourage process improvement • Both break projects into smaller processes that are iterated • Both encourage team collaboration	

Lean

Roots of Agile

The Agile Manifesto has its roots in the **Lean** methodology and approach.

Lean was born in post–World War II Japan and introduced to the automotive industry in the United States in the form of the **Toyota Production System**.

Lean is also the foundation for two other popular agile frameworks: **Scrum** and **Kanban**.

Features of Lean

Eliminate waste

Build quality

Fast delivery

Delay making decisions

Empower team

Optimize the whole

Extreme Programming (XP)

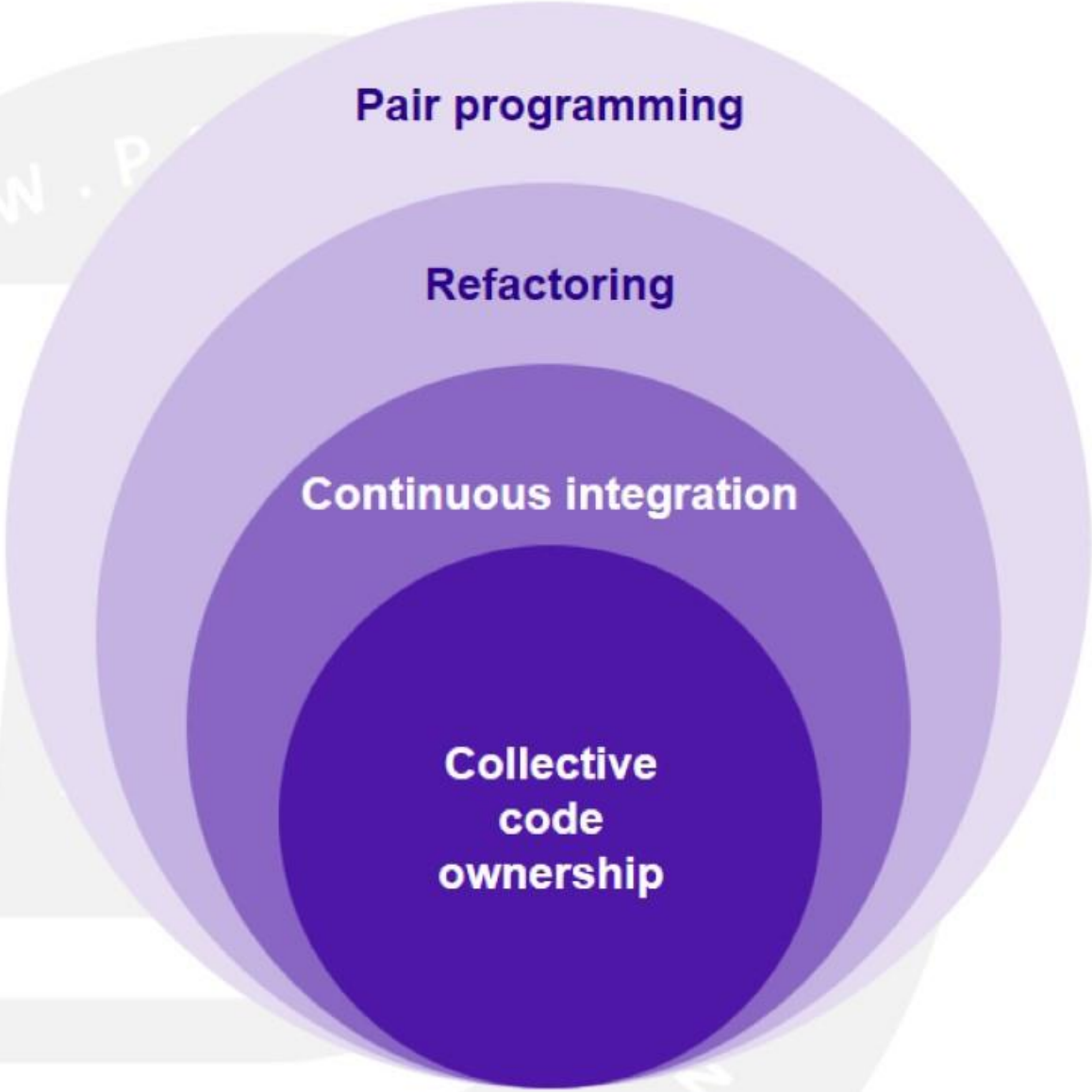
Extreme Programming

Abbreviated as “XP” and sometimes shown as “eXtreme”

Iterative, incremental framework popular for software development

Shares many features with Scrum

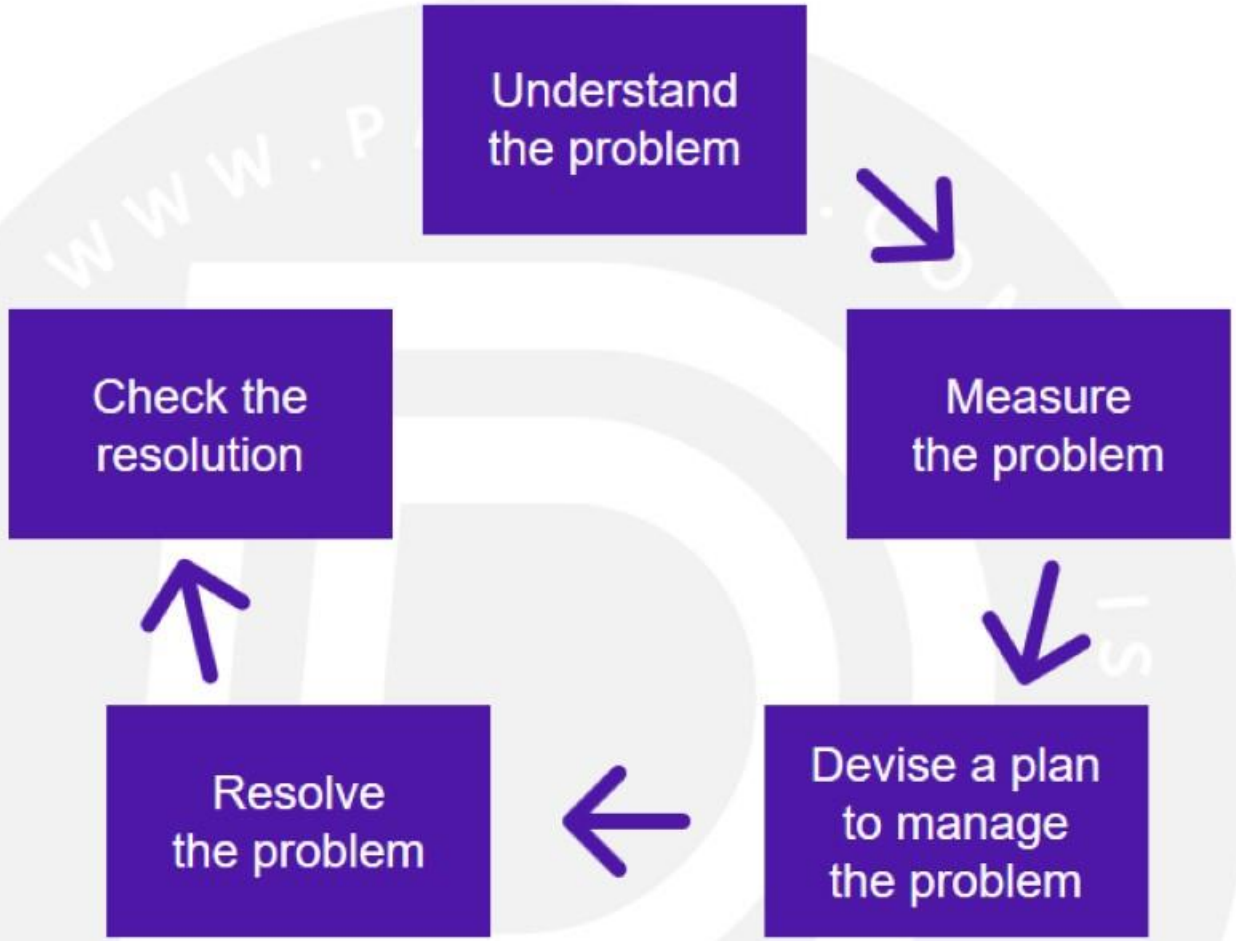
Four Critical Concepts of Extreme Programming



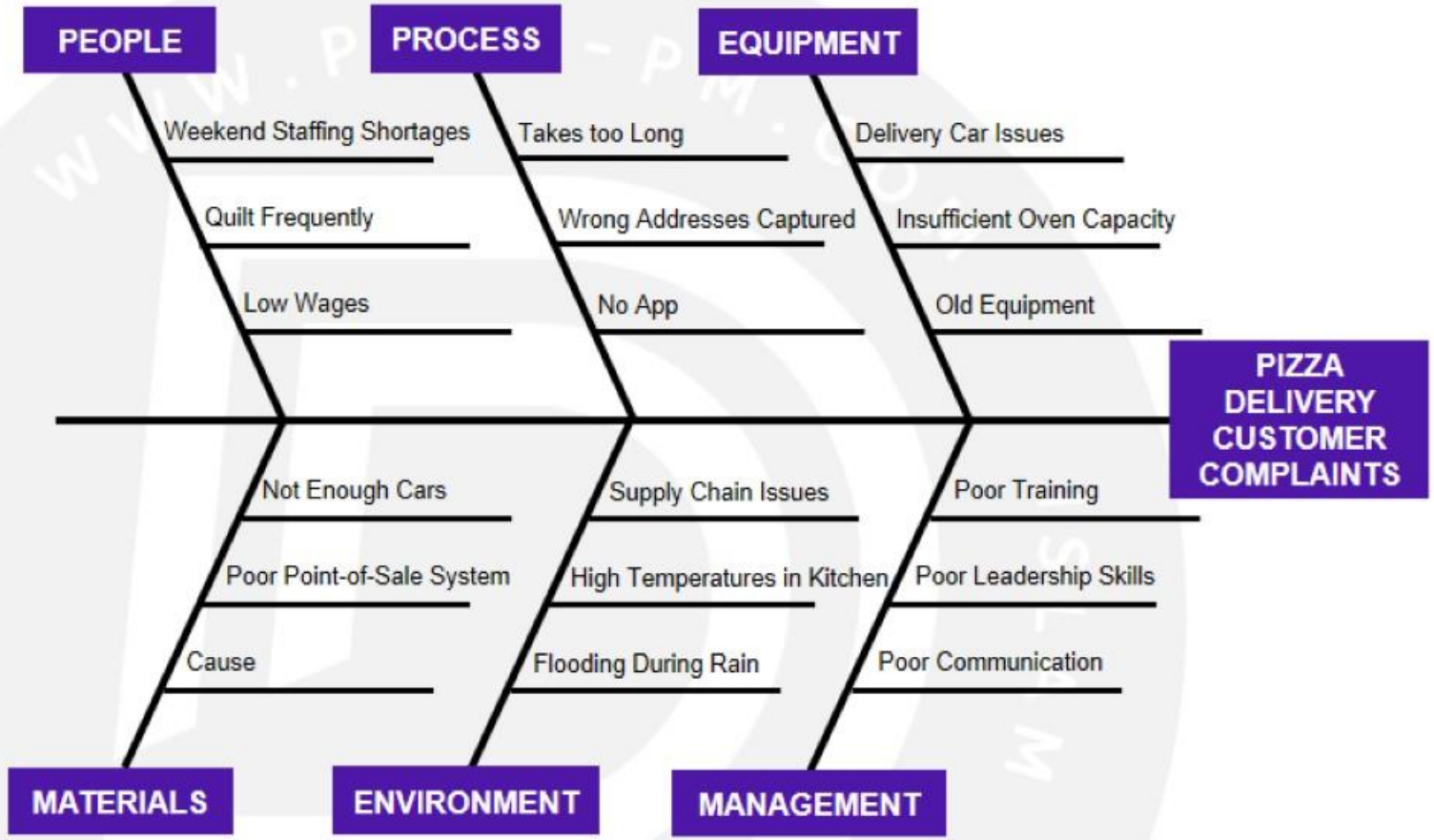
Measurement, Tracking, and Managing Uncertainty

Detect and Resolve Problems

Understanding and Resolving Problems



Tools and Techniques for Understanding and Measurement



Measuring Performance

What Makes a Metric Effective?

Specific

Measurable

Achievable

Relevant

Timely

Benefits of Metrics

Tracking progress

Supporting decision making

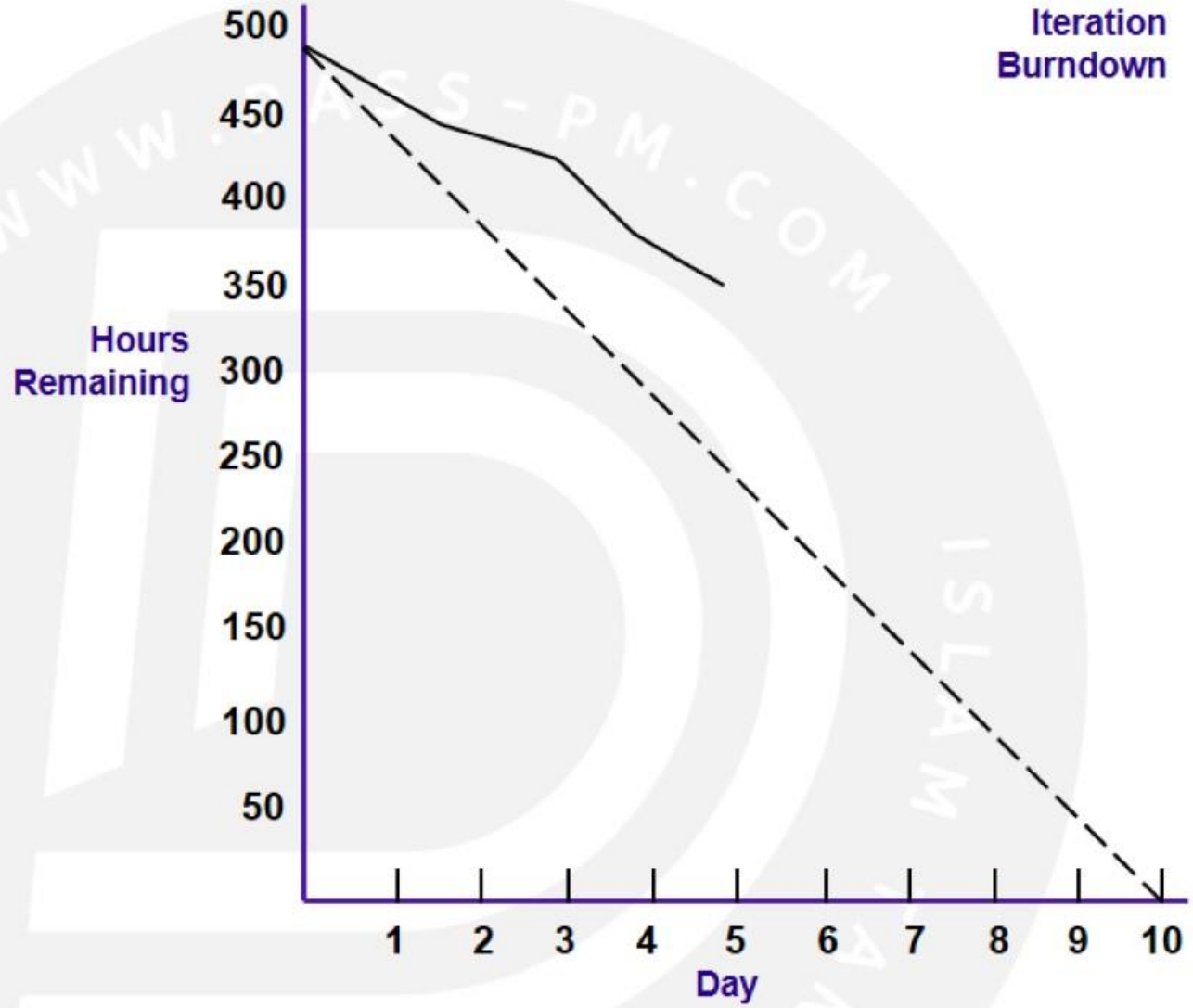
Communicating Project Metrics

Burndown chart

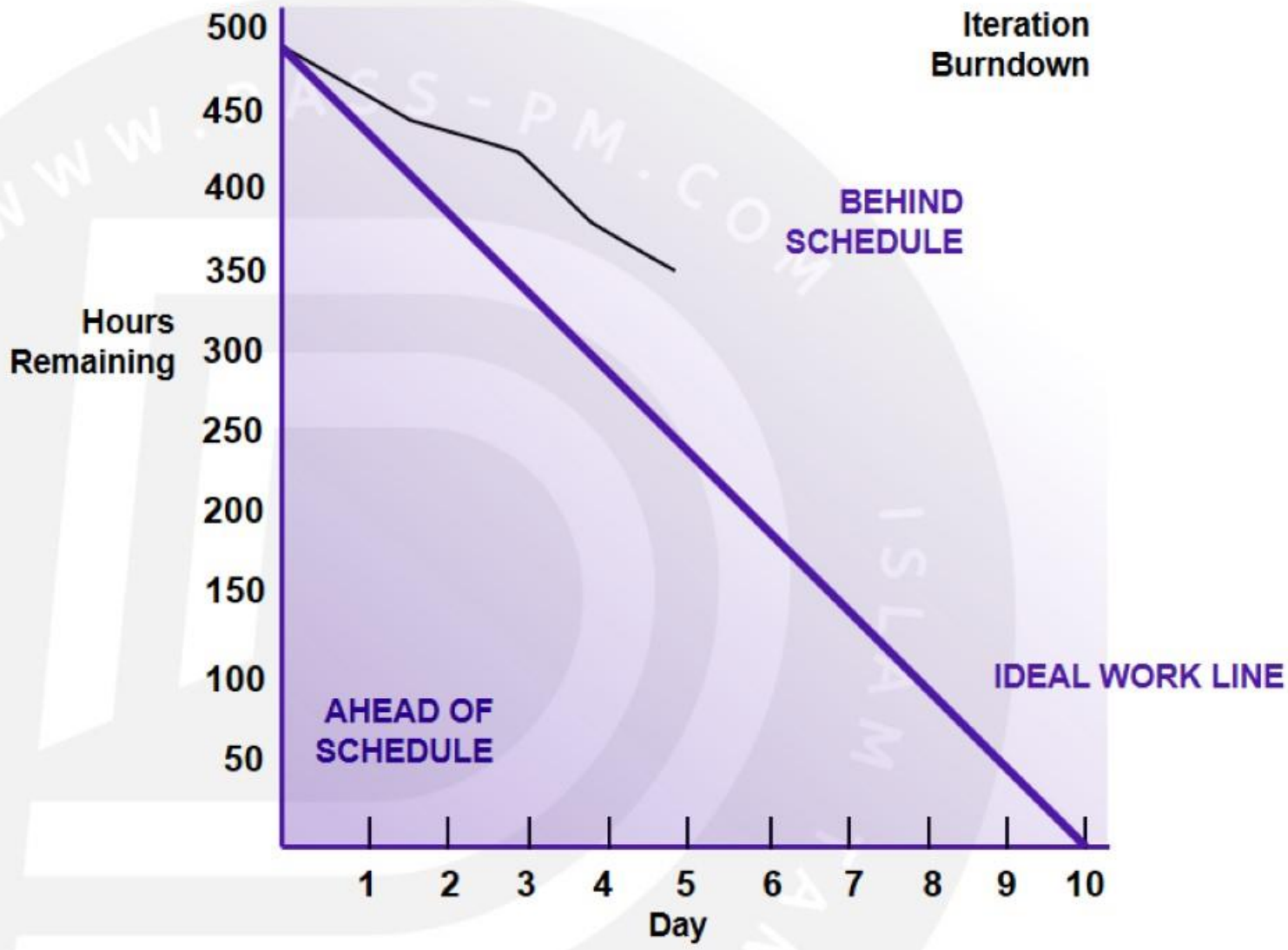
Velocity chart

Cumulative flow diagram

Burndown Chart



Ideal Work Line

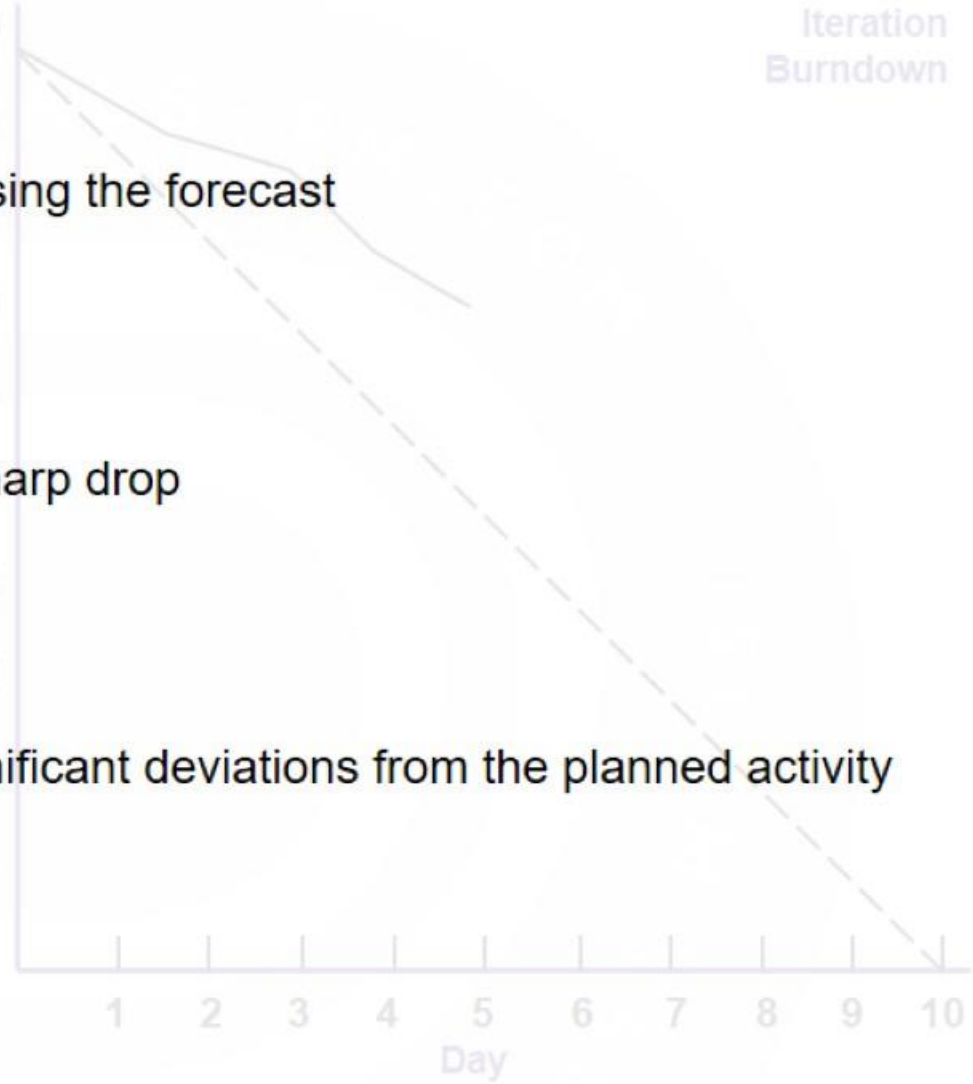


Learning From Burndown Charts

Missing the forecast

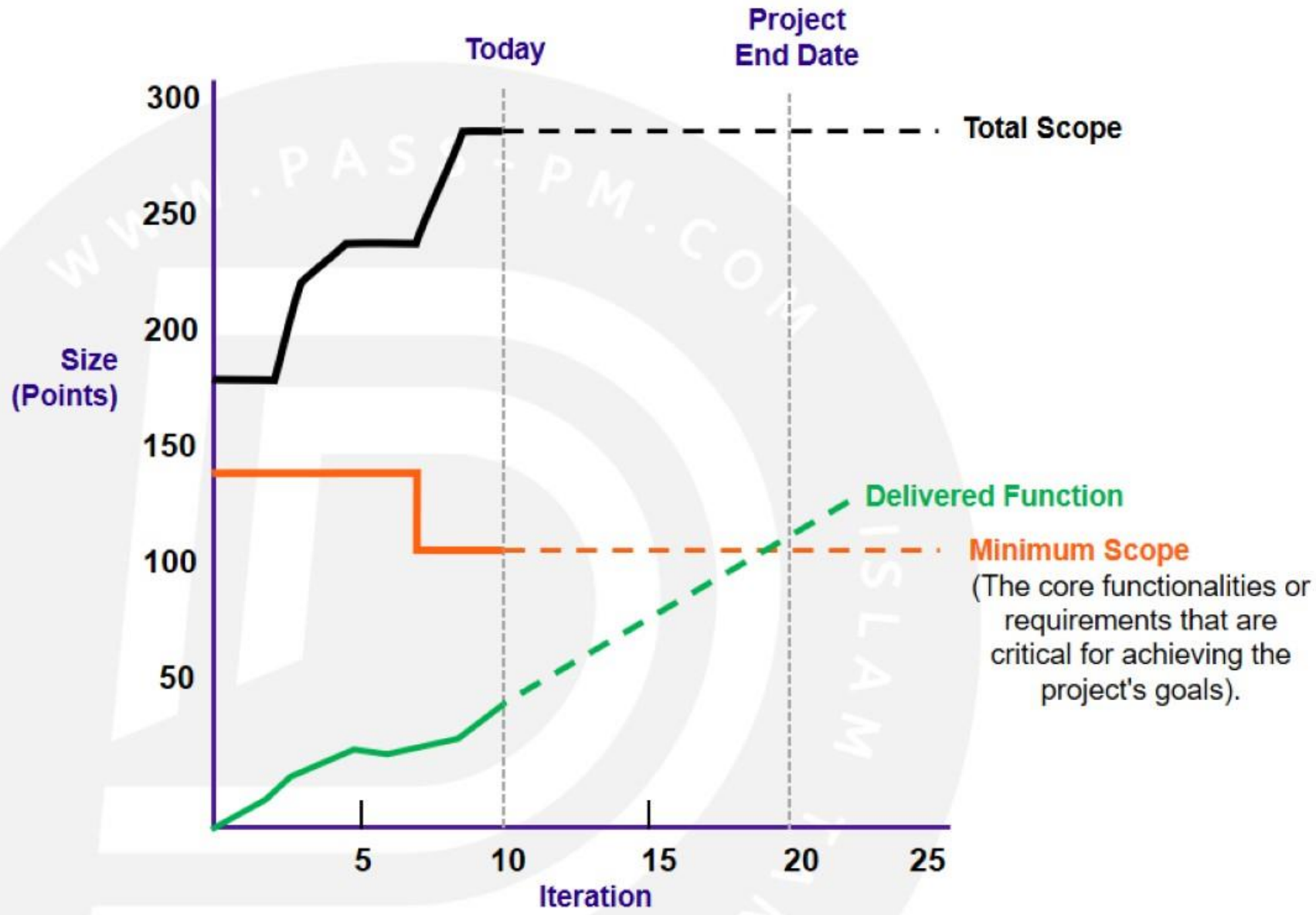
A sharp drop

Significant deviations from the planned activity



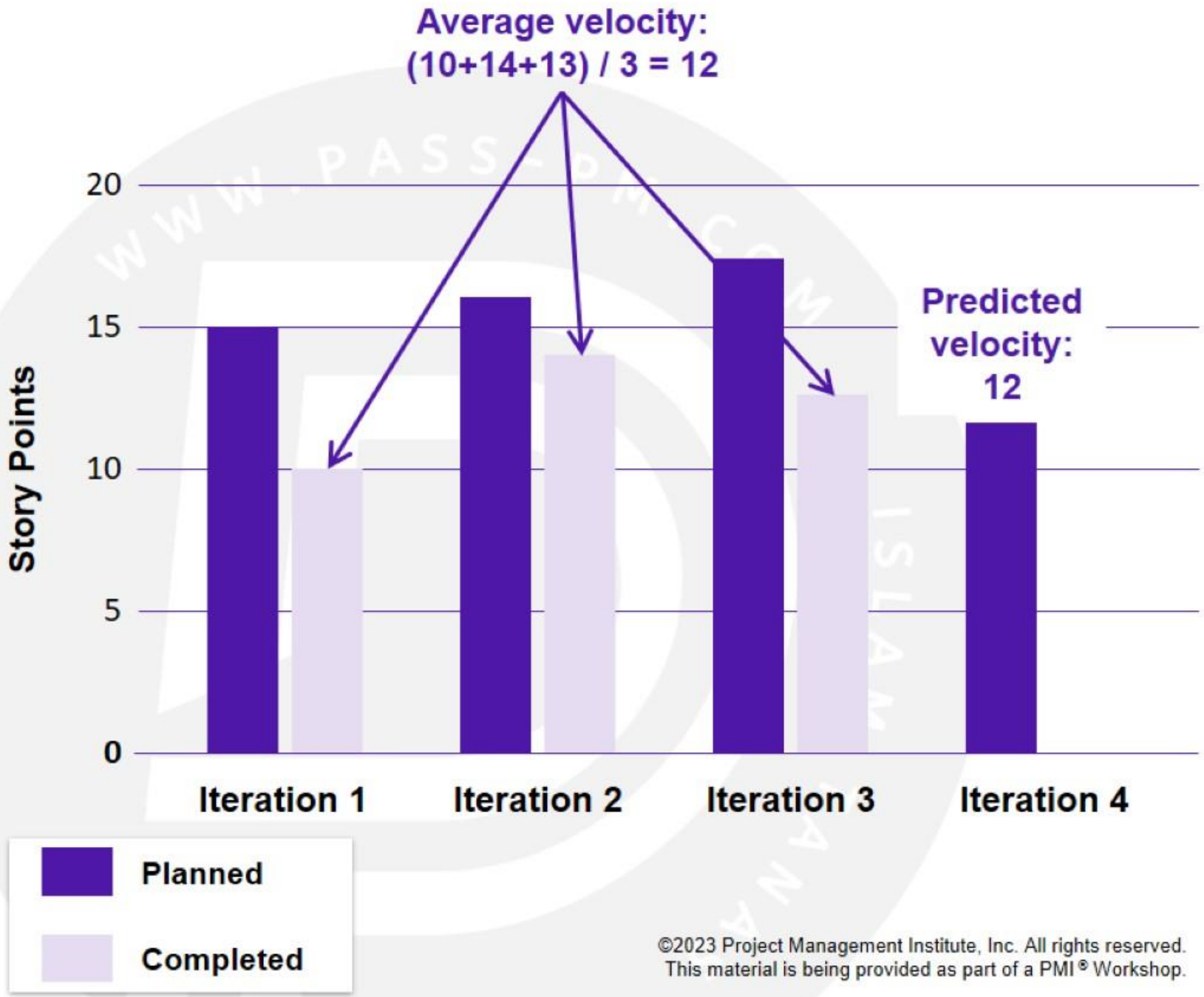
Burnup Charts

- The burnup chart shows the work completed instead of the work remaining.
- Disadvantages: do not show what tasks are still in progress. And they do not reflect how close the team is to completing the work.



Velocity

Predicting Velocity



Forecasting Iterations Needed

Average velocity: **12 story points**

Remaining backlog: **60 story points**

$$60 / 12 = 5$$

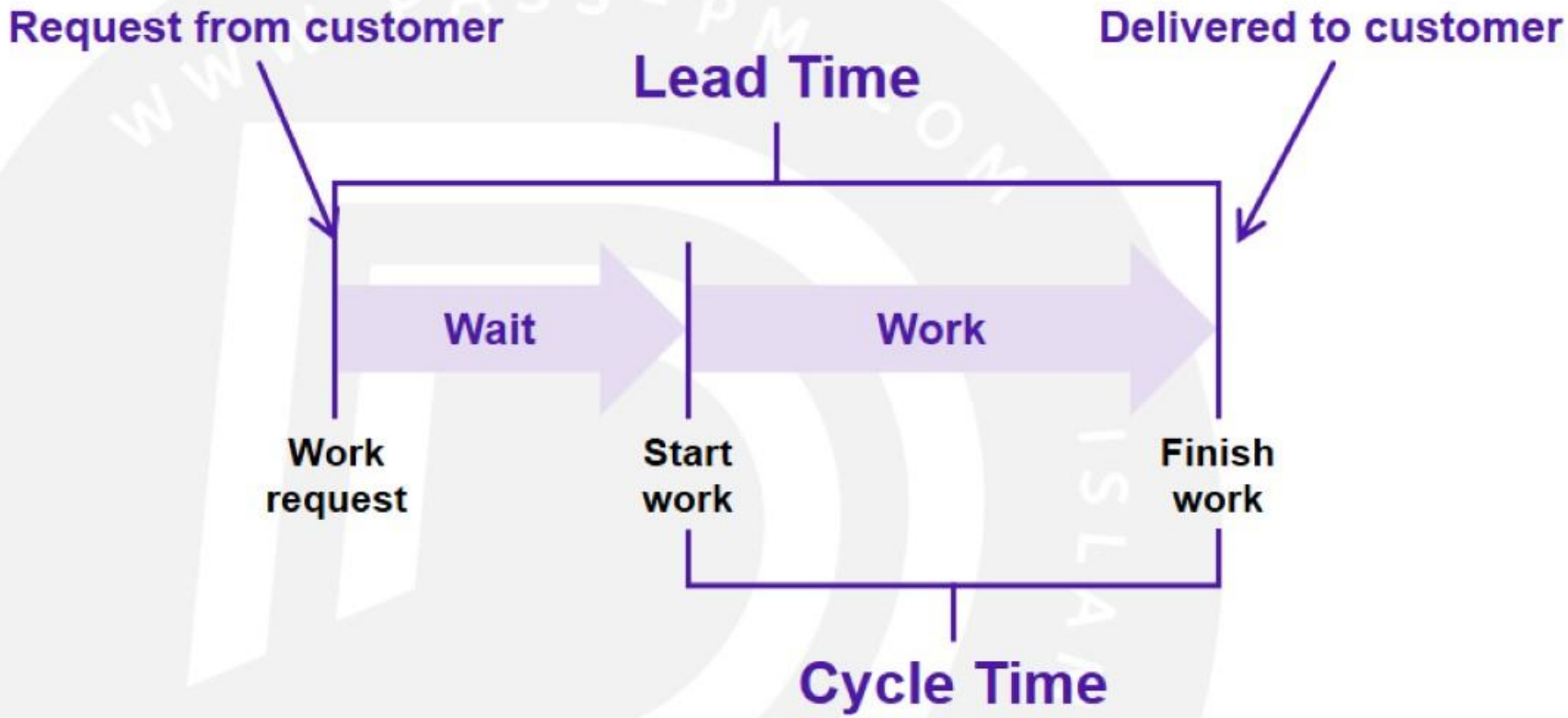
Velocity **should not** be used to **compare** the productivity of **different teams**.

Throughput

The **number of units** produced during a **specific interval**

80 cars / 8-hour shift = 10 cars per hour

Cycle Time and Lead Time



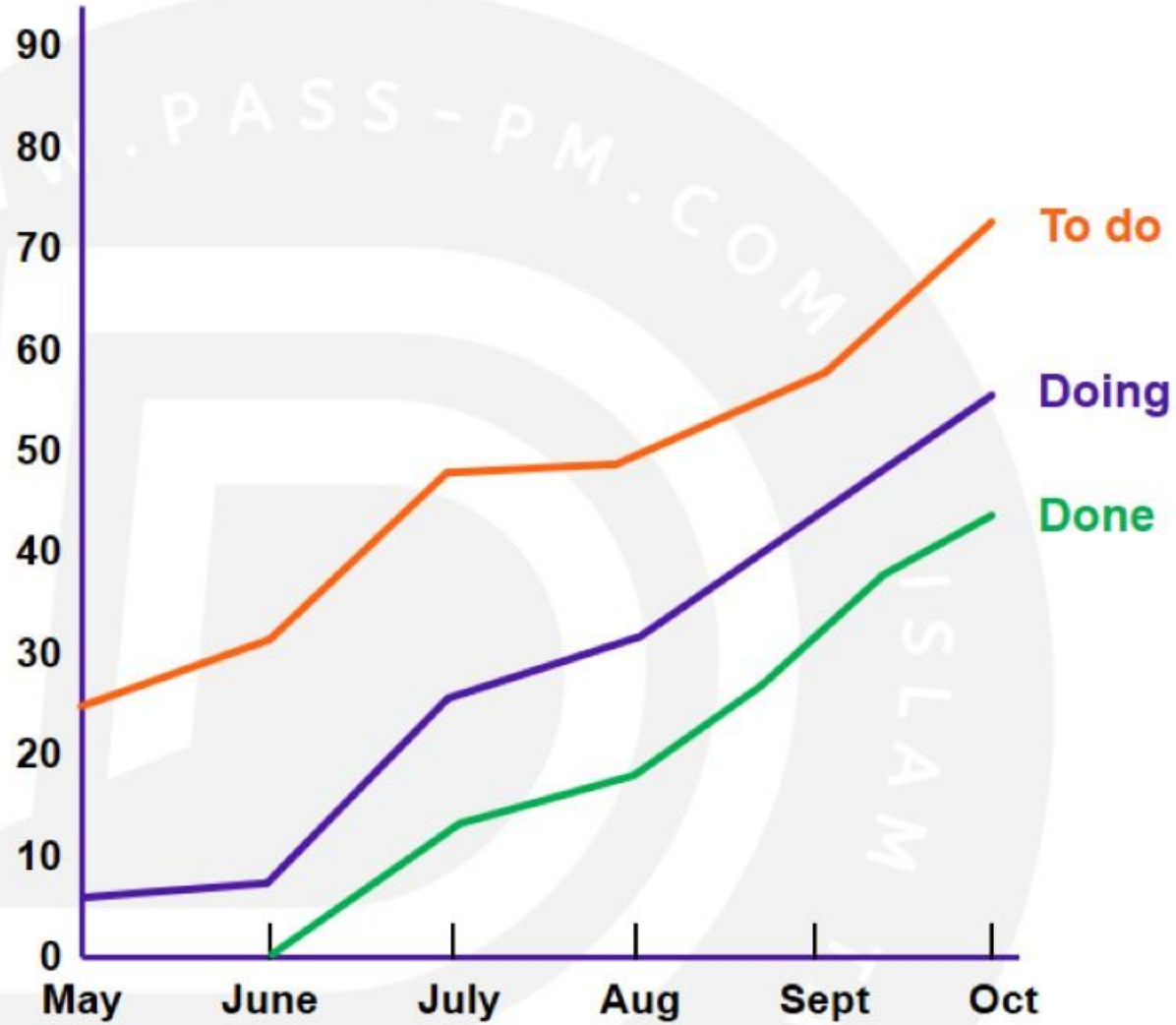
Cycle Times

Control charts are used to monitor cycle times.

Lead time is a direct function of **how many tasks are entering the system** and how often.

The cycle time is closely associated with **work in progress**, which is a count of all started tasks that have not yet been finished.

Cumulative Flow Diagram



Information Radiators

The Uncertainty Performance Domain

102

Tracking and Managing Risk in Adaptive Projects

Establish a regular rhythm of review and feedback sessions

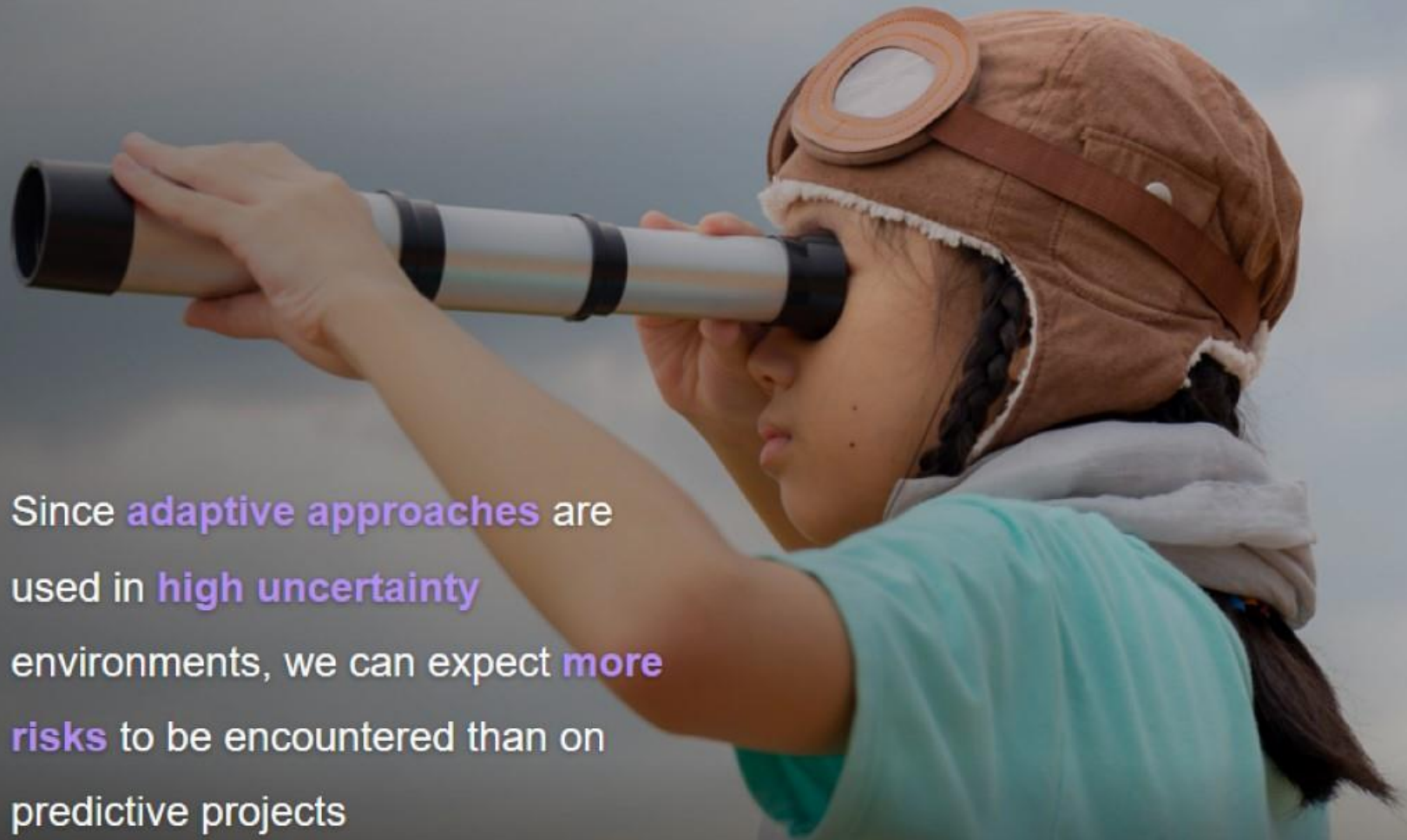
Use daily standup meetings identify potential threats and opportunities

Demonstrate product or service increments frequently

Test the effectiveness of candidate solutions with short proof-of-concept experiments or spikes

Tracking Progress to Manage Risk

Roadblocks or issues are reported **at the daily meeting** and escalated to the scrum master or project manager for resolution.



Since **adaptive approaches** are used in **high uncertainty** environments, we can expect **more risks** to be encountered than on predictive projects

Impediments List

		January			February		
ID	Risk	Impact	Probability	Severity	Impact	Probability	Impact
1	Permits not obtained	3	3	9	3	2	6
2	Site not ready	2	2	4	2	0	0
3	Early road thaw	3	2	6	3	1	3
4	…						

Calculating the Severity of Risk

Risk Severity = Impact x Probability

A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition

Wrapping Up

Summary

Detect and resolve problems

Setting priorities

Measuring performance

Dealing with uncertainty

Tracking and managing risk

Up Next:
Business Analysis Frameworks