Agile Scrum Master

Agile Way of Thinking

Learning Objectives

By the end of this lesson, you will be able to:

- Explain model and list the types of project management models for effective planning, executing, and controlling projects to achieve desired outcomes
- Define Agile and explain its fundamental principles to enhance project management and teamwork
- List the core values as outlined in the Agile Manifesto to improve collaboration and product delivery
- Perform activities on different Agile frameworks and methodologies to effectively manage and execute projects in a real-world environment

Introduction to Model

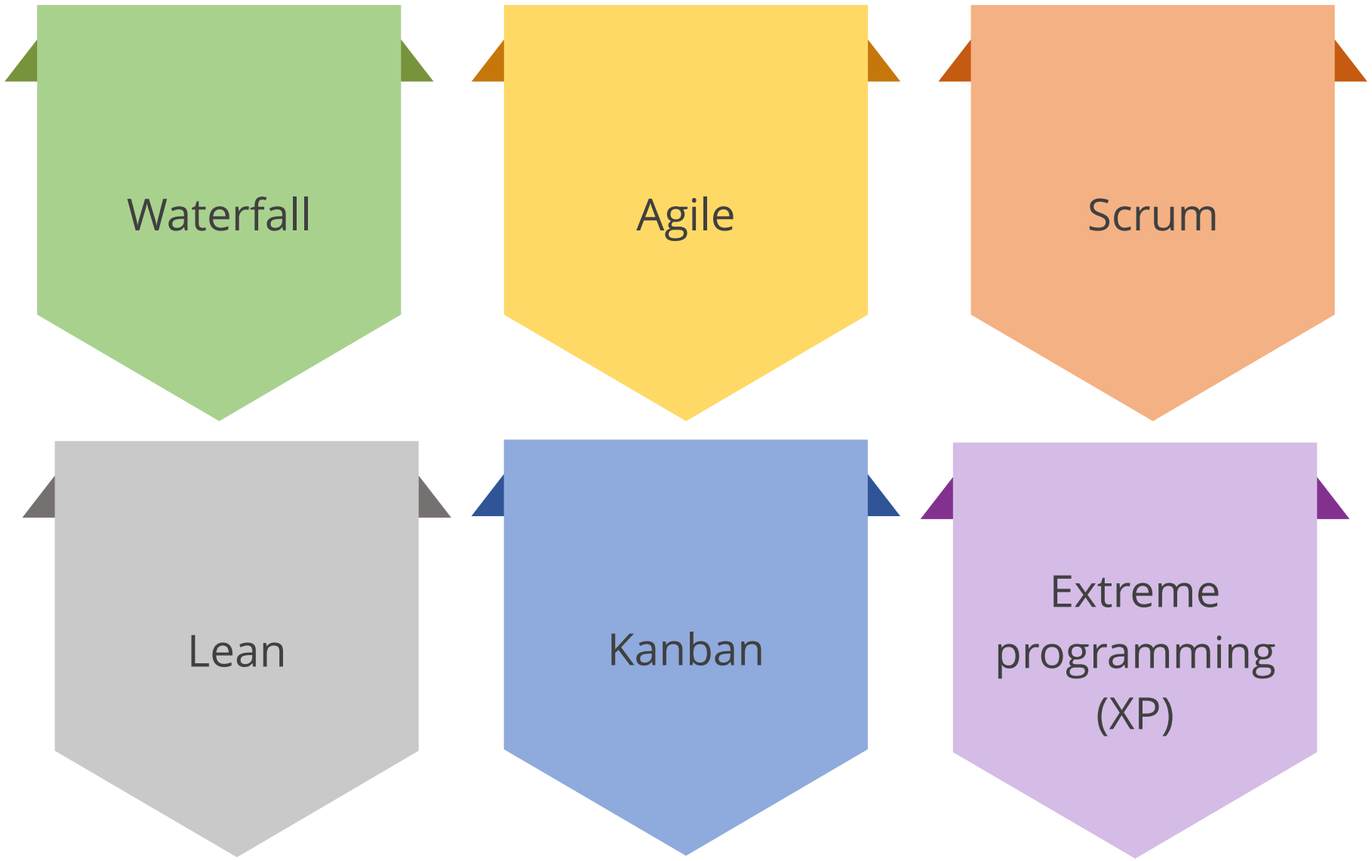
What Is a Model?

It is a thinking strategy to explain a process, framework, or phenomenon.

A project management model is a framework or approach used to plan, execute, and control a project.

Types of Project Management Models

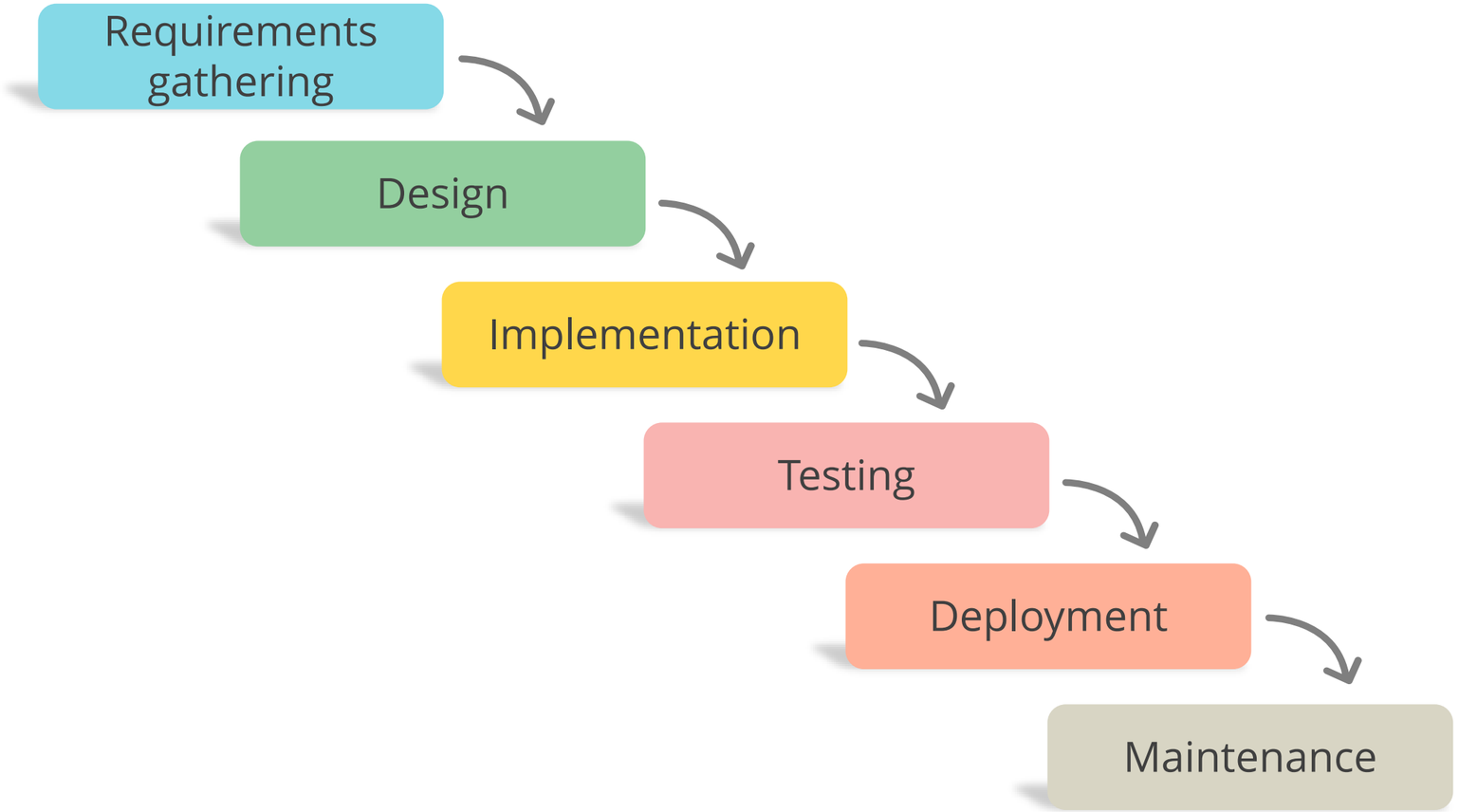
Some of the commonly used project management models are:



Introduction to Waterfall and Agile Model

Waterfall Model

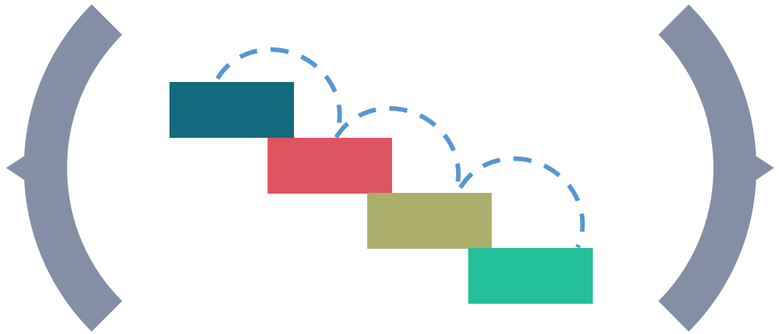
It is a traditional project management method that follows a set sequence, where each stage must be completed before moving to the next, as shown below:



Waterfall Model

In a waterfall model:

Entire scope is known and stable



Plan and design is decided up front

Waterfall Model: Benefits

Clear project structure

Set costs

Simplify tracking

Establish a replicable process

Create comprehensive project documentation

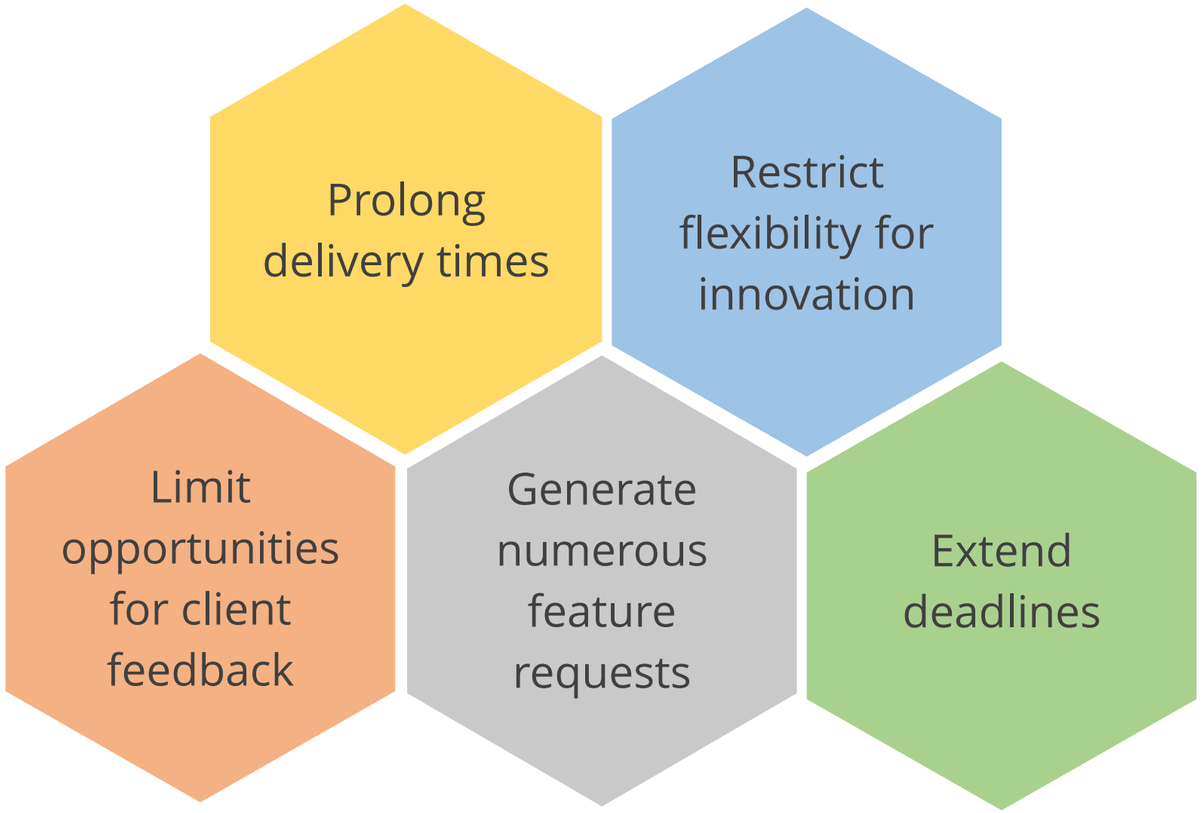
Improve risk management

Enhance responsibility and accountability

Ensure precise execution for a non-expert workforce

Reduce delays due to additional requirements

Waterfall Model: Limitations

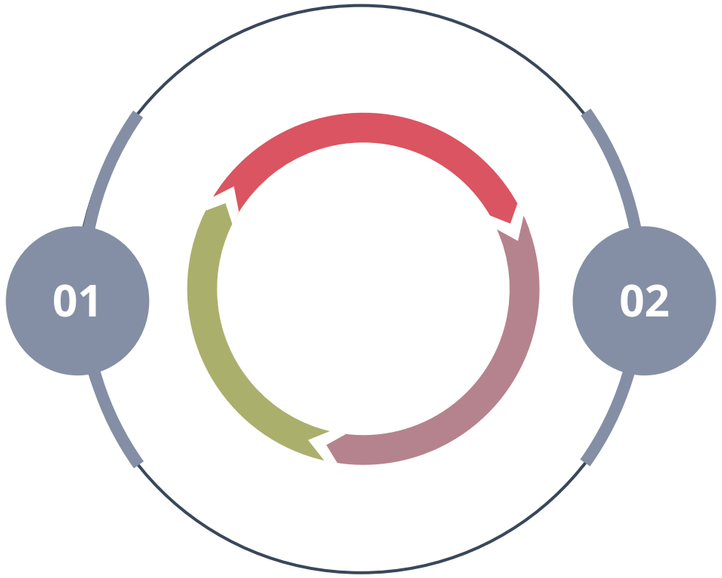


Agile Model

It refers to short increments with feedback. It was designed in response to the inflexibility of traditional methods.

It usually applies to:

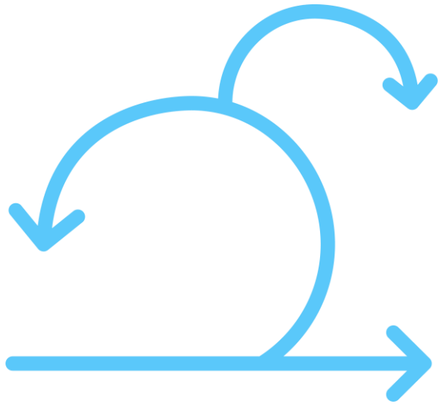
Complex projects



Changing requirements

Why to Choose Agile Model?

Although the waterfall model is traditional and structured, the agile model has become popular for its flexibility, adaptability, and emphasis on customer satisfaction.



The best choice for a project depends on its specific characteristics and the preferences of the team and stakeholders.

Introduction to Agile

What Is Agile?

It is a methodology used for managing projects, and it emphasizes flexibility, collaboration, and customer-centric approach.

Agile is needed for:

- Iterative development
- Customer collaboration
- Cross-functional teams
- Flexibility and adaptability
- Continuous improvement

Agile Manifesto and Its History

The Agile Manifesto is a declaration of the values and principles that underpin Agile methodologies.

1990

Evolution of Agile

It was a shift from a family of lightweight, quality-driven access to software development.

Agile methodology was initially designed for software development. It has evolved to encompass different types of projects.

2001

Agile Manifesto

It was signed by 17 leading software developers.

Key Values of Agile Manifesto

The four core values that underpin the Agile methodologies are:

Value individuals and interactions over processes and tools

Use working software over comprehensive documentation

Respond to change over following a plan

Prioritize customer collaboration over contract negotiation

Agile Principles

The highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Changing requirements must be welcomed, even late in development. Agile processes harness change for the customer's competitive advantage.

Working software must be delivered frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile Principles

Businesspeople and developers must work together daily throughout the project.

Projects should be built around motivated individuals. They should be provided with the necessary environment and support and trusted to complete the job.

The most efficient and effective method of conveying information to and within a development team is via face-to-face conversation.

Agile Principles

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Agile Principles

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

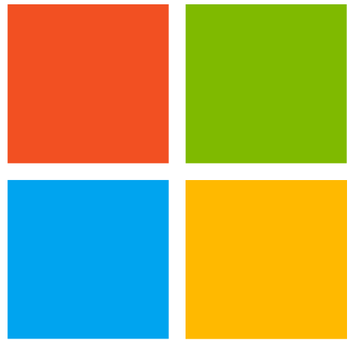
The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behavior accordingly.

Use Case: Microsoft’s Shift from Waterfall to Agile

Use Case: Background

Microsoft is a global leader in software development. Traditionally, they relied on the Waterfall model for product development, which was also used for their flagship operating system, Windows.



Microsoft

For many years, this approach worked well. However, as the tech industry began to evolve more rapidly, Microsoft encountered new challenges.

Challenges with Waterfall

Lengthy release cycles

For each new version of Windows, Microsoft followed the Waterfall model. This led to long development cycles, often requiring two to three years between releases.

Competitive pressure

As the industry shifted towards more frequent and iterative releases, especially with the rise of mobile operating systems, Microsoft struggled to compete with rivals who could release updates more quickly.

Challenges with Waterfall

Delayed customer feedback

In the Waterfall model, user feedback was typically received only when the product was nearly complete, often resulting in last-minute changes and rework.

Adapting to modern demands

When Microsoft began developing Windows 10, they realized the need for a new approach to meet modern user demands and stay in competition in the rapidly evolving tech industry.

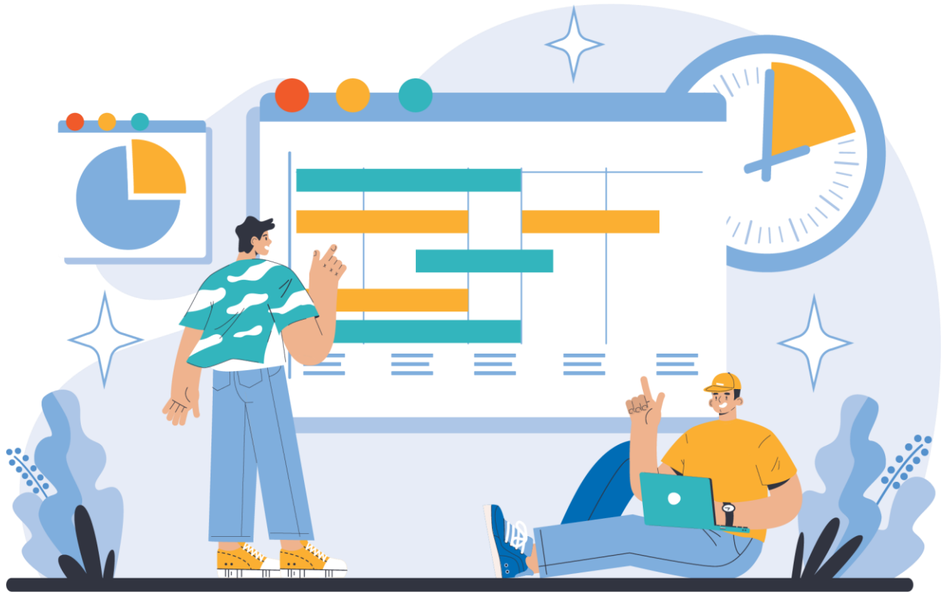
Microsoft’s Shift from Waterfall to Agile

With Windows 10, Microsoft shifted from major releases to continuous delivery, adopting an Agile approach.

- Microsoft shifted to smaller updates for Windows 10, delivering two major updates per year instead of massive updates every few years.
- They launched the Windows Insider Program to gather real-time user feedback on new features, establishing a feedback loop.
- They adopted Agile's collaborative approach for successive developments.
- They split Windows 10 development into shorter cycles, focusing on specific features and refining them based on user feedback.

Outcome of Microsoft's Agile Transformation

Microsoft’s shift to an Agile approach with Windows 10 marked a significant transformation in how they develop and deliver software.



- Smaller, more frequent updates in Windows 10 allowed Microsoft to release features faster and respond more quickly to user needs.
- The Windows Insider Program gave users a voice in the development process, boosting engagement and satisfaction.
- Frequent updates helped catch bugs early, reducing the risk of major failures and the need for rework.
- By updating Windows 10 more efficiently, Microsoft became more adaptable to security threats and emerging technologies.

Conclusion

- Microsoft’s shift to an Agile approach with Windows 10 marked a significant transformation in how they develop and deliver software.
- The adoption of continuous delivery, real-time feedback, and cross-functional collaboration allowed Microsoft to meet the demands of modern users and compete effectively in the tech landscape.

Activity: Solving Problems with Agile Principles

Solving Problems with Agile Principles

Scenario:
Your project team is struggling with timely deliveries and customer satisfaction. Deadlines are frequently missed, and the delivered products often do not meet customer expectations. This has led to frustration among team members and stakeholders, creating a tense work environment.

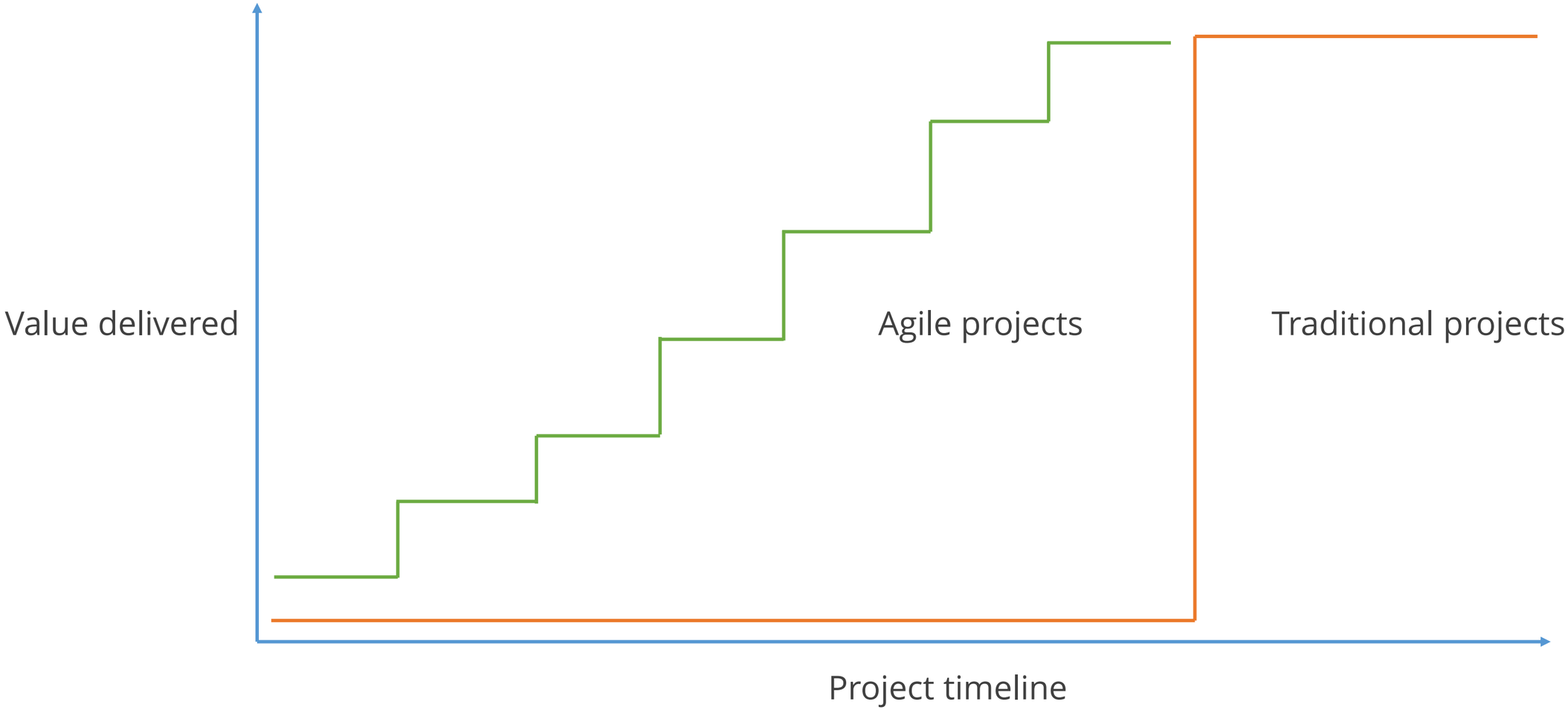
Task:

- Read the principles
- Select the appropriate Agile principles to address the issues your team is facing
- Explain how each principle will help resolve the problems

Importance of Agile

Faster Delivery of Value

Agile projects provide quicker delivery of value by following an incremental approach to delivery. Customers do not have to wait too long to see the value delivered.



Benefits of Agile Methods

Increased flexibility and adaptability

It allows teams to adapt to changes in customer requirements, market conditions, or technology.

Enhanced customer satisfaction

It improves customer satisfaction by involving the customer or client throughout the development process.

Improved product quality

It identifies and fixes defects early and often through continuous integration, regular testing, and frequent iterations.

Benefits of Agile Methods

Accelerated time to market

It releases significant features to the market faster than traditional models.

Improved risk management

Due to its incremental nature, it detects and resolves issues earlier, reducing the likelihood of a project failure.

Streamlined project control

It provides a high degree of control and visibility into the project's progress and potential issues.

Benefits of Agile Methods

Enhanced team morale

It promotes a collaborative and empowering work environment.

Optimized resource utilization

It prioritizes high-value features and eliminates tasks that do not directly add value to the project.

Activity: Comparing Agile and Waterfall

Comparing Agile and Waterfall

Scenario:

Bright Future Pharmaceuticals, a leading company in the pharmaceutical industry, has traditionally used the Waterfall methodology for its research and development projects. However, to keep pace with the rapid advancements in medical technology and regulatory changes, the company is considering a shift to Agile project management to enhance flexibility and responsiveness. While some team members and middle management are enthusiastic about the switch to Agile, the upper management is hesitant due to concerns about the cost of training and potential disruptions during the transition.

Based on above scenario:

- Write a report comparing the Agile and Waterfall methodologies
- Highlight the key differences, benefits, and challenges of each one of them

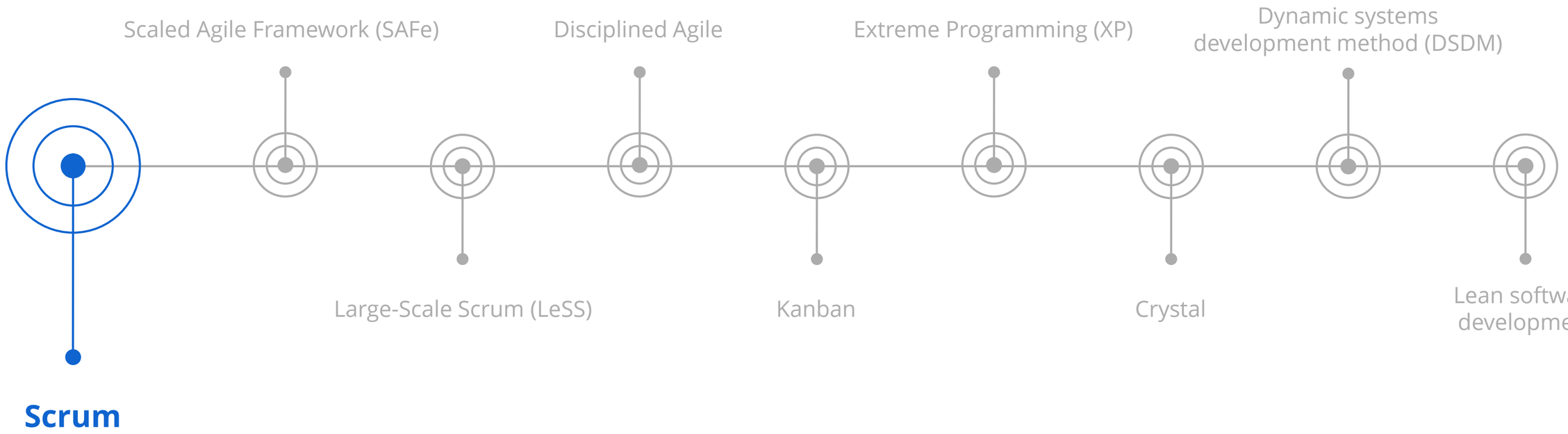
Types of Agile Frameworks and Methodologies

Agile Frameworks and Methodologies

They are structured approaches to project management and product development that emphasize flexibility, collaboration, and iterative progress. The most popular Agile frameworks and methodologies are:

- Scrum
- Kanban
- Scaled Agile Framework (SAFe)
- Extreme Programming (XP)
- Large-Scale Scrum (LeSS)
- Crystal
- Disciplined Agile
- Dynamic systems development method (DSDM)
- Lean software development

Agile Frameworks and Methodologies



Scrum

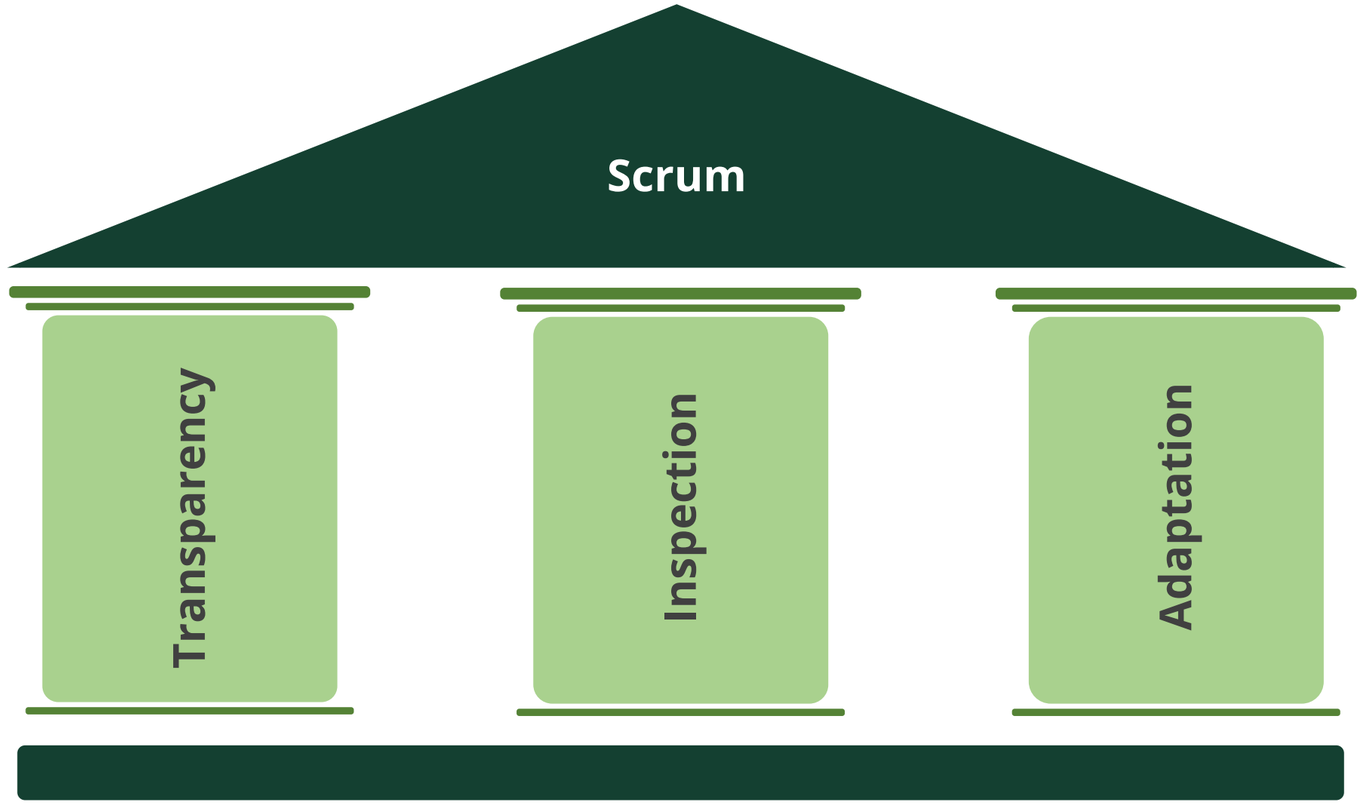
It is a high-level Agile framework designed to help organizations deliver complex projects incrementally and iteratively through short, time-boxed iterations known as sprints.

Sprint is a set period during which specific work must be completed and made ready for review.

Scrum allows people to address complex adaptive problems while delivering products of the highest possible value in a productive and creative manner.

Pillars of Scrum

Scrum is a widely adopted Agile framework and is founded on three core pillars that are:



Pillars of Scrum

Transparency	Teams share information openly and maintain complete transparency.
Inspection	The product and the process are subjected to frequent inspections.
Adaptation	Based on the findings of the inspection, teams will make the necessary adjustments.

Values of Scrum

Scrum is built on five core values that guide the behavior and actions of individuals within a Scrum team. The following are the five core values of Scrum:

- Commitment
- Focus
- Openness
- Respect
- Courage

Values of Scrum

Commitment	Every team member should be committed to the team and to the work that will be undertaken.
Focus	Team members must individually and collectively focus on getting the job done.
Openness	Team members should openly admit mistakes, share obstacles or gaps in knowledge, share feedback, and learn from each other.

Values of Scrum

Respect	Team members must treat each other with mutual respect and help each other where and whenever they can.
Courage	Team members must show the courage to do the right thing and face tough problems.

These values support to create a collaborative, adaptive, and efficient working environment.

Use Case: Implementing a Major Payment Gateway for an E-commerce Platform

Use Case: Background

A Scrum team at a mid-sized e-commerce company was tasked with integrating a new payment gateway to support international transactions. The integration was critical, as the company was expanding into new markets.

In this use case, the Scrum values are essential to guiding the team's behavior and ensuring they can deliver a high-quality solution while navigating challenges effectively. These values help the team stay aligned, communicate openly, and collaborate effectively throughout the project.

Values of Scrum

Commitment	At the beginning of the sprint, the team committed to completing the payment gateway integration within two sprints.
Focus	Despite last-minute promotional feature requests from marketing, the team stayed focused on the payment gateway integration.
Openness	The team openly discussed the challenges they faced with the product owner and stakeholders in their sprint review.

Values of Scrum

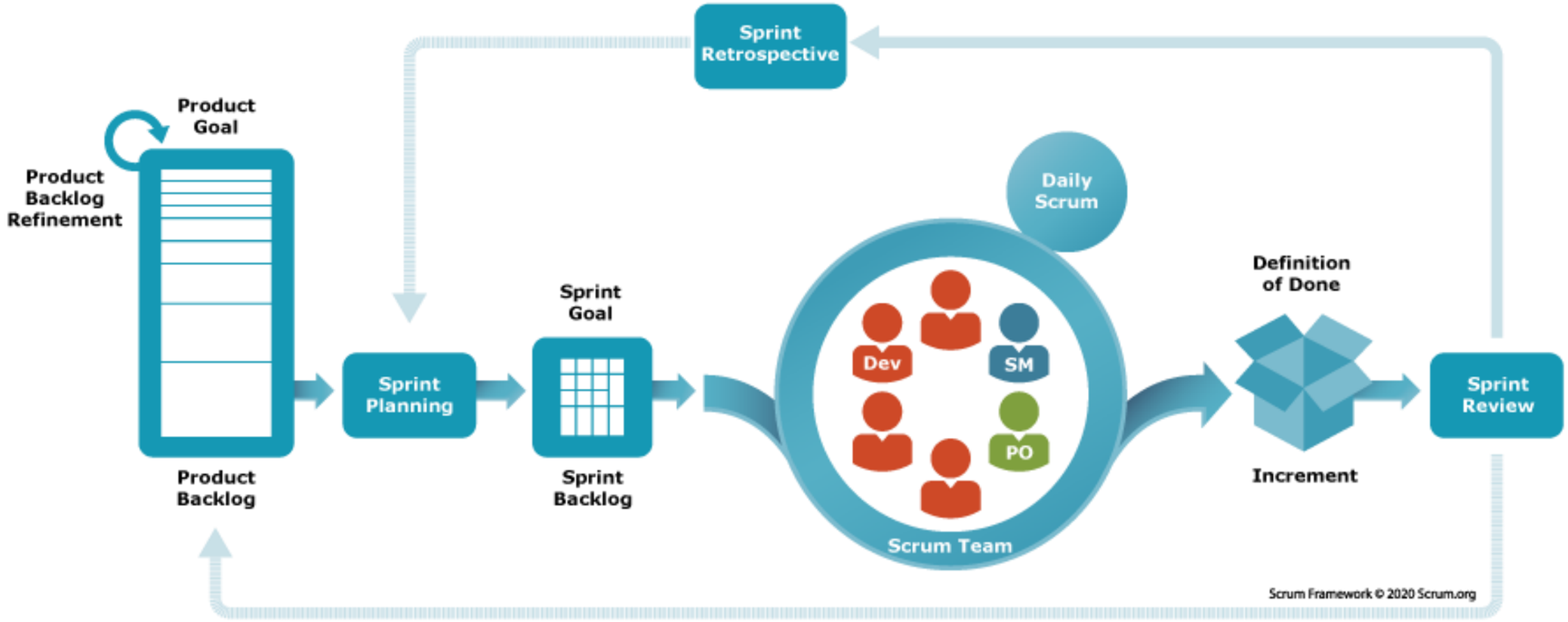
Respect	Throughout the sprint, the team demonstrated deep respect for each other’s roles and expertise.
Courage	During the sprint, the team encountered a significant issue: the gateway API had undergone changes that were not documented.

Outcome

- The payment gateway was successfully integrated within the two-sprint timeframe by overcoming challenges through open communication, focus on the sprint goal, and mutual respect for each other’s expertise.
- The stakeholders were impressed by the technical achievement and how the team handled challenges without losing focus on the overall objective.
- This real-world use case shows how all five Scrum values work together to help a team navigate challenges and deliver value.

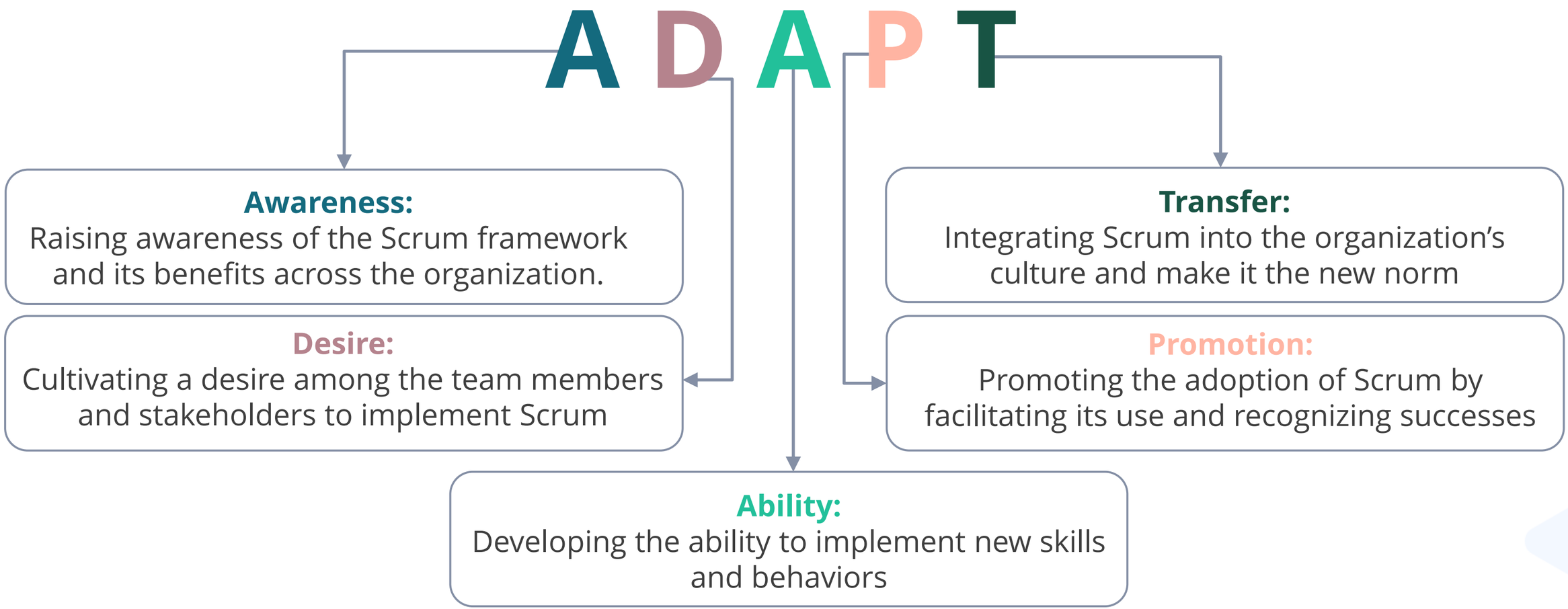
Scrum Workflow

It can be viewed as follows:



Process of Adopting Scrum

It involves a structured and phased approach. One common method for this transition is:



Scenarios for Scrum Implementation

Objective

To deliver working software frequently, prioritize transparency, and empower self-organizing teams

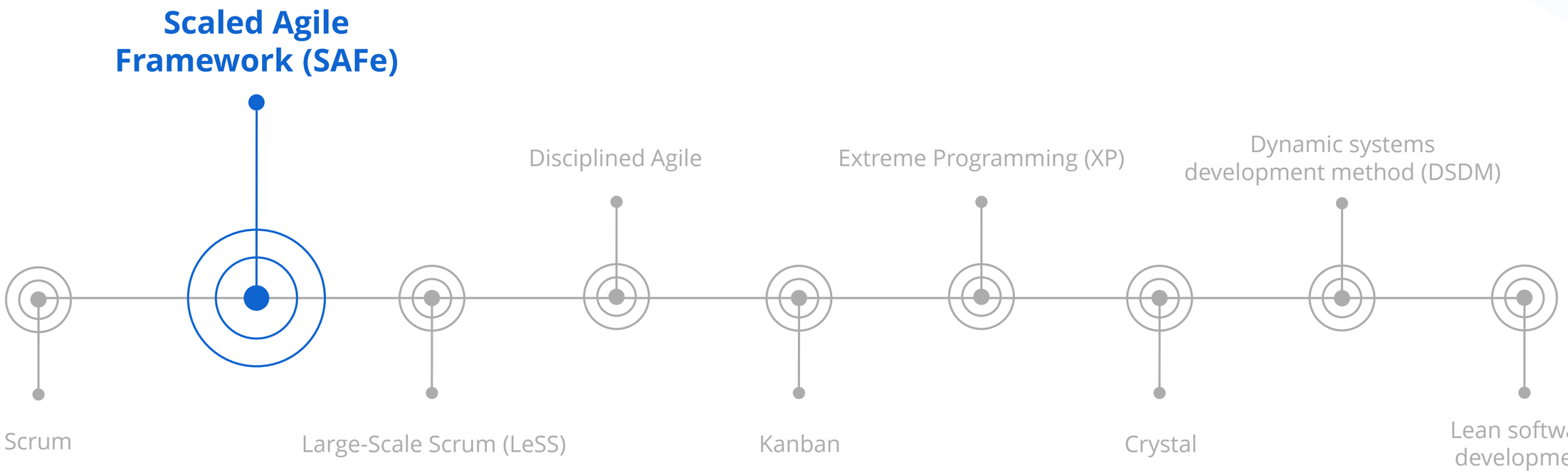
Best for:

Small to medium-sized teams, projects having flexibility and well-defined requirements

Use case

A mobile app startup with a small, cross-functional team (developers, designers, testers) building a fitness-tracking app where requirements evolve based on customer feedback

Agile Frameworks and Methodologies



Scaled Agile Framework (SAFe)

It is a collection of organizational and workflow patterns designed to implement Agile practices on an enterprise-wide scale. The four configurations of Scaled Framework are:

- Essential SAFe
- Portfolio SAFe
- Large solution SAFe
- Full SAFe

The framework is designed to help organizations deliver faster, better, and more affordable solutions, making it ideal for complex projects with multiple teams, large systems, or critical compliance requirements.

Core Principles of SAFe

1. Take an economic view
2. Apply systems thinking
3. Assume variability; preserve options
4. Build incrementally with fast, integrated learning cycles
5. Base milestones on objective evaluation of working systems

Core Principles of SAFe

6

Make value flow without interruptions

7

Apply cadence, synchronize with cross-domain planning

8

Unlock the intrinsic motivation of knowledge workers

9

Decentralize decision-making

10

Organize around value

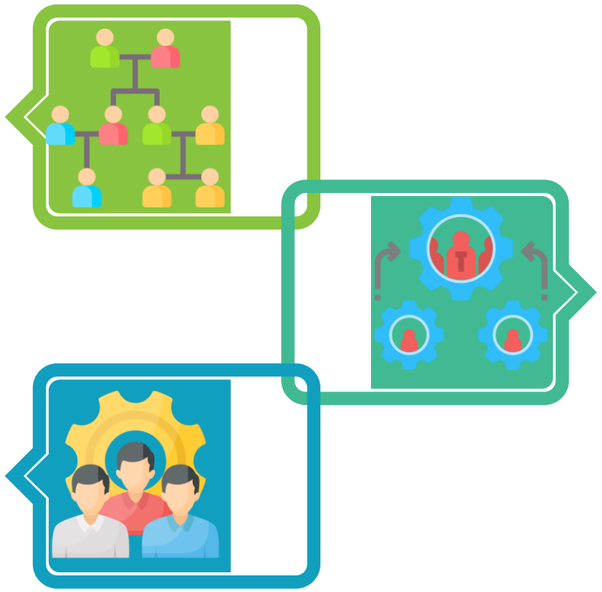
Levels of SAFe

Portfolio level

This is the highest level of the framework, focusing on enterprise-wide operations.

Team level

This is one of the crucial levels where the real development and delivery of value occur.



Program level

This level involves aligning multiple Agile teams to a shared business and technology mission.

Scenarios for SAFe Implementation

Objective

To scale Agile practices to large-scale projects, improve collaboration across teams, and achieve enterprise-wide agility

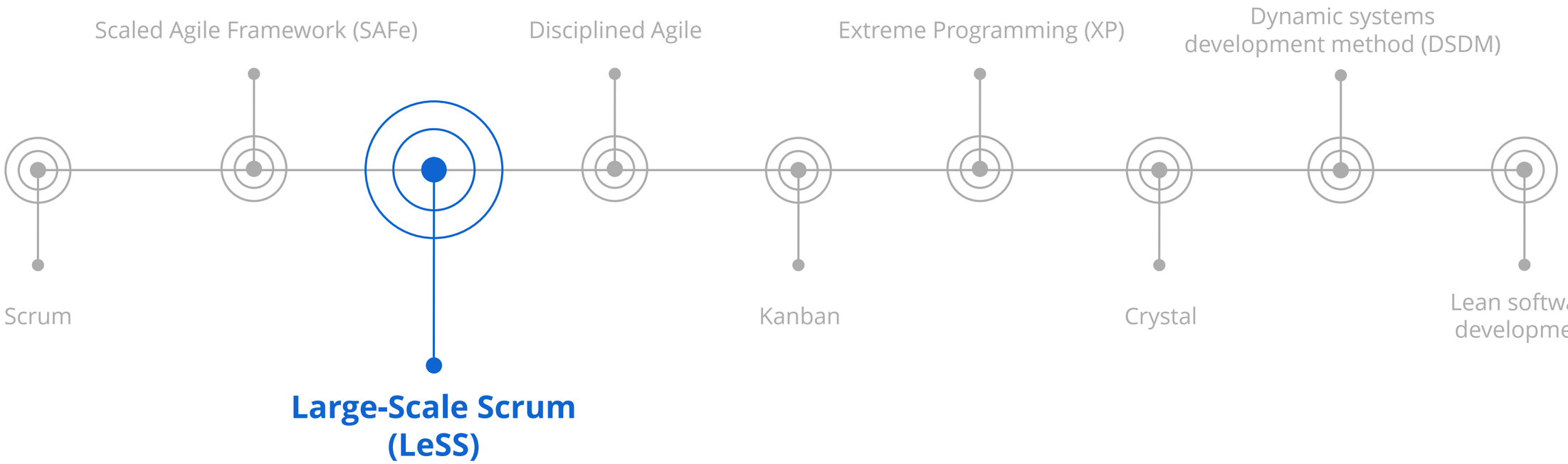
Best for:

Large enterprises with complex systems, multiple teams, and a need for alignment across the organization

Use case

A large multinational company manages multiple teams to develop an enterprise-wide banking and insurance system, where integration between several teams and layers of management is essential

Agile Frameworks and Methodologies



Large-Scale Scrum (LeSS)

It is a lightweight framework that is used for scaling Scrum while trying to stay true to the Scrum structure.

It is designed to simplify complex organizational processes using Scrum's principles of empiricism and cross-functional, self-managing teams at scale.

Core Principles of LeSS

- Queueing theory
- Empirical process control
- System thinking
- Lean thinking
- Continuous improvement towards perfection

- Large-Scale Scrum is Scrum
- Transparency
- More with less
- Whole product focus
- Customer-centric

Scenarios for LeSS Implementation

Objective

To scale Scrum while preserving its core principles and avoiding excessive complexity

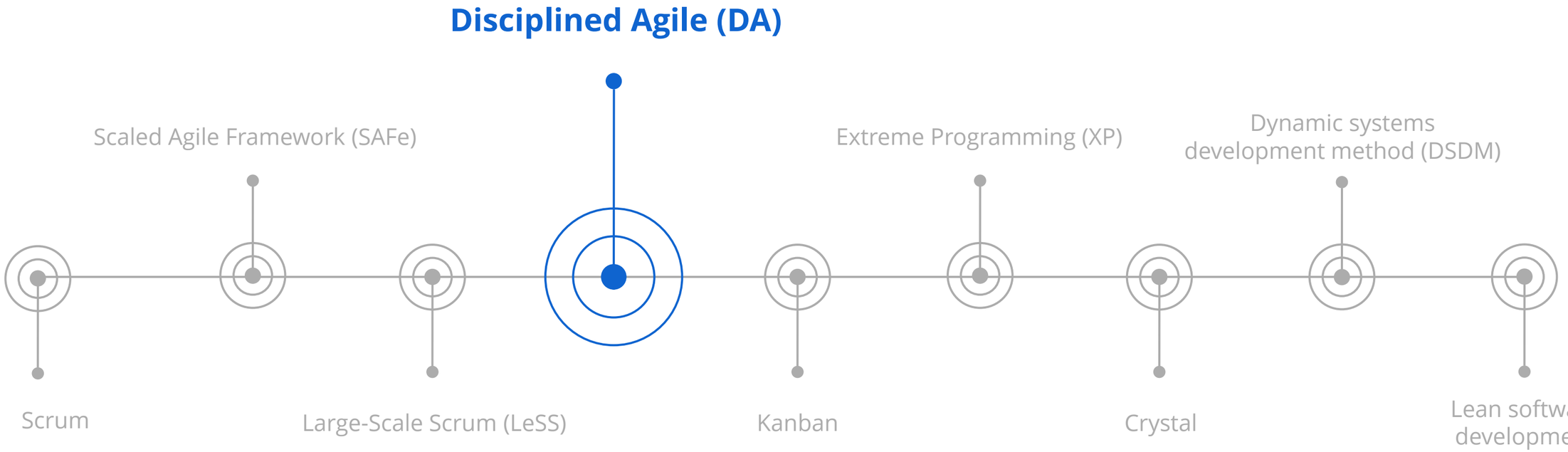
Best for:

Large-scale projects with multiple teams, where minimizing overhead and maintaining a simple approach are priorities

Use case

A mid-sized company has several Scrum teams working on a single product, like a new video conferencing tool, where the goal is to keep the process lightweight and Scrum-based

Agile Frameworks and Methodologies

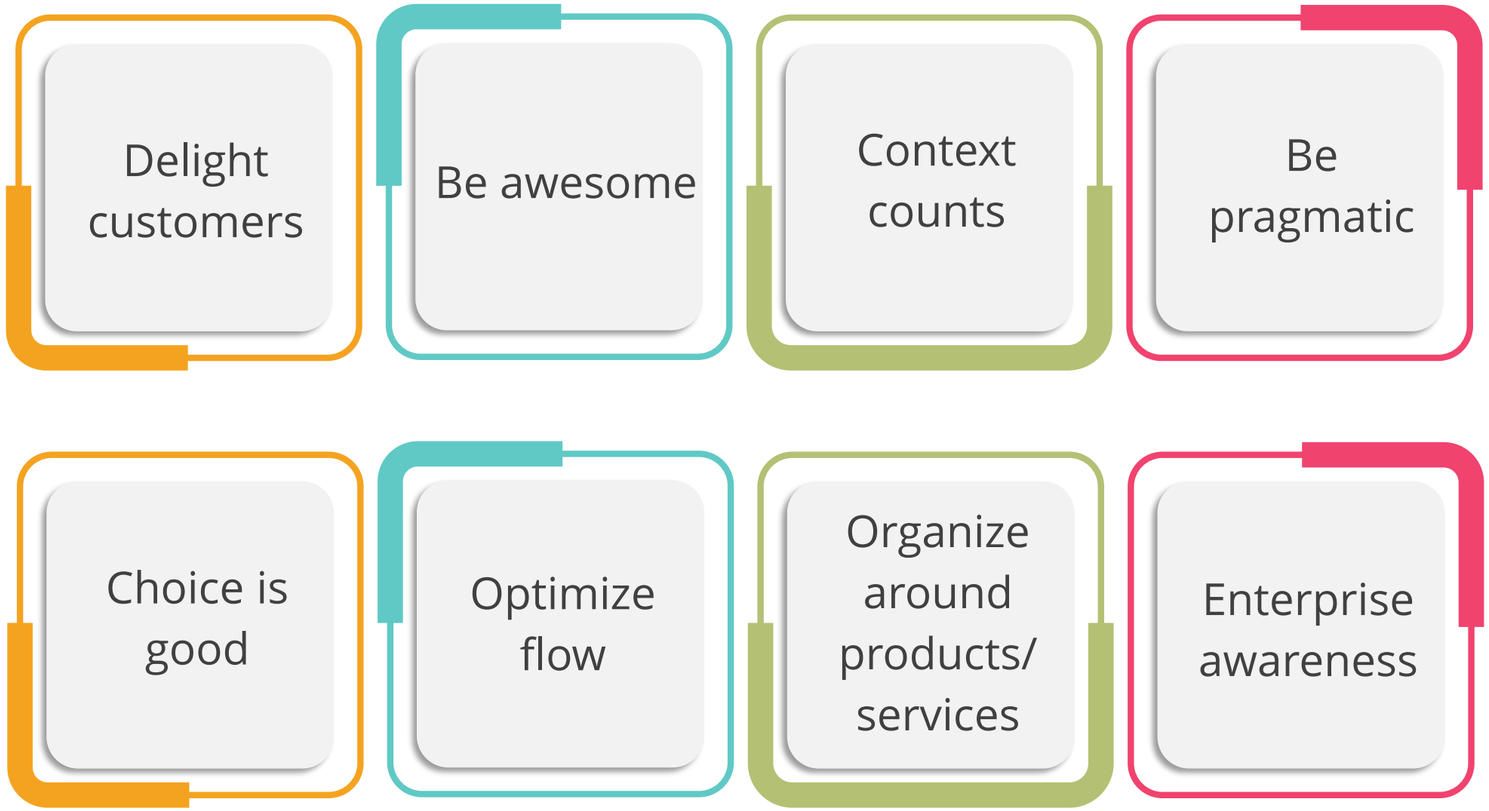


Disciplined Agile (DA)

It is a toolkit that leverages hundreds of Agile practices to guide the best way of working for a team or organization.

It offers clear guidance to help organizations optimize their processes in a context-aware way, establishing a strong foundation for business agility.

Core Principles of Disciplined Agile (DA)



These principles are designed to provide a philosophical foundation for business agility.

Scenarios for DA Implementation

Objective

To tailor Agile practices to the specific context, balance agility with governance, and improve process efficiency

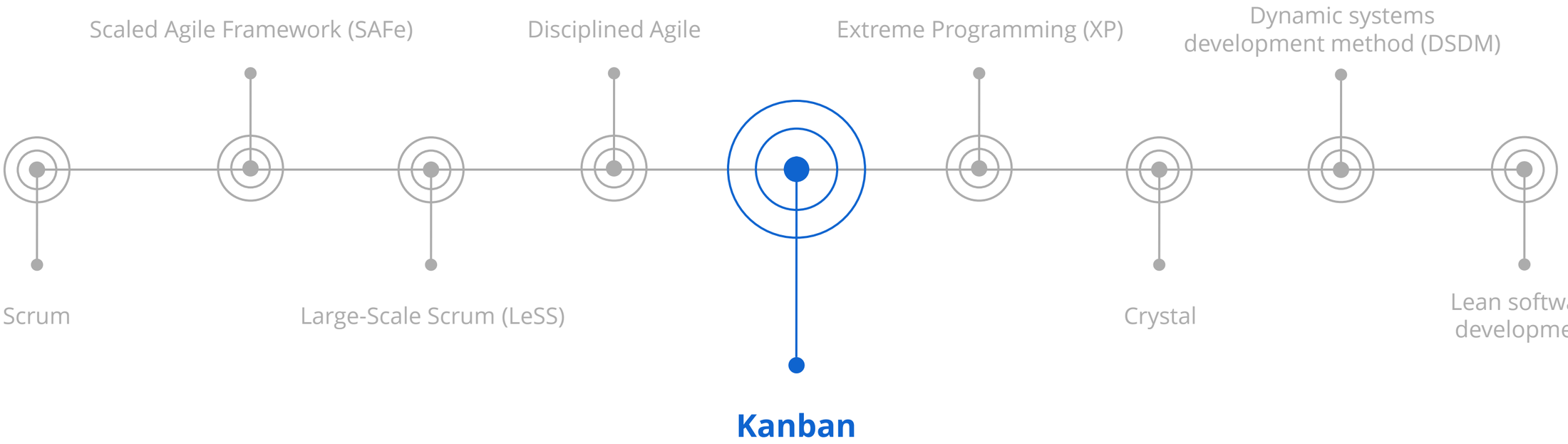
Best for:

Organizations that need a customizable and adaptable approach to Agile

Use case

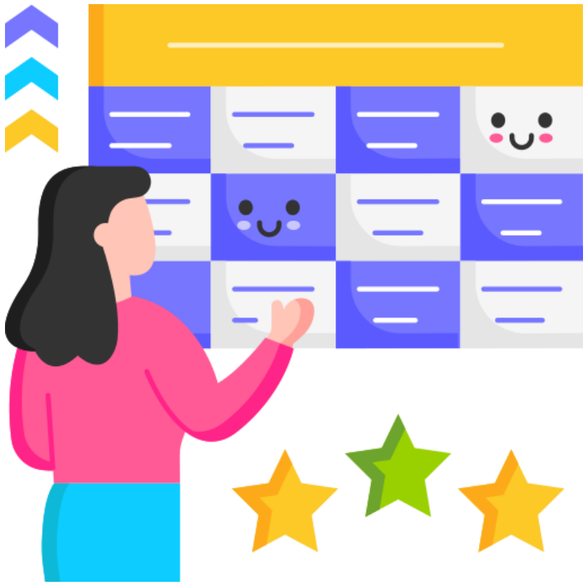
A financial institution blending Agile with traditional project management and regulatory compliance to build a secure online banking platform

Agile Frameworks and Methodologies



Kanban

It is a means to design, manage, and improve flow systems for knowledge work. The method allows organizations to start with their existing workflow and drive evolutionary change.



It requires real-time communication of capacity and full transparency of work. These values support to create a collaborative, adaptive, and efficient working environment.

Core Principles of Kanban

Change management principles

- Understand current processes and respect existing roles, responsibilities, and titles
- Agree to pursue improvement through evolutionary change
- Encourage acts of leadership at every level

Service delivery principles

- Understand and focus on customers' needs and expectations
- Manage the work; let people self-organize around it
- Evolve policies to improve customer and business outcomes

Kanban Terminologies

Kanban board	A visual tool that depicts the entire workflow of a process
Card	A visual representation of a task or work item on the Kanban board
WIP limits	Constraints placed on the number of tasks that can be in a particular stage of the workflow at one time
Swimlanes	Horizontal lanes on a Kanban board that can represent different types of work or priorities

Kanban Terminologies

Throughput	The number of cards (tasks) completed in each time frame
Lead time	The time taken for a single task to move from the start to the finish of the workflow
Cycle time	The amount of time a task spends in the workflow after work begins on it, which can help in identifying process improvement opportunities
Blocker	Anything that prevents a task from moving forward in the workflow, which can cause delays and increase lead time

Scenarios for Kanban Implementation

Objective

To focus on continuous delivery, improve efficiency, and minimize bottlenecks

Best for:

Teams that need to visualize workflow, improve flow, and reduce waste

Use case

A team managing a customer support system where tasks come in unpredictably, and the team needs to manage continuous work in progress

Activity: Analyzing a Kanban Board

Analyzing a Kanban Board

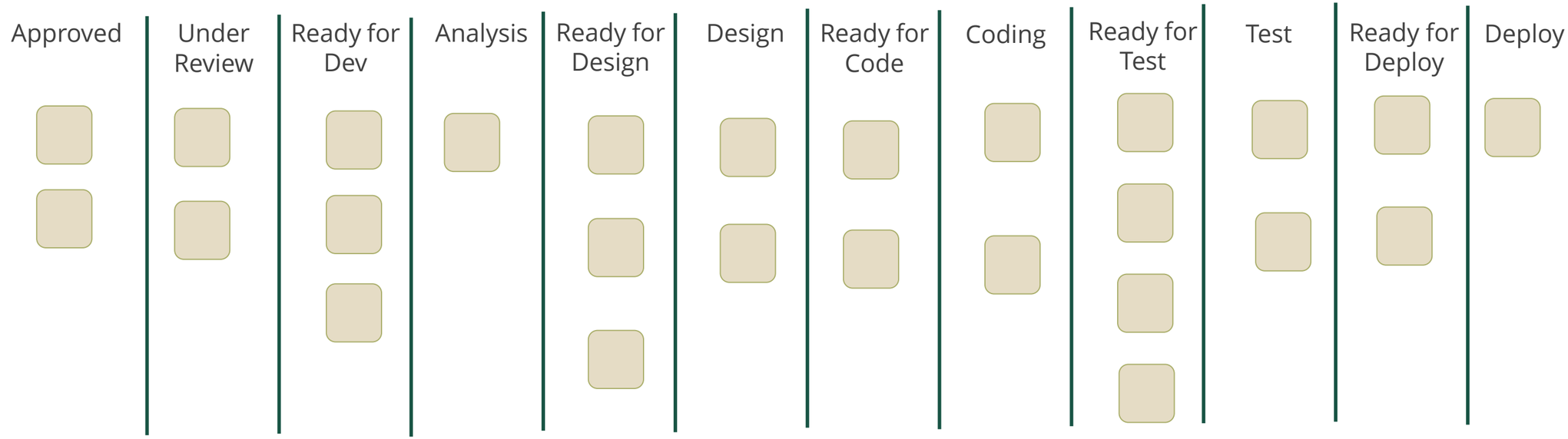
Scenario:

The IT team at Hi-tec Learning uses a Kanban board to manage tasks. However, it has noticed delays in deliveries. They are analyzing the board to identify and address workflow bottlenecks.

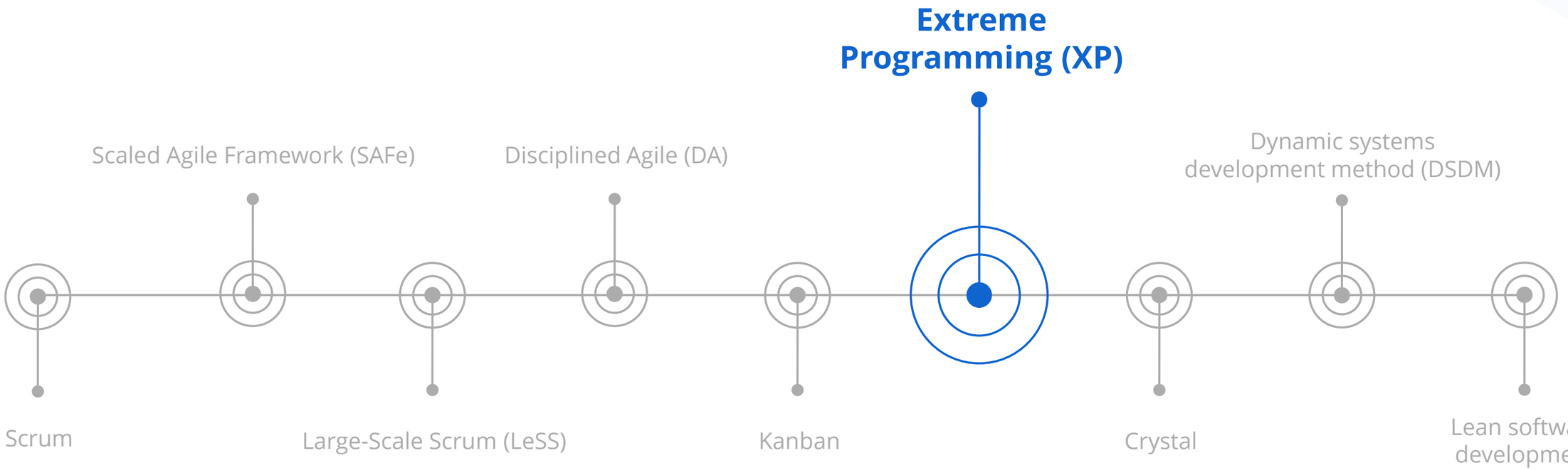
Task:

Study the board and answer these questions:

- Where is the biggest bottleneck in the system?
- What can be done to improve flow and reduce inventory?



Agile Frameworks and Methodologies



Extreme Programming (XP)

It is a software development framework that aims to produce higher-quality software and a higher quality of life for the development team. XP is appropriate when:

01 Software requirements are subject to frequent changes.

02 Fixed-time projects involving new technology carry inherent risks.

03 The development team is small, colocated, and experienced.

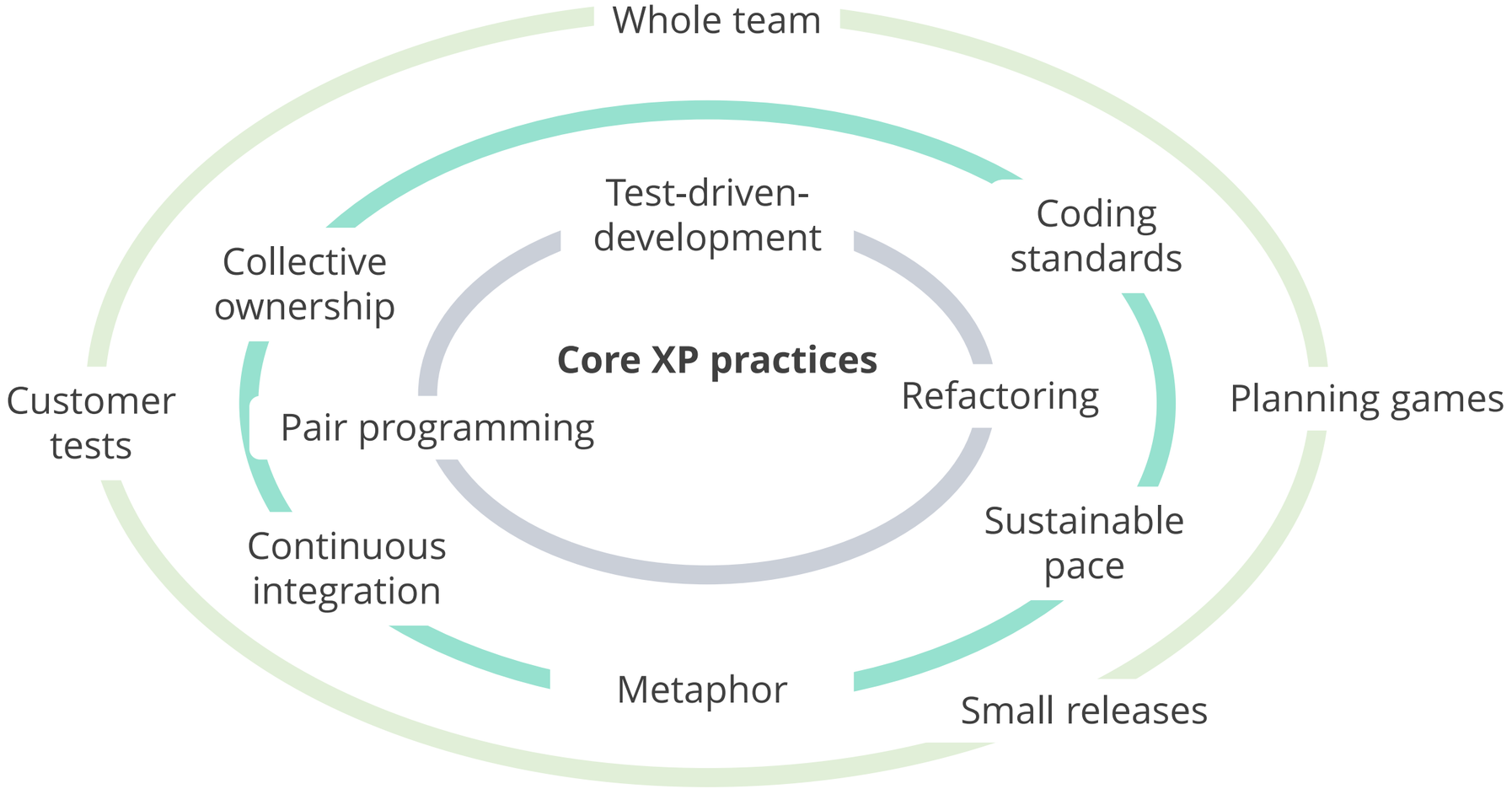
04 The technology supports automated unit and functional testing.

Values of Extreme Programming (XP)

The five values of XP are:

Extreme Programming Practices

There are several core practices that make up the extreme programming method.



XP Team Roles

Different team roles followed in XP are:

Customer

- The customer is responsible for making all the business decisions regarding the project.

Developer

- As XP does not have much need for role definition, everyone on the team is labeled a developer.

XP Team Roles

Different team roles followed in XP are:

Tracker

- The tracker monitors key metrics to track progress and identify improvements.

Coach

- A coach is an external mentor who guides the team in XP practices and ensures self-discipline.

Scenarios for XP Implementation

Objective

To prioritize quality, customer satisfaction, and frequent releases

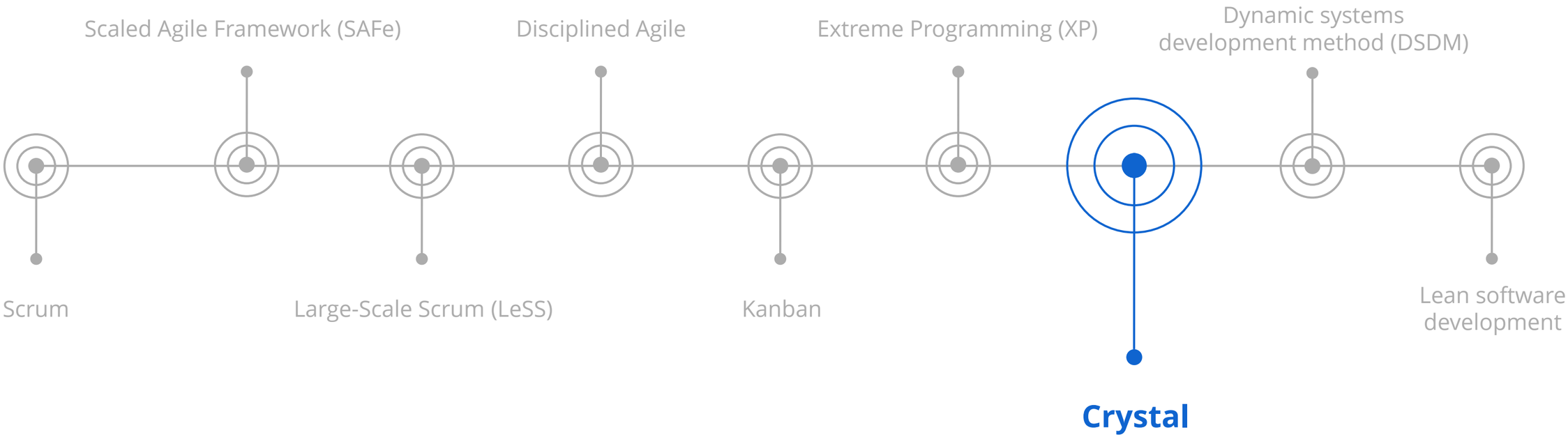
Best for:

Small teams working on complex projects with rapidly changing requirements

Use case

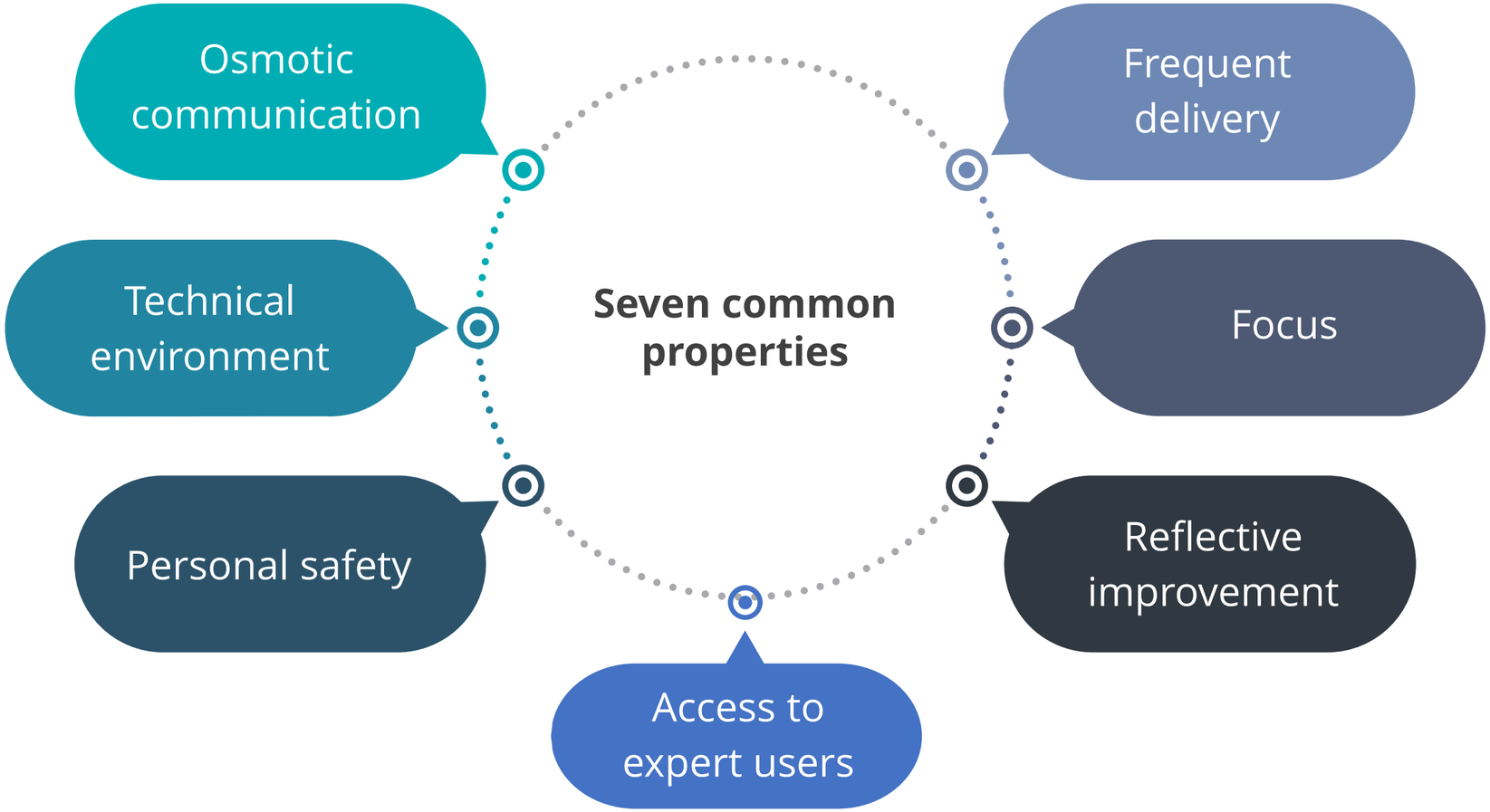
A software company developing a highly technical and complex AI system, requiring constant testing, pair programming, and quick feedback to ensure quality and functionality

Agile Frameworks and Methodologies



Crystal Methodology

It is an adaptable, lightweight, human-centric Agile framework, which emphasizes people and communication over strict processes.



Crystal Methodology: Properties

Osmotic communication	Ensure easy access to information in a shared space, facilitating spontaneous information flow and quick issue resolution
Technical environment	Provide a supportive and productive technical setup, including tools, practices, and infrastructure for efficient development and collaboration
Personal safety	Create an environment where team members feel safe to express ideas, raise concerns, and take risks without fear of blame or consequences

Crystal Methodology: Properties

Access to expert users	Guarantee regular and direct access to users or domain experts for valuable insights, feedback, and requirement clarification
Frequent delivery	Promote delivering functional software in short cycles for regular feedback, progress validation, and early issue detection
Focus	Ensure the team stays focused on goals, priorities, and tasks, avoiding distractions and scope changes
Reflective improvement	Encourage continuous learning and improvement through regular retrospectives and feedback-based adaptations

Scenarios for Crystal Implementation

Objective

To have a lightweight and adaptable approach that can be tailored to the specific needs

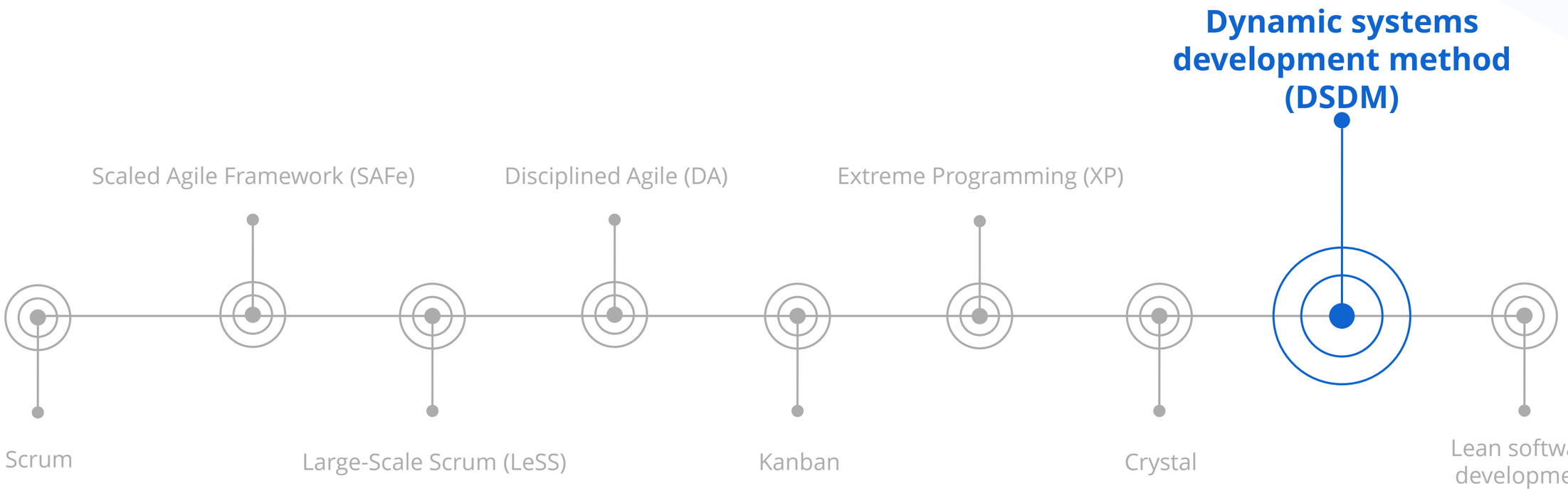
Best for:

Small teams with a focus on people and interactions, particularly where communication is key to project success

Use case

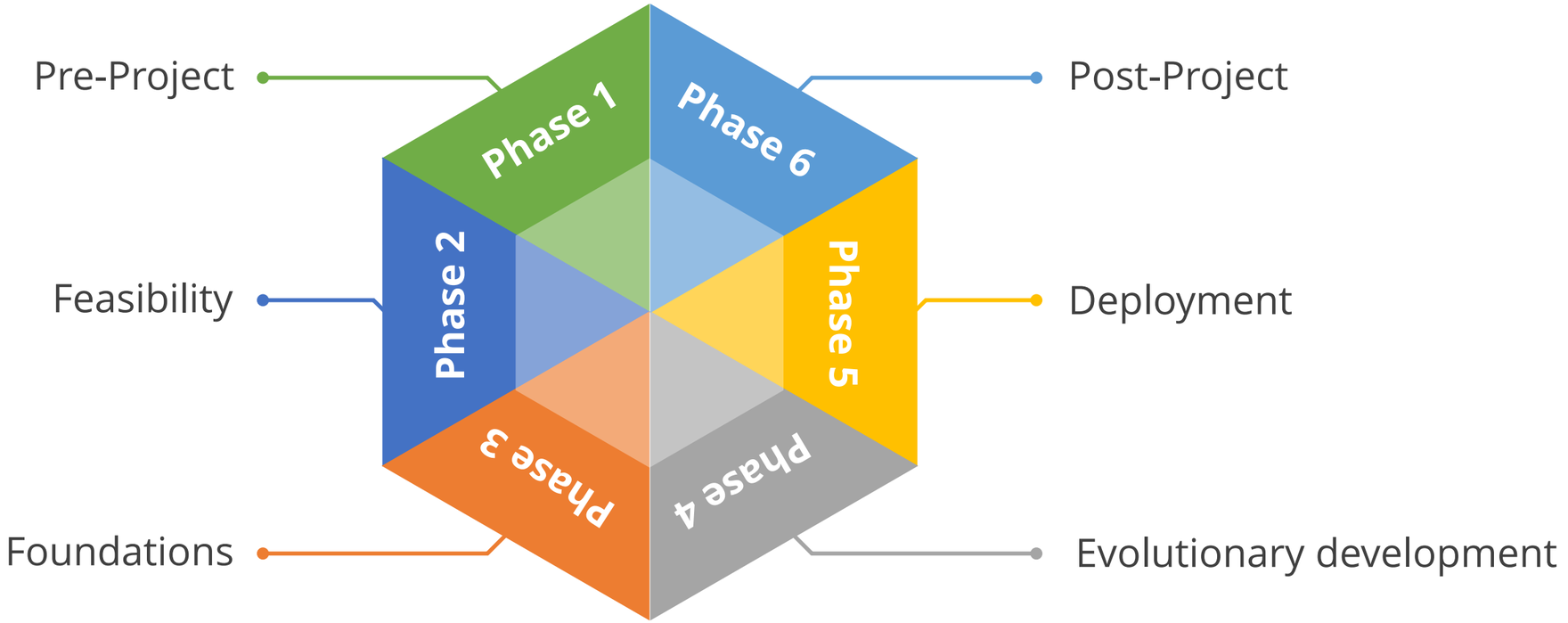
A small team working on a customized software solution for a specific client, where human interaction and adaptability are more important than formal processes

Agile Frameworks and Methodologies



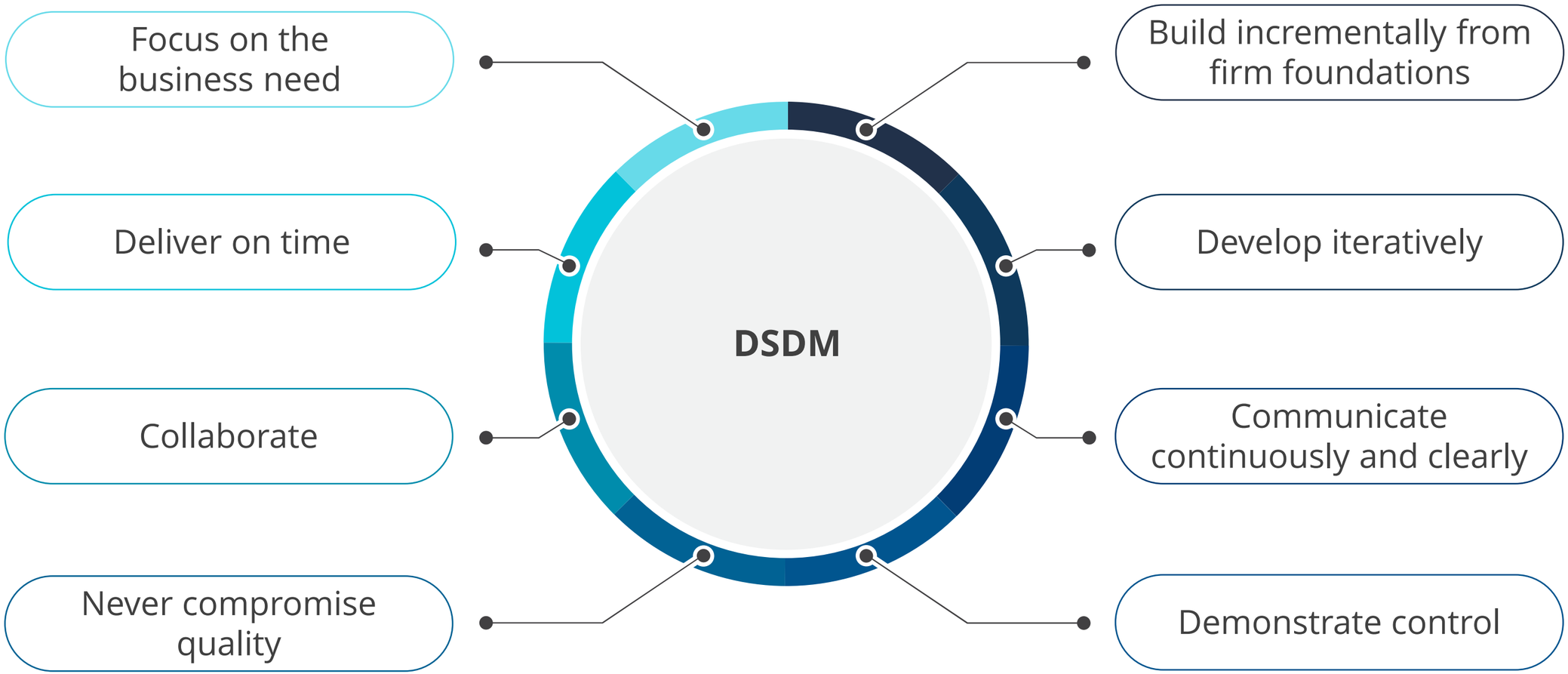
Dynamic Systems Development Method (DSDM)

It is a framework that focuses on the full project life cycle. It comprises of six phases that are:



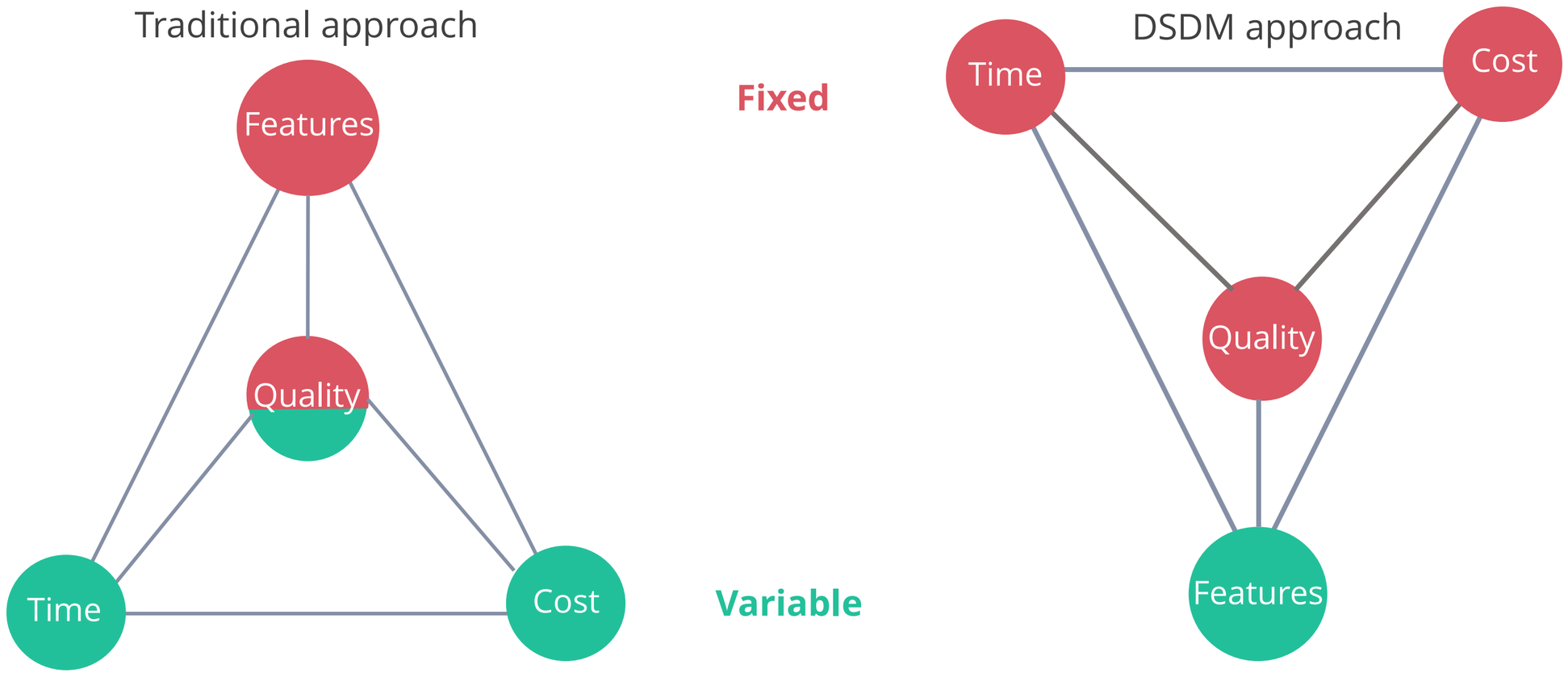
This framework is an iterative and incremental approach that embraces principles of Agile development, including continuous user or customer involvement.

Principles of DSDM



DSDM: Planning Philosophy

DSDM projects are timeboxed. Time and cost are fixed, and features are allowed to vary.



This approach makes DSDM effective in environments with strict time and budget constraints and adjustable project scope without compromising end goals.

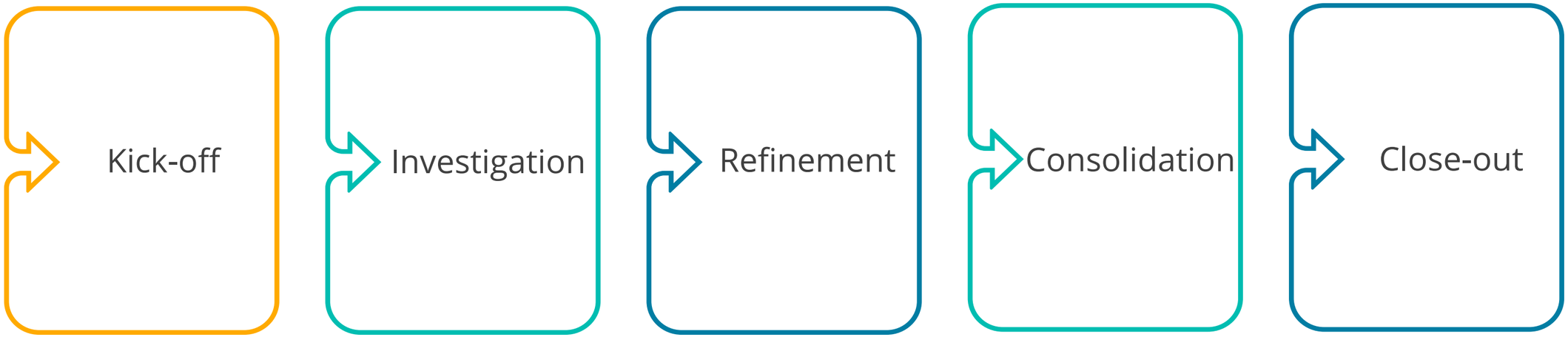
MoSCoW

It is a time management technique often used in DSDM for helping to understand and manage priorities. The letters stand for:

Must have	These are non-negotiable deliverables that the project requires to be considered a success. Without these, the project will fail.
Should have	These are important but not critical items, and the project would be usable without them but would be better if included.
Could have	These are desirable but not necessary items that will have a minor impact if omitted.
Won't have	These are least critical, often included in a wish list for future developments.

Timeboxing

It is a well-defined process to support the creation of low-level products in an iterative but controlled way. It comprises the following steps:



It incorporates frequent review points to ensure the quality of those products and the efficiency of the iterative development process.

Scenarios for DSDM Implementation

Objective

To deliver projects on time and within budget, while still allowing for flexibility

Best for:

Projects with fixed deadlines and budgets, but where requirements may change

Use case

A project to build a business intelligence dashboard for executives, where there’s a clear need for on-time delivery and involvement of business stakeholders throughout

Activity: Applying MoSCoW and Timeboxing in Project Management

Applying MoSCoW and Timeboxing in Project Management

Scenario:

You are a project manager for a team developing a new e-commerce website with a strict 4-week deadline. Prioritize tasks using the MoSCoW method and allocate timeboxes to ensure timely delivery of critical features.

Applying MoSCoW and Timeboxing in Project Management

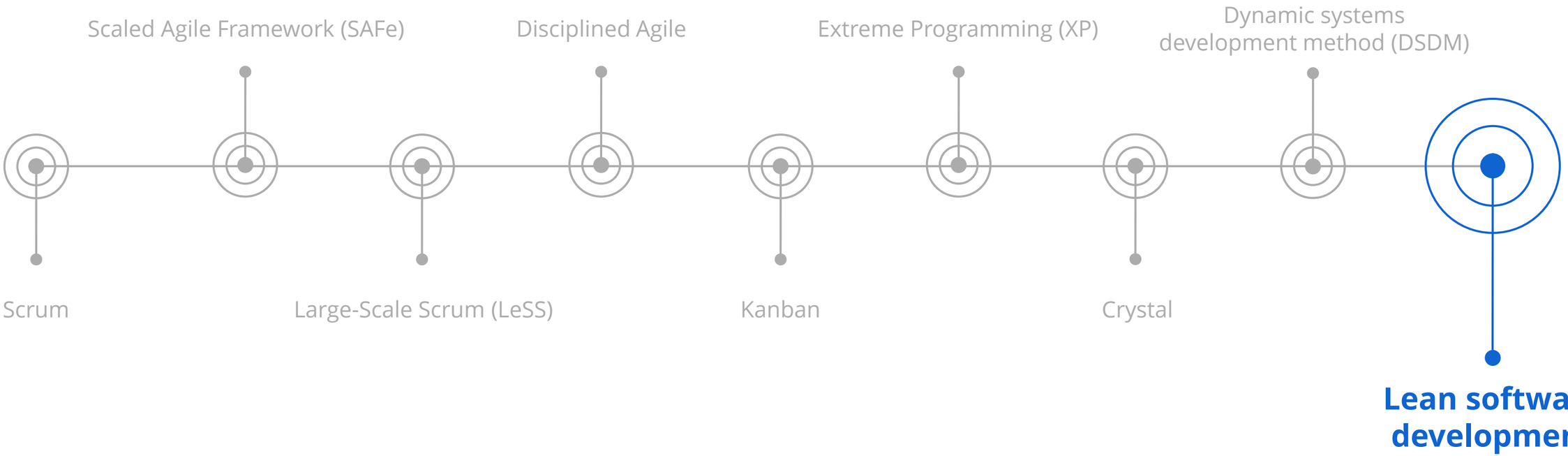
Task:

Below are the tasks for the e-commerce website with estimated hours:

- Using the MoSCoW method, categorize each task into the following categories: must have, should have, could have, and would not have.
- Submit your prioritized task list with the MoSCoW categories and assigned timeboxes. Provide a brief explanation for your prioritization and timeboxing decisions.

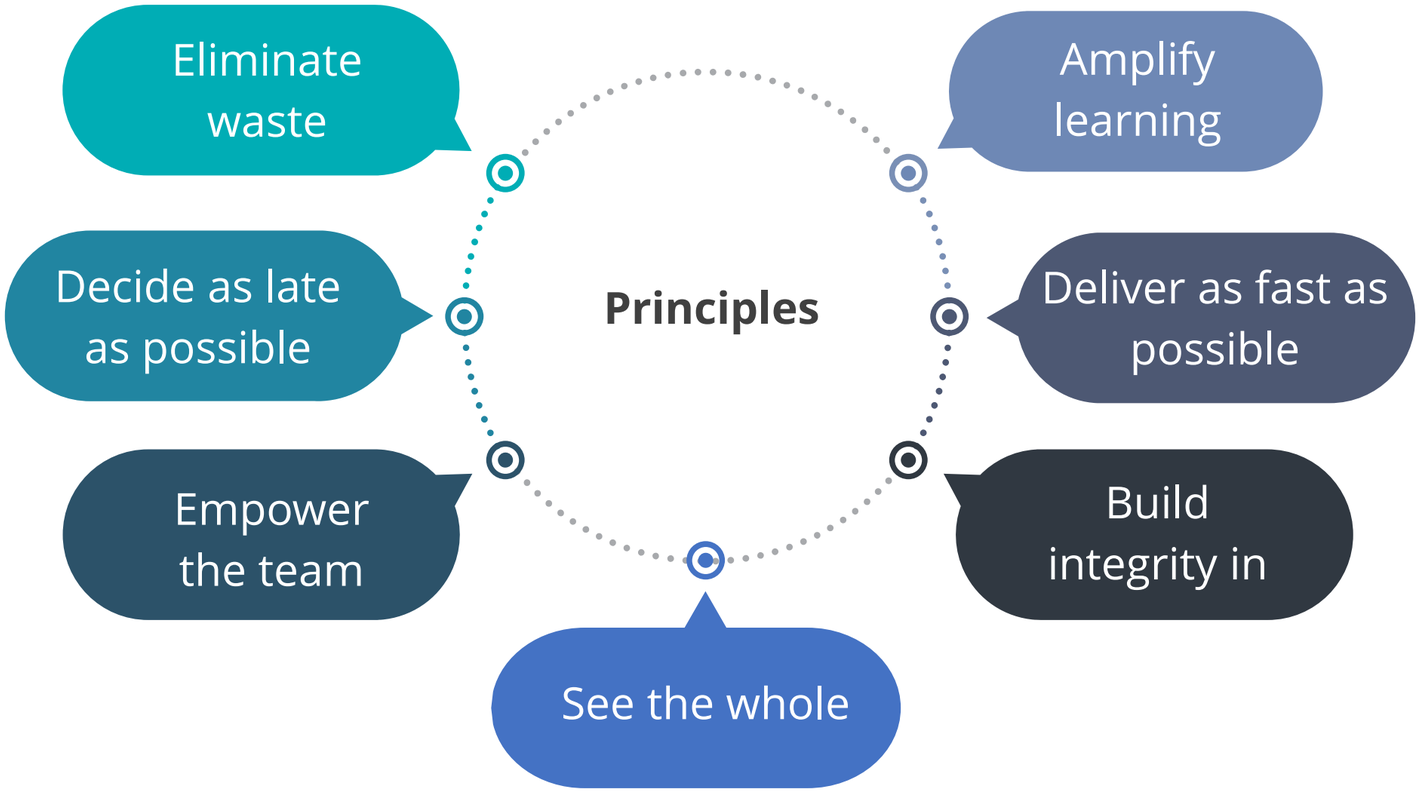
Task	Estimated duration (hours)
User registration	10
Product listing	15
Shopping cart	12
Payment gateway integration	20
Search functionality	8
User profile management	10
Order management	15

Agile Frameworks and Methodologies



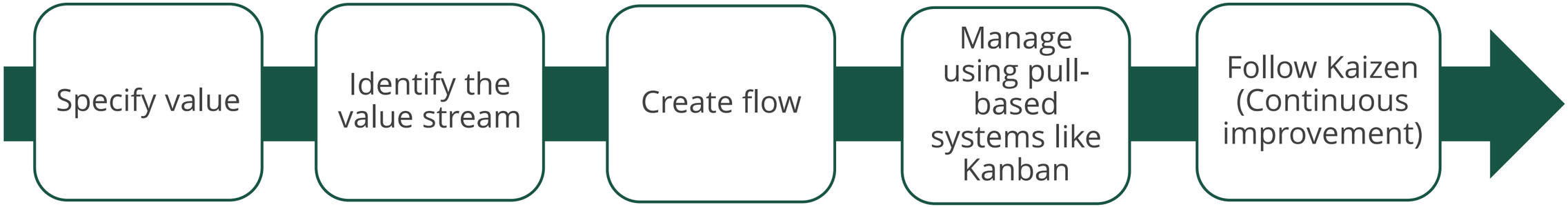
Lean Software Development

It is an Agile framework used to streamline and optimize the development process.



5-Step Process to Become Lean

A simple 5-step process to make a journey toward adopting a lean culture is:



Value Stream Mapping

It is a lean manufacturing technique to analyze, design, and manage the flow of materials and information required to bring a product to a customer.

It defines an end-to-end process that depicts how value is added to the customers.

It maps the **as-is** process steps in as much detail as possible.

It involves the team in drawing the value stream map.

It helps identify those steps that do not directly add customer value (waste).

It encourages brainstorming about how to improve the value stream map by reducing waste.

Scenarios for Lean Software Development Implementation

Objective

To focus on continuous improvement, reduce overhead, and deliver products that meet customer needs

Best for:

Teams that want to eliminate waste, improve efficiency, and deliver value quickly

Use case

A SaaS company optimizing its payment processing system to eliminate waste, reduce overhead, and improve flow without adding unnecessary features

Activity: Creating a Value Stream Map

Creating a Value Stream Map

Scenario:

You are a process improvement consultant hired by a small manufacturing company. The company produces custom-made furniture and wants to optimize its production process to reduce delays and increase efficiency. Your task is to create a value stream map of their current process and identify areas for improvement.

Creating a Value Stream Map

Task:

Following are the steps involved in the furniture production process, along with the estimated lead time (LT), cycle time (CT), and defect rate for each step.

- Create the value stream map
- Analyze the value stream map
- Write your analysis and improvement suggestions
- Ensure that your analysis includes calculating the total lead time and cycle time, identifying bottlenecks and high defect rates, and a rationale for your proposed improvements

Step	Lead time (LT)	Cycle time (CT)	Defect rate
Order received	2 hours	0.5 hours	0.05
Design customization	8 hours	4 hours	0.1
Material preparation	10 hours	6 hours	0.08
Cutting and shaping	15 hours	8 hours	0.12
Assembly	20 hours	10 hours	0.15
Finishing	10 hours	5 hours	0.1
Quality inspection	5 hours	2 hours	0.05
Packaging and shipping	5 hours	1 hour	0.02

Key Takeaways

- Agile is a methodology used for managing projects, that emphasizes flexibility, collaboration, and customer-centricity.
- Scrum is a high-level framework that was created to help organizations to deliver complex projects incrementally.
- SAFe (Scaled Agile Framework) is a set of organization and workflow patterns for implementing Agile practices at scale.
- LeSS is a lightweight framework which is used for scaling Scrum while trying to stay true to the Scrum structure.
- Disciplined Agile (DA) is a hybrid Agile framework that provides a comprehensive toolkit for organizations to design their Agile and lean process in a context-sensitive manner.

Key Takeaways

- Extreme programming is a streamlined methodology that emphasizes simplicity, communication, feedback, and courage.
- Crystal methodology is an adaptable, lightweight, human-centric Agile framework emphasizing people and communication over strict processes.
- DSDM is a framework that focuses on the full project life cycle. It is expressed in the form of five principles.
- Value stream mapping is a lean manufacturing technique to analyze, design, and manage the flow of materials and information required to bring a product to a customer.

Knowledge Check

Knowledge Check 1

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 product

Knowledge Check 1

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 product

The correct answer is **D**

According to the Agile Manifesto, a working product is valued the most. Ultimately, it all boils down to whether there is a working product to show or not.

Knowledge Check 2

Which of the following is a team transitioning to Scrum most likely to lose?

A. Flexibility and adaptability

B. Productivity and quality

C. Detailed up-front planning and architecture

D. Ability to respond to business changes

Knowledge Check 2

Which of the following is a team transitioning to Scrum most likely to lose?

A. Flexibility and adaptability

B. Productivity and quality

C. Detailed up-front planning and architecture

D. Ability to respond to business changes

The correct answer is **C**

A Scrum team usually exhibits higher productivity and quality with the ability to respond to changes, flexibility, and adaptability.

Knowledge Check 3

An organization wonders whether it should use Kanban or Scrum for a particular team. In general, you could say that Kanban is a better approach if:

A. An iterative approach is favored.

B. Work is highly innovative and research oriented.

C. The focus is on speed of delivery.

D. Scrum Masters are available in the team.

Knowledge Check 3

An organization wonders whether it should use Kanban or Scrum for a particular team. In general, you could say that Kanban is a better approach if:

A. An iterative approach is favored.

B. Work is highly innovative and research oriented.

C. The focus is on speed of delivery.

D. Scrum Masters are available in the team.

The correct answer is **C**

Kanban focuses on speed of delivery by increasing flow. Scrum, on the other hand, focuses on inspect and adapt through frequent feedback from timeboxed iterations.