

Agile Estimating, Planning, Monitoring, and Controlling



Learning Objectives

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

- 🔗 Define and apply Agile estimation techniques for accurate project planning
- 🔗 Analyze and prioritize product backlogs using the Kano model, relative weights method, and value risk matrix to ensure focus on high-value features
- 🔗 Identify different levels of Agile planning to create a structured project roadmap
- 🔗 Describe and implement Agile monitoring methods, such as burndown and burnup charts, to track project progress
- 🔗 List down the communication strategies with stakeholders to ensure clarity on project value and progress





Agile Estimating

What Is Agile Estimating ?

It is a technique used in Agile project management to forecast the effort required to complete tasks, features, or user stories within a project.



It focuses on relative estimation, where the complexity, size, or effort of one task is compared to others.



Essential Terms for Agile Estimation

What Is User Persona?

A user persona:



A user persona's perspective is used to write a user story.



Activity: Writing User Personas

Activity: Writing User Personas

Scenario:

You are part of an Agile Scrum team responsible for developing a new mobile application for a fitness company. The goal of the app is to help users maintain a healthy lifestyle by providing personalized workout plans, nutrition advice, and progress tracking. To ensure that the app meets the diverse needs of its users, the team needs to identify key user personas that will interact with the app. These personas will guide the development of features that align with the users' fitness goals, routines, and preferences.

Create user personas for the fitness app. Each persona should represent a different type of user who might engage with the app. When creating the personas, consider the following aspects:

Demographics: It include age, gender, occupation, fitness level, and other relevant characteristics.

Goals: What does this user aim to achieve with the app ? (For example, weight loss, muscle gain, and general fitness)

Pain points: What challenges or obstacles might this user face while trying to achieve their fitness goals? (For example, lack of time, motivation, or access to equipment)

Behavioral patterns: How frequently will they use the app? What features of the app will they use frequently? (For example, workout plans, tracking tools, and community support)

Sample Solution

Name: Emily

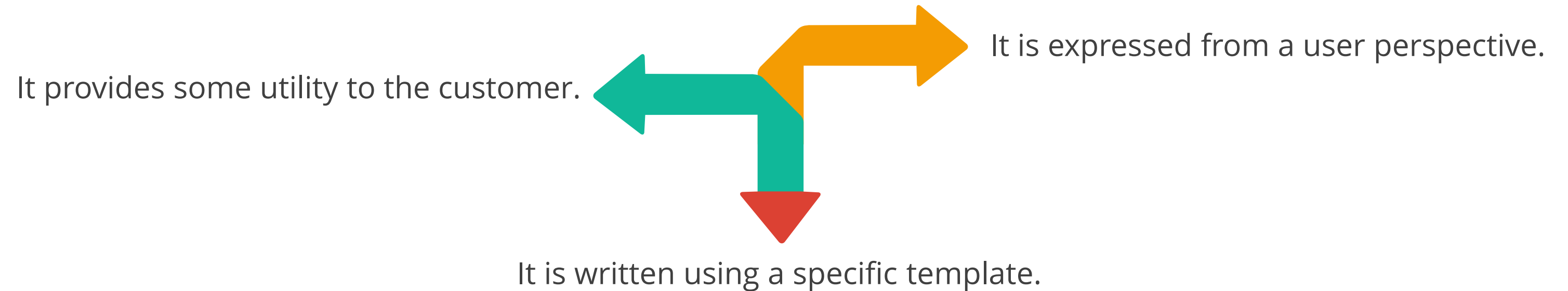
Age: 28, F

Designation: Marketing manager

- **Demographics:** Emily is a 28-year-old marketing manager living in a busy city. She has an intermediate fitness level and works long hours, which often leaves her feeling stressed and short on time.
- **Goals:** Emily wants to maintain weight and reduce stress by incorporating regular exercise into her routine. She is also interested in improving her strength and flexibility.
- **Pain points:** Emily struggles with finding the time and motivation to work out consistently. She often feels overwhelmed by the number of fitness options available and is unsure which exercises are best for her goals.
- **Behavioral patterns:** Emily would use the app 3-4 times a week, primarily in the evenings. She would likely engage with features like personalized workout plans, quick stress-relief exercises, and progress tracking. She would also benefit from reminders and motivational content to keep her on track.

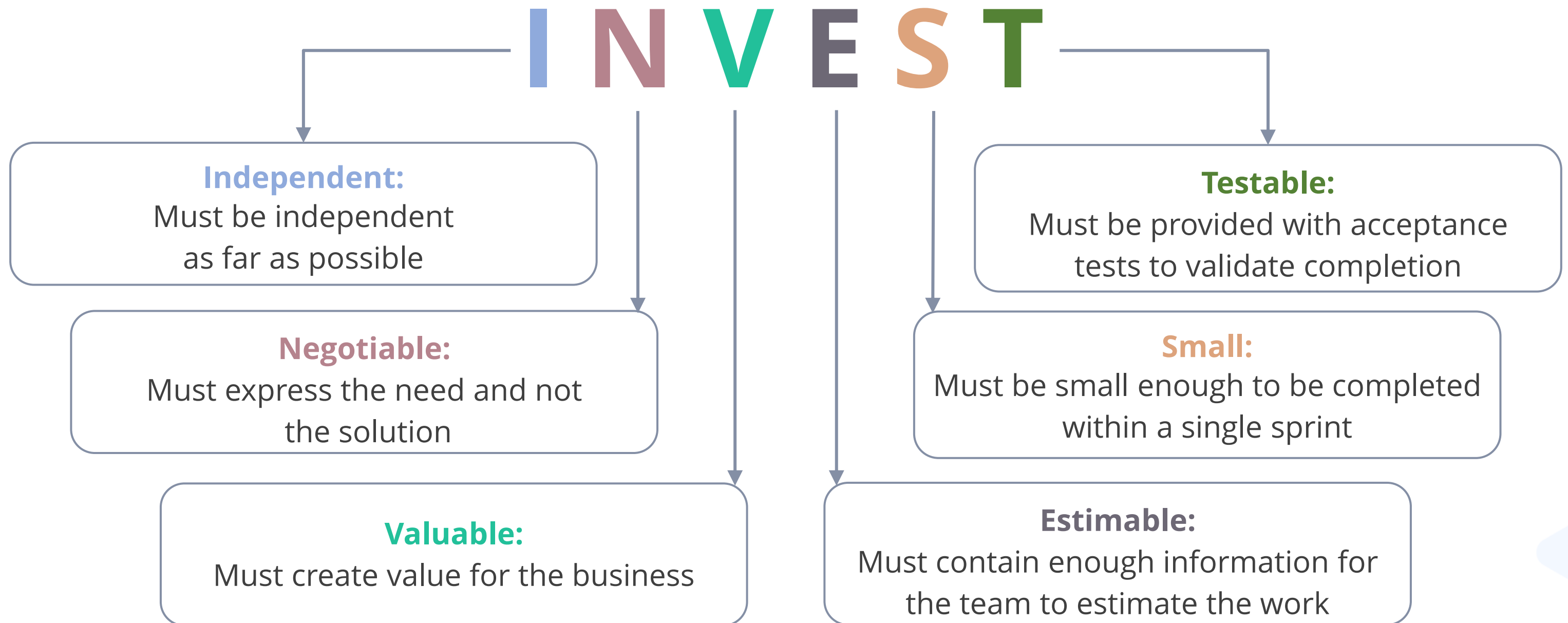
What Is User Story?

It is a short, simple description of a feature from the user's perspective, explaining what they need and why to guide development.



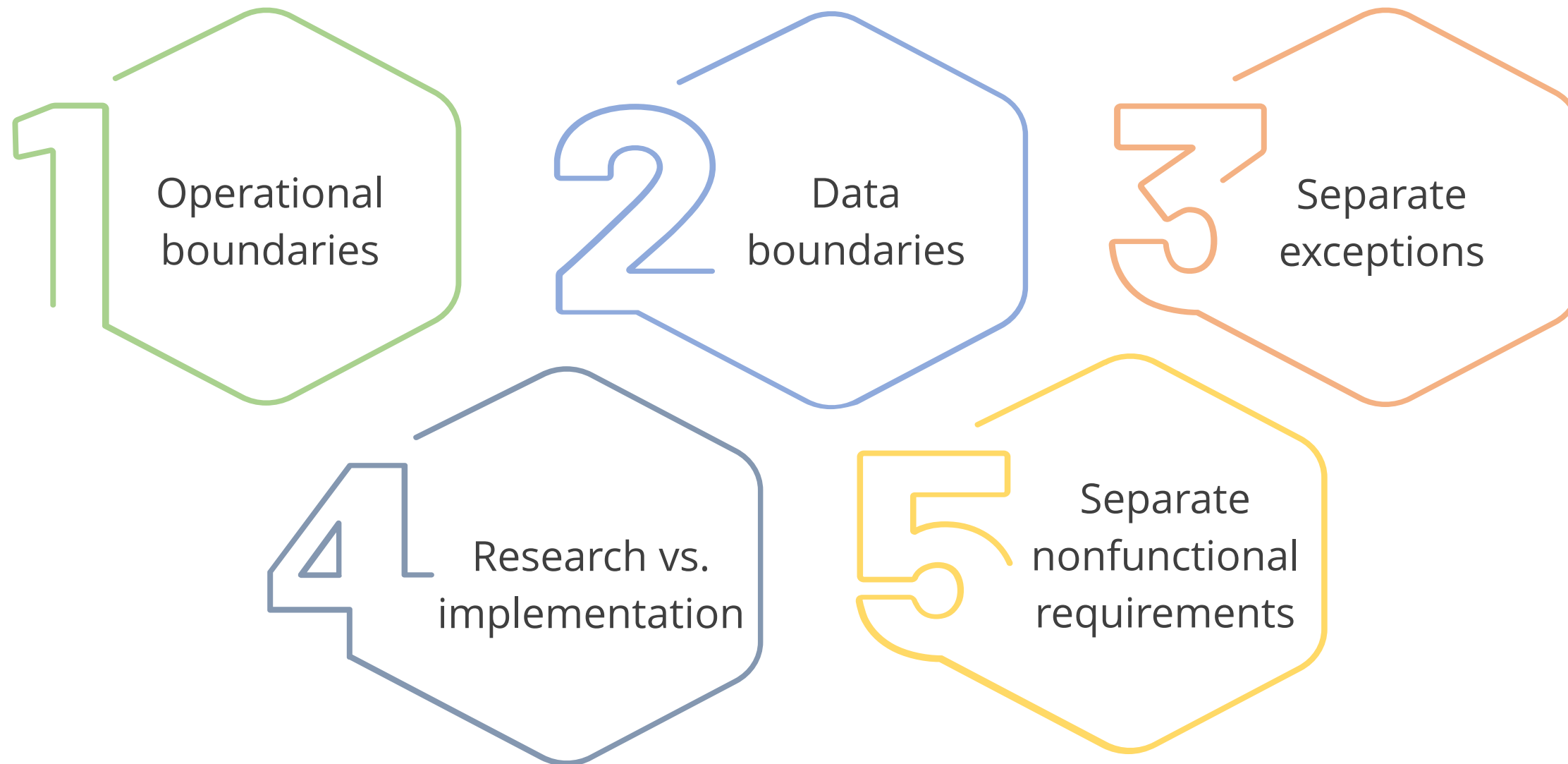
INVEST Model

It provides some clues on how to write good user stories.



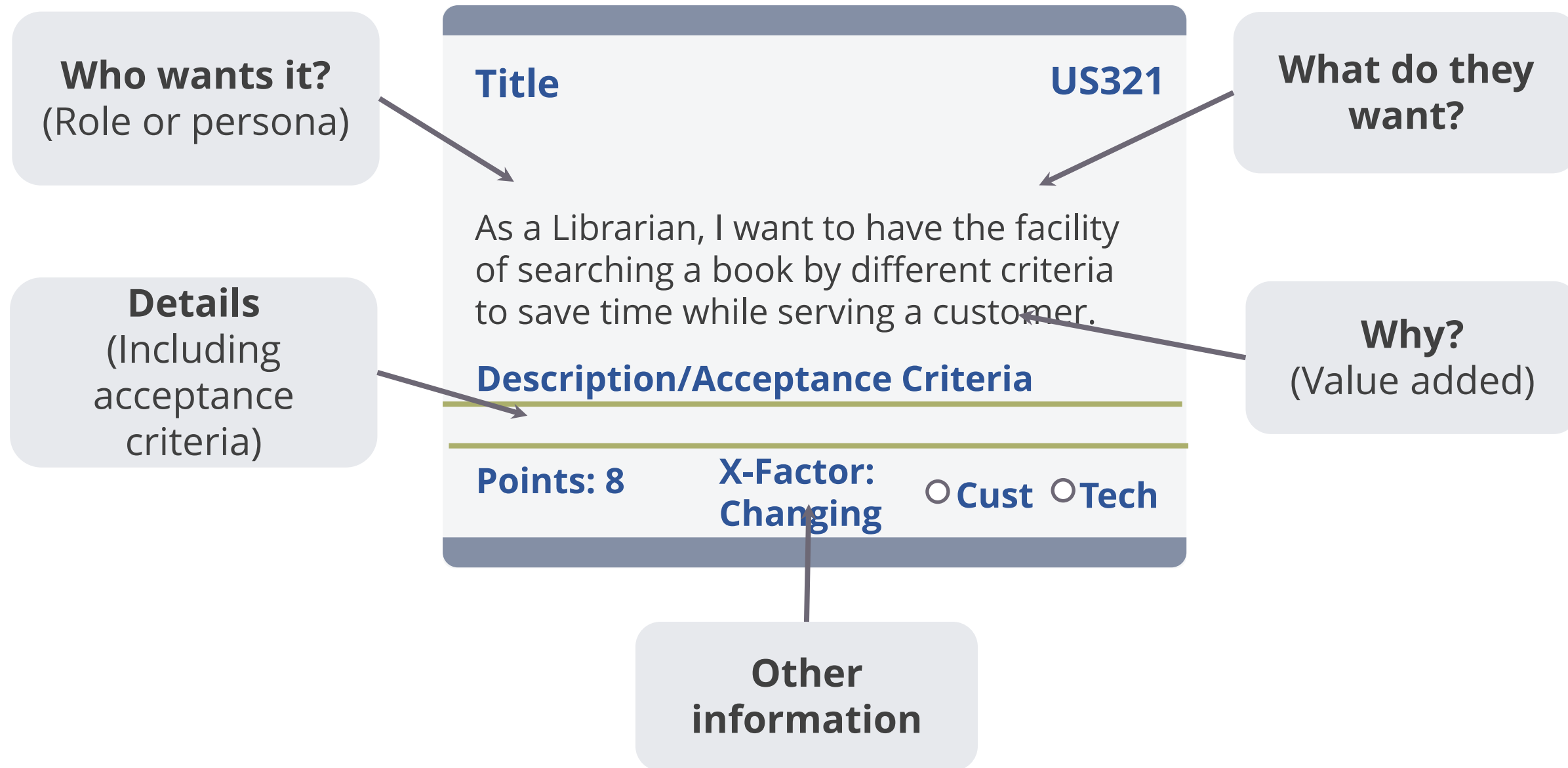
Splitting User Stories

The following methods represent different ways of splitting user stories into smaller units.



User Story Format

Stories are often written on cards. A typical story card will contain the following information:





Activity: Writing User Stories

Activity: Writing User Stories

Scenario:

You are part of an Agile Scrum team working on developing a new mobile application for a fitness company. The app is intended to help users achieve their fitness goals by offering personalized workout plans, nutrition advice, and progress tracking. To ensure that the app's features meet the needs of its users, you have to write clear and actionable user stories that guide the development process.

When writing your user stories, consider the following:

User type: Who is the user? (For example, a beginner, an experienced athlete, or someone looking to improve general fitness)

Feature or action: What does the user want to do with the app? (For example, create a workout plan, track progress, and receive nutrition tips)

Benefit: Why is this feature important to the user? (For example, to stay motivated, reach specific fitness goals, and ensure balanced nutrition)

Sample Solution

User Story 1:

- **User Type:** Beginner
- **Feature or action:** The user wants to create a personalized workout plan
- **Benefit:** The user will be able to follow a structured routine that helps them to start exercising safely and effectively.

User Story 2:

- **User Type:** Experienced athlete
- **Feature or action:** The user wants to track the workout progress over time
- **Benefit:** The user will be able to monitor the performance and make data-driven decisions to improve their fitness levels.

What Is an Epic?

It is a large body of work that can be broken down into smaller, more manageable tasks or user stories.

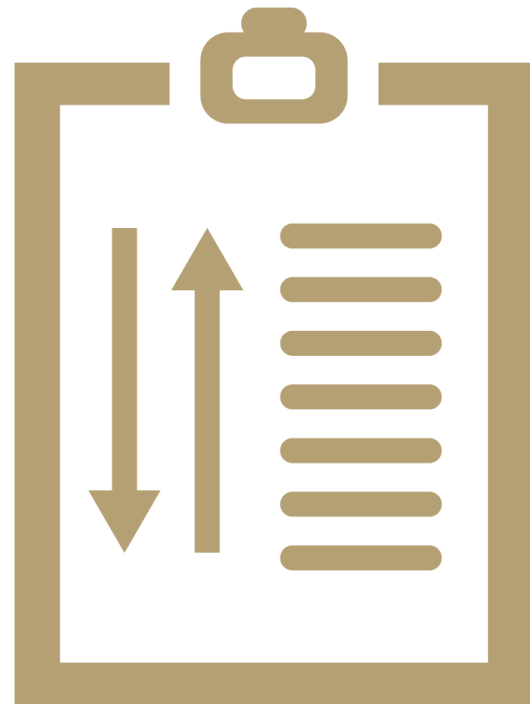


They are typically used to group related work under a single category, making it easier to manage and prioritize tasks.

Prioritization Models

These are crucial for prioritizing tasks, user stories, and features.

Some common prioritization models used in Agile Scrum are:



Kano model

Relative weights method

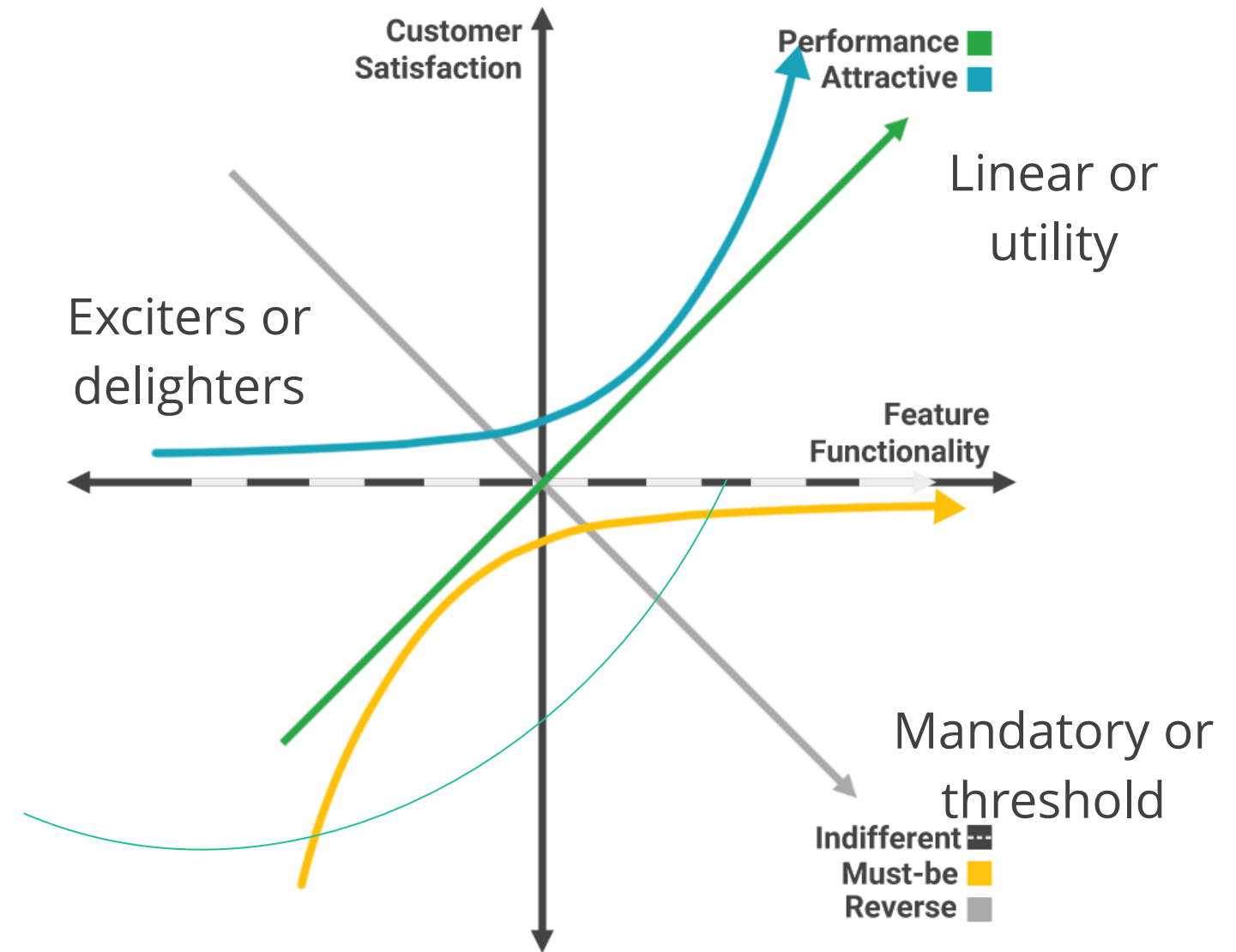
Value risk matrix

Kano Model

It is used to prioritize features based on customer satisfaction and the level of functionality provided.

Delight
↑
Immediate happiness
↑
Not unhappy
↑
Disappointed

Levels of customer satisfaction

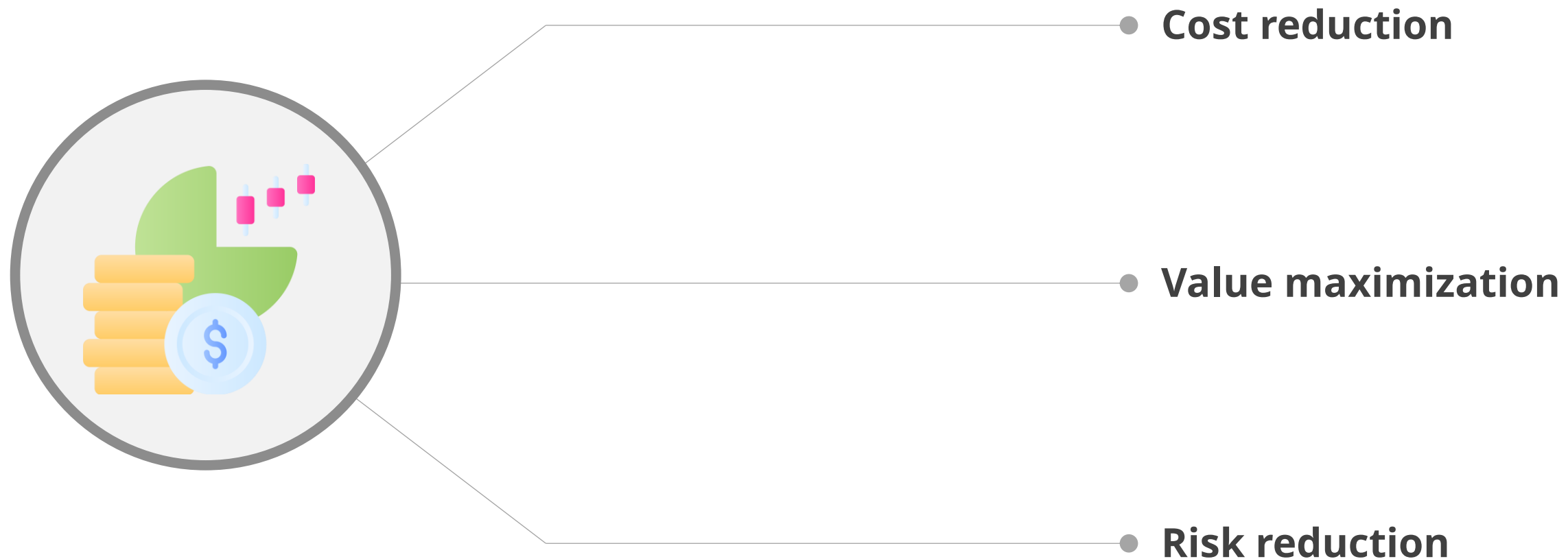


Customer satisfaction vs.
Feature functionality

Relative Weights Method

It is a technique used to prioritize features by comparing their value and cost to focus on the most impactful work.

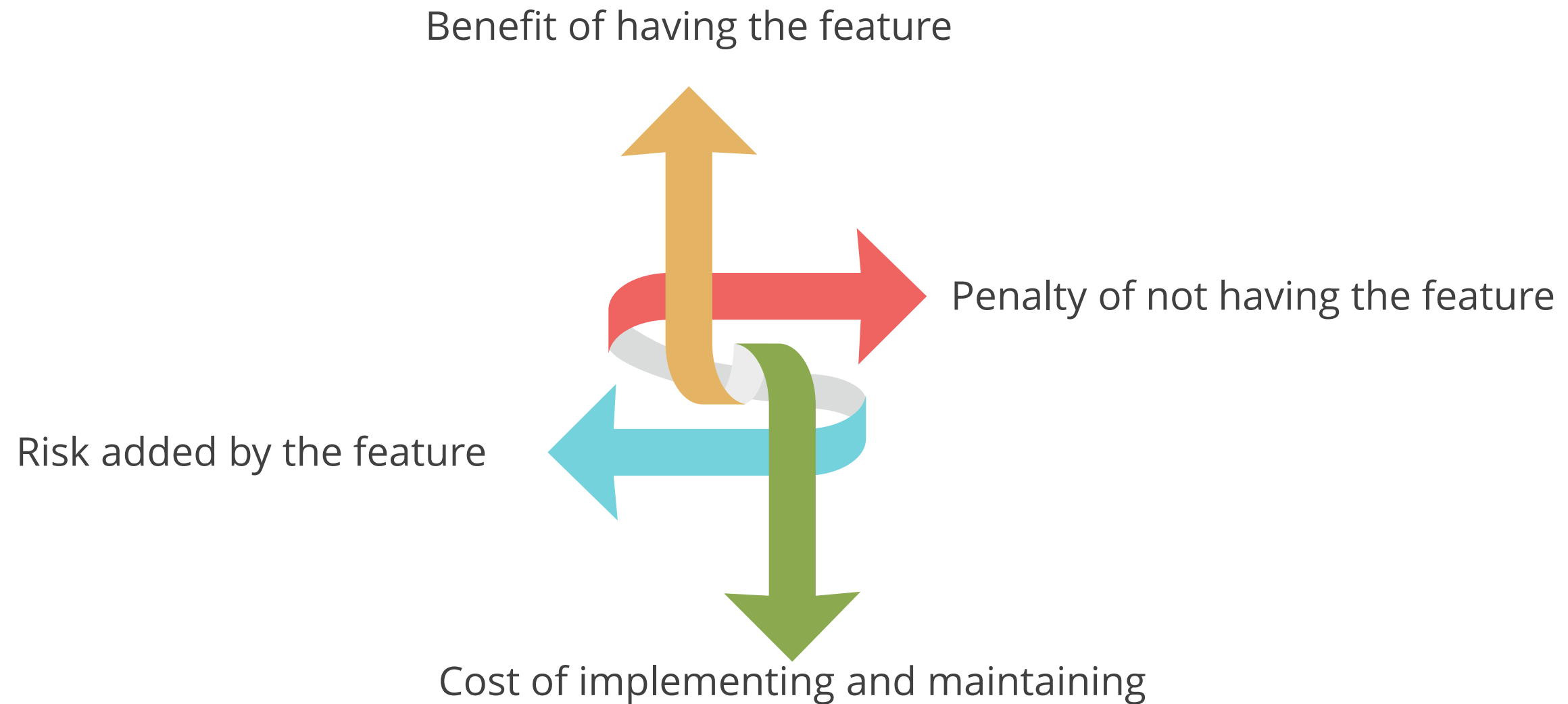
The key factors used in the relative weights method to prioritize work are:



It provides numeric rankings for different parameters and combines them in different proportions to get a numeric priority score.

Relative Weights Method

Ranks (scores) are assigned per feature based on:



Value score = Benefit score + Penalty score
(Value is added either by giving benefits or removing penalties)

Relative Weights Method

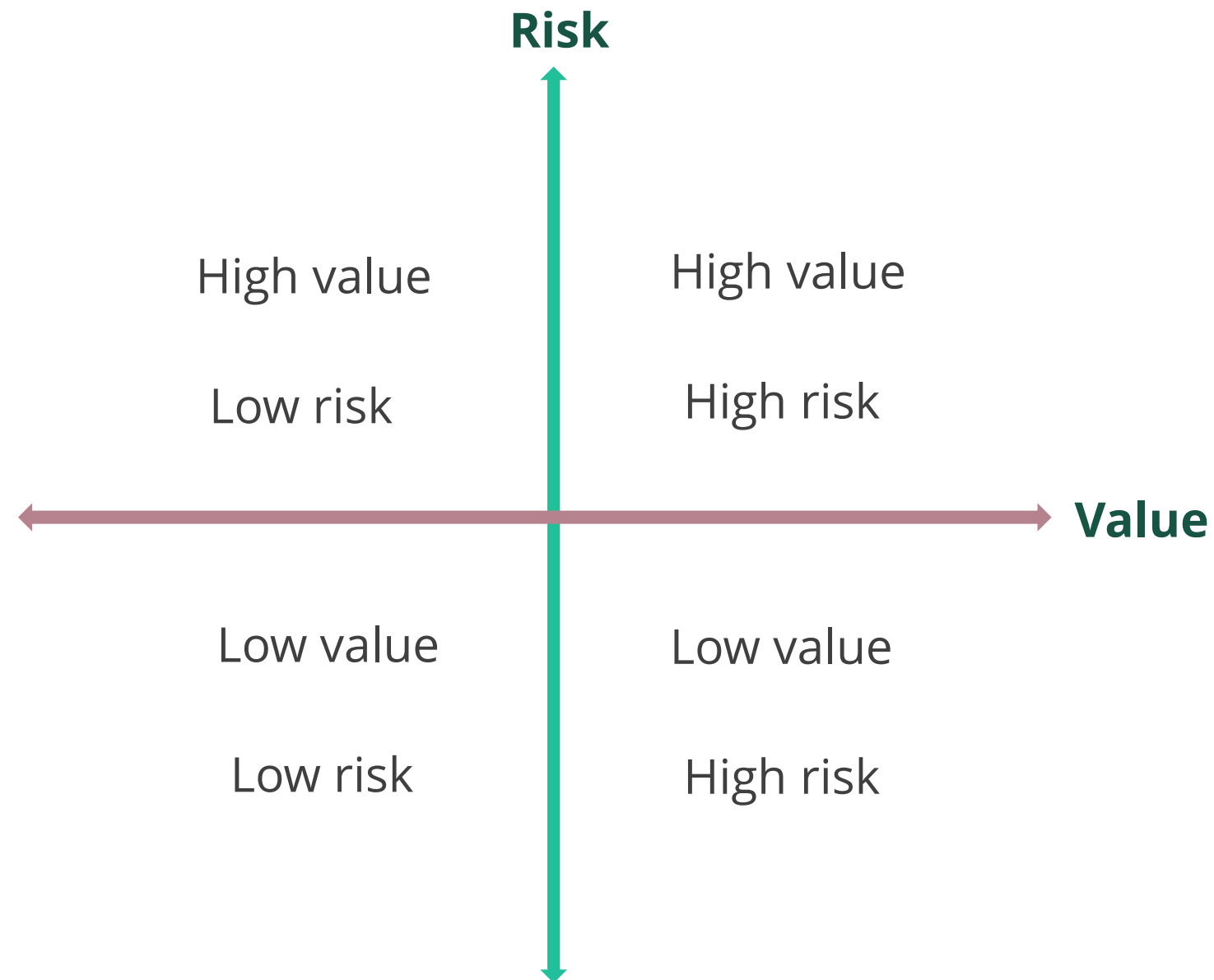
Numeric priority is calculated based on the formula below:

$$Priority = \frac{Value\ Percentage * Value}{Cost\ Percentage * Cost\ weight + Risk\ Percentage * Risk\ Weight}$$

Note: Refer to this method, especially when considering multiple factors, such as value, cost, and risk, to determine the priority of tasks or features

Value Risk Matrix

It prioritizes features by mapping their benefits against associated risks to guide decision-making and resource allocation.



Risk-value prioritization matrix

The features or tasks can be prioritized based on the value risk matrix as below:

1. High risk - high value
2. Low risk - high value
3. Low risk - low value
4. High risk - low value



**Activity: Prioritizing the Backlogs Using
Value Risk Matrix and Kano Model**

Activity: Prioritizing the Backlogs Using Value Risk Matrix and Kano Model

Prioritize the six requirements in the backlog as given in the table using the value risk matrix and Kano model

Requirements	Value (H/L)	Risk (H/L)	Mandatory	Linear	Exciter	Priority
Simplify navigation by adding bookmarks and back or forward buttons						
Modify architecture to support 10,000 learners						
Payment via American Express cards in addition to Visa and Mastercard						
Improve search to show relevant pages for every question						
Add spot quizzes based on audio-visual cues to engage learners						
Add a live chat option for learner support						
Total						

Sample Solution

Requirements	Value (H/L)	Risk (H/L)	Mandatory	Linear	Exciter	Priority
Simplify navigation by adding bookmarks and back or forward buttons	High	Low	Yes	Yes	No	High
Modify architecture to support 10,000 learners	High	High	Yes	Yes	No	Medium
Payment via American Express cards in addition to Visa and Mastercard	Medium	Low	No	Yes	No	Medium
Improve search to show relevant pages for every question	High	Medium	Yes	Yes	No	High
Add spot quizzes based on audio-visual cues to engage learners	Medium	Medium	No	No	Yes	Low
Add a live chat option for learner support	High	Medium	No	No	Yes	Medium
Total	4 High, 2 Medium	3 Low, 3 Medium, 1 High	3 Yes, 3 No	4 Yes, 2 No	3 No, 3 Yes	2 High, 3 Medium, 1 Low

Story Points

They are a relative unit of measure used to estimate the effort required to implement a product backlog item or any other piece of work.

They help to:



Ideal Time Estimates

One of the units of estimation used on Agile projects is ideal time.



- It is the time taken to complete a piece of work under ideal circumstances.
- Distractions depend upon who is doing the work and other preoccupations.
- Estimation work is challenging enough, but it is more challenging to estimate distractions.
- To convert ideal time to actual (elapsed) time, the user needs a conversion factor.

Ideal Hours

It refers to the estimated time to complete a task if there are no interruptions, distractions, or delays.



This estimation assumes a perfect, focused working environment where the team can dedicate their full attention to the task at hand.

Ideal Days

It represents the number of full, uninterrupted days required to complete a task or set of tasks under perfect conditions.



It provides a higher-level estimate useful for planning larger tasks or entire sprints.

Velocity

It is the amount of work that a team completes in a single sprint.

Velocity is used to calculate:



Number of stories that should be accepted in each sprint



Number of sprints required to complete a set of stories

Velocity Calculation: Example

A team completed five stories of sizes 5, 3, 8, 13, and 2. What is the velocity of the team?



Answer

$$\text{Velocity} = 5 + 3 + 8 + 13 + 2 = 31$$

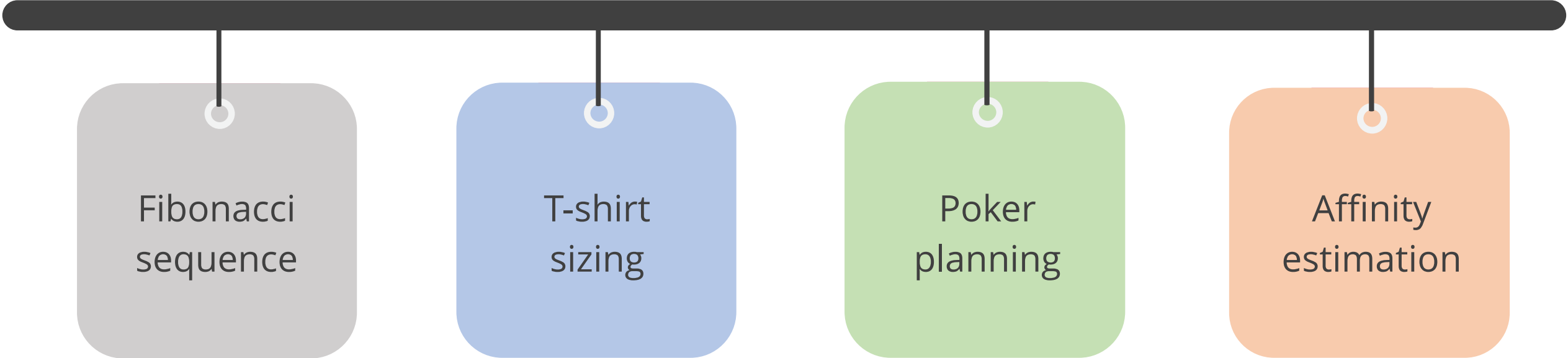


Agile Estimation Methods

Agile Estimation Methods

These methods help teams predict the effort required to complete tasks.

Common estimation methods include:



Fibonacci
sequence

T-shirt
sizing

Poker
planning

Affinity
estimation

Fibonacci Sequence

It is a sequence of numbers often used in Agile estimation to reflect the increasing complexity or uncertainty in tasks as they grow larger.

0 1 1 2 3 5 8 13 21

$$0 + 1 = 1$$

$$1 + 1 = 2$$

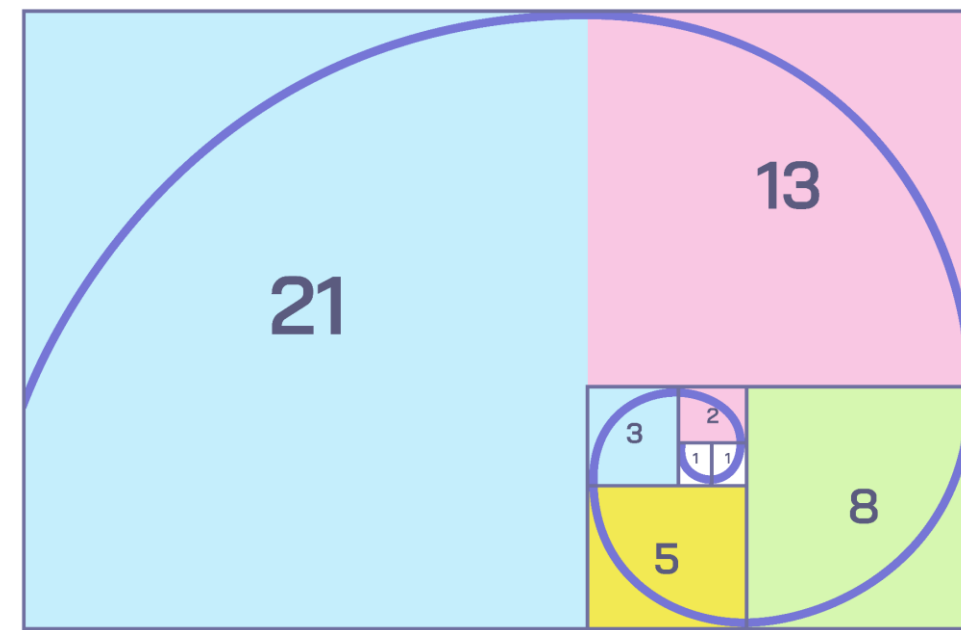
$$1 + 2 = 3$$

$$2 + 3 = 5$$

$$3 + 5 = 8$$

$$5 + 8 = 13$$

$$8 + 13 = 21$$



The sequence typically used in Agile is 1, 2, 3, 5, 8, 13, 21, and so on.

T-Shirt Sizing

It is a relative estimation technique used in Agile to quickly categorize the size or effort of tasks or user stories without getting into precise details.



T-shirt size	Number of story points
XS	1
S	2
M	3
L	5
XL	8
XXL	13
XXXL	21

This is often used in the early stages of a project when high-level estimations are needed or when teams need to estimate many user stories quickly.

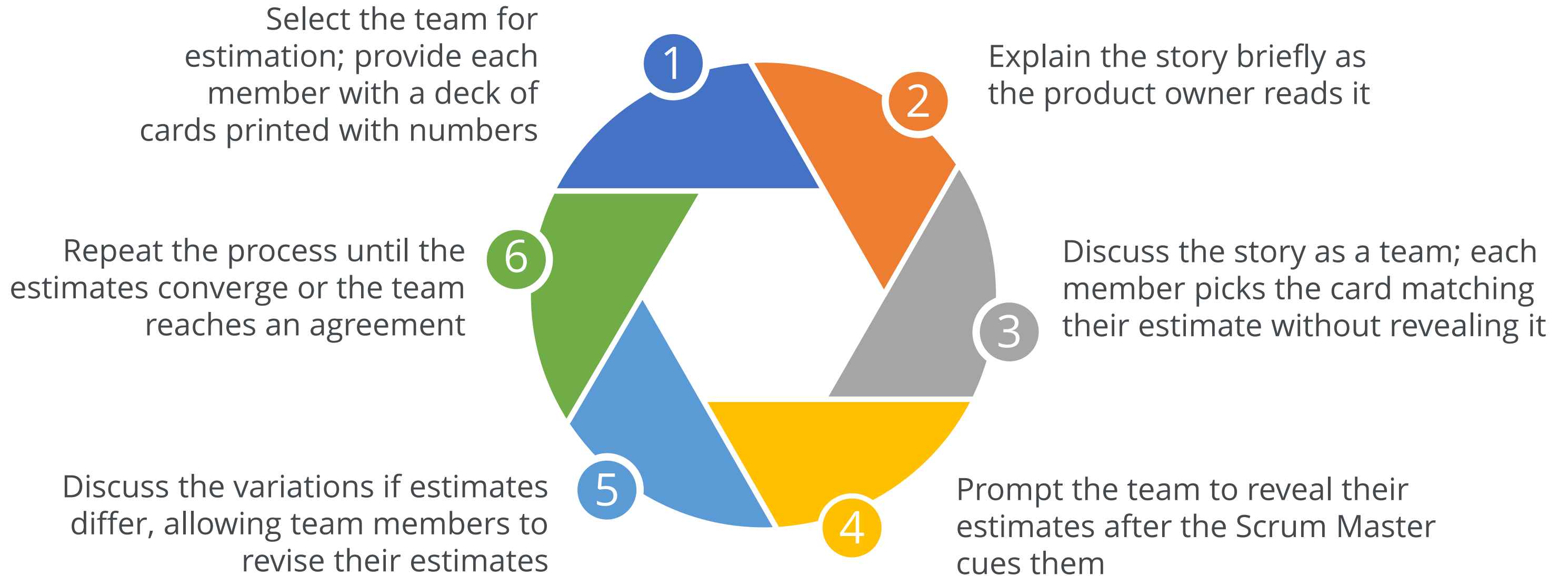
Poker Planning

It is a popular, team-based estimation method that generates useful discussion.



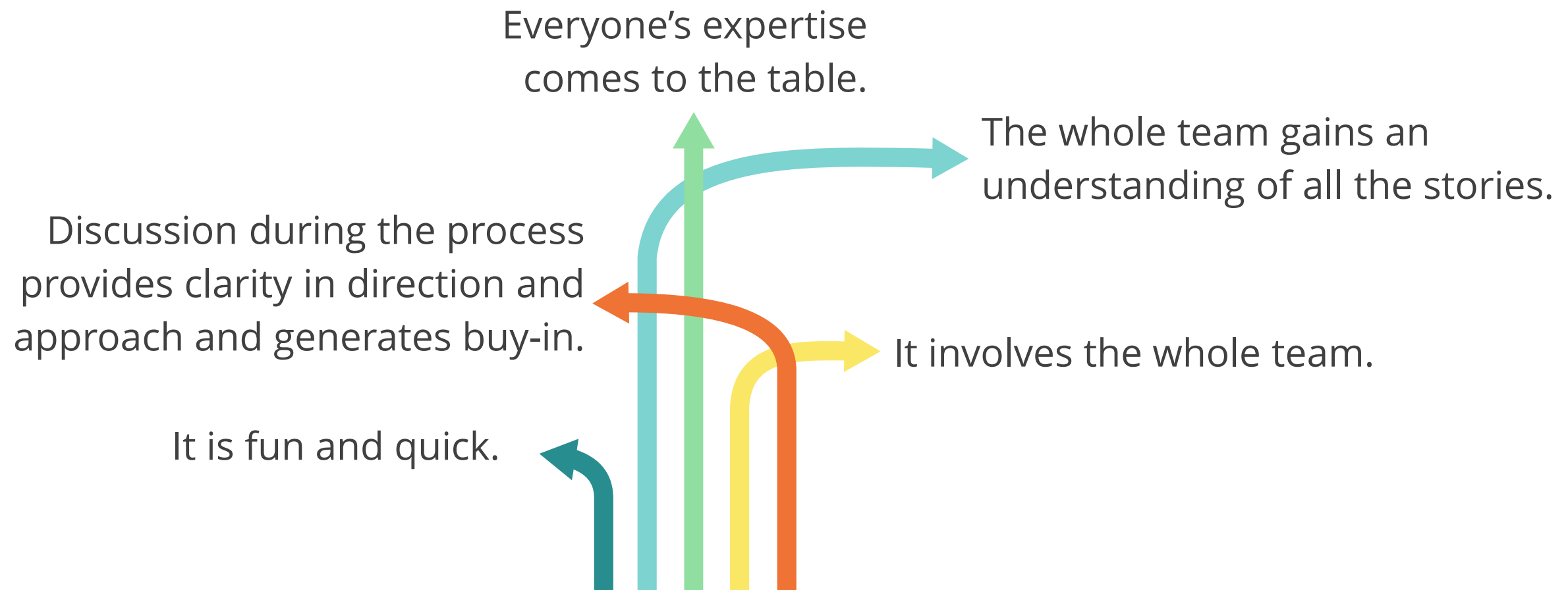
It combines individual input with group discussion, promoting balanced and informed estimates to encourage team alignment on the effort required for tasks.

Steps for Poker Planning



Why Poker Planning Works?

The major advantages of the Poker planning technique are:





Activity: Poker Planning Simulation

Activity: Poker Planning Simulation

Scenario:

Your team is working on enhancing an e-learning platform that has been gaining popularity among learners. However, with the rapid increase in users, the platform needs several improvements to maintain its quality and user experience. The product owner has outlined a few critical tasks, and it's time for the development team to estimate these tasks using Poker planning, a method that ensures everyone's voice is heard in the estimation process. You are part of the Agile team responsible for improving the e-learning platform.

Activity: Poker Planning Simulation

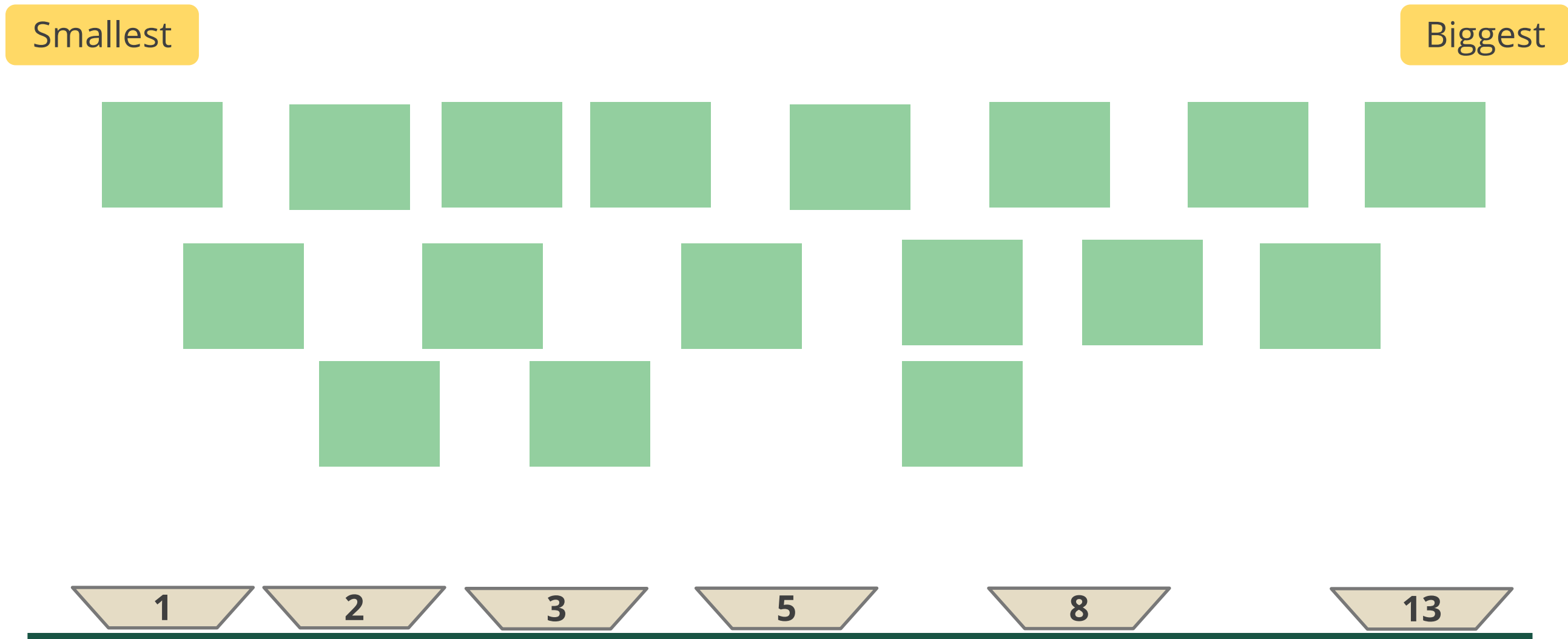
Task:

- Simplify navigation by adding bookmarks and back or forward buttons
- Modify architecture to support 10,000 learners
- Enable payment via American Express cards in addition to Visa and Mastercard
- Improve search to show relevant pages for every question
- Add spot quizzes based on audio-visual cues to engage the learners
- Add a live chat option for learner support

Affinity Estimation

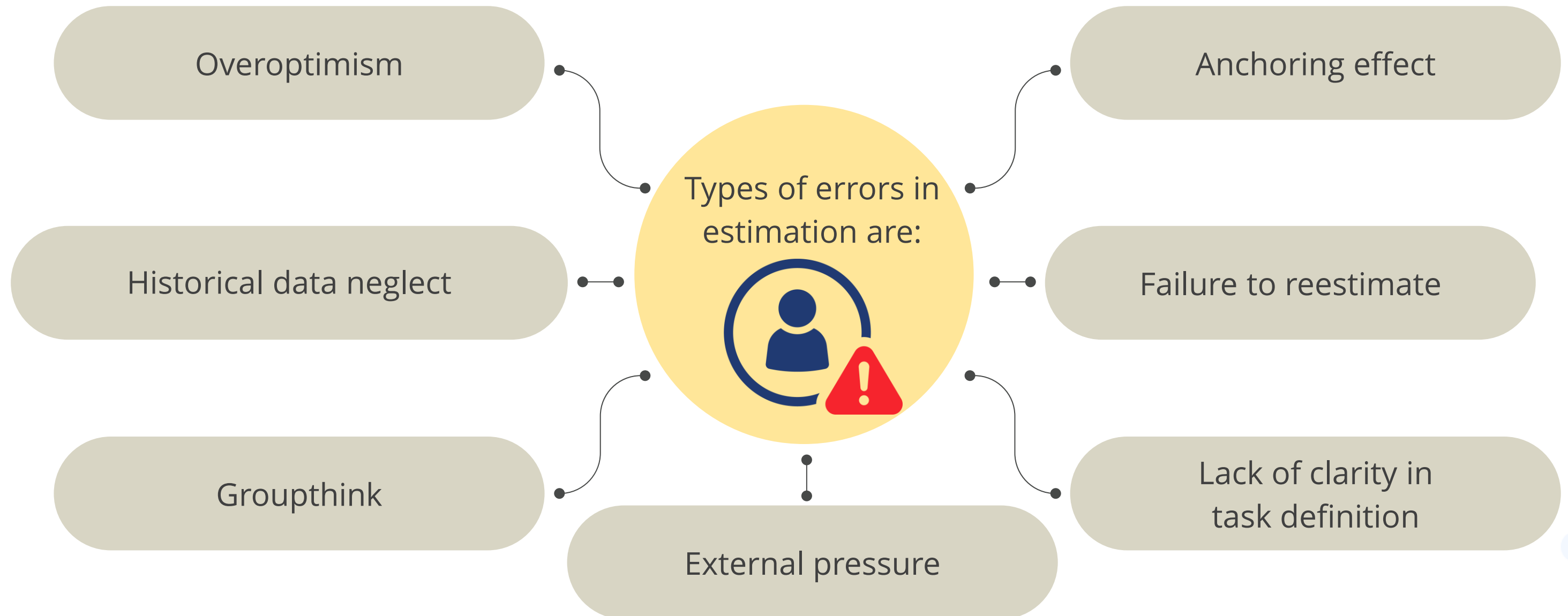
It is a quick, team-based estimation method based on relative sizing that helps the team work out quick and reliable estimates.

Shown below are stories that are arranged in order of their size:



Errors in Estimation

The errors can stem from several sources; understanding them can refine the estimation process to ensure project success.



Types of Errors in Estimation

Overoptimism

Overestimating the efficiency of task completion leads to unrealistic timelines and increased pressure.

Historical data neglect

Ignoring past project data and lessons learned results in inaccurate estimates.

Groupthink

Conforming to consensus without independent evaluation can skew estimations.

Anchoring effect

Initial estimates disproportionately influence final estimates, regardless of accuracy.

Types of Errors in Estimation

Failure to reestimate

Poorly defined tasks make accurate estimation difficult.

Lack of clarity in task definition

Not adjusting estimates with new information leads to discrepancies between estimated and actual requirements.

External pressure

External deadlines can force underestimation, risking quality and deadlines.



Agile Planning

Introduction to Agile Planning

It is a dynamic and iterative approach to project management that focuses on delivering value to customers through frequent and incremental updates.



Different Levels of Agile Planning

Agile planning functions on several levels, each with its specific purpose and timeframe. These levels help teams balance between their long-term goals and immediate tasks.

The different levels of planning in Agile are:



What Is Portfolio Planning?

It encompasses selecting and prioritizing different projects and initiatives to fulfill the organization's strategic objectives.



Portfolio management and planning determine which products support the organization's goals and objectives.



Activity: Product Backlog Prioritization Discussion

Activity: Product Backlog Prioritization Discussion

Scenario:

Your Agile team is preparing for the next sprint, and the product backlog contains a variety of user stories, features, and bug fixes. The list includes items such as implementing a new search functionality, enhancing security measures, fixing a critical bug, and improving the user interface. As the Scrum Master, you need to facilitate a discussion with your team to prioritize these items based on their business value, urgency, and technical dependencies. The goal is to ensure that the team focuses on the most impactful tasks that align with the project's goals.

Task:

Facilitate a discussion to prioritize the two main items in the backlog: the critical bug fix and the new search feature.

Consider the following factors:

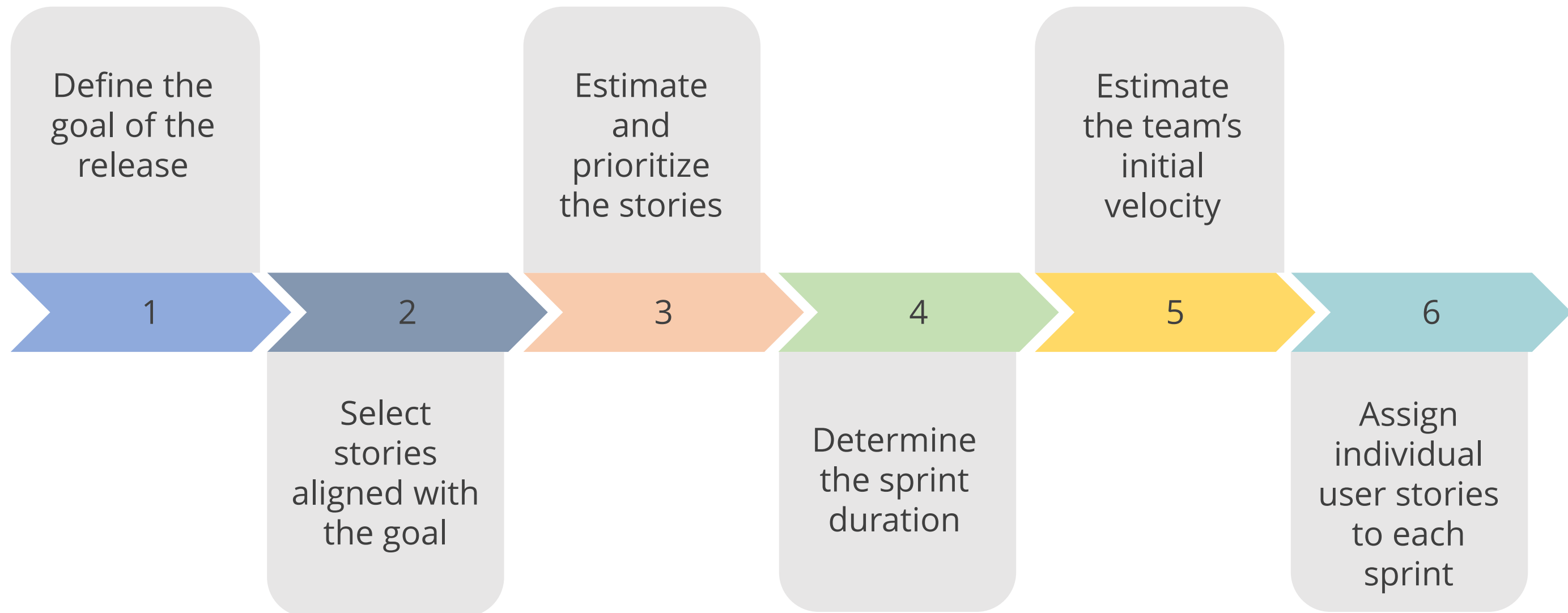
How urgent is each task? Is the bug affecting a lot of users? Does the search feature need to be live by a certain deadline?

Which task will provide the most value to users and stakeholders?

What Is Release Planning?

It defines the scope and schedule for upcoming product releases.

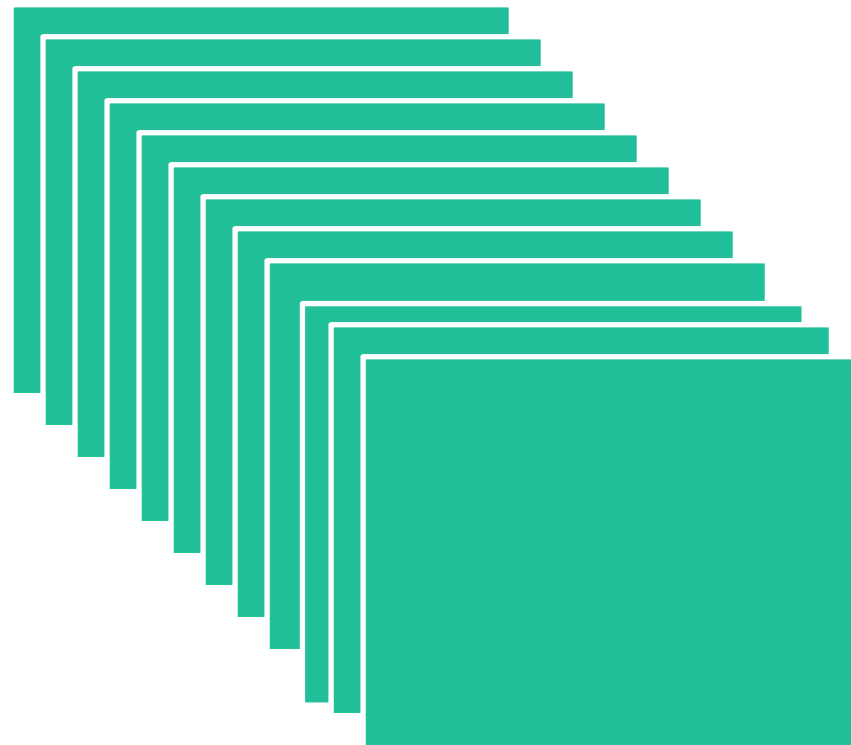
Steps for release planning:



Release Planning

In a release planning session, stories are tentatively assigned to different sprints or iterations.

Stories targeted at the release



Release planning meeting

Stories tentatively assigned to specific iterations

Iteration 1



Iteration 2



Iteration 3



Iteration 4





Activity: Creating a Release Planning

Activity

Scenario:

The product owner at Hi-Tec Learning would like to plan the release of the online math course during the summer vacation. The course is to be accessed through a computer. The team has listed the stories required and has estimated the velocity for the first four sprints as follows:

- Sprint 1: 40 points
- Sprint 2: 45 points
- Sprint 3: 40 points
- Sprint 4: 40 points

Instructions

Help the team produce a release plan; apply the good practices of Agile planning

No.	Stories	Estimates (Points)
1	Develop the detailed curriculum and storyboard for the course	13
2	Collect the basic images and text	13
3	Collect the exercises and grade by difficulty	8
4	Create metadata for plugging into the learning portal	5
5	Finalize the lesson content	21
6	Provide keyboard and mouse support to access the course	13
7	Implement access controls for authorized and free users	13
8	Create database setup for storing test scores, levels, and other information	8
9	Integrate into the main learning portal	5
10	Add pretests to decide the skill level of student	5
11	Add value-added modules for students with high aptitude	5
12	Implement feedback from beta users	13
13	Provide hooks for advertisements	5
14	Implement billing based on portal pricing policies	5
15	Implement feedback from early customers	8

Sample Solution

Sprint allocation:

Sprint 1 (40 points):

- Story 1: Develop the detailed curriculum and storyboard for the course (13 points)
- Story 2: Collect the basic images and text (13 points)
- Story 3: Collect the exercises and grade by difficulty (8 points)
- Story 4: Integrate into the main learning portal (5 points)
- **Total:** 39 Points

Sprint 2 (45 Points):

- Story 1: Finalize the lesson content (21 points)
- Story 2: Provide keyboard and mouse support to access the course (13 points)
- Story 3: Implement access controls for authorized and free users (13 points)
- **Total:** 47 Points

What is Sprint Planning?

It is a collaborative meeting with the Scrum team, including the product owner, Scrum Master, and development team, to determine what work will be accomplished during the upcoming sprint.

There are two possible approaches to sprint planning:

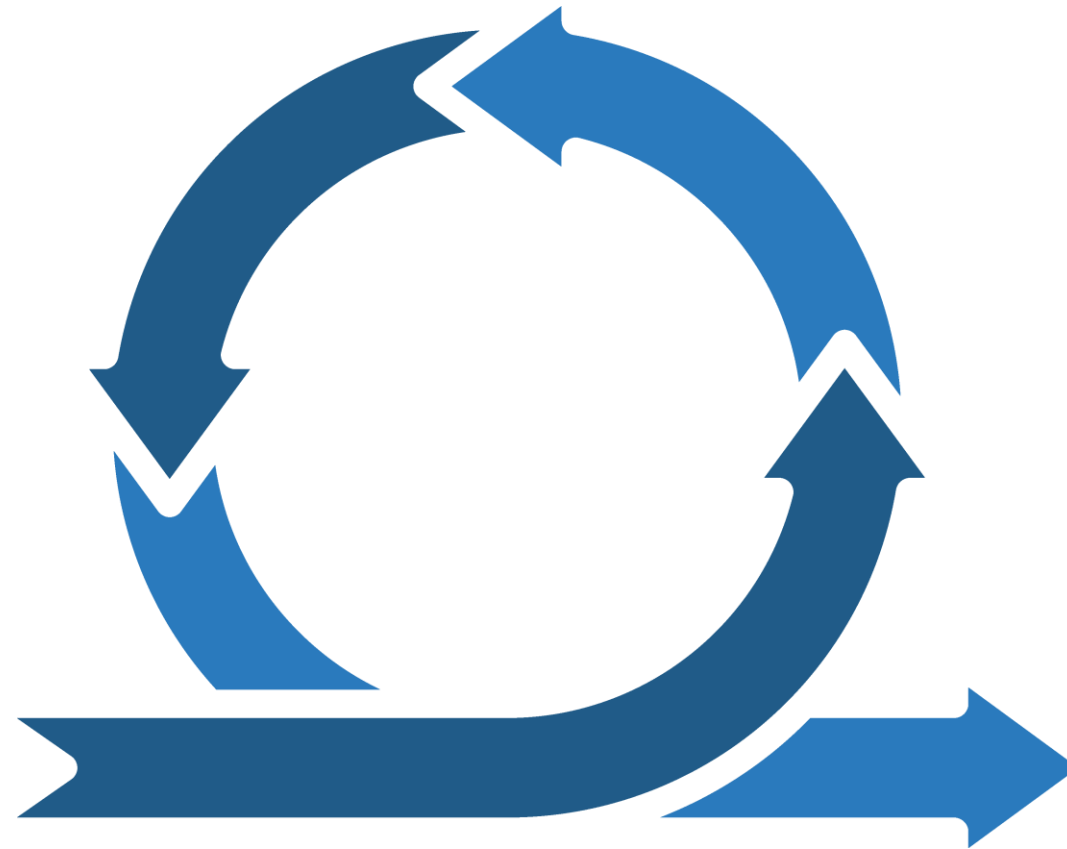
Commitment-driven



Velocity-driven

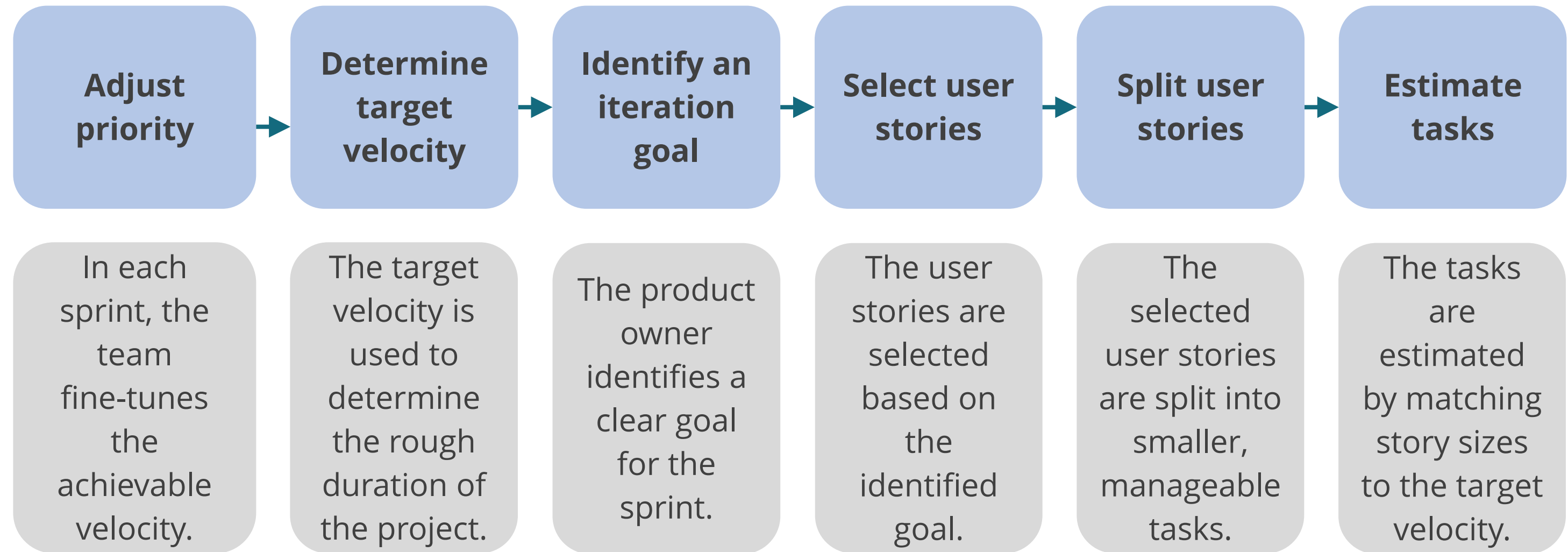
Velocity-Driven Sprint Planning

Velocity is a measure of the amount of work the team can do in each sprint or iteration.



The velocity-driven approach works only when the team knows its velocity.

Steps in Velocity-Driven Sprint Planning



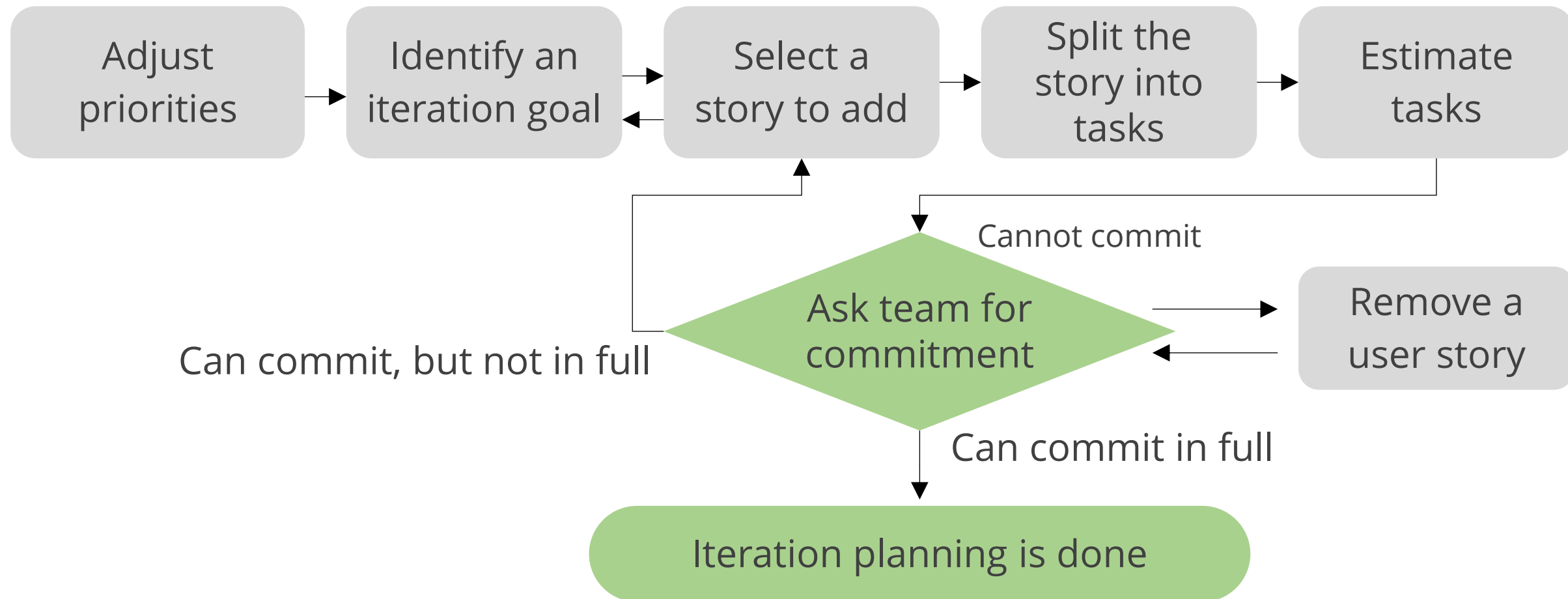
Commitment-Driven Sprint Planning

It is preferred for projects with strict deadlines or when timely delivery is crucial for external partners.



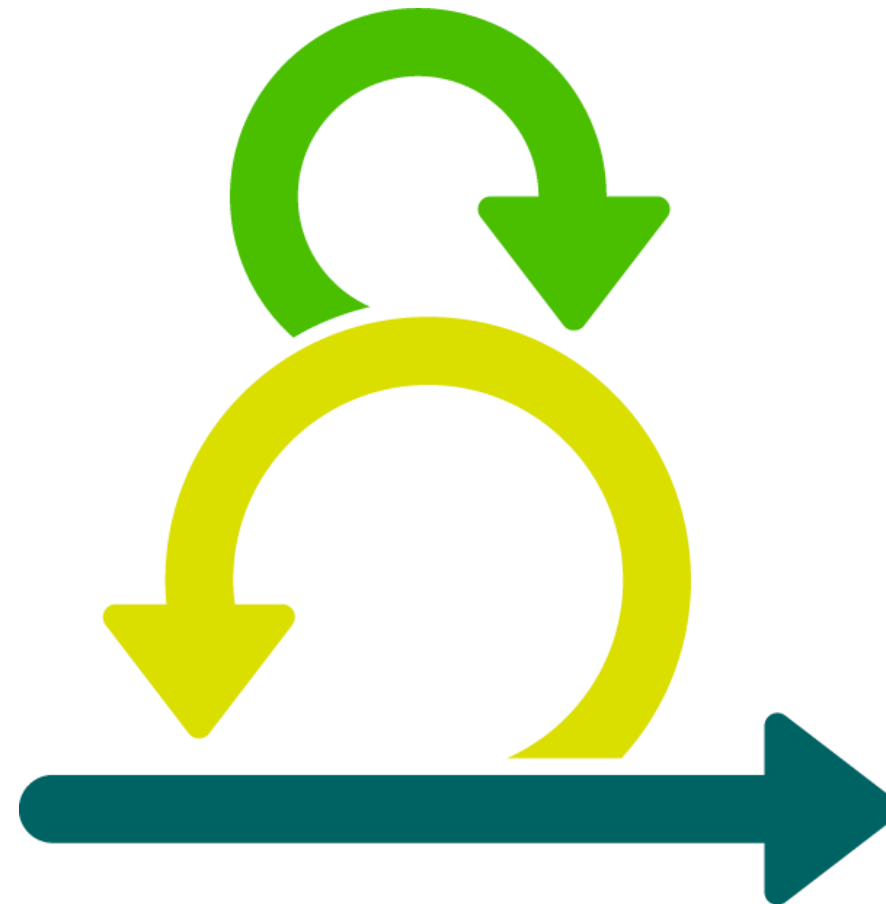
Once the team is unable to commit to additional stories, the addition of story stops, and sprint backlog is finalized.

Steps in Commitment-Driven Sprint Planning



What Is Daily Planning?

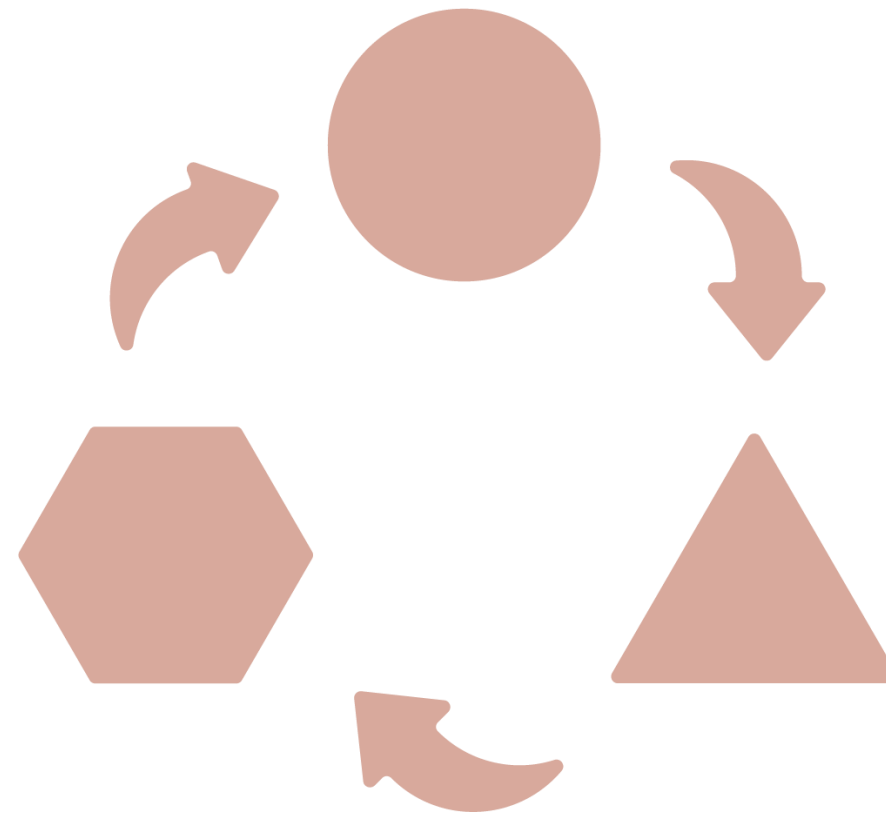
It happens during the daily Scrum meetings where team members plan their work for the day.



Progress, impediments, and coordination of activities are discussed to ensure the sprint goals are met.

What Is Adaptive Planning?

It involves continuously reviewing and adjusting plans based on feedback, new information, and changing priorities.



It emphasizes responsiveness to change and the ability to pivot quickly to address emerging needs or opportunities.



Agile Monitoring

Introduction to Agile Monitoring

It is the ongoing process of tracking and assessing the progress and performance of an Agile team.



It involves regularly reviewing completed work, identifying obstacles, and ensuring alignment with project goals.

Reestimation

It is the process of revisiting and updating the original estimates for tasks or user stories in a sprint or project.

Key considerations for reestimation are:





Challenges Faced While Monitoring

Roadblocks

Roadblocks to be aware of during Agile monitoring include:

Lack of clarity on requirements or user stories

Dependencies on external teams or resources

Technical debt or legacy code

Unforeseen issues or bugs

Team member unavailability or skill gaps

Roadblocks

Roadblocks to be aware of during Agile monitoring include:

Issues in the development or testing process

Challenges in the infrastructure or environment

Lack of access to necessary tools or resources

Communication barriers within the team or
with stakeholders

Organizational policies or procedures that
hinder progress

Deviations

Deviations to be aware of during Agile monitoring include:

Changing project scope or priorities

Varying between estimated and actual effort for tasks or user stories

Delaying the delivery of critical features or milestones

Adding unplanned work or scope creep

Experiencing inconsistent or unpredictable velocity



Information Radiators

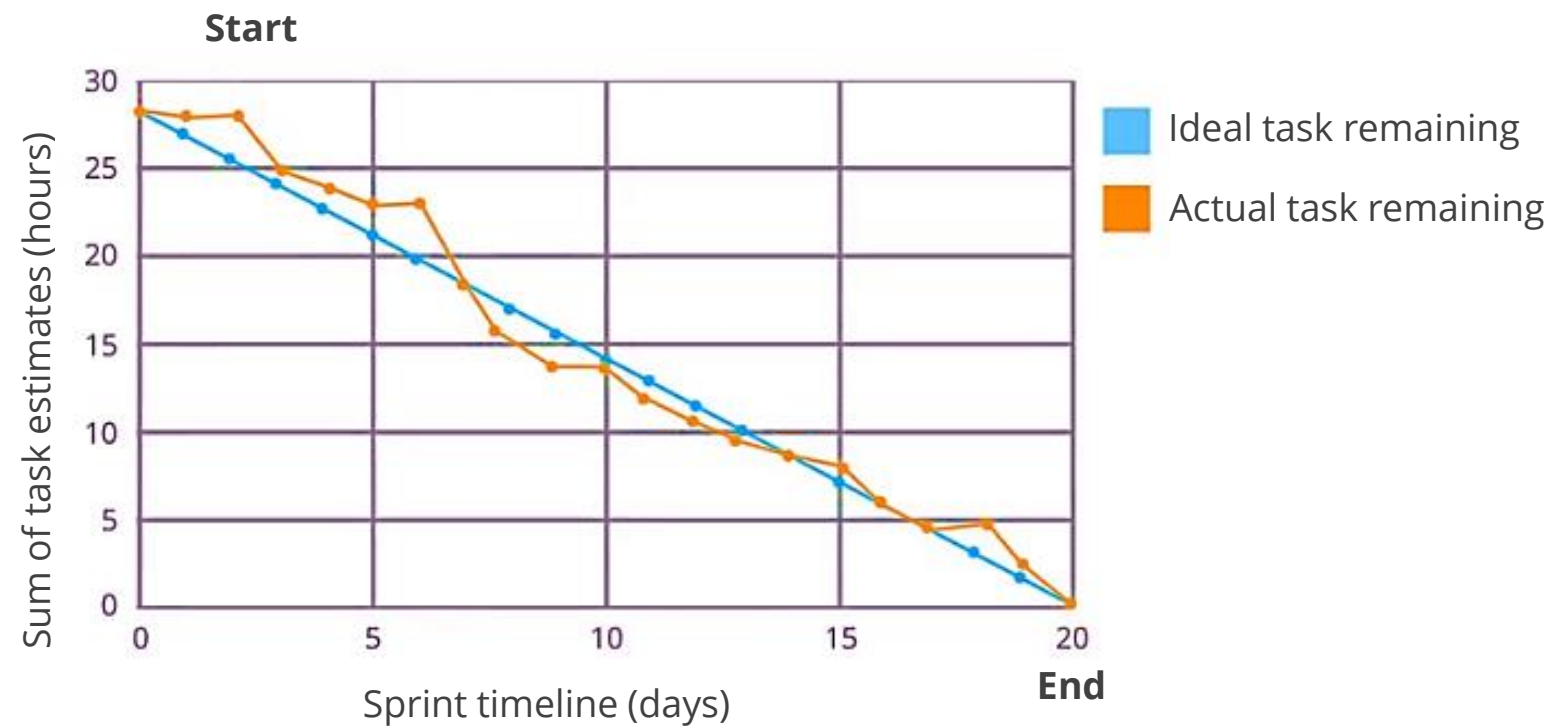
Information Radiators

These visual tools display key project information to the entire team and stakeholders.

Types of information radiators are:

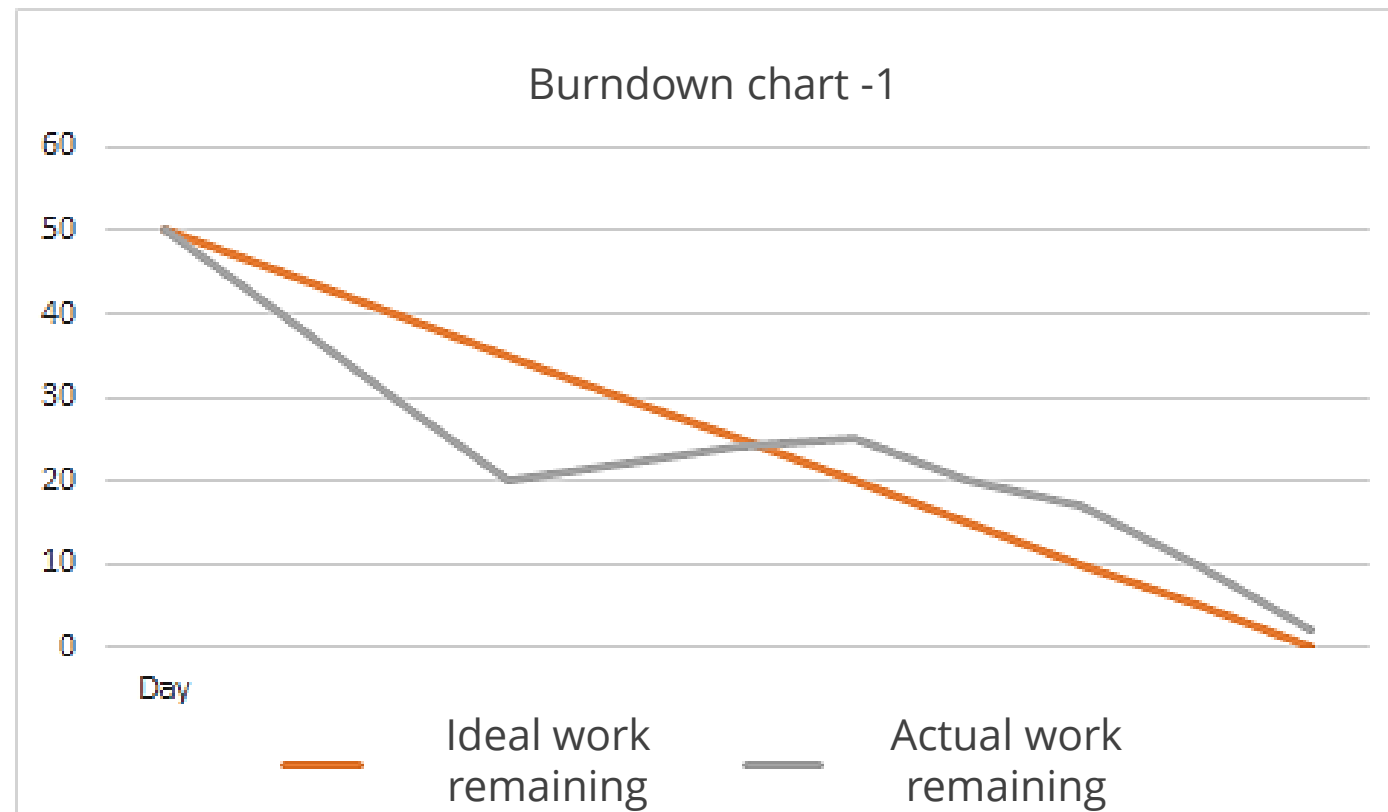


Burndown Charts



- The vertical axis is the work remaining (points or hours).
- The horizontal axis is the timeline (sprint or project).
- The blue line represents the steady burn rate.
- The orange line represents the actual burn rate.

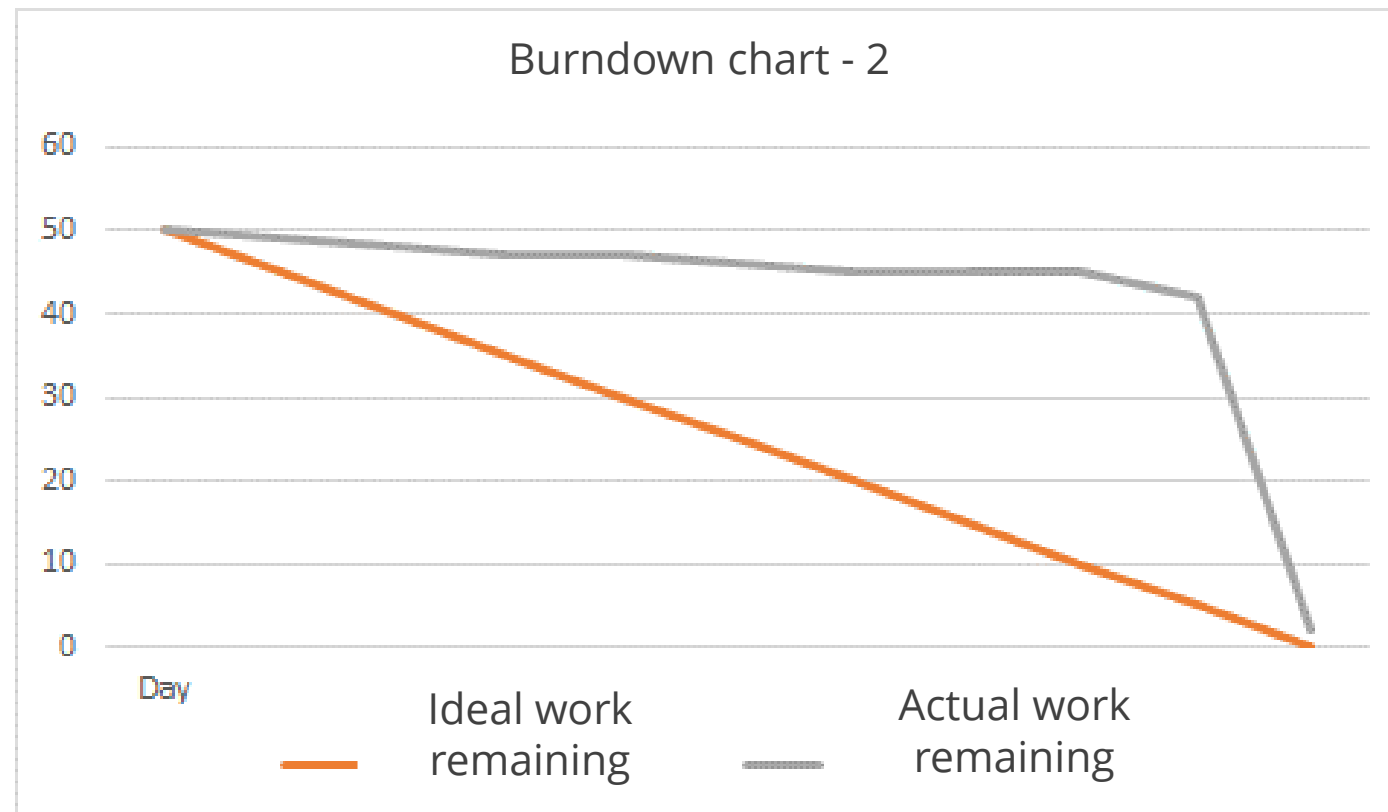
Burndown Chart Interpretation



Scenario: The grey line representing the actual work goes up suddenly.

Interpretation: Unexpected challenges or issues may be causing delays for the team.

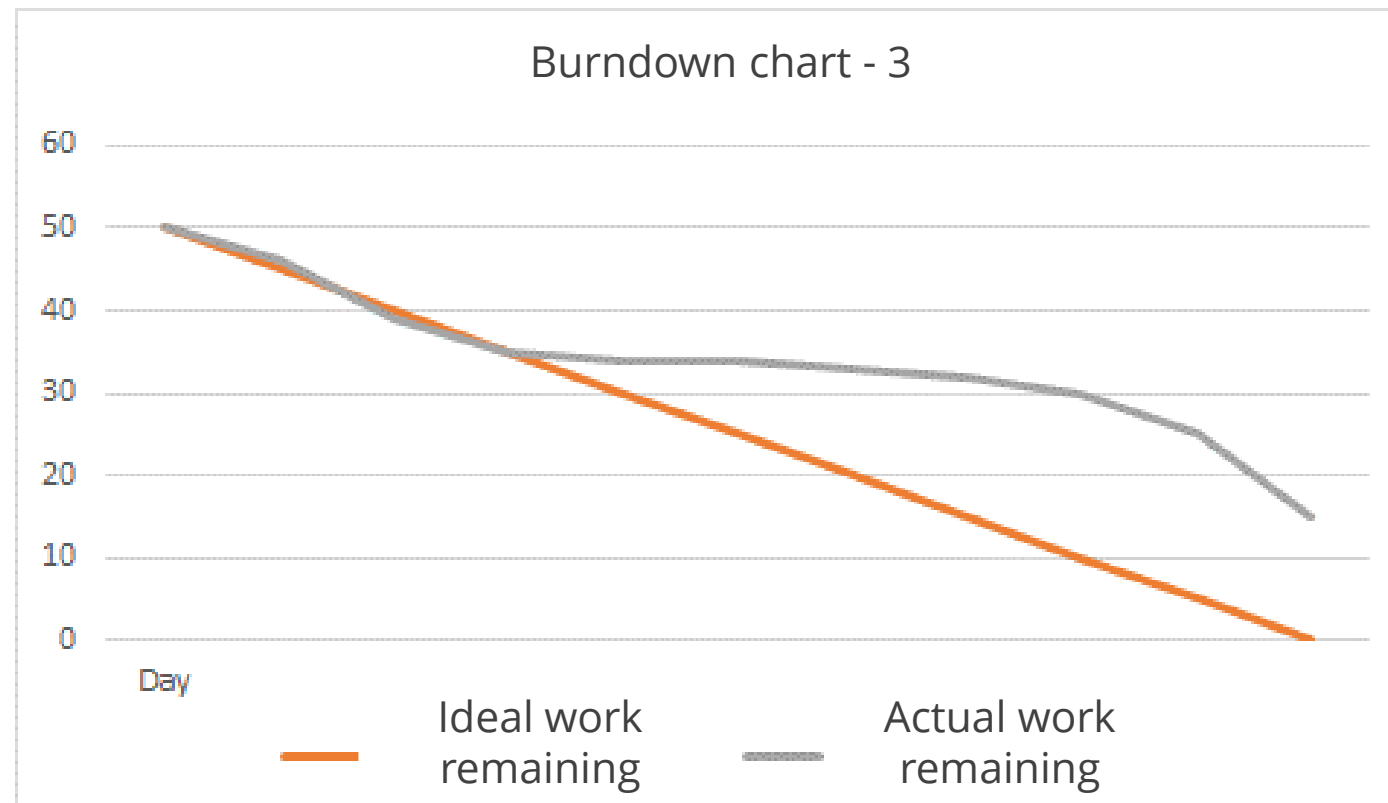
Burndown Chart Interpretation



Scenario: The grey line representing the actual work drops dramatically close to zero towards the end.

Interpretation: The work is either stuck for a long or the team is not updating the remaining work.

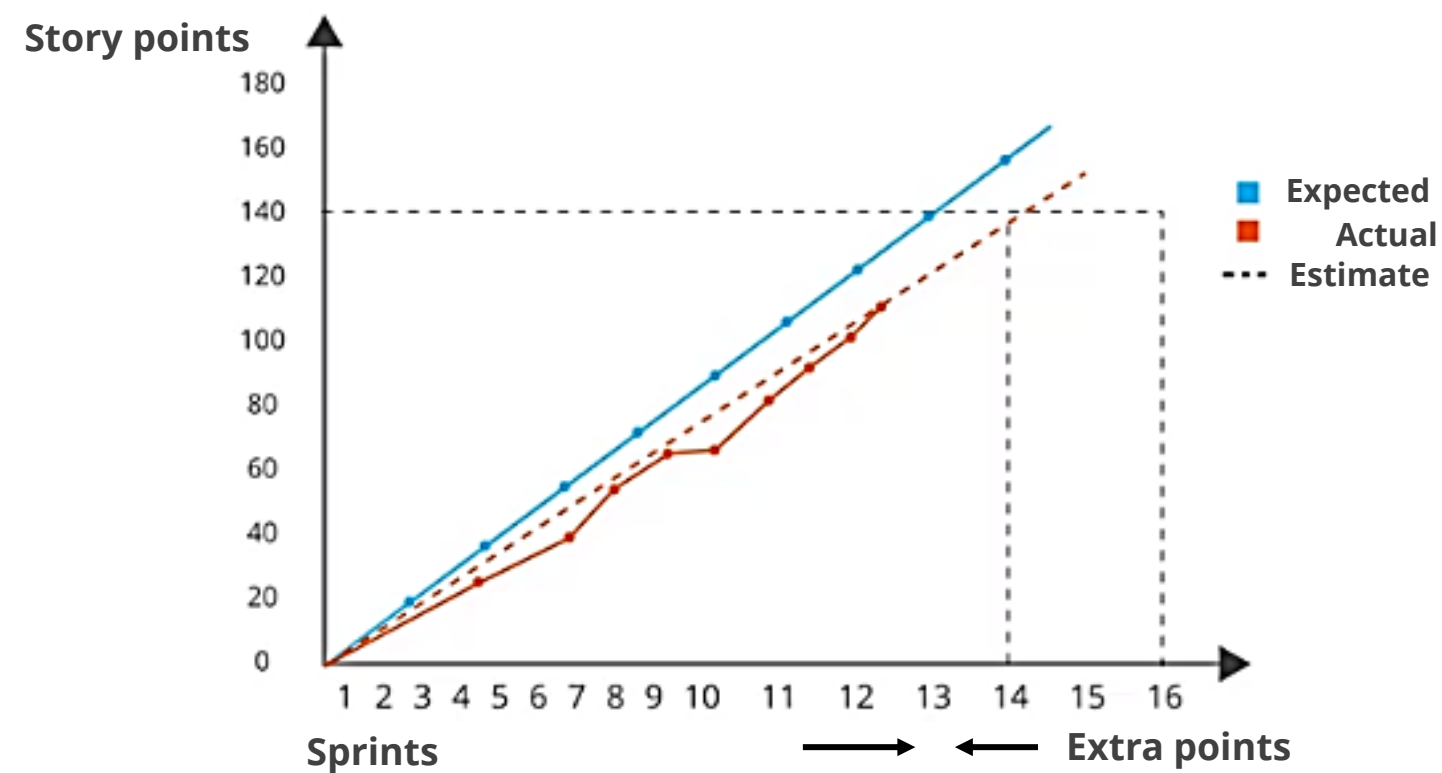
Burndown Chart Interpretation



Scenario: The grey line representing the actual work line goes down but then gets stuck and levels off.

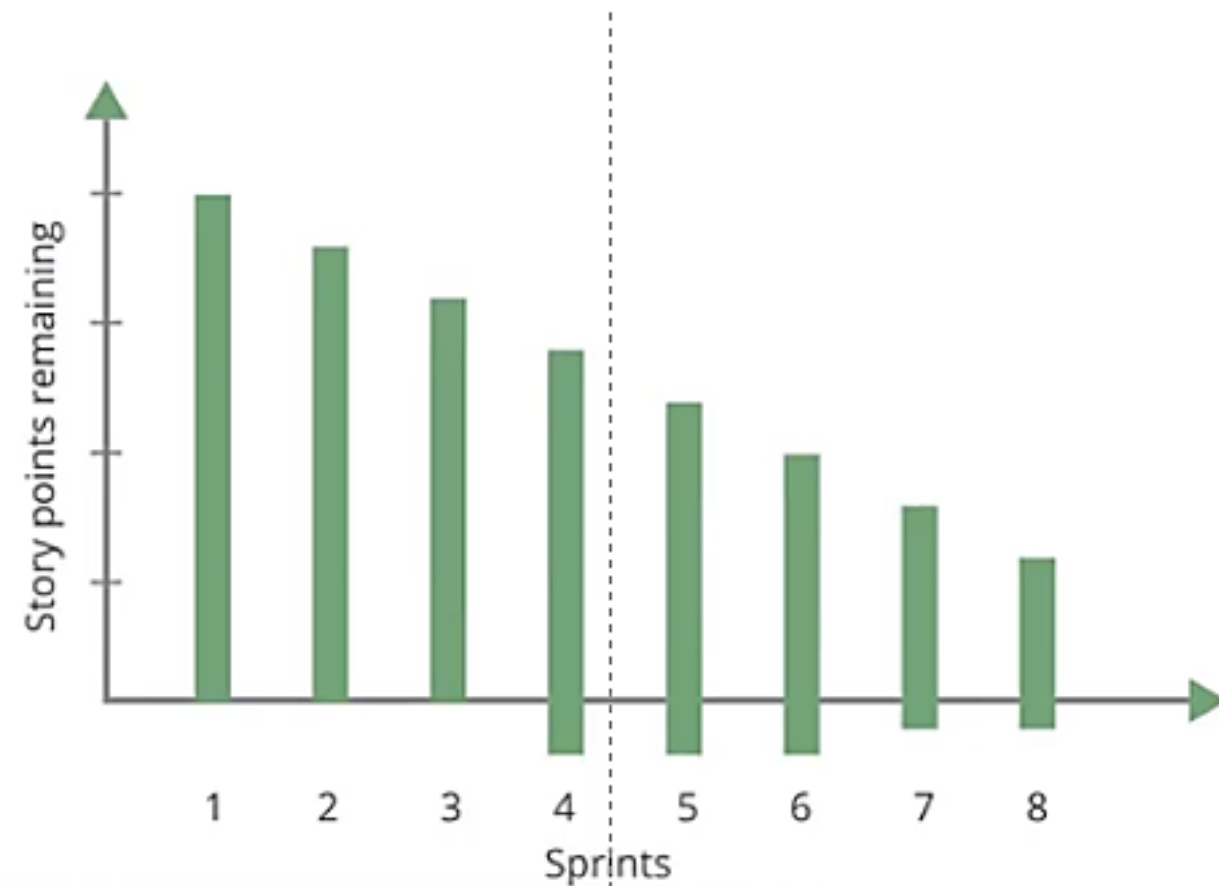
Interpretation: There is a blocking issue slowing the team down.

Burnup Charts



- A burnup chart tracks the work completed against a timeline.
- The blue line is the expected (steady) completion rate.
- The red line is the actual completion rate.
- The red dotted line indicates where the users are likely to end up at the current rate.
- The extra points refer to additional story points that have been added to the scope of the project after the initial estimation.

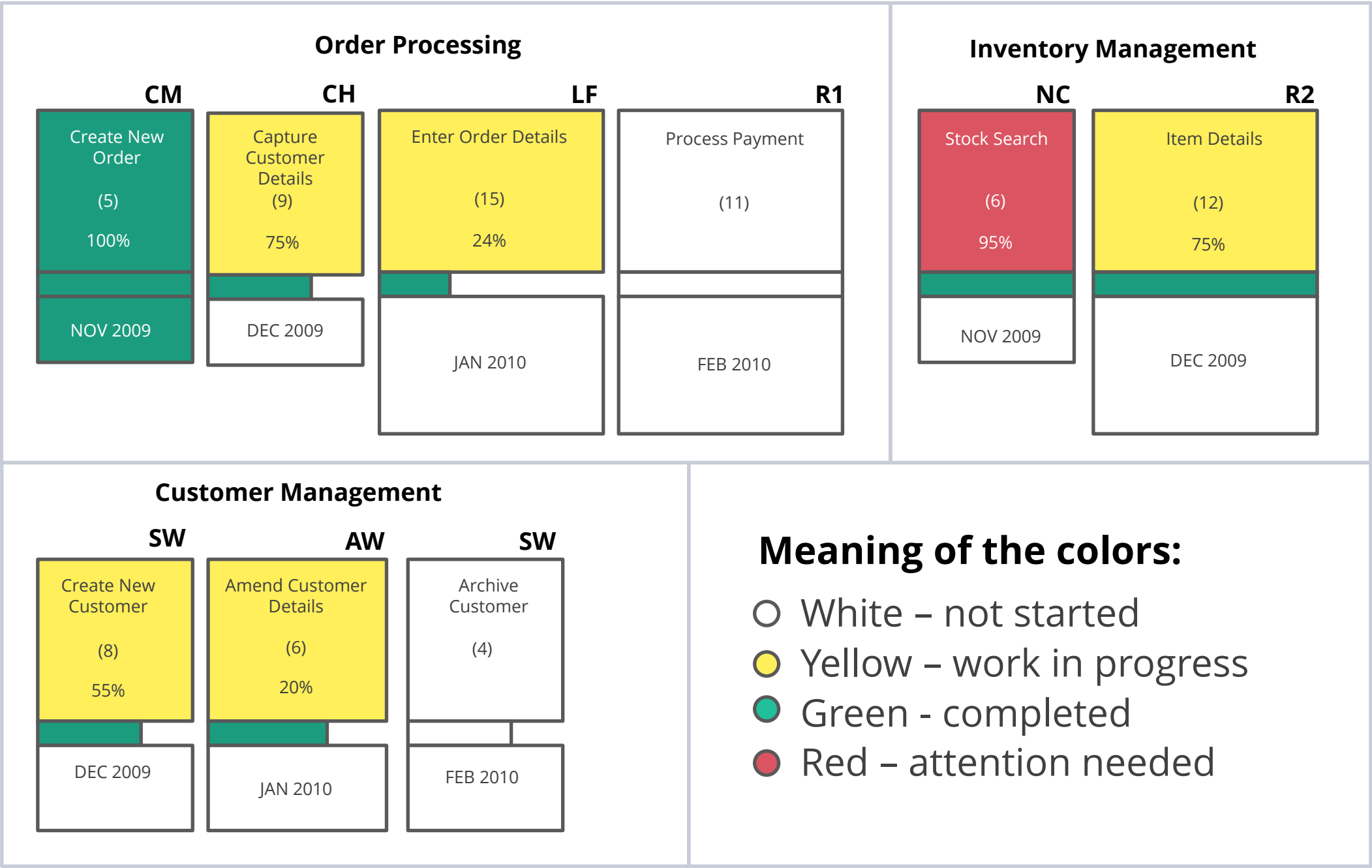
Burndown Bar Charts



- It helps visualize both the progress of work and any changes in scope during a release or project.
- The length of the bars indicates the overall work remaining in the project.
- After each sprint, points of work completed are taken out from the top of the bar.
- If the work is added to the project, the bottom of the bar extends down; if work is removed from the project, the bottom of the bar moves up.

Parking Lot Charts

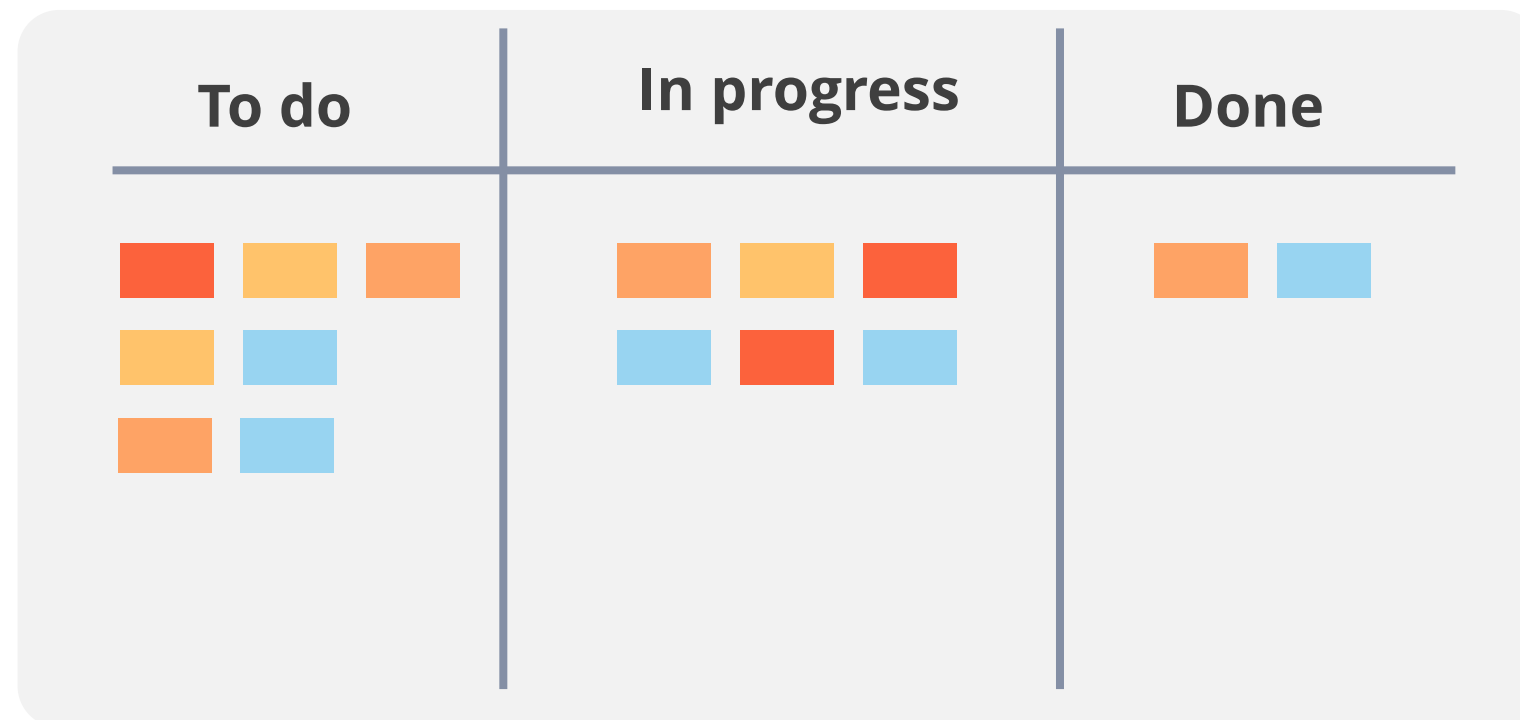
These visual tools track and highlight the status of key features or tasks, showing what is completed, in progress, or pending. They provide a clear overview of project progress.



This image represents an order progress tracking chart.

Scrum Board


























It is a visual tool that tracks the progress of tasks in a sprint, organizing them into columns like to do, in progress, and done.



- Each card represents an item of work (story, task, or ticket).
- As work moves from one step to another, the card is moved from left to right.
- Columns represent the steps in the development process.
- The users can have as many columns as steps in the value stream.
- The Scrum board indicates how much work is stuck at each step.

Niko-Niko Calendar

It is a visual tool used primarily in Agile project management to track team members' moods or emotions over time.

Users	Day 1	Day 2	Day 3	Day 4
User 1				
User 2				
User 3				
User 4				
User 5				
User 6				
User 7				



Agile Controlling

What Is Agile Controlling?

It is the process of monitoring and managing an Agile project's progress and performance flexibly and adaptively.



It focuses on real-time tracking, feedback, and adjustments.

Manage Issues and Bugs

In an Agile Scrum environment, managing issues and bugs is crucial for maintaining the quality and performance of the project.

Steps to manage issues and bugs effectively include:



Identifying and Logging Issues and Bugs

The process of identifying and logging issues and bugs involves:

Documentation

Log each issue or bug in a centralized tracking system.

Detection

Issues or bugs may be identified by team members, automated tests, or user feedback.

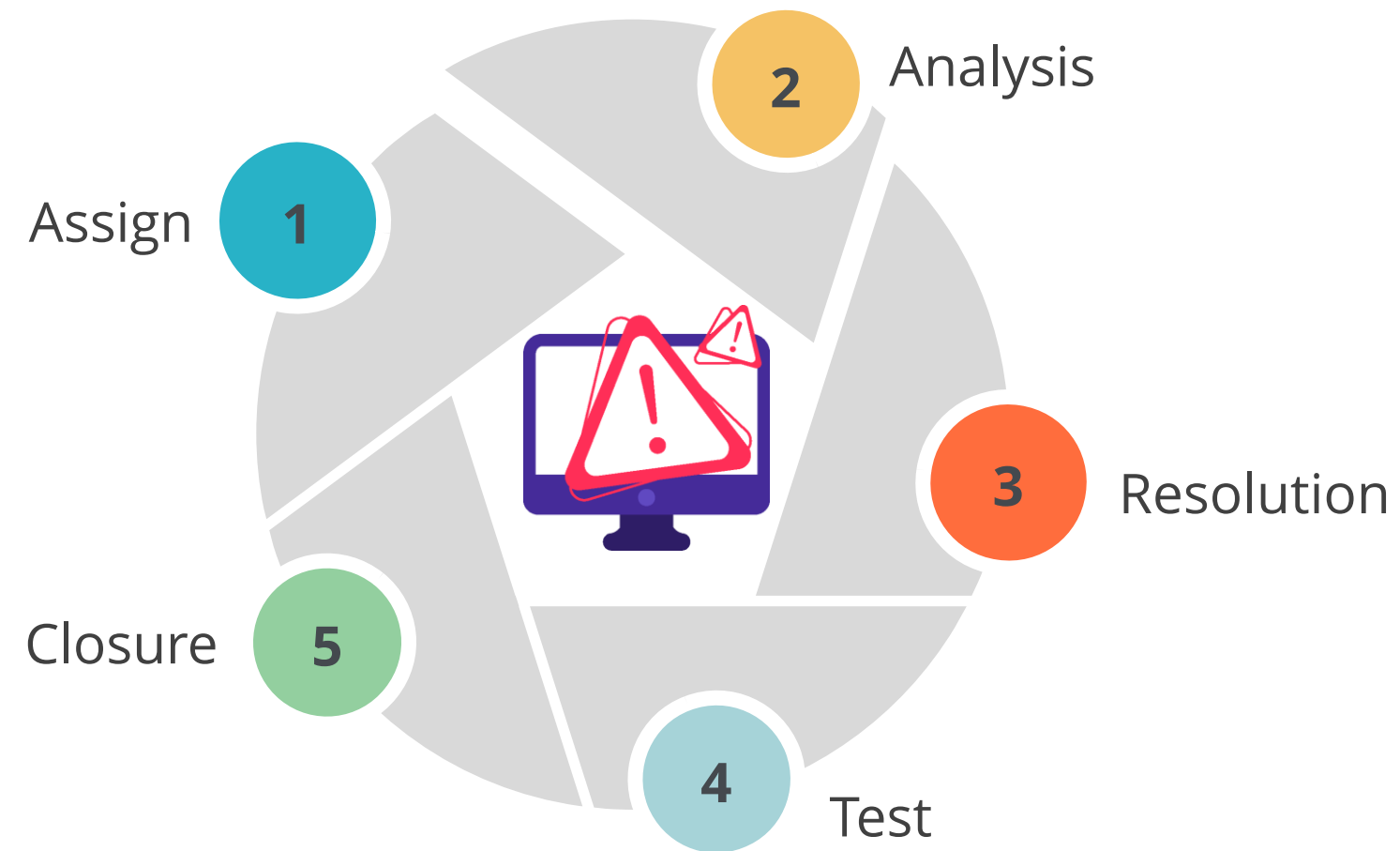


Prioritization

Use the Agile triage process to prioritize issues based on their impact on the project.

Handling and Resolving Issues and Bugs

The process of handling and resolving issues and bugs involves:



Informing Stakeholders

The process of informing stakeholders about issues and bugs includes:

Maintaining transparency
Inform stakeholders about the status of issues and bugs

Developing a communication plan
Include regular updates

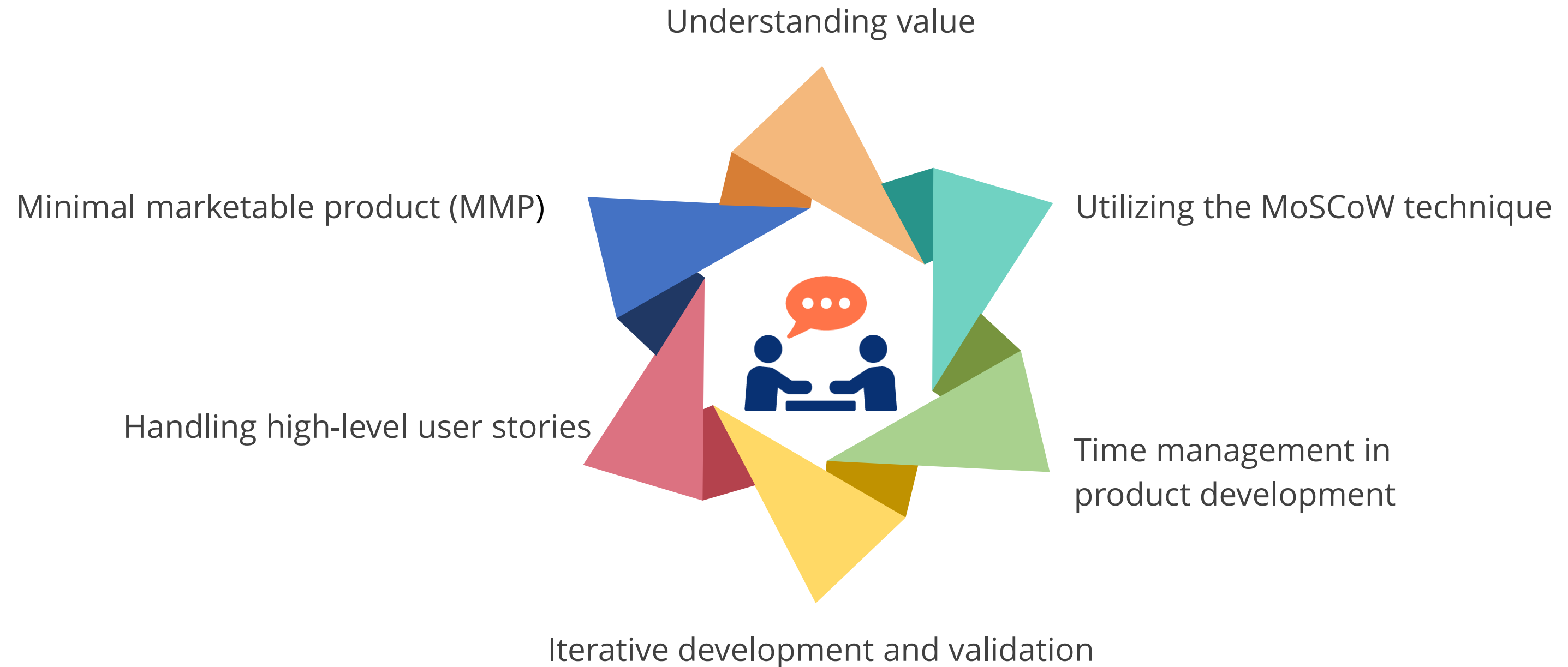
Creating a feedback loop
Provide input on the prioritization and handling of issues to stakeholders

Creating a concise report
Inform stakeholders about identified issues, their severity, fix status, and potential impacts on project timelines



Communication with Stakeholders

Adopting effective communication strategies with stakeholders is crucial for ensuring successful project outcomes. These strategies include:



Continuous Improvement Backlog

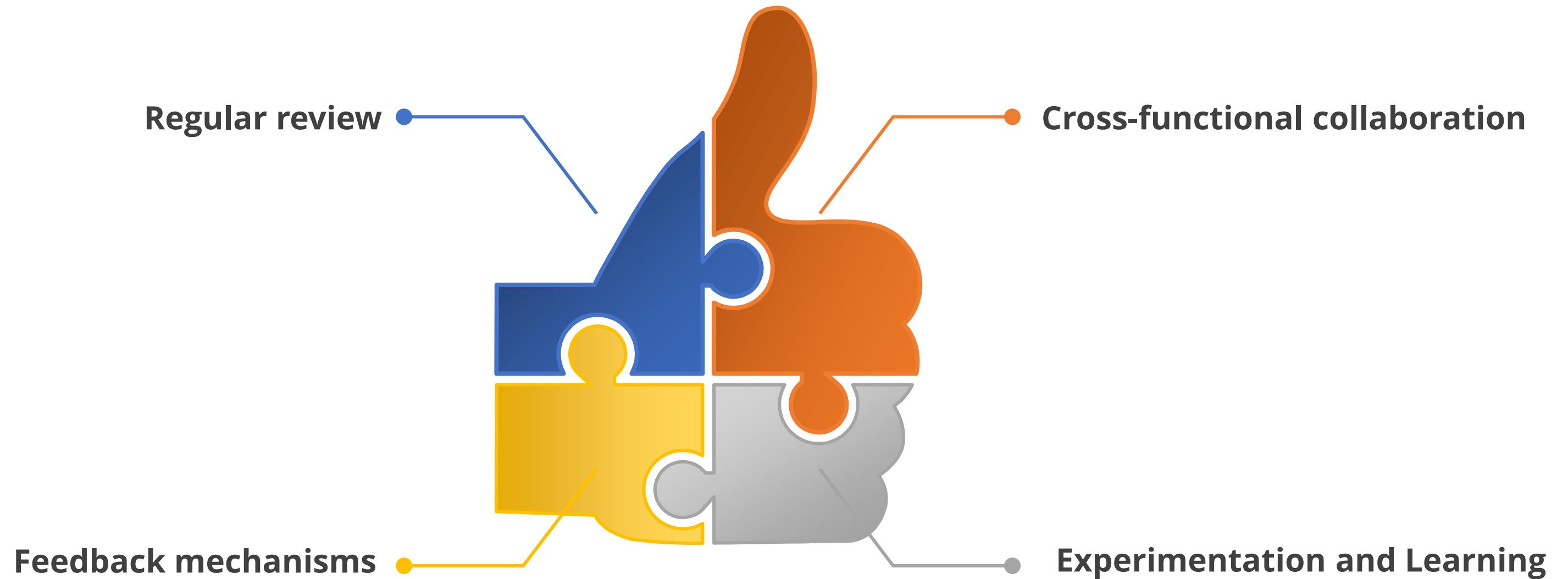
The result and outcome of the stakeholder meeting will suggest what can be implemented with the help of the continuous improvement backlog.



It is a prioritized list of enhancements and optimizations to refine processes and practices over time.

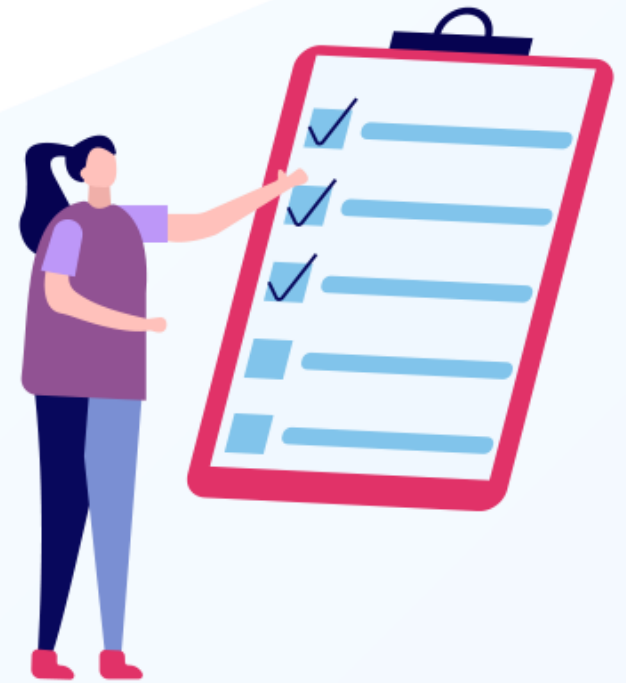
Continuous Improvement Backlog: Best Practices

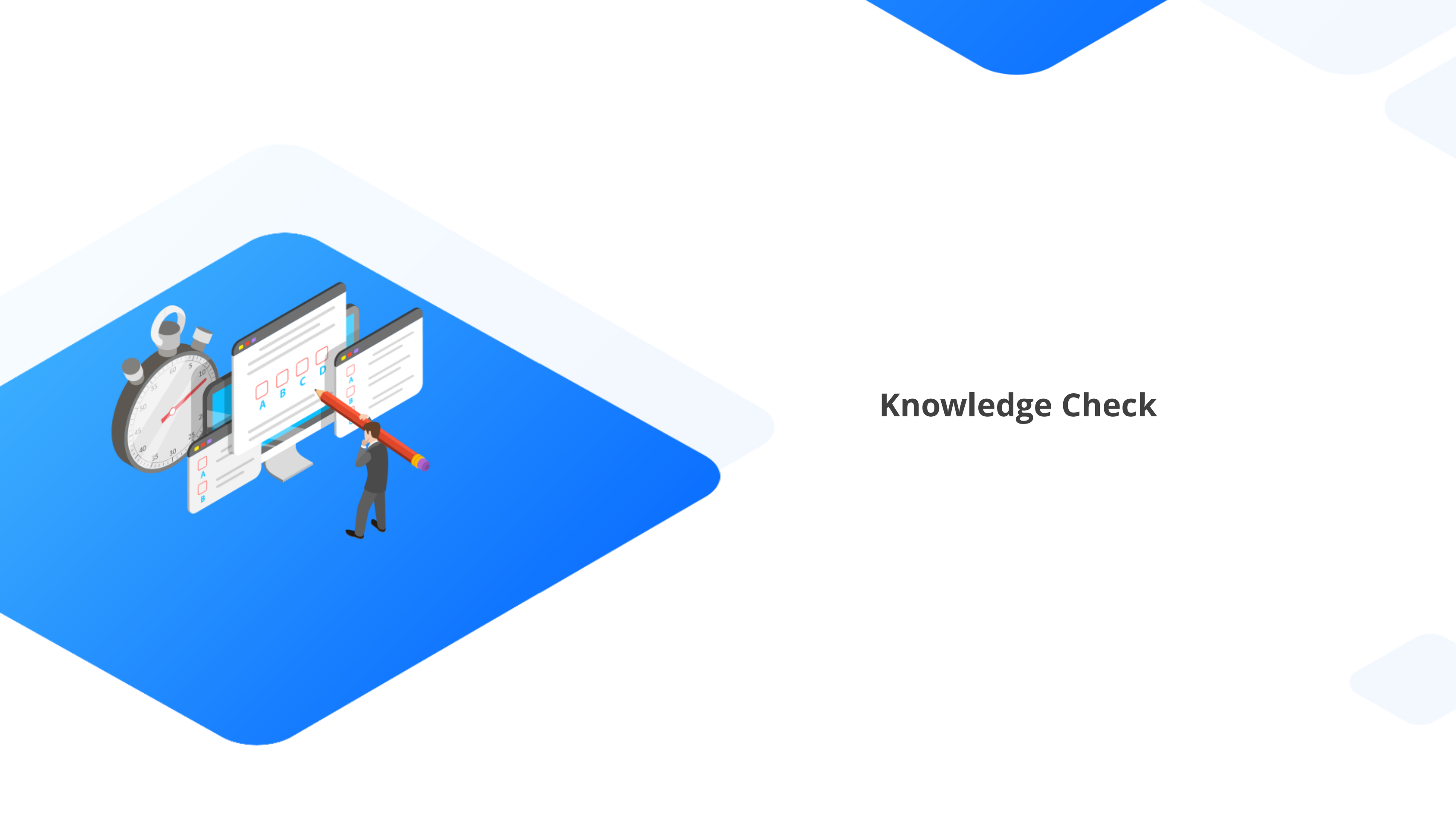
Best practices for managing the continuous improvement backlog are:



Key Takeaways

- Agile estimating is a technique used in Agile project management to forecast the effort required to complete tasks, features, or user stories within a project.
- Story points are a relative unit of measure used to estimate the effort required to implement a product backlog item or any other piece of work.
- Agile planning is a dynamic and iterative approach to project management that focuses on delivering value to customers through frequent and incremental updates.
- Agile monitoring is the ongoing process of tracking and assessing the progress and performance of an Agile team.
- Agile controlling is the process of monitoring and managing an Agile project's progress and performance flexibly and adaptively.





Knowledge Check

Knowledge Check

1

Which of the following techniques is commonly used in Agile estimation to evaluate the effort required for a user story?

- A. Waterfall analysis
- B. Poker planning
- C. Critical path method
- D. Gantt chart



Knowledge Check

1

Which of the following techniques is commonly used in Agile estimation to evaluate the effort required for a user story?

- A. Waterfall analysis
- B. Poker planning
- C. Critical path method
- D. Gantt chart



The correct answer is **B**

Poker planning is the technique commonly used in Agile estimation to evaluate the effort required for a user story.

Knowledge Check

2

You are planning the work for the next sprint and need to ensure it aligns with the upcoming product release. What type of planning are you engaging in?

- A. Portfolio planning
- B. Release planning
- C. Daily planning
- D. Tactical planning



Knowledge Check

2

You are planning the work for the next sprint and need to ensure it aligns with the upcoming product release. What type of planning are you engaging in?

- A. Portfolio planning
- B. Release planning
- C. Daily planning
- D. Tactical planning



The correct answer are **B**

Release planning is the type of planning that aligns with the upcoming product release.

**Knowledge
Check**
3

A team's average velocity is 10, with a maximum of 12 and a minimum of 8. The release backlog is sized at 72. How many sprints should the team commit to?

- A. 7
- B. 9
- C. 8
- D. 6



**Knowledge
Check**

3

A team's average velocity is 10, with a maximum of 12 and a minimum of 8. The release backlog is sized at 72. How many sprints should the team commit to?

- A. 7
- B. 9
- C. 8
- D. 6

The correct answer are **C**

Considering the average velocity, the number of sprints needed is $72/10 = 7.2$. If we round it up to the higher whole number, eight sprints would be needed to complete the release.



Knowledge Check

4

What is the primary purpose of a burnup chart in Agile project management?

- A. To identify the critical path of the project
- B. To track the progress of completed work against the total scope
- C. To manage project risks and issues
- D. To outline the sequence of project tasks



Knowledge Check

4

What is the primary purpose of a burnup chart in Agile project management?

- A. To identify the critical path of the project
- B. To track the progress of completed work against the total scope
- C. To manage project risks and issues
- D. To outline the sequence of project tasks



The correct answer are **B**

The primary purpose of a burnup chart in Agile project management is to track the progress of completed work against the total scope.