

# User Stories

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA

A.A. 2024-2025

# Introduction to Agile Hierarchy

## Agile Hierarchy Overview:

- Agile work items are organized into a hierarchy to manage features, functionality, and tasks efficiently.
- The hierarchy typically starts with broad, high-level concepts (Themes), and becomes more detailed (Tasks) as you break them down.

## Levels:

- **Themes → Epics → User Stories → Tasks**

## Purpose:

- Helps teams structure large bodies of work into manageable pieces that can be delivered incrementally.

# Breaking Down Large User Stories

<https://geekbot.com/blog/epics-stories-themes-tasks/>



# What is a Theme?

## Definition:

- A Theme is a broad, high-level objective that encompasses several Epics and User Stories.
- Themes group related functionality or goals that align with a larger business need or product vision.

## Purpose:

- Themes help align the work of multiple teams toward achieving large-scale business objectives.

## Example:

- *Theme: Improve the user experience of the e-commerce platform.*

# What is an Epic?

## Definition:

- An Epic is a large body of work that can be broken down into smaller, more manageable user stories.
- It represents a significant feature or functionality but is too large to complete within a single Sprint.

## Purpose:

- Epics allow teams to work on large features incrementally by breaking them into smaller parts (User Stories).

## Example:

- *Epic: Implement a new checkout process.*

# What is a User Story?

## Definition:

- A User Story is a small, specific piece of functionality that delivers value to the end user.
- It follows the template: "As a [user], I want to [perform an action], so that [benefit]."

## Purpose:

- User Stories break down Epics into smaller, actionable work items that can be completed in a single Sprint.

## Example:

- *As a user, I want to save my payment information, so that I can check out faster.*

# What is a Task?

## Definition:

- A Task is a specific piece of work needed to implement a User Story.
- Tasks are technical in nature and describe what needs to be done to complete the story.

## Purpose:

- Tasks allow the Development Team to break down the work required to complete a User Story into small, actionable steps.

## Example:

- *Task: Write the API to store payment information securely.*
- *Task: Implement front-end form validation for payment details.*

# Relationship Between Themes, Epics, User Stories, and Tasks

## Hierarchy:

- **Themes** group related Epics around broad objectives.
- **Epics** are large features or functionalities that need to be broken down into smaller parts.
- **User Stories** represent individual pieces of functionality, focused on user needs.
- **Tasks** are the actionable steps required to complete each User Story.

## Example:

- *Theme: Improve customer experience.*
- *Epic: Redesign the account management section.*
- *User Story: As a user, I want to update my email address, so that I can receive notifications.*
- *Task: Create the front-end form for updating email addresses.*



# Example of a Theme and its Breakdown

## Theme: Enhance Mobile Experience

- **Epic:** Create a mobile-friendly version of the checkout process.
- **User Story 1:** As a user, I want to use mobile payment options, so that I can check out easily on my phone.
- **User Story 2:** As a user, I want the product images to resize on mobile, so that I can see details clearly.
- **Task 1:** Implement the Apple Pay and Google Pay APIs.
- **Task 2:** Write CSS for responsive product images.

# Benefits of Using Themes, Epics, User Stories, and Tasks

## 1. Focus on Value:

- This hierarchy ensures that all work items are focused on delivering value to the user.

## 2. Manageable Work:

- Breaking down work into Themes, Epics, User Stories, and Tasks ensures that large objectives can be delivered incrementally.

# Benefits of Using Themes, Epics, User Stories, and Tasks

## 3. Clear Organization:

- The hierarchy provides a structured approach to organizing work, making it easier to track progress.

## 4. Flexibility:

- Teams can reprioritize or adapt User Stories and Tasks as new information becomes available, without affecting the broader Theme or Epic.

# Best Practices for Managing Themes, Epics, User Stories, and Tasks

## 1. Prioritize Based on Value:

- Focus on Epics and User Stories that deliver the most value to the customer.

## 2. Refine Epics and Stories Regularly:

- Continuously refine and break down Epics and Stories into smaller, more actionable items as the project evolves.

# Best Practices for Managing Themes, Epics, User Stories, and Tasks

## 3. Keep Stories Small and Actionable:

- User Stories should be small enough to be completed within a single Sprint.

## 4. Ensure Tasks are Clear:

- Make sure that each Task has a clear definition and can be easily understood by the team.

# What are User Stories?

## Definition:

- A User Story is a short, simple description of a feature told from the perspective of the user or customer.
- It describes what the user needs and why, written in non-technical language.

## Goal:

- The purpose is to capture user requirements without detailing the technical implementation.

## Structure:

- User stories are part of the Product Backlog in Agile, helping the team focus on the value delivered to the user.

# The Structure of a User Story

## Template:

*As a [type of user], I want to [perform an action], so that [achieve a goal].*

## Elements:

- **User Role:** The type of user interacting with the feature.
- **Action:** What the user wants to do.
- **Goal/Benefit:** The reason why the user wants to perform the action.

## Example:

- *As a customer, I want to receive a confirmation email after placing an order, so that I know my order was successful.*

# Who Writes User Stories?

## Product Owner:

- The Product Owner is typically responsible for creating and prioritizing user stories.
- They ensure the stories align with the overall product vision and customer needs.

## Development Team:

- The team collaborates with the Product Owner to refine and estimate the user stories.
- They ensure that the stories are actionable and technically feasible.

## Users/Stakeholders:

- Users and stakeholders may contribute by providing feedback or specific needs that the Product Owner turns into user stories.



# Example of a Good User Story

## Example:

*As a frequent traveler, I want to save my payment information, so that I can check out faster when booking flights.*

## Why It's Good:

- Clearly identifies the user (*frequent traveler*).
- Specifies the action (*save payment information*).
- Provides a benefit (*check out faster when booking flights*).

## Focus on Value:

- This story focuses on delivering value to the user by simplifying the checkout process.

# INVEST Criteria for Good User Stories

## INVEST Acronym:

- **I**ndependent: The story should be self-contained.
- **N**egotiable: The details can be discussed and modified.
- **V**aluable: It should deliver clear value to the user.
- **E**stimable: The team should be able to estimate the effort required.
- **S**mall: It should be small enough to be completed in a single iteration.
- **T**estable: The story should have clear acceptance criteria to validate completion.

## Example:

- "As a user, I want to reset my password, so that I can regain access to my account."
- Testable criteria: "The system sends a password reset email when a user requests it."

# Acceptance Criteria

## What are Acceptance Criteria?

- These are specific conditions that must be met for the user story to be considered complete.
- Acceptance criteria define the scope and boundaries of the story, ensuring that it meets the user's expectations.

### Example:

*"As a user, I want to be able to search for products, so that I can find items quickly."*

### Acceptance Criteria:

- The search function should display relevant products based on the search term.
- The search results should appear within 2 seconds.
- The user should be able to filter search results by category and price.

# Breaking Down Large User Stories

## Why Break Down Large Stories?

- Some user stories, known as "epics," are too large to be completed in a single iteration.
- Breaking down epics into smaller stories ensures that each piece of work can be completed within a Sprint.

## Example:

- Epic: "As a user, I want to manage my profile, so that I can update my personal information."
- Smaller Stories:
  - ▶ "As a user, I want to update my email address, so that I can receive notifications."
  - ▶ "As a user, I want to update my profile picture, so that I can personalize my account."

# Common Mistakes to Avoid

## 1. Epics that are too vague:

- Avoid writing Epics that lack enough detail to break down into Stories.

## 2. Stories that are too large:

- Ensure User Stories are small and actionable, rather than trying to accomplish too much in a single Story.

# Common Mistakes to Avoid

## 3. Tasks that lack clarity:

- Every Task should be clearly defined with specific actions, so the team knows exactly what needs to be done.

## 4. Losing sight of the Theme:

- Always ensure that the work aligns with the broader Theme, keeping the end goal in mind.

# Example of a User Story Lifecycle

## Lifecycle Stages:

- **Drafting:** The Product Owner writes an initial user story based on stakeholder input.
- **Refinement:** The Development Team collaborates to refine the story, clarify requirements, and add acceptance criteria.
- **Development:** The story is selected for the Sprint and the team works on implementing the feature.
- **Review:** The feature is demonstrated during the Sprint Review, and stakeholders provide feedback.
- **Completion:** The story is marked as done once it passes the acceptance criteria and feedback is incorporated.

# Common Mistakes in Writing User Stories

## Too Vague or Broad:

- "As a user, I want a better dashboard."
- Solution: Focus on specific improvements (e.g., "As a user, I want to see my account balance on the dashboard, so that I can quickly check my finances.").

## Technical Language:

- Avoid writing stories in technical jargon (e.g., "As a user, I want an AJAX request to update the page content asynchronously.").
- Solution: Keep it user-focused (e.g., "As a user, I want the page content to update without reloading, so that I have a smooth experience.").



# Best Practices for Writing User Stories

## Keep it User-Focused:

- Write from the perspective of the user or customer, not the system or developer.

## Collaborate:

- Involve the team in refining and breaking down stories to ensure they are actionable.

# Best Practices for Writing User Stories

## Prioritize Value:

- Focus on the stories that provide the most value to the user or customer.

## Clear Acceptance Criteria:

- Ensure every user story has well-defined acceptance criteria to confirm when the story is complete.

# User Stories

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA

A.A. 2024-2025