

Introduction to Agile Methodologies

**Presented by: uchechi
chibuzor**

Learning Objectives

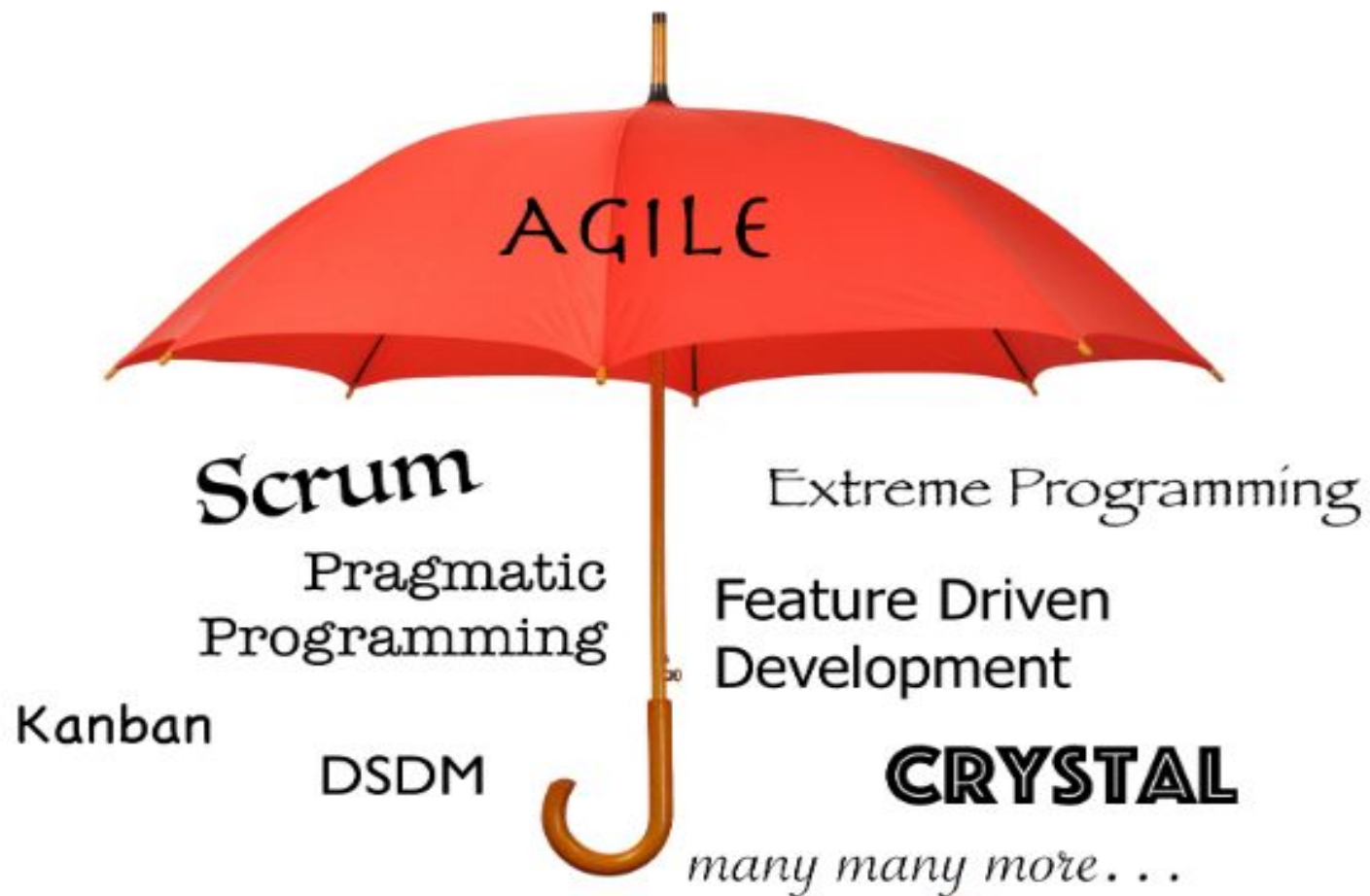
- Understand what agile methodology is about
- Learn the key values and principles of Agile
- Differentiate Scrum from other Agile frameworks
- key roles in a Scrum team
- Recognize how Scrum supports continuous delivery, team collaboration, and customer feedback
- Apply Scrum concepts to a simple product-building scenario

Introduction to Agile

Agile methodologies are a group of software development approaches based on iterative development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Agile promotes adaptive planning, evolutionary development, early delivery, and continual improvement, all while encouraging flexible responses to change.

Agile Manifesto - Four Core Values:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan



AGILE

Scrum

Pragmatic
Programming

Kanban

DSDM

Extreme Programming

Feature Driven
Development

CRYSTAL

many many more...

Frameworks in Agile Methodology

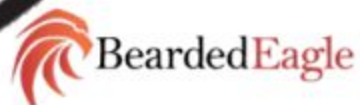
- Scrum
- Kanban
- Lean/Agile software development
- Scaled Agile Framework (SAFe)
- ETC.

Although there are several frameworks in agile, our focus would be on

scrum

Scrum Defined

A lightweight framework
that helps people, teams and
organizations generate value
through adaptive solutions for
complex problems.



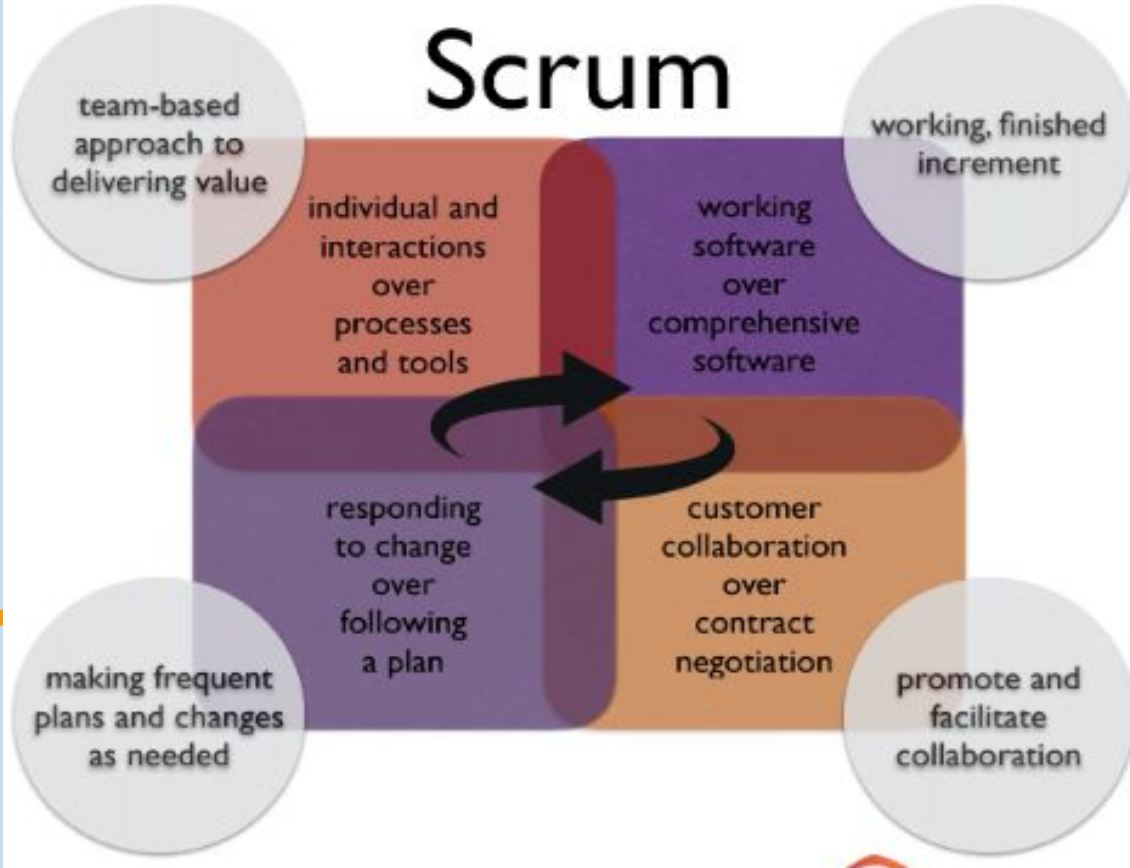
Introduction to Scrum

What is Scrum?

Scrum is an agile framework that software development teams use to work in a collaborative way. These are the attributes of scrum:

- Fast feedback
- Quicker innovation
- Continuous improvement
- Rapid adaptation to change
- More delighted customers
- Accelerated pace from idea to delivery

Scrum



SCRUM PROCESS

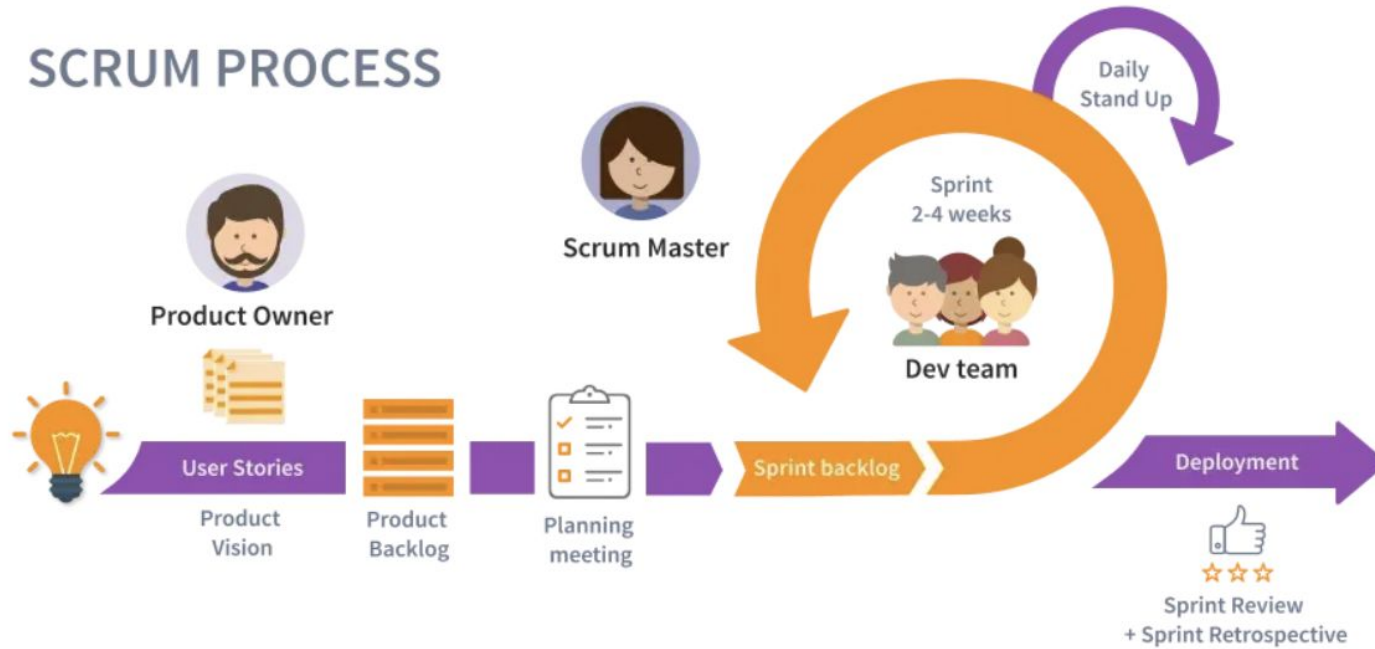


Image: Scrum Process

Source: Internet

Why Scrum Stands Out?

- Structured yet flexible (fixed sprints but adaptable scope)
- Continuous improvement (retrospectives refine processes)
- Clear accountability (defined roles prevent chaos)

What is a Sprint?

Sprint is at the center of Scrum. Another way to put it is that Sprint is the heart and soul of Scrum.

A sprint is a timeboxed event or period of development. Each sprint (an increment) should bring the product closer to the product goal.

Sprint Planning

Sprint planning is a scrum event that's carried out before a sprint starts. This is where the team and the PO determines what'd go into the sprint.

Things that happens in sprint planning:

What

How

Who

Input and

Output

Sprint Planning Meeting

Backlog



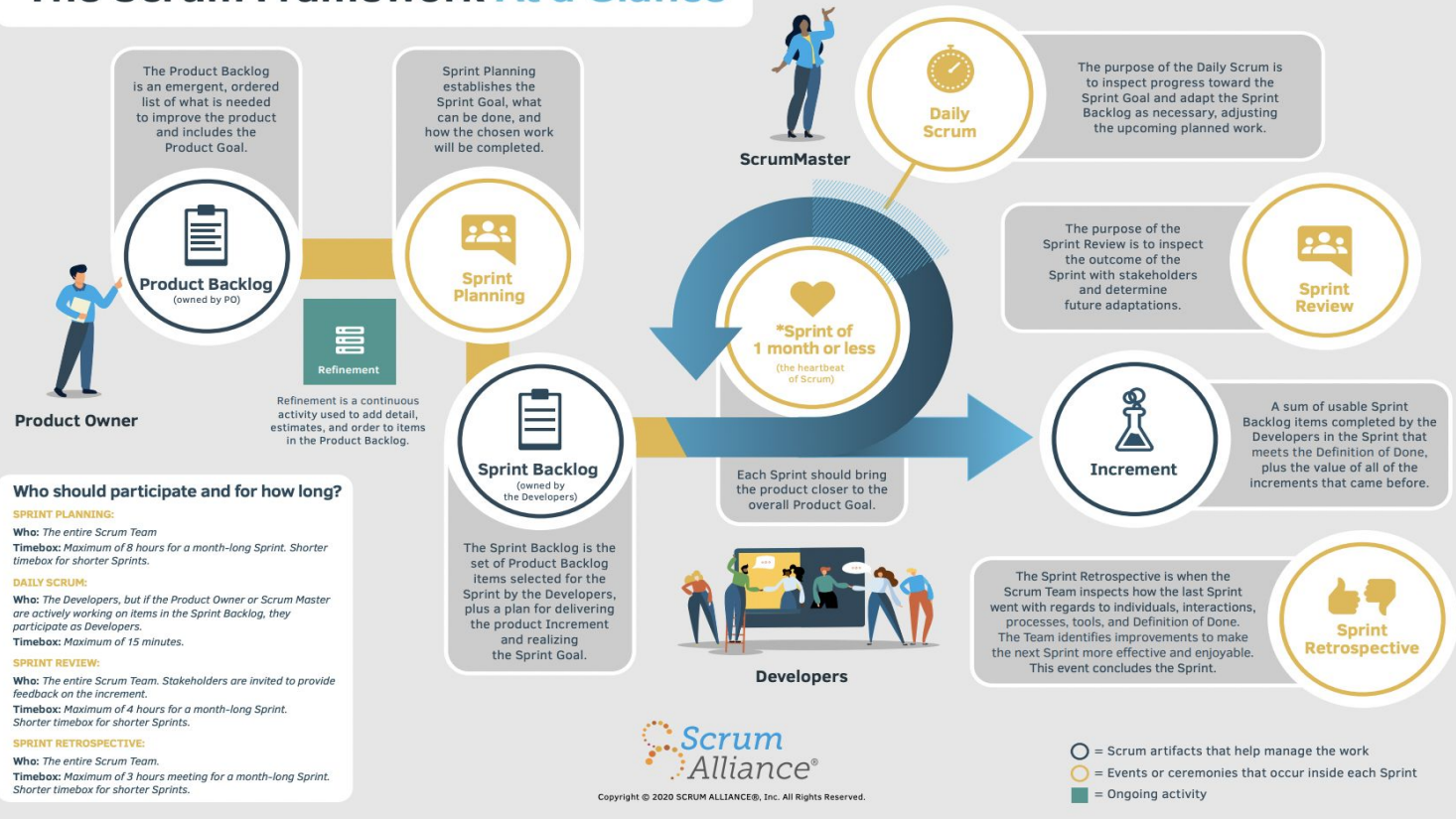
Sprint Planning Meeting



Sprint Backlog



The Scrum Framework At a Glance



Scrum Ceremonies

- Sprint
- Sprint Planning
- Daily Scrum – Stand Up
- Sprint Review
- Sprint Retrospective

Scrum Roles

- Developers – engineers that are to work on the features or items of the product
- Product Owner – the driver in charge of what needs to be done in line with the product vision or goal
- Scrum Master – helps the team makes the best use of scrum to achieve the product goal

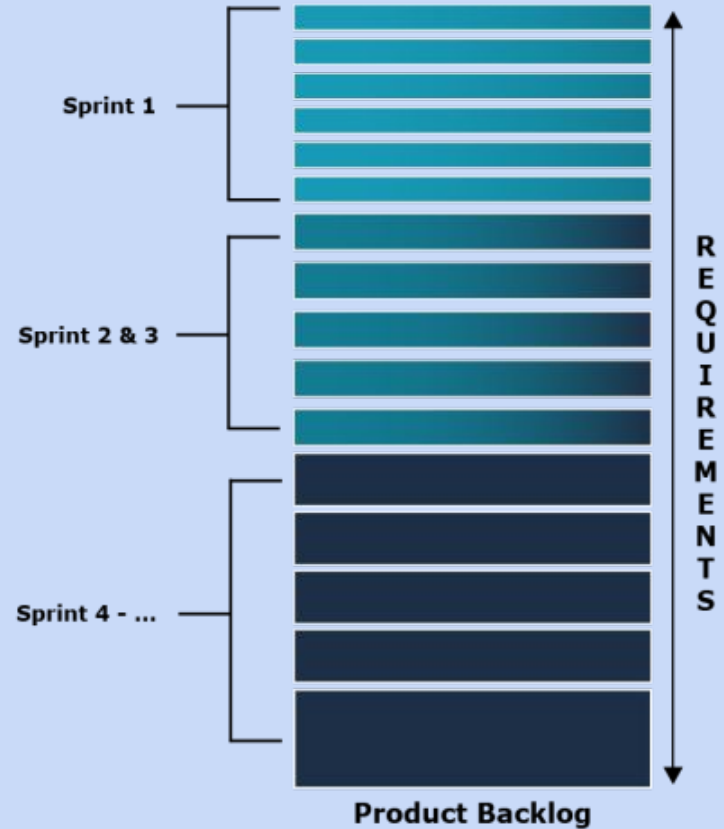
Scrum Artifacts

- Product Backlog – an ordered list of items or features for a product
- Sprint Backlog – an ordered list of items or features that made it to a sprint (was selected for the sprint) that'd be delivered as increments
- Increments – what was just built in the last sprint, plus what was previously built and how they are working together and are ready to be delivered

Product Backlog?

- A product backlog is a prioritized list of work items or features that help you meet product goals and set expectations among teams.
- It is a list of tasks required to support a larger strategic plan. For example, the product development context contains a prioritized list of items. The product team agrees to work on these projects next. Typical items on a product backlog include user stories, changes to existing functionality, and bug fixes.

- The product backlog is managed by the product manager and is never considered complete. It is always open to adding, updating, or removing items as the product increases and business requirements or customer needs changes



'Product Backlog' vs 'Sprint Backlog'

The key differences between two common artifacts.



Anything that needed to accomplish the project vision



Product Owner owns the Product backlog



Contains requirements, defects, tasks.



Everyone contributes to the product backlog



Product backlog refinement meeting is to refine the product backlog



Anything that is needed to fulfil the sprint goal.



Development team owns the sprint backlog



Subset of product backlog items defined as priority by product owner.



Only development team contributes to the sprint backlog



Sprint Planning meeting is to refine the sprint backlog items

Product Backlog

VS

Sprint Backlog

Product Backlog

DEEP

Technique

- * Detailed
- * Estimated
- * Emergent
- * Prioritized

Userstory#1

Userstory#2

Userstory#3

Userstory#4

Userstory#5

Userstory#6

Sprint Backlog

Sprint #1

Userstory#1

Userstory#2

Userstory#3

Sprint #2

Userstory#4

Sprint #3

Sprint backlog

▼ Engagement and Data Export 7 Jun – 24 Jun (7 issues)

 SD-80 Implement monitoring systems for Squareroof's servers

 SD-81 Implement Squareroof's Admin roles and permissions

 SD-153 Data Export

 SD-154 Engagement Emails

 SD-103 Instalment Offering Button

 SD-113 Agent request feature

 SD-98 Update Feature on Admin

+ Create issue

Product backlog

Backlog



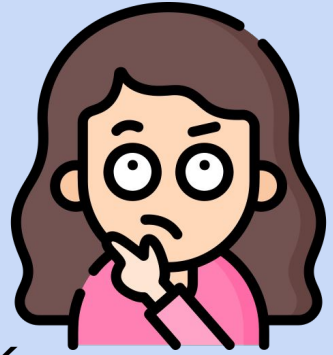
Epic ▾

Type ▾

Insights

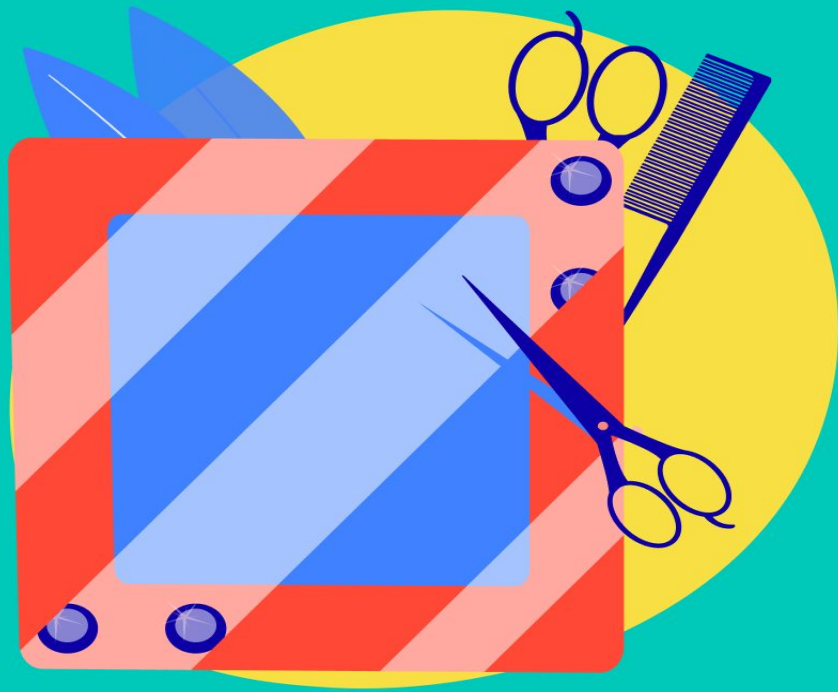
SD-77 Project listing - verification	TO DO ▾	AA
SD-53 Buy shares	TO DO ▾	
SD-54 Sell shares	TO DO ▾	
SD-55 Sales Order (Admin)	TO DO ▾	
SD-49 Auto Invest	TO DO ▾	
SD-48 Reinvest after maturity	TO DO ▾	
SD-52 Roof Agent Up REFERRAL	TO DO ▾	
SD-75 Roof Developer Mark Up	- TO DO ▾	...
SD-11 Returns Calculator PROJECTS	TO DO ▾	
SD-40 Transaction Notification NOTIFICATION	TO DO ▾	
SD-41 Market Rules PROJECTS	TO DO ▾	
SD-47 User transaction TRANSACTION (ADMIN)	TO DO ▾	

Does the backlog have a purpose?



1. Provide a single source of truth for the team's planned work
2. Facilitate team discussion
3. Make it easier to assign work

Backlog Grooming



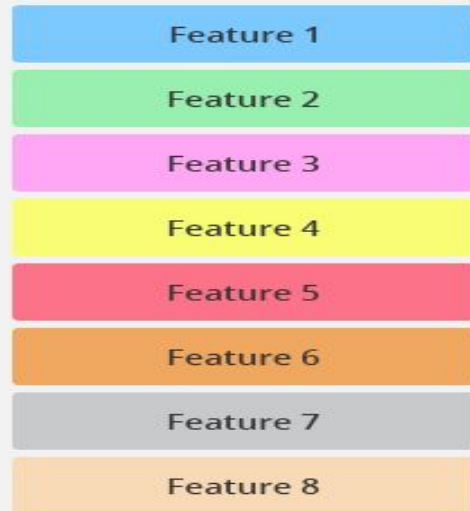
- Backlog grooming or refinement is a regular session where backlog items are discussed, reviewed, and prioritized by product managers, product owners, and the rest of the team.
- The primary goal of backlog grooming is to keep the backlog up-to-date and ensure that backlog items are prepared for upcoming sprints.
- Additionally, the process helps product managers explain and align the organization behind the strategy that informs the backlog items.

The most common tactical activities that occur during backlog grooming include:

- Removing outdated user stories and tasks.
- Adding new user stories that reflect newly discovered user insights.
- Breaking down broad user stories into smaller items.
- Reordering user stories based on their priority.
- Explaining and clearly defining user stories and tasks to avoid uncertainty and “black box” communication.
- Assigning or re-assigning story points and estimates.
- Identifying roadblocks and minimizing risks related to backlog items

PRODUCT BACKLOG REFINEMENT

PRODUCT BACKLOG



PRIORITISED PRODUCT BACKLOG



Product Backlog

Work Item

Work Item

Work Item

Work Item

Work Item

Work Item

Work Item

Work Item

Sprint Backlog

Work Item

Task

Task

Task

Task

Work Item

Task

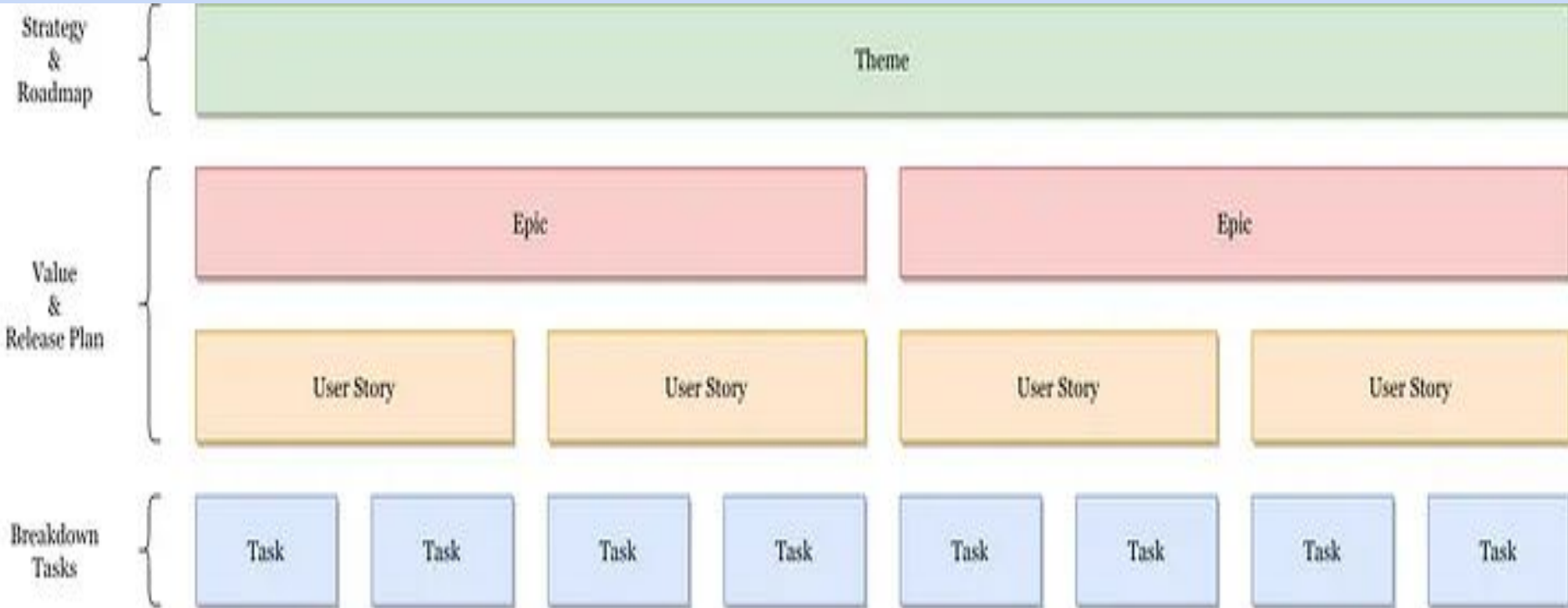
Task

Task

Why Backlog Grooming



Hierarchies in Agile



Epic?

An epic is a feature or functionality that consist of multiple building blocks and scenarios. Epics are derived from themes or initiatives and can be broken down or segmented into smaller pieces called user stories.

An epic can span across multiple sprints, teams, and even projects. The theme, epic, and user stories share the same strategic goal at different levels of the focus area.

Consider the example of building a house. If an initiative is like building the ground floor, an epic is like creating a kitchen and a user story is like building a wall, with each brick being a task.



Initiative



Theme



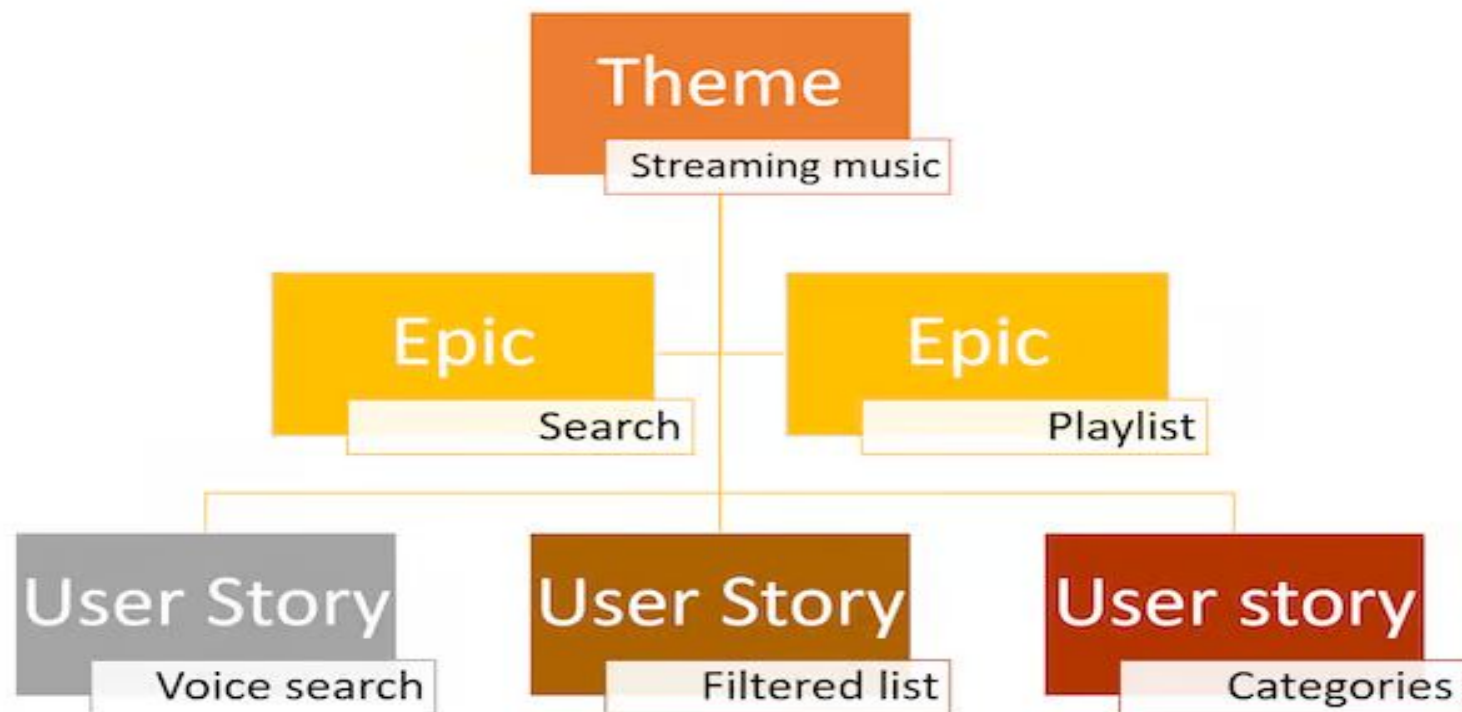
Epic



User story



Task



Initiative

```
graph TD; Initiative[Initiative] --> Epic1[Epic 1]; Initiative --> Epic2[Epic 2]; Initiative --> Epic3[Epic 3]; Epic1 --> ST1_1[Story/Task 1]; Epic1 --> ST1_2[Story/Task 2]; ST1_1 --> Subtask1_1[Subtask 1]; Epic2 --> ST2_1[Story/Task 1]; ST2_1 --> Subtask1_2[Subtask 1]; ST2_1 --> Subtask2_2[Subtask 2]; Epic3 --> ST3_1[Story/Task 1]; Epic3 --> ST3_2[Story/Task 2]; Epic3 --> ST3_3[Story/Task 3]; ST3_1 --> Subtask1_3[Subtask 1]; ST3_2 --> Subtask2_3[Subtask 2];
```

Epic 1

Story/Task 1

Subtask 1

Story/Task 2

Epic 2

Story/Task 1

Subtask 1

Subtask 2

Epic 3

Story/
Task 1

Subtask 1

Story/
Task 2

Subtask 2

Story/
Task 3

Feature?

A Feature is a service that fulfills a stakeholder need. Each feature includes a benefit hypothesis and acceptance criteria, and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a Program Increment (PI).

User Stories

User stories are a popular method of capturing requirements and describing features in software development projects. They are concise, non-technical descriptions of a feature or functionality written in a specific format. User stories help to create a shared understanding of what the software should do and why it should be built, and serve as a starting point for discussions among stakeholders, product owners, and development teams.

A typical user story consists of three parts:

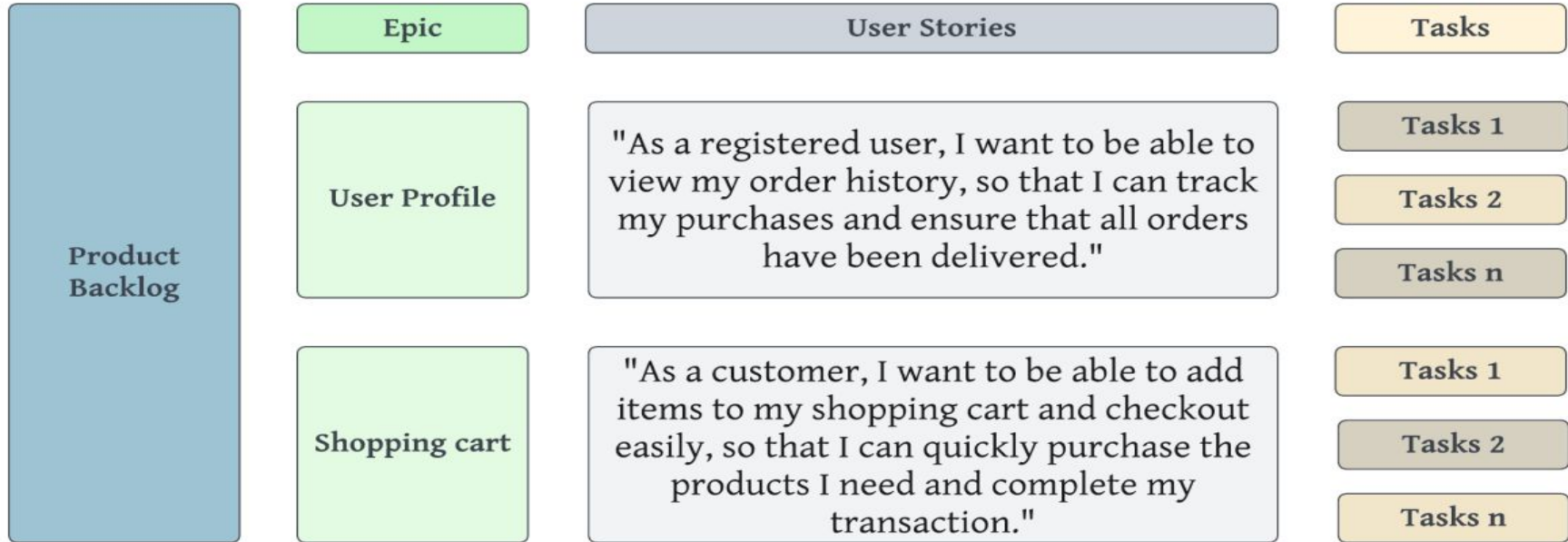
1. The user or customer who needs the feature or functionality.
2. The action or goal the user wants to achieve.
3. The benefit or reason the user wants to achieve the goal.

"As a [persona or user], I want to [action or goal], so that [benefit or reason]."

For example, a user story for an e-commerce website could be:

"As a customer, I want to be able to add items to my shopping cart and checkout easily, so that I can quickly purchase the products I need and complete my transaction."

User Stories



Wishlist

Theme

As a customer, I want to be able to have wishlists so that I can come back to buy products later

Epic

As a customer, I want to be able to save a product in my wishlist so that I can view it again later

As a customer, I want to be able to view my wishlist so that I can buy items from it

Stories

Put 'Add to wishlist' button on each product page

Create new db to store wishlist items

Create page to display user's wishlist

Add 'View wishlist' link to homepage

Tasks

ACCEPTANCE CRITERIA

Acceptance criteria (AC) are the conditions a software feature must meet to be accepted by a user, a customer, or other systems. They are unique for each user story and define the feature behavior from the end-user's perspective.

Well-written acceptance criteria help avoid unexpected results at the end of a development stage and ensure that all stakeholders and users are satisfied with what they get. In terms of:

Clarity: They should be straightforward and easy to understand for all team members, avoiding any confusion.

Conciseness: The criteria should communicate the necessary information without unnecessary detail.

Testability: Each criterion must be verifiable, allowing testers to clearly determine whether it has been met.

Result-oriented: The focus should be on delivering results that gratify the customer, emphasizing the end benefit or value.

User story: *As a user, I want to be able to register online, so that I can start shopping online.*

Acceptance criteria:

- ☐ User can only submit a form by filling in all required fields
- ☐ The email user provided must not be a free email
- ☐ Submission from same IP can only be made three times within 30 minutes
- ☐ User will receive a notification email after successfully registration

FOR <target customers> Executive assistants, Employees, Students, Healthcare professionals and anyone in a meeting in Nigeria

WHO <statement of need> needs an organized note taking tool to help execute seamless tasks and workflows

THE <product name> FOB Notes

IS A <product category> Cloud service note taking tool

THAT <key benefit, reason to buy> helps users organize thoughts, take notes, capture ideas and improve productivity

UNLIKE <competition, alternative> Otter.io

OUR PRODUCT <differentiating statement> provide digital space to store, manage, share notes with co-workers / teams, and integrate with other tools such as calendars and agile platforms



Adobe Stock | #51072375

