**User stories**

* A user story is an informal, general explanation of a software feature written from the perspective of the end user.
* Its purpose is to articulate how a software feature will provide value to the customer.
* It's tempting to think that user stories are, simply put, software system requirements. But they're not.
* A key component of agile software development is putting people first, and a user story puts end users at the centre of the conversation.
* These stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building, what they're building, and what value it creates.

**1.Agile project methodologies**

* Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration.
* Software teams that embrace agile project management methodologies increase their development speed,expand collaboration, and foster the ability to better respond to market trends.
* Stemming from Toyota's lean manufacturing concept of the 1940s, software development teams have embraced agile methodologies to reduce waste and increase transparency, while quickly addressing their customers' ever-changing needs

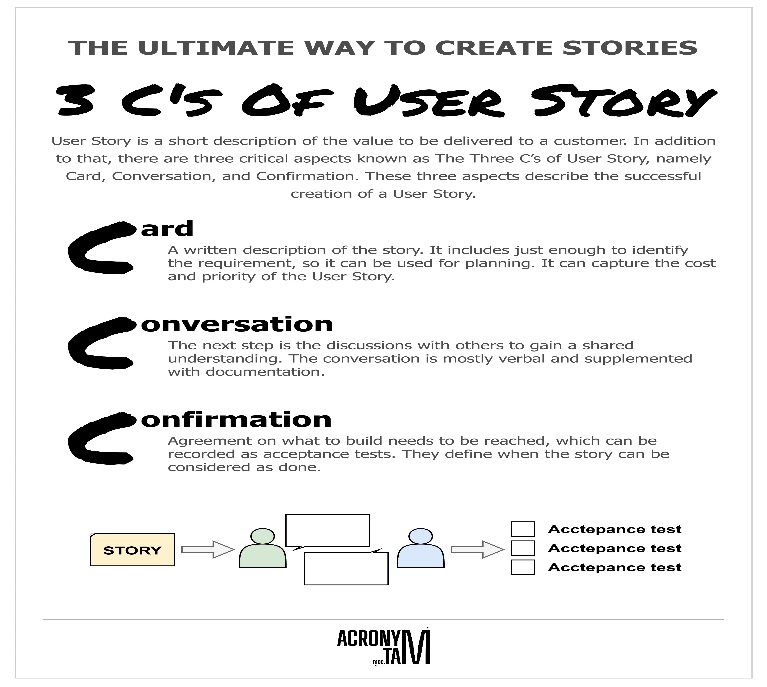
2.**Scrum** is an agile project management framework that helps teams structure and manage their work through a set of values, principles, and practices.

* Much like a rug by team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.
* While the scrum I’m talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork.
* This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.
* Scrum is a framework for agile project management that uses fixed-length iterations of work,called **sprints**. There are four ceremonies that bring structure to each sprint.

3.**Kanban** is a framework for agile project management that matches the work to the team's capacity. It's focused on getting things done as fast as possible, giving teams the ability to react to change even faster than scrum.

Unlike scrum, kanban has no backlogs (usually). Instead, work sits in the To Do column. This enables kanban teams to focus on continuous releases,which can be done at any time. All work is visible, scoped, and ready to execute on so that when something is completed, the team immediately moves on to the next. The amount of work is matched to the team's capacity through WIP limits, which is a predefined limit of work that can be in a single column at one time (except the To Do column).

3c’s of user story--



User story template—

* As a (who wants to accomplish something)
* I want to (what they want to accomplish)
* So that (why they want to accomplish that thing)

Example--

Bank loan management--

1. As a loan officer, I want to be able to view a customer’s up-to-date credit history, so I can make informed loan decisions.
2. As a borrower, I want to calculate my loan eligibility based on my income and credit score, so I know how much I can borrow.
3. As a loan processor, I want to be able to track the status of a loan application, so I can manage my work effectively.

Acceptance criteria—

Acceptance criteria are a set of predefined conditions that a product or feature must meet to be accepted by the customer, project stakeholders, or the product management team.

## Why are acceptance criteria important?

* **Provide clear guidelines**
* **Facilitate and understanding across teams**
* **Form a basis for testing**
* **Help manage customer expectations and enhance satisfaction**

Given that the customer wants to apply for a loan, when they loan officer check the transaction history of the customer, then borrower will be provided with the loan.

* First the loan officer check on with the customers whole transaction history,
* Then the customer should calculate the loan eligibility and should know about how much should he/she should borrow
* Next the loan processor should track the loan status
* When the customer is eligible for the loan
* Then the customer will be provided with the required loan

INVEST

## What is INVEST?

The acronym [**INVEST**](http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/) helps to remember a widely accepted set of criteria, or checklist, to assess the quality of a [**user story**](https://www.agilealliance.org/glossary/user-stories/). If the story fails to meet one of these criteria, the team may want to reword it, or even consider a rewrite (which often translates into physically tearing up the old story card and writing a new one).

A good user story should be:

* “I” ndependent (of all others)
* “N” egotiable (not a specific contract for features)
* “V” aluable (or [**vertical**](http://guide.agilealliance.org/guide/incremental.html))
* “E” stimable (to a good approximation)
* “S” mall (so as to fit within an iteration)
* “T” estable (in principle, even if there isn’t a test for it yet)

**Independent (I)**

All the user stories should be self-contained single units and independent of each other. This segregation between each story makes prioritization, evaluation, addition or elimination easier from an iteration.

#### Negotiable (N)

In Agile, a story is not supposed to be an agreement and the crux of the same lies in delivering what the customers desire.

#### Valuable (V)

Each story is required to be of value to the customer (it could be the user or the purchaser). And the best way to do the same is to make the customers compose them.

#### Estimable (E)

If the story is too big, the development team will not be able to understand it, to make an estimation. The story should enable the development team to make a judgment about the complexities of work and the efforts required.

#### Small (S)

As it’s known, agile stories are smaller units of work.

But what is the expected story size?

Ideally, the development team should be able to deliver a unit of work task within 1 Sprint (Single sprint duration – not more than 2 to 4 weeks). The stories which exceed this duration and efforts tend to have errors with estimation.

#### Testable (T)

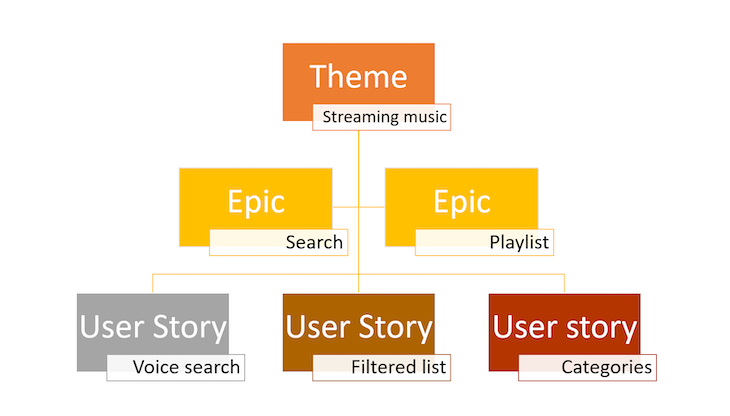
To go through the confirmation, the story has to be testable.

Epic--An Epic can be defined as a big chunk of work that has one common objective. It could be a feature, customer request or business requirement.

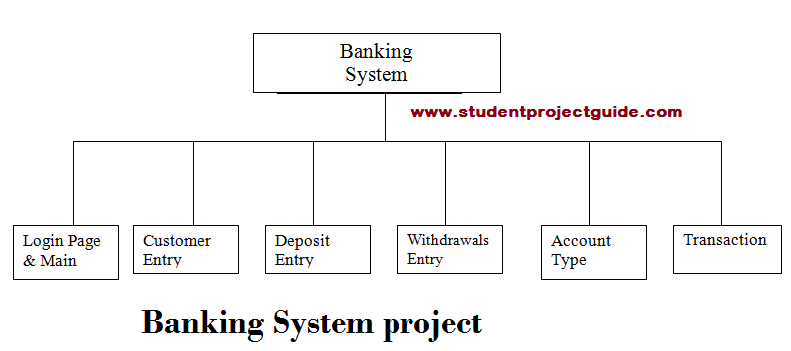
The Basic unit of work defined in Scrum is User story. But very often, when Product Owner writes a user story for a feature or against customer request, that looks simple in the beginning. But, while covering all related work and scenarios, same user story expands so much that it can not fit either in a week or a sprint time-frame. It is the time to consider this big user story as epic and start slicing it in smaller user stories. This way, Agile teams get better effort estimate and get smaller but concrete output in single sprint.

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.

Example--



User Stories--



Register and login—

As a User I want to create a new bank account using the mobile app so that I can carry out the financial transactions.

As a customer, I can register for the mobile application by entering email,Username and password ,so that I can login to the app.

As a customer, I can login to the application by using email and the password so that I can be able to access the application.

Customer entry—

As a customer, I can view my profile information so that I can manage my account effectively.

As a customer, I can apply for the debit card so that I can be able to transaction easily.

As a customer, I want to know about my bank balance, so that I check the balance through ATM or mobile application by entering the pin.

Deposit entry—

As a customer, I want to deposit the money on other account so that I can use the deposit cheque or Demand draft for depositing the money.

As a customer I want to send money to the other account so that I can use ATM machine for deposition.

As a customer, I can use Banking application so that I can send money to the other account .

Withdraw entry--

As a customer, I can use ATM card for Withdrawals so that I will have no need to go the bank .

As a customer, I want to withdraw the money on my account so that I can use the withdrawal cheque.

Account type--

As a customer, I have to know about what type of account to be opened so that I can open an account like savings, salary, current account.

Transaction--

As a customer, I can make transaction through a demand draft or a cheque

As a customer, I can view my account statement and transaction history in the application so that I can make further transactions and manage my finance.

A sprint backlog is a list of work items your team plans to complete during a project sprint.

Sprint is a fixed-length iteration and an agile scrum event, during which a cross-functional scrum team works to complete a set amount of work.

A product backlog is a prioritized list of work items or features that help you meet product goals and set expectations among teams.

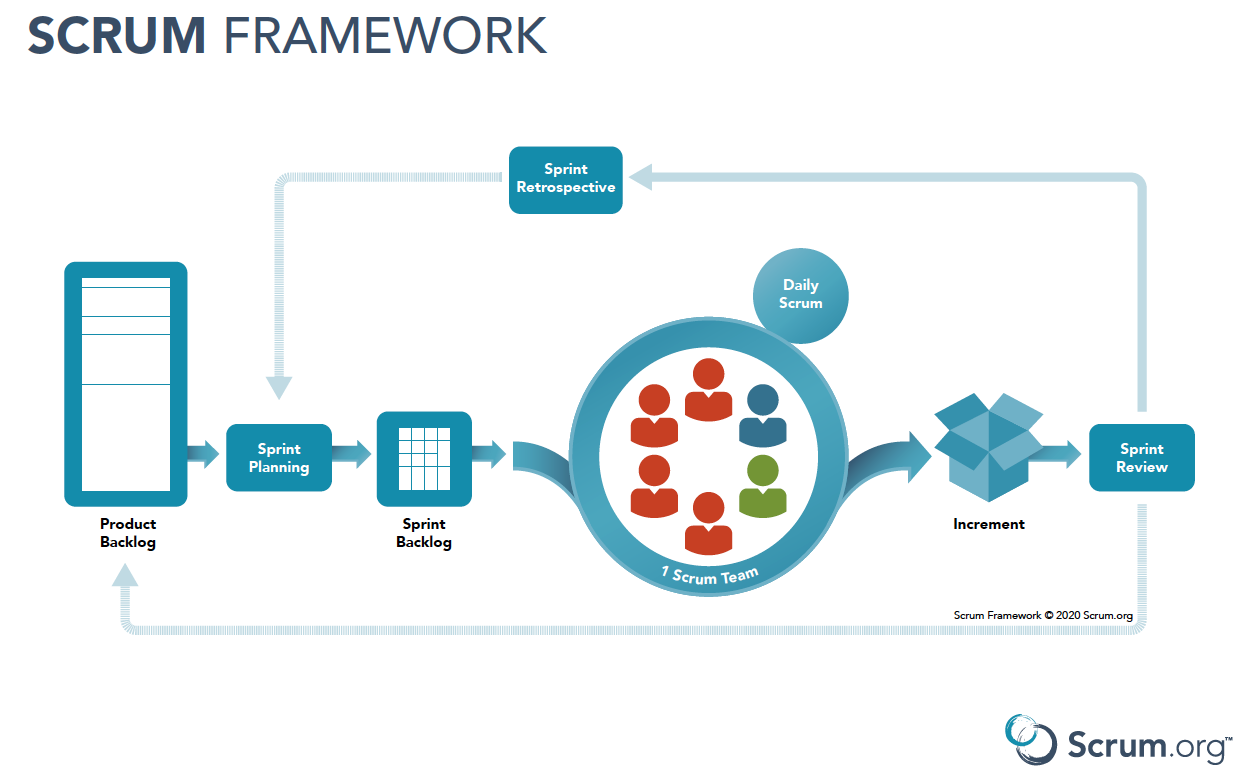
## What does a Scrum Master do?

## Scrum is an [Agile](https://www.coursera.org/articles/what-is-agile-a-beginners-guide) framework for developing complex projects, most often software. The Agile project management methodology uses short development cycles, called sprints, that result in the continuous improvement of a product or service. There are many Agile frameworks, and Scrum is a popular option for fast-moving projects. The methodology is highly collaborative and requires efficient processes, and the results of the process depend upon the expertise of the Scrum Master.

### **What does a Product Owner do?**

The Product Owner is accountable for effective [Product Backlog](https://www.scrum.org/node/8119) management, which includes:

* Developing and explicitly communicating the Product Goal
* Creating and clearly communicating Product Backlog Items
* Ordering Product Backlog Items
* Ensuring that the Product Backlog is transparent, visible and understood



* **Verification** evaluates software artifacts (such as requirements, design, code, etc.) to ensure they meet the specified requirements and standards. It ensures the software is built according to the needs and design specifications.
* **Validation**evaluates software to meet the user’s needs and requirements. It ensures the software fits its intended purpose and meets the user’s expectations.

“A test of a system to prove that it meets all its specified requirements at a particular stage of its development.”

Verification tests must be run at every stage of development before any feature is implemented.

“An activity that ensures that an end product stakeholder’s true needs and expectations are met.”

Unlike verification testing, which occurs at every stage in development, validation testing occurs at the end of a specific module or even after the software has been entirely built. Its primary intent is to ensure the final product matches the stakeholder and customer requirements.

For example--

 Imagine yourself going to a restaurant/diner and ordering maybe blueberry pancakes. When the waiter/waitress brings your order out, how can you tell that the food that came out is as per your order?

**Verification is all when you are yet to eat but are checking on a few things by reviewing the subjects.**

* Does the food look like what pancakes typically appear to be?
* Are the blueberries to be seen?
* Do they smell right?

**Validation is when you actually eat the product to see if it is right.**

**Verification answers the question, “Did we build the right system?” while validations addresses, “Did we build the system right?”**