INVEST Principle:

Independent:

User stories should be self-contained and not dependent on another story to be completed.

Example:

As a user, I want to be able to reset my password so that I can regain access to my account if I forget it.

Negotiable: User stories should be open to negotiation and discussion between the development team and stakeholders.

Example:

As a user, I want to filter search results by price range so that I can find products within my budget.

Valuable: User stories should deliver value to the end user or customer.

Example:

As a customer, I want to receive email notifications when my order ships so that I can track its delivery status.

Estimable: User stories should be small and well-defined enough to allow for accurate estimation of effort. Teams should be able to estimate how long it will take to implement a user story and what resources will be required to complete it.

Example:

As a user, I want to be able to add items to my shopping cart so that I can purchase them later.

Small: User stories should be small enough to be completed within a single iteration or sprint. Break down the larger feature into smaller so it can help to reduce the risk.

Example:

As a user, I want to be able to view product reviews on the product detail page so that I can make informed purchasing decisions.

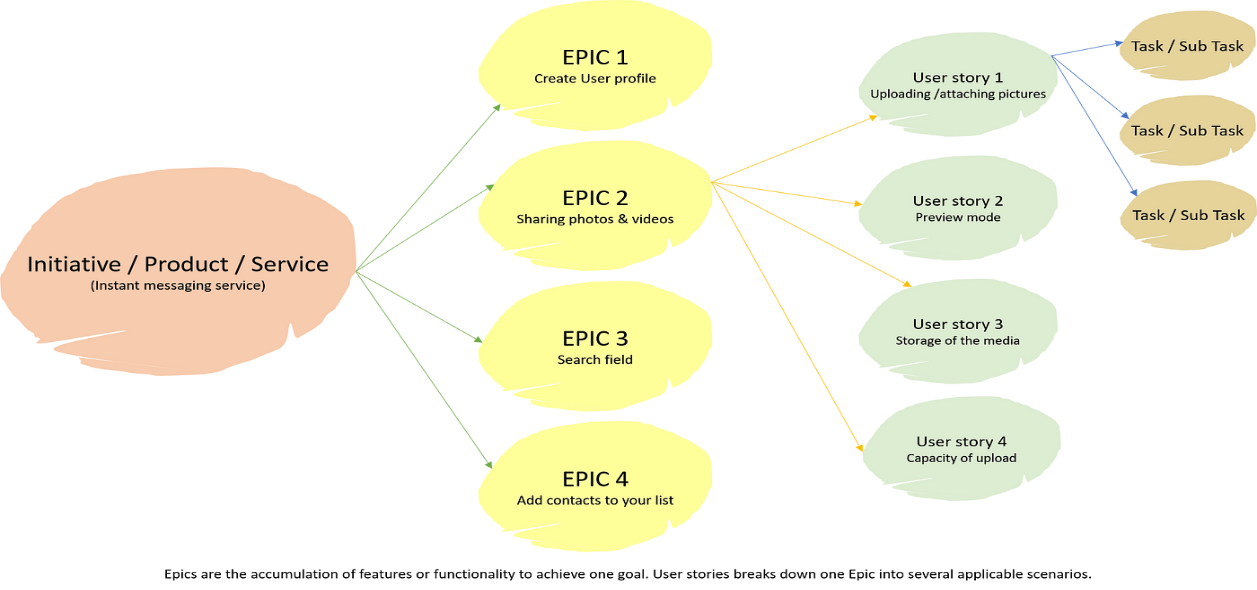
Testable: User stories should be written in a way that allows for testing to confirm whether the acceptance criteria are met.

Example:

As a user, I should be able to add items to my shopping cart" is testable by verifying that items are successfully added and reflected in the cart.

EPIC:

Epics are large chunks of work that can be broken down into a number of smaller pieces.



Product Backlog:

The product backlog is the **list of all the work that needs to get done to deliver a product**. It usually contains user stories, bugs, technical tasks, and knowledge acquisition.

The product backlog is typically managed by the product owner, who is responsible for prioritizing the items based on their value, risk, and dependencies.

Scrum:

Scrum is a framework for agile software development that emphasizes iterative and incremental delivery, collaboration, and continuous improvement. It provides a structured approach for teams to work together on complex projects while adapting to changing requirements and feedback.