

# **EPICS USER STORIES STORY WRITING**

**People who want IT must communicate  
with people who can build IT.**

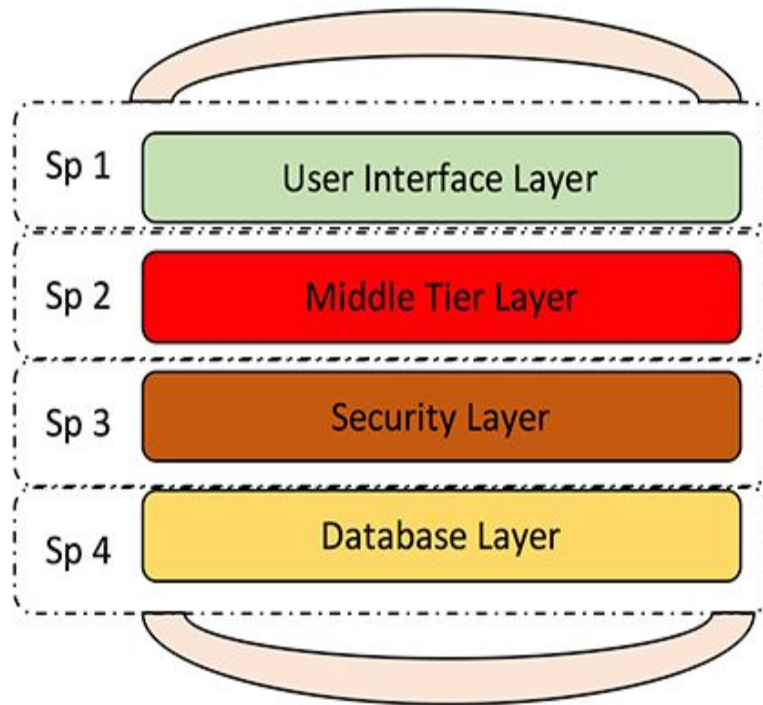
# NEED

There should be a way to:

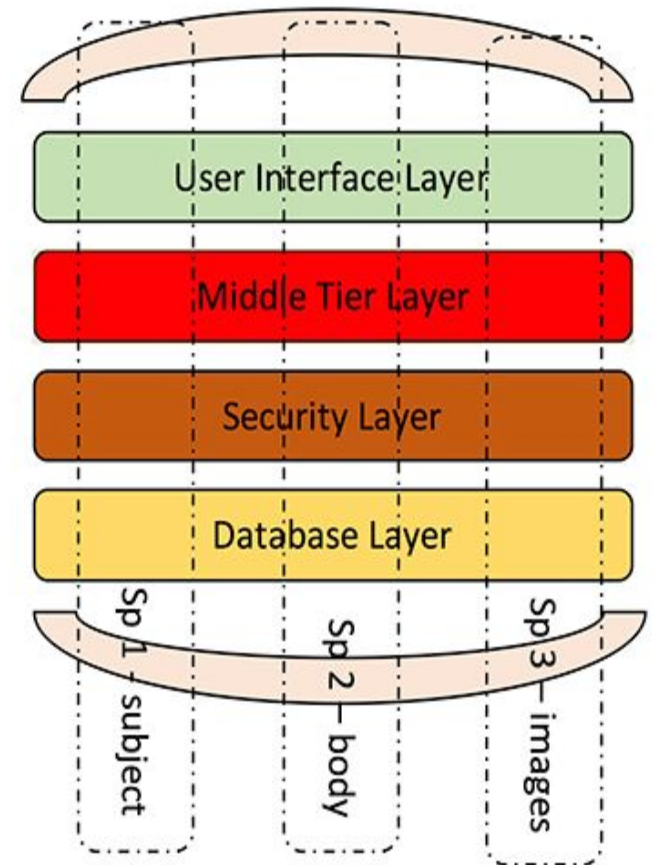
- Define requirements / features at high level
- Break high level requirements into smaller understandable pieces
- Quickly Estimating of schedule (both short term and long term)
- Prioritizing requirements of higher business value over lower ones
- Communicate requirements to development team more simply / effectively

# Delivering Customer Value

Would you rather have this?



Or this?



# USER STORY

The **user story** focuses on the **experience** — what the person using the product wants to be able to do.

A **traditional requirement** focuses on functionality — what the **product should do**

A user story is a short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system .

A user story is a technology and/or business requirement described in terms of the user, providing context as to what they want and why. Stories in practice typically require multiple people to contribute to their completion.

- Is short and simple
- User Stories are NOT part of the Scrum framework
- User Stories are an eXtreme Programming technique
- User Stories capture Who, What and Why of any requirement
- Conversation rather than documentation

# Writing Stories

- Follow this basic construct:

<b>WHO</b> are we building it for? Who is the user?	As a <type of user>
<b>WHAT</b> are we building? What is the intention?	I want <some goal or objective>
<b>WHY</b> are we building it? What is the value for the customer?	So that <benefit/value>

# Writing Stories - Guidelines

- Collaboration: write the story with your team, involve all aspects to reduce waste
- Story break down: work with your team to review, question and break down the stories
- Decompose your stories until they are ready
  - Break your epics into smaller, detailed stories until they are ready: clear, feasible, and testable
  - Everyone should have a shared understanding of the story's meaning; the story should not be too big, and there has to be an effective way to determine if the story is done
  - Avoid confusing and ambiguous terms
  - Focus on what's important and leave out non-essentials

# User Stories - Examples

As an agile learner I want to understand the role of user stories so that I can articulate customers needs

As a staff member I need to buy a parking pass that lets me park anywhere on campus so that I can get on meetings ontime

As an online shopper, I want to add an item to my cart, so that I can purchase it

As a marketer I want create automated email campaigns So that I can keep evaluators engaged

As an ABC Bank marketing executive I want Advt. on the home page of bank so that I can increase revenue of the bank

## Writing Stories - Guidelines

**I**

***Independent***

**N**

***Negotiable***

**V**

***Valuable***

**E**

***Estimatable***

**S**

***Sized Appropriately***

**T**

***Testable***



# Writing Stories - INVEST

## **Independent**

Dependencies lead to problems estimating and prioritizing. Ideally you can work on a single story without pulling in lots of other stories.

## **Negotiable**

Stories can be changed and rewritten as needed until they are pulled into a sprint (iteration).

## **Valuable**

Valuable to users or customers, not developers. Rewrite developer stories to reflect value to users or customers

## **Estimable**

We need to be able to estimate our user stories so that we can use them to create a plan.

## **Small/Sized Appropriately**

A story is sized appropriately when it can be completed in one iteration

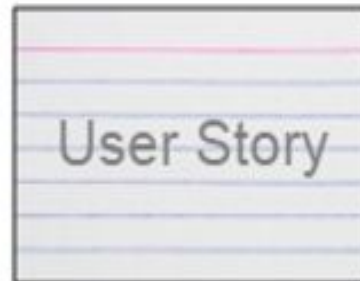
## **Testable**

You should have an easy and binary way of knowing when a story is finished. Done or not done; not partially finished or “done... except...”

# USER STORY – 3 C's

## C

ard



- ☐ Token for Conversation
- ☐ Physical or Virtual
- ☐ Captures Who, What & Why

---

## C

onversation



- ☐ Gain Shares Understanding
- ☐ Elaborate and Record Details

---

## C

onfirmation



- ☐ Product Owner's conditions of satisfaction (Acceptance Criteria)

# Acceptance criteria to Stories

Acceptance criteria or 'conditions of satisfaction' provide a detailed scope of a user's requirements.

They help the team to understand the value of the story and set expectations as to when a team should consider something done

## Acceptance Criteria Goals:

- To clarify what the team should build before they start work
- Define the boundaries of a user story
- Help developers know when to STOP adding more functionality
- To ensure everyone has a common understanding of the requirement
- To help the team members know when the story is complete
- Help developers and testers to derive tests

# User story with Acceptance criteria - Example

**User Story :** As an online banking customer, I want strong a strong password, so that my credit card information is secure

## **Acceptance Criteria :**

- The password must be at least 8 characters
- The password must contain at least 1 character from each of the following groups: lower case, alphabet, upper case alphabet, numeric, special characters (!, @, #, \$, %, ^, & , \*)

# User story with Acceptance criteria - Example

**User Story** : As an enrolled student, I need the ability to pay with a credit card that has separate billing information so that I can pay using my parent's card.

## **Acceptance Criteria:**

- Accept Discover, Visa, MC
- Validate CC# when entered
- Validate expiration date and CVV
- Validate billing address
- Generate success and failure messages after processing

# User story with Acceptance criteria - Example

**User Story** : As an enrolled student, I need the ability to pay with a Discover credit card that has separate billing information so that I can pay using my parent's card.

## **Acceptance Criteria:**

- Validate CC# when entered
- Validate expiration date and CVV
- Validate billing address
- Generate success and failure messages after processing

# User story with Acceptance criteria - Example

**User Story** : As an enrolled student, I need the ability to pay with a Master credit card that has separate billing information so that I can pay using my parent's card.

## **Acceptance Criteria:**

- • Validate CC# when entered
- Validate expiration date and CVV
- Validate billing address
- Generate success and failure messages after processing

# User story with Acceptance criteria - Example

**User Story** : As an enrolled student, I need the ability to pay with a Visa credit card that has separate billing information so that I can pay using my parent's card.

## **Acceptance Criteria:**

- Validate CC# when entered
- Validate expiration date and CVV
- Validate billing address
- Generate success and failure messages after processing



# USER STORY - BENIFITS

## Key benefits of user stories

- Precise
- Stories keep the focus on the user
- Stories enable collaboration
- Stories drive creative solutions
- Stories create momentum
- Very useful for Sprint planning
- Understood equally well by everyone

# User stories & Tasks

User stories	Tasks
a user story = the WHAT	the task = the HOW
user stories describe a piece of functionality from the point of view of the user	“what are the activities we need to perform in order to deliver outcomes (user stories)”
divided features into business processes	tasks are individual pieces of work

# EPIC

**An epic is a group of user stories in an Agile requirement, stated as a short description of something a customer will do.**

- Is a label applied to a large story that is broken into smaller user stories
- Product Backlog item or User Story too big to complete in 1 Sprint
- need to be broken down so that the team can deliver value in a given Sprint
  - This is done during Backlog Refinement
- One Epic can be made up of many user stories, but a user story can only relate to one Epic
- Backlog Refinement meeting is where the Team works with the PO to break the Epic down appropriately
- Business Value can be best estimated at this level

# EPIC Slicing Example

**Epic** : As a frequent flyer I want to book flights customized to my preferences, so I save time

**User Story 1** : As a frequent flyer I want to book a trip using miles so that I can save money

**User Story 2** : As a frequent flyer I want to easily book a trip I take often so that I can save time

**User Story 3** : As a premium frequent flyer I want to request an upgrade so I can be more comfortable

**Epic** : As a Jira user I would like to have the reporting function in order to take take right decisions

**User Story 1** : As a Jira user I would like to have my sprint report in order to conduct retrospectives effectively

**User Story 1** : As a Jira admin I should be able to integrate latest gadgets from markets so that stakeholders can benefit from automated dashboards

# Taxonomy

