

User Story

A user story describes something that the user wants to accomplish by using the software product.

As a [persona],

I want to [do something],

so that I can [derive a benefit]

Who

What

Why

The persona is a vivid, humanised, yet operational description of a user.

This is a some action you assume the user wants to do.

This is benefit / motivation / reason why the user wants to perform the action.

not HOW?

Requirements in Agile / Scrum may not follow “user story” format but it should cover information of WHO-WHAT-WHY.

Acceptance Criteria

Pass/Fail conditions providing a story is done. AC for the larger story can become a new, smaller user stories with their own acceptance criteria. AC can be:

- Functional: identify specific business rules, user tasks, functions or business processes that must be in place.
- Non Functional: identify specific non-functional conditions the implementation must meet. E.g., “User’s action buttons comply with the button design in UI & UX Guideline”

Purpose of Acceptance Criteria

- Define the boundaries for a user story / feature
- Help the Product Owner answer details of the value provided by the story
- Help the team gain a shared understanding of the story / feature

Acceptance Criteria should state intent, not a solution. The criteria should be independent of the implementation. E.g., “A manager can approve or disapprove an audit form” rather than “A manager can click an ‘Approve/Disapprove’ radio button to approve an audit form”.

User Story

As a customer,

I want to be able to cancel my reservation,

so that I do not lose all the money if an incident occurs.

Card

Describe the user story in short form to allow common understanding of the user need

3C

Conversation

User stories shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

Confirmation

Confirm the understanding is correct. Confirm the story is ready for next step of development.

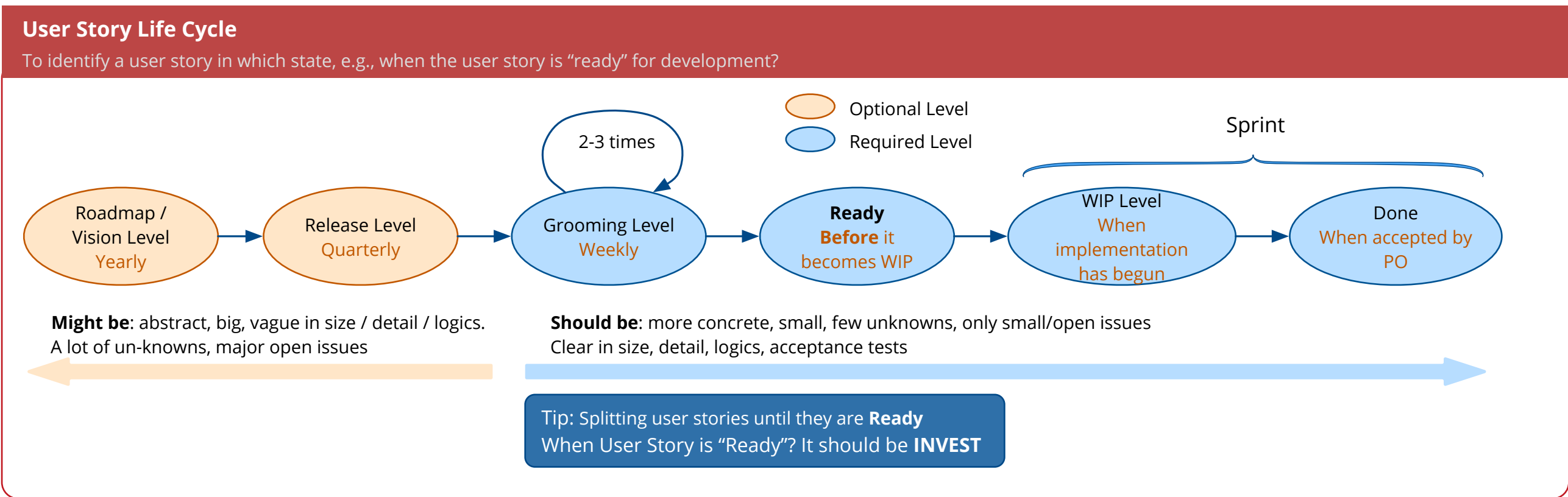
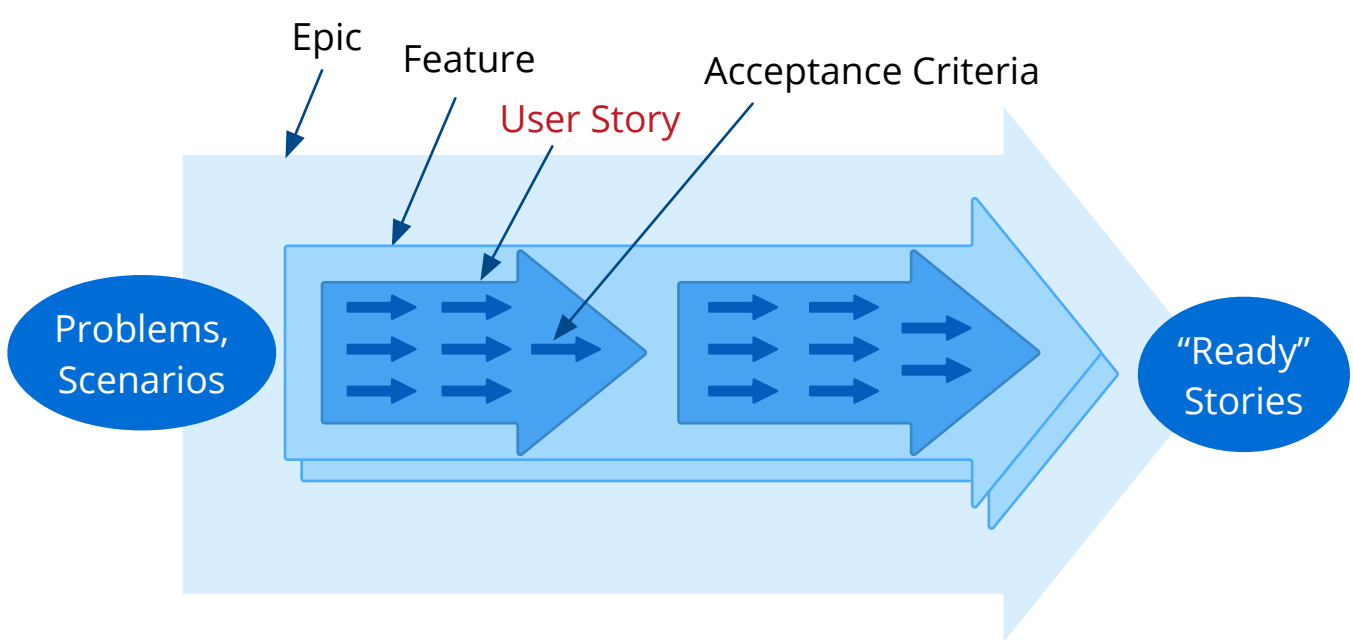
Acceptance Criteria

Verify that a premium member can cancel the same day without a fee.

Verify that a non-premium member is charged 10% for a same-day cancellation.

Verify that an email confirmation is sent.

Verify that the hotel is notified of any cancellation.



Feature / Story Mapping

How we organise features & their stories into sprints?

Benefits

- Provides visibility of the workflow across the system
- Points out relationships between stories
- Helps to spotlight missing stories
- Provides a prioritization mechanism
- Release planning is improved by focusing on valuable slices

Step 1

Arrange features in the business sequence they are done.

Step 2

Identify stories that support features

Step 3

Identify missing stories and fill in

Step 4

Prioritise stories: arrange stories so most necessary and importance are at the top.

Step 5

Create horizontal slice: pick necessary stories that are required for the first release

Step 6

Make a skeleton release / Create multiple horizontal slices: prioritisation considers business workflow and value of stories within the context of the whole system.

Business Usage Sequence

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Feature 1

Feature 2

Feature 3

Feature 4

User Story

Horizontal Slices

Factor	Waterfall/Traditional BA	Agile BA
Focus	Requirements documentation	Value delivery in the form of business outcomes
Artifact	Requirement Documents (SRS, Use Case Specifications, Functional Documents)	Backlog Documents, Backlogs (e.g., managed in Jira)
Requirements Definition	All requirements defined upfront	Defined continuously in small batches (i.e., features)
Requirement Approach	Use Case, Functional Decomposition	User Story (and Acceptance Criteria)
Sign-Off	Formal sign-off obtained	Quick feedback
Requirement Changes	Strictly controlled after baseline	Changes are anticipated and welcome at any time

Scope

Timeline

Cost

Risk

Value

Scope

Timeline

Cost

Value

Value = Customer Value + Knowledge Value

Domain Knowledge / Client's Business Awareness

Domain knowledge helps us understand IF WE ARE DOING THE RIGHT THING.

What should we focus in Agile development? How to find the trade-off?

Build the right thing (WHAT)

Build the right things right and fast. But it's hard to find a balance

Quick start but not good quality. Great for short-term usable products. But may be huge problems in future.

Build it fast

Try to build a perfect product with a perfect architecture. May miss market opportunities or run to cost problems.

Build the thing right (HOW)

High quality, but no stakeholder needs. Product failed.

Product Owner / BA

Dev Team

Scrum Master / Managers

INVEST Principle

- Independent: No dependencies between user stories
- Negotiable: Describes functionality to be negotiated between the customer and development team. Must deliver “value” to the end-users.
- Valuable: Must be able to be sized so that it can be planned
- Small: Small enough to be planned inside a sprint
- Testable: Worded in a way they can be tested

Prepare the input story

Apply splitting patterns

Evaluate the split

INVEST is validated?

Yes

No

User Story Splitting / Agile Requirements Decomposition

Technique 1 Workflow

Examine the workflow of a process, and create stories for steps in that workflow.

Before

As a grocery store cashier, I want to checkout a customer, so that I can complete the sales transaction.

After

As a grocery store cashier, I want to...

- ...calculate the total amount that will be charged to customer
- ...specify the method of payment preferred by the customer
- ...enter the credit card details
- ...print a receipt for the customer

Technique 2 Data Details

For complex data or user interfaces, write stories for related elements

Before

As a student, I want to view my grades for this semester's courses, so that I can see how I'm performing.

After

As a student, I want to view...

- ...my numeric grade for this semester's courses, so that I can quantify my performance.
- ...my letter grade for this semester's courses, so that I can calculate my GPA easily
- ...the class average for this semester's courses, so that I understand my relative performance.

Technique 3 Happy Path

For complex workflows, create a story to complete the happy path, i.e., the case where everything goes perfectly, and then create stories which deal with the deviations from the happy path (i.e., alternative flows).

Before

As a UBER-taxi customer, I want to view information about my booked taxi, so that I can track its movement.

After

As a UBER-taxi customer, I want to view information about...

- ...an on-time taxi, so that I can track its movement
- ...a delayed taxi, so that I can track its movement
- ...a cancelled taxi, so that I can re-book another one..

Technique 4 Core+Enhance

Do the core, then enhance: Create a story for the difficult part of the work where key learnings will happen; then create stories which complete the work

Before

As a Salesforce user, I want to create revenue, profit, and growth reports, so that I can perform monthly forecasting.

After

As a Salesforce user, I want...

- ...to create a revenue report for a month, so that I can view the revenue generated in that month
- ...to create revenue, profit, and growth reports for all months, so that I can perform forecasting for the next month.

Technique 5 Business Rules

Many user stories involve a number of explicit or implicit business rules. Break each rule in a user story.

Before

As a shop owner, I want to track & control the orders submitted from the customer in my store, so that I'm aware of what status of the order is

After

As a shop owner, I want to...

- ...decline orders below 10 dollars, because I don't make any profit on them;
- ...decline customers from outside the US, because the shipping expenses make these orders unprofitable;
- ...reserve ordered products from stock for 48 hours, so other customers see a realistic stock;
- ...automatically cancel orders for which I have not received payment within 48 hours, so I can sell them again to other customers;

Technique 6 Operation

Stories often involves a number of default operations, e.g., CRUD. These operations are very prevalent when functionality involves the management of entities.

Before

As shop owner I want to manage products in my online shop, so I can update price and product information if it is changed.

After

As a shop owner, I want to...

- ...add new products, so customers can purchase them;
- ...update existing products, so I can adjust for changes in pricing or product information;
- ...delete products, so I can remove products that I no longer stock;
- ...hide products, so they cannot be sold for the time being;

Agile = Iterative + Incremental

What Iterative & Incremental means? How it relates to BA work in Agile / Scrum?

Agile

Don't try to get it all right from the beginning

Iterative

Iterative Development (Scrum)

Timebox

Iterations

Daily Meeting

Three Questions

Burndown Chart

Definition of Done

Definition of Ready

Point Estimate

Relative Estimate

Backlog

Backlog Grooming

Incremental

Incremental Development

INVEST Principle

3C

User Stories

Story Decomposition

Story Splitting

Story Mapping