User Story

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

As a **[persona]**, **Who** The persona is a vivid, humanised, yet operational description of a user.

not HOW?

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.

I want to **[do something]**, so that I can **[derive a benefit]**

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

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

**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

Acceptance Criteria should state intent, **not a solution**. The criteria should be independent of the implementation.

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

**3C**

**Card**

Help the Product Owner answer details of the **value provided** by the story

Help the team gain a **shared understanding** of the story / feature

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”.

y**As a** customer, r

o

t

Describe the user story in short form to allow

**I want to** be able to cancel my reservation,

S

r

e

s

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

common understanding of the user need

**Conversation**

User stories shift the focus from writing about features **to discussing them**. In fact,

Acceptance in form of Given-When-Then

The Given-When-Then formula is a template intended to guide the writing of **acceptance criteria** for a user story:

(Given) some context

(When) some action is carried out

**Given** my bank account is in credit, and I made no withdrawals recently, **When** I attempt to withdraw an amount less than my card's limit,

**Then** the withdrawal should complete

a

ir

e

t

ir

C

e

c

n

a

t

p

e

c

c

A

Verify that a premium member can cancel the same AC1

day without a fee.

Verify that a non-premium member is charged 10% AC2

for a same-day cancellation.

Verify that an email confirmation is sent.

AC3

Verify that the hotel is notified of any cancellation. AC4

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.

(Then) a particular set of observable consequences should obtain

without errors or warnings

Epic

Problems,

Feature

User Story

Acceptance Criteria

“Ready”

**User Story Life Cycle**

To identify a user story in which state, e.g., when the user story is “ready” for development?

2-3 times

Roadmap /

Optional Level Required Level

**Ready**

Sprint

WIP Level

Done

Scenarios

Stories

Vision Level Yearly

Release Level Quarterly

Grooming Level Weekly

**Before** it

becomes WIP

When

implementation has begun

When accepted by PO

**Might be**: abstract, big, vague in size / detail / logics.

A lot of un-knowns, major open issues

**Feature / Story Mapping** How we organise features & their stories into sprints?

Provides **visibility** of the workflow across the system

s

t

**Should be**: more concrete, small, few unknowns, only small/open issues

Clear in size, detail, logics, acceptance tests

Tip: Splitting user stories until they are **Ready**

When User Story is “Ready”? It should be **INVEST**

Prepare the

input story

Points out **relationships between stories**

fi

e

Helps to spotlight missing stories

n

e

B

Provides a **prioritization** mechanism

Release planning is improved by focusing on **valuable slices**

**INVEST** Principle

**Independent**: **Negotiable**:

**Valuable**:

No dependencies between user stories

Describes functionality to be negotiated between the customer and development team

Must deliver “value” to the end-users.

Apply splitting patterns

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.

**Estimable**: **Small**:

**Testable**:

Must be able to be sized so that it can be planned Small enough to be planned inside a sprint Worded in a way they can be tested

No

Yes

Evaluate the split INVE**S**T is

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

validated?

**User Story Splitting / Agile Requirements Decomposition**

Technique 1

Feature 1 Feature 2 Feature 3 Feature 4

t

n

a

t

r

Sprint 1 User Story User Story User Story User Story o

p

User Story User Story Sprint 2 User Story

m

i

s

e

e

c

r

i

l

o

S

M

l

a

User Story Sprint 3

User Story User Story

t

n

t

o

n

z

User Story

i

a

r

t

o

r

o

H

User Story User Story User Story Sprint 4User Story

p

m

i

s

s

e

L

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

After

Before As a grocery store cashier, I want to…

As a grocery store cashier, I want

...calculate the total amount that will be charged to customer

to checkout a customer, so that I

...specify the method of payment preferred by the customer

can complete the sales

...enter the credit card details

transaction.

...print a receipt for the customer

Technique 2

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

After

Before As a student, I want to view…

Factor

Focus

Waterfall/Traditional BA

Requirements documentation

As a student, I want to

view my grades for this

semester’s courses, so

that I can see how I’m

performing.

Agile BA

Value delivery in the form of

...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.

business outcomes

Specifications, Functional Documents) Artifact Backlog Documents, Backlogs (e.g.,

Requirement Documents (SRS, Use Case

managed in Jira)

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).

All requirements defined upfront Requirements Definition

Use Case, Functional Decomposition Requirement

Defined continuously in small batches (i.e., features)

Before

As a UBER-taxi customer, I want

After

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

Approach User Story (and Acceptance Criteria) Sign-Off Formal sign-off obtained Quick feedback

to view information about my booked taxi, so that I can track

…an on-time taxi, so that I can track its movement …a delayed taxi, so that I can track its movement

Strictly controlled after baseline Requirement Changes

e

e

p

p

o

o

c

c

S

S

Changes are anticipated and welcome at any time

its movement.

Technique 4 **Core+Enhance**

Before

…a cancelled taxi, so that I can re-book another one..

**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

After

**Cost**

Risk

Timeline

Value

**Cost**

Value

Timeline

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

forecasting.

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.

Value = Customer Value + Knowledge Value

Technique 5 **Business Rules**

Before

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

After

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?

Try to build a perfect

product with a perfect

**architecture**. May miss

market opportunities or

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

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;

Build the **right** thing

run to cost problems.

(WHAT) Build the

Technique 6

Stories often involves a number of default operations, e.g., CRUD. These operations are very

**Product**

**Owner / BA**

Build the **right**

**things right and**

**Dev**

**Team**

thing **right** (HOW)

**Operation** Before

prevalent when functionality involves the management of entities. After

**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.

**Agile = Iterative + Incremental**

**Scrum Master / Managers**

Build it **fast**

High quality, but no stakeholder needs. Product failed.

As shop owner I want to manage products in my online shop, so I can update price and

product information if it is changed.

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;

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

**e**

**li**

**gA**

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

**Iterative**

**Incremental**

**Iterative**

**Development** Timebox Iterations Daily

(Scrum)

Meeting

**Incremental**

**Development** INVEST

Three

Questions

Burndown Chart

Definition of Done

Definition of Ready

Point

Estimate

Relative

Estimate Backlog

Backlog

Grooming

Principle 3C User Stories Story

Story

Story

Don’t build it all at once

Decomposition

Splitting

Mapping