COMP 1531

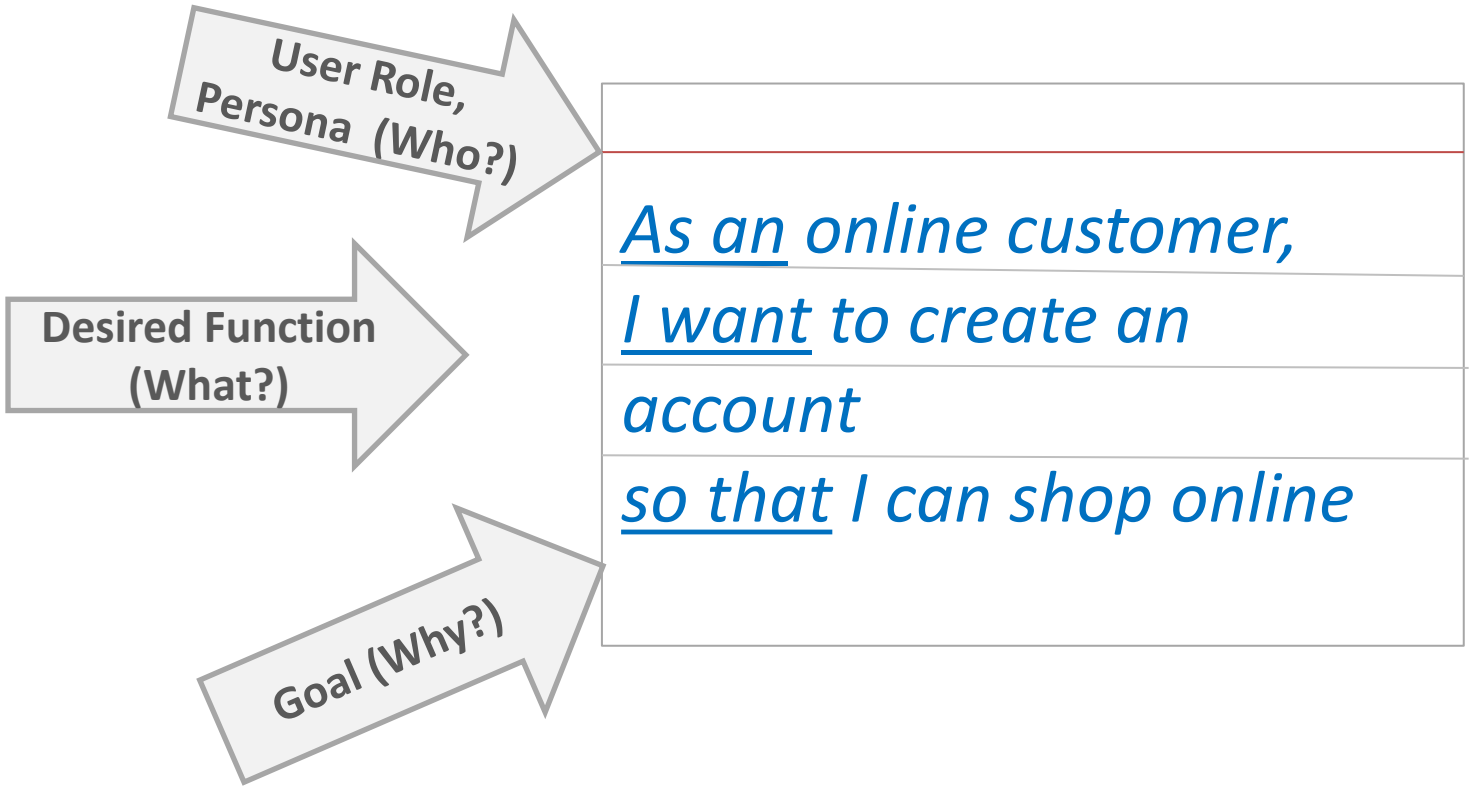
Software Engineering Fundamentals

Week 03

The Art of Writing User-Stories

User Story at a Glance

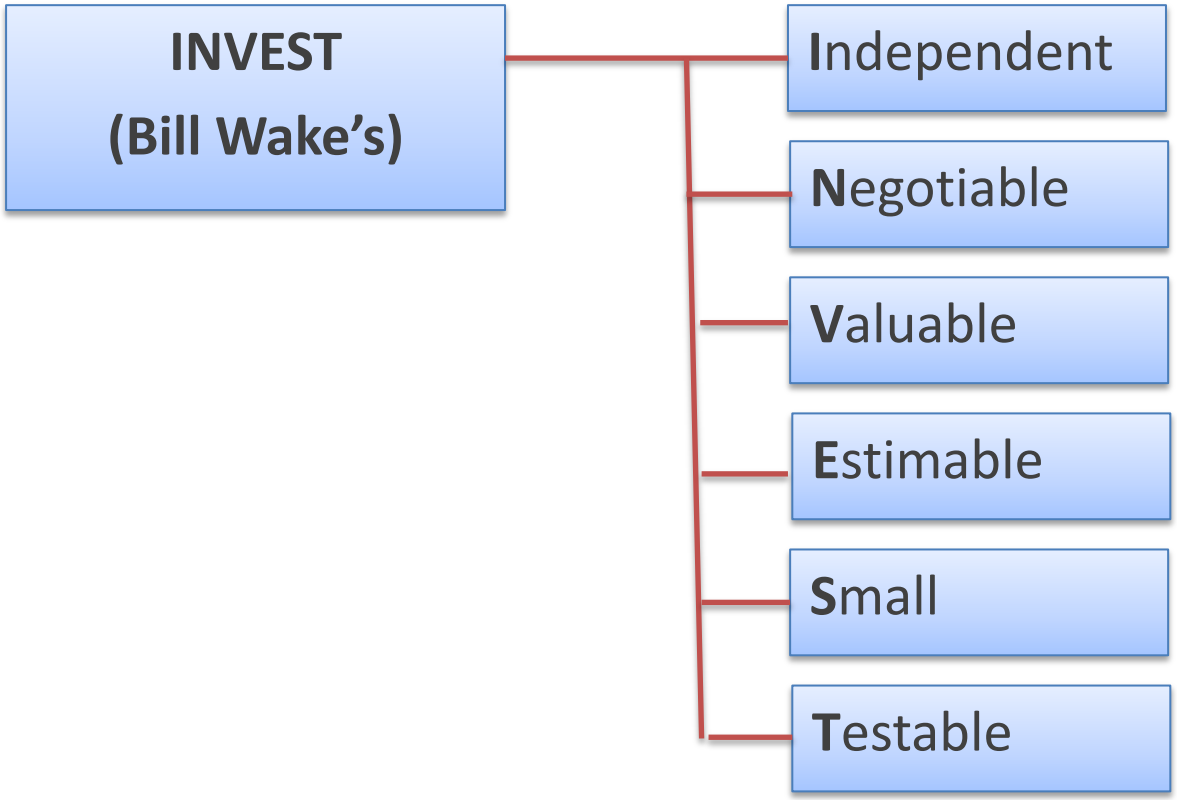
Basic user story has a simple template – a high level description of desired functionality and goal



Attributes of a good user-story

The WHO, WHAT and WHY of user-story must be at the right granularity and add value to the business and user

INVEST – an acronym that helps evaluate whether you have a high quality user story

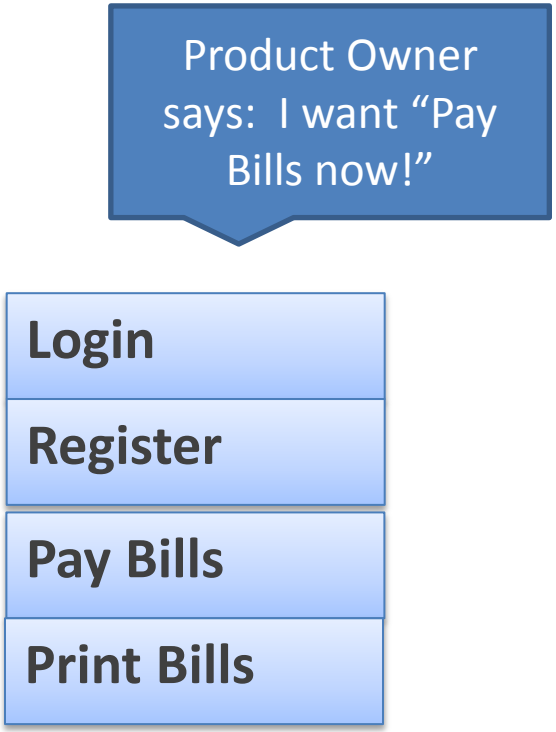


Attributes of a good user-story

- I = *Independent*: user story could be developed independently and delivered separately
- N = *Negotiable*: avoid, too much detail. user story should be discussable further, keep them flexible
- V = *Valuable*: the product owner should be "clear" on the "why" of the original statement (value of the user story)
- E = *Estimable*: user story should be understandable enough so could be divided into task and could get estimated
- S = *Small*: user story should be small, deliverable within an iteration (i.e., designed, coded and tested within the iteration)
- T = *Testable*: user story should be defined with clear *acceptance criteria*, both the correct functionality and the error conditions which leads to test-cases

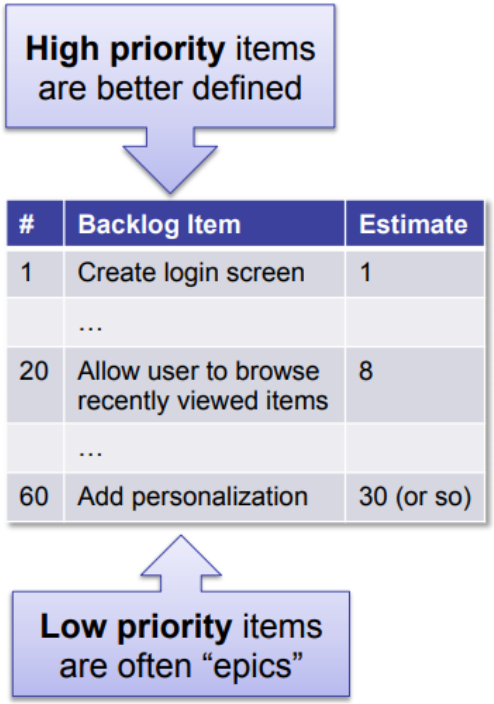
Invest: Independent

- Ideally, a user-story must be self-contained, independent (no inherent dependency on other PBI)
- If all independent, then any PBI can be developed in isolation, not always feasible!
- So, identifying right granularity, prioritizing PBI and minimising dependencies results in a better backlog
- Which user-story comes first?



Invest: **N**egotiable

- Leave room for negotiation, defer collecting details when you have more context
- Not fleshed out completely, need only enough to allow prioritisation
  - Higher priority user-stories should be more precisely defined
  - Defer details for lower priority stories



US#2
As a delivery driver,
I want to get directions to customer homes
so that I can get there quickly

Acceptance Criteria
Show location on maps
~~*Show location on Google Maps*~~

Update acceptance criteria, when you are close to implementation

Invest: **V**aluable

- The user-story must have value to the customer and the business

US
As a user
I want to have my previous orders stored
so that I can re-order those items again quickly

There is clearly value to the customer, but is there value to the business?

US
As a customer
I want to save 75% of all purchases
so that I can save money

Invest: **E**stimable/**S**mall

- Stories need to be clear enough to estimate (without being too detailed)
- If user-story is not at the right granularity or vague, it is difficult to estimate it
- Stories should be built in a small amount of time, a few person-days

US

As a customer
I want to login with user-name/pwd
so that I can access my account securely

Break a large EPIC into precise, estimable, atomic user-stories

US

As a user,
I want to login successfully with a correct user-name and password
so that I can access my account securely

US

As user
I want to be able to reset my login details when my login fails
so that I am not blocked from accessing the site

US

As store-owner,
I want to block users that login incorrectly three-times in a row
so that I can protect my site from un-authorised access

Defining Acceptance Criteria

- Tests that defined “conditions of satisfaction” described from the perspective of the customer to determine what is required for the business and product owner to accept the user story as being “done”

US
As a customer
I can cancel a booking
So that I don’t have to pay

Acceptance Criteria
1. *A premium-alliance member can cancel same day without incurring a fee*
2. *A non-premium member incurs 25% of first day for same day cancellation*
3. *Email confirmation is sent to user to email-address used in original booking*

Invest: **T**estable

- Need clarity on the story specific “done criteria”
- Stories need to be clear, concise and complete

US

????

As a user
I want a better looking homepage
so that the colours are aesthetically pleasing

US

More Precise

As a whole-sale food supplier
I want to list different food items in different colours
so that I can quickly identify food items by type

Acceptance Criteria

Each food-type should be displayed in specific RGB colors

Detail AC

Acceptance Criteria

1. *All meats are displayed in #FF0000*
2. *All grains are displayed in #A52600*
3. *All fruits and vegetables are displayed in #807600*

Epic and User Stories

Stories in Product Backlog

Atomic user-stories pulled into a iteration

Epic Story
As a customer,
I want to create an account
so that I can shop online
Estimate 10 story points
Priority 1 (High)

US#1
As a customer,
I want to enter my personal information
so that I can create an account to shop online
Estimate 3 story points
Priority 1 (High)

US#2
As a customer,
I want to save my billing information in my account so that I don't have to enter the payment details every-time
Estimate 4 story points
Priority 1 (High)

Epic and User Stories

Stories in Product Backlog

Atomic user-stories pulled into a iteration

Epic Story
As a customer,
I want to create an account
so that I can shop online
Estimate 10 story points
Priority 1 (High)

US#1
As a customer,
I want to enter my personal information
so that I can create an account to shop online
Estimate 3 story points
Priority 1 (High)

US#2
As a customer,
I want to save my billing information in my account so that I don't have to enter the payment details every-time
Estimate 4 story points
Priority 1 (High)

Class Exercise: Which story is better?

As a repeat customer I want to access old orders so that I can rapidly purchase the same products again	As a user I want to have my previous orders stored in the database so that there is a permanent record *(user does not need to know that a database is used for persistence)*
As a driver I want to find directions to a store on Google Maps So I can get there quickly *(too detailed)*	As a driver I want to find the store with the shortest drive time So I can get there quickly
As a user I want a nice looking site so my aesthetics are satisfied *(subjective – cannot define acceptance criteria)*	As a colour blind user I want dark text & light background So that I can easily read the text
As a the marketing manager, I would like logins to time out and log off after a preset number of seconds in case users leave their computers unattended. *(It is not clear, who is benefited by this)*	As a customer who is logged in, I would like my login to time out and log off after a preset number of seconds so that I can leave my computer unattended and still have some measure of protection against unauthorized use.
As the system, I need to store customer account info and their order lists in the database. *(Not a user-story at all, it's a technical decision)*	

User-Stories (US) are not Use-Cases (UC) (1)

One obvious difference is their scope

e.g.

User Story:

> "As a recruiter, I can pay for a job posting with a credit card."

Use Case

Use Case Title: Pay for a job posting.
Primary Actor: Recruiter
Level: Actor goal
Precondition: The job information has been entered but is not viewable.
Minimal Guarantees: None
Success Guarantees: Job is posted; recruiter's credit card is changed.

Main Success Scenario

1. Recruiter submits credit card number, date, and authentication information.
2. System validates credit card.
3. System charges credit card full amount.
4. Job posting is made viewable to job seekers.
5. Recruiter is given a unique confirmation number.

Extensions

2a: The card is not a type accepted by the system.
2a1: The system notifies the user to use a different card.

User-Stories are not Use-Cases (2)

User Story:

"As a recruiter, I can pay for a job posting with a credit card."

Acceptance Criteria:

- Test with Visa, MasterCard, and American Express (pass)
- Test with Diner's Club (fail)
- Test with missing card ID numbers
- Test with expired cards
- Test with different purchase amounts (including one over the card's limit)

US and UC really differ in their level of completeness

- Some argue that there is some correlation between a user-story and use-case
- AC tests look like the extensions of main usage scenario

US and UC serve different purposes

- UC serve as a document contract, while US are placeholders for conversation

US and UC vary in longevity

- UC tend to be permanent artifacts, while US may be thrown out

Agile Requirements Engineering - Summary