IT 314: Software Engineering

User Stories & Acceptance Criteria

Saurabh Tiwari

User Stories

seek to combine the strengths
of written and verbal communication,
where possible supported by a picture.

User Story

- A user story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective.

- The user story describes the type of user, what they want and why.

- A user story helps to create a simplified description of a requirement.

- Agile reduces the risk by delivering the value of the project very early

Specification - User Stories

User Story Description

As a **[user role]** I want to **[goal]** so I can **[reason]**

For example:

- As a registered user I want to log in so I can access subscriber-only content

User Story Description

- **Who** (user role)

- **What** (goal)

- **Why** (reason)
  - gives clarity as to why a feature is useful
  - can influence how a feature should function
  - can give you ideas for other useful features that support the user's goals

User Story - Example

As a [user role] I want to [goal] so I can [reason]

As a driver, **I want** to block badly behaved passengers **so** they are never shown me again.

As a **passenger**, **I want** to link the credit card to my profile **so that** I can pay for a ride faster, easier and without cash.

As a **driver**, **I want** to add photos of my car in my profile **so that** I can attract more users.

As a **passenger**, **I want** several available drivers to be displayed **so that** I can choose the most suitable option for me.

Acceptance

An acceptance criteria is a set of conditions that are used to confirm when a Story is completed.

Acceptance Criteria

As a passenger, I want several available drivers to be displayed so that I can choose the most suitable option for me.

What acceptance criteria can be applied to this Story?

- The app shows drivers that were online within last 20 minutes and don't have an ongoing ride.

- The app shows only 5 drivers that are closest to the user.

- A user can browse profiles of these drivers, including their photos and rates.

User Story Example: Front of Card

#0001 | USER LOGIN | Fibonacci Size # 3

As a [registered user], I want to [log in], so I can [access subscriber content].

For new features, annotated wireframe. For bugs, steps to reproduce with screenshot. For non-functional stories, explain scope/standards.

User Login

Username: [_____] — User's email address. Validate format.

Password: [_____]

Remember me [ ] — Authenticate against SRS using new web service.

Store cookie if ticked and login successful.

[message] Forgot password? — Go to forgotten password page.

Display message here if not successful. (see confirmation scenarios over)

Further information is attached to this story on VSTS Product Backlog.

User Story Example: Back of Card

Confirmation

1. Success – valid user logged in and referred to home page.
   a. 'Remember me' ticked – store cookie / automatic login next time.
   b. 'Remember me' not ticked – force login next time.

2. Failure – display message:
   a) "Email address in wrong format"
   b) "Unrecognised user name, please try again"
   c) "Incorrect password, please try again"
   d) "Service unavailable, please try again"
   e) Account has expired – refer to account renewal sales page.

User Story Card

Front of Card

As a student I want to purchase a parking pass so that I can drive to school

17B
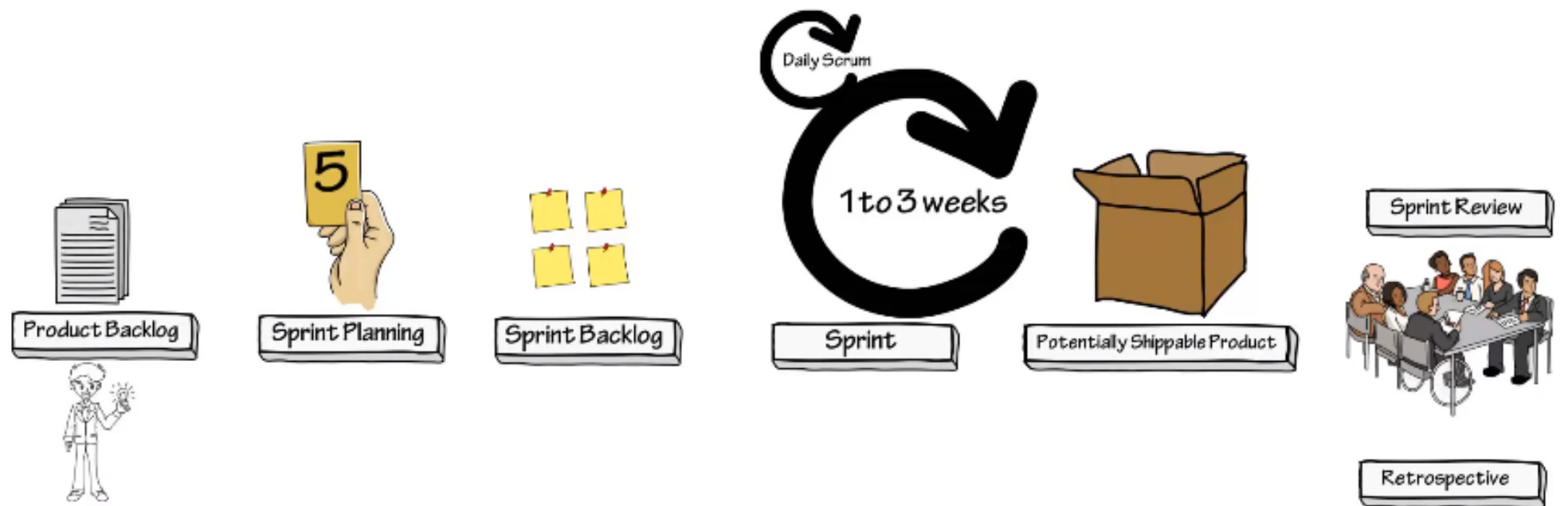
Priority: ~~XXXX~~ Should
Estimate: 4

Back of Card

Confirmations!

~~The student must pay the correct amt~~

One pass for one month is issued at a time

The student will not recieve a pass if the payment isn't sufficient

The person buying the pass must be a currently enrolled student.

The student may only buy one pass per month.

How detailed should a User Story be?

Detailed enough for the team to start work from,
and further details to be established and clarified
at the time of development.

How detailed should a User Story be?

One of the benefits of agile user stories is that they can be written at varying levels of detail.

Here is an epic agile user story example from a desktop backup product:

- As a user, I can backup my entire hard drive.

The epic above could be split into dozens (or possibly hundreds), including these two:

- As a power user, I can specify files or folders to backup based on file size, date created and date modified.
- As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved

Rules to Remember

One action per a Story. If you want to write something like "as a customer I want to browse items and add them to the cart" you'd better split it into 2 separate Stories.

Describe an intention, not a feature. For example, instead of "I want to manage my profile" create a few Stories like "I want to be able to register", "I want to upload my profile photo", "I want to link my credit card to my profile" - each Story will have a different value.

Keep it short. Users don't care what library you will use to let them browse the list of items so leave all the tech details aside.

Avoid describing UI. We've defined Stories as negotiable, remember? That's why all good User Story examples don't include any UI details. So don't try to compose any special way to implement them (we'll do this later).

Rules to Remember

If you can't answer what value this feature brings to end users and your product as well, then you're doing something wrong.

For instance, there are a few User Stories examples with a well-written value for our ongoing food ordering app project:

- As a customer, I want to get notifications when there are new hot offers so that I never miss the best deals. **[how it affects KPIs: users get notified ➡ they use the app more often ➡ retention rate grows].**
- As a restaurant manager, I want to complement dish description in the menu with a photo so that it looks more attractive to the customers. **[how it affects metrics: users are satisfied that they can see photos ➡ sales grow ➡ your revenue also grows]**

Constraints or non-functional requirements can be easily handled as user stories.
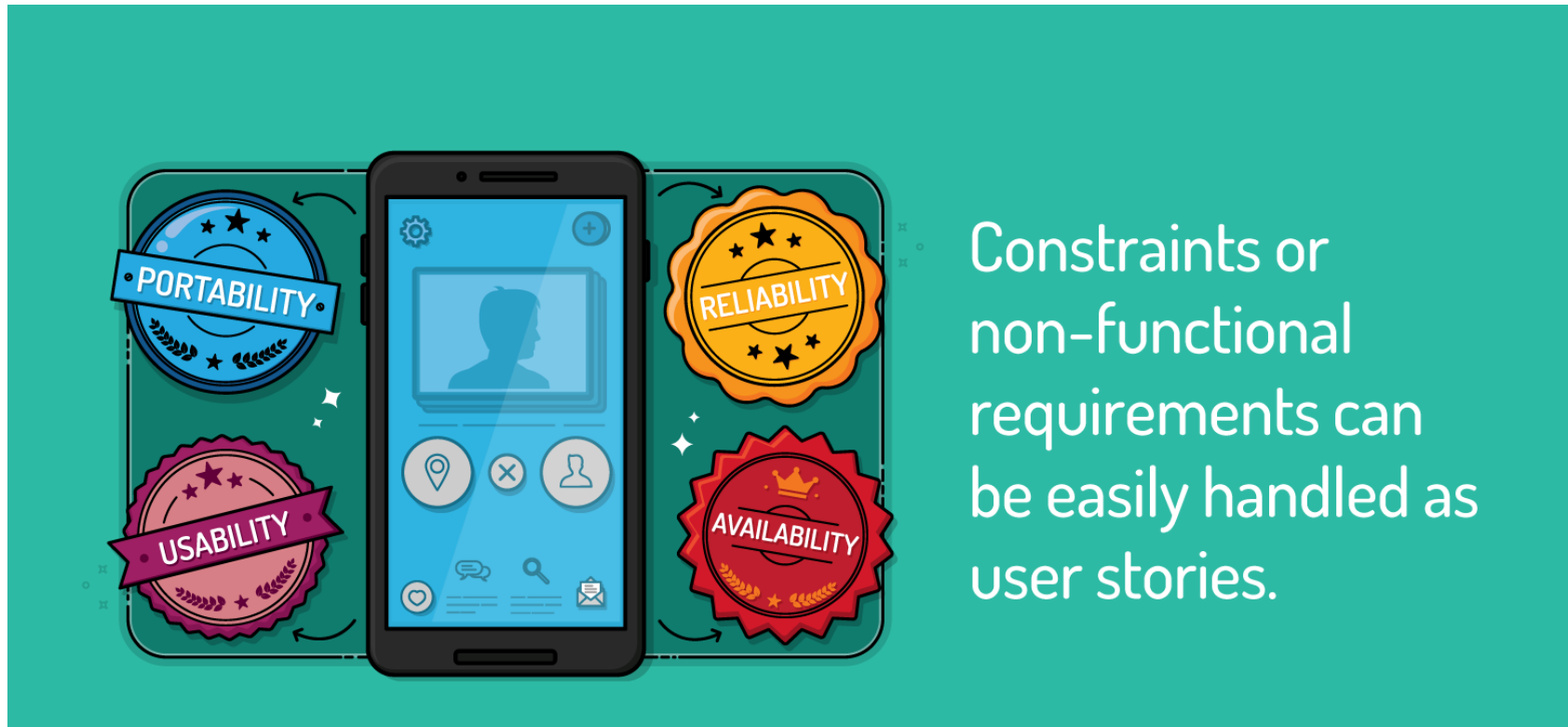
Non-functional Aspects

think of non-functional requirements as "constraints" we put on the system. When a product owner says, "this system must perform adequately with 100,000 concurrent users," the product owner is putting a constraint on the development team.

- As a customer, I want to be able to run your product on all versions of Windows from Windows 95 on.

- As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.

- As a user, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.

- As someone who speaks a Latin-based language, I might want to run your software someday.

- As a user, I want the driving directions to be the best 90 percent of the time, and reasonable 99 percent of the time.

Example

TASK: To create small user stories having all possible roles and tasks of a QA for an ecommerce website

FUNCTIONALITY: Customer can purchase items from the website.

RULES TO PONDER/ASSUMPTIONS

- All the stories that have been broken down are mostly **independent** , hence making sure that they can be parallelly developed.
- Not every minute detail has been included in the cards, encouraging developers to seek further clarification and ensuring **negotiability**.
- Stories have been written in regular business language that is equally **valuable** to both stakeholders and developers.
- It is assumed that all the user stories mentioned below can be converted into working pieces of code within the working period of 1 to 3 weeks and hence are **estimable** and **small**.
- It is also written in a way where the completion of the story and its working is quantifiable and can be tested, hence being **testable**.

Example....

Sno	User Story	Acceptance Criteria
1	Customer can create account on website. *Note :* *Customer accounts should be verified. Discuss types of verifications.*	• Test with illegal entries • Test with empty entries • Test with different ways of verification(email / phone number) • Test with fake email and phone numbers • Test with registered accounts
2	Customer should be able to log into website with a registered account.	• Test login with unregistered account. • Test login with correct and incorrect combinations of email/phone and passwords

Other user stories and their acceptance criteria...