

The Agile Modules

CSC3045 & CSC3052

User Stories

**School of Electronics, Electrical Engineering &
Computer Science**

Queen's University, Belfast

Dr Darryl Stewart

What are User Stories?

- ▶ A User Story is a form of Agile Requirements specification

“A concise, written description of a piece of functionality that will be valuable to a user (or owner) of the software.”

- ▶ They define what will be built in the project
- ▶ Formulated in one or two sentences in everyday language understood by both developers and customers/product owners
- ▶ Should be written by the customers (with guidance and refinement from developers)
- ▶ The work/effort required to implement them is estimated by the developers
- ▶ Prioritised by the customer after estimation is carried out

Benefits of User Stories



- ▶ They are quick to write compared to formalised requirements specifications
- ▶ Easier to modify and maintain when requirements change than more formal specifications
- ▶ Enable a quick response to changing requirements (main idea behind Agile methods)
- ▶ Encourage face-to-face communication
- ▶ Minimal investment of time needed until really necessary
- ▶ Allows projects to be broken up into '*sprintable*' chunks
- ▶ Gives a unit of work that developers can estimate

How to write User Stories

- ▶ The following general format is commonly used when writing user stories:

As [role], I can [feature] so that [reason]

For example:

As an unregistered user I can register so that I can log in

As a registered user I can log in so that I can see my profile page

As a registered user I can log in so that I can start a chat conversation

How to write User Stories

- ▶ Sometimes the “so that” part can seem a little redundant and it can be left off
- ▶ The format does not have to be followed strictly

For example, some alternative formats could be:

- ▶ As a **[role]**, I want **[feature]** because **[reason]**
- ▶ As a **[role]**, I want **[feature]** in order that **[reason]**
- ▶ As a **[role]**, I can **[feature]**
- ▶ As **[role]**, I can **[feature]** so that **[reason]**

- ▶ Etc.

The 3 C's

- ▶ It is recommended that User Stories consist of three Cs

Card

- Stories are traditionally written on index cards (usually 3x5 inches)
- Card is annotated with estimates, notes etc.

Conversation

- Details of the story are learned during conversations with the customer.

Confirmation

- Acceptance tests confirm the story was implemented correctly

Story Cards

The following can be captured on the card:

- ▶ Story Number – just the next consecutive number for this project
- ▶ Story Name – simple name that can be used to identify it in conversation
- ▶ Story Description – “As a ...”
- ▶ Developers Estimate – in story points
- ▶ Conversation Notes – can include a sketch of UI etc.
- ▶ Confirmation details (acceptance test details - often written on the reverse side)

#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:

Password:

☐ Remember me

[Forgot password?](#)

[message]

Store cookie if ticked and login successful.

User's email address. Validate format.

Authenticate against SRS using new web service.

Go to forgotten password page.

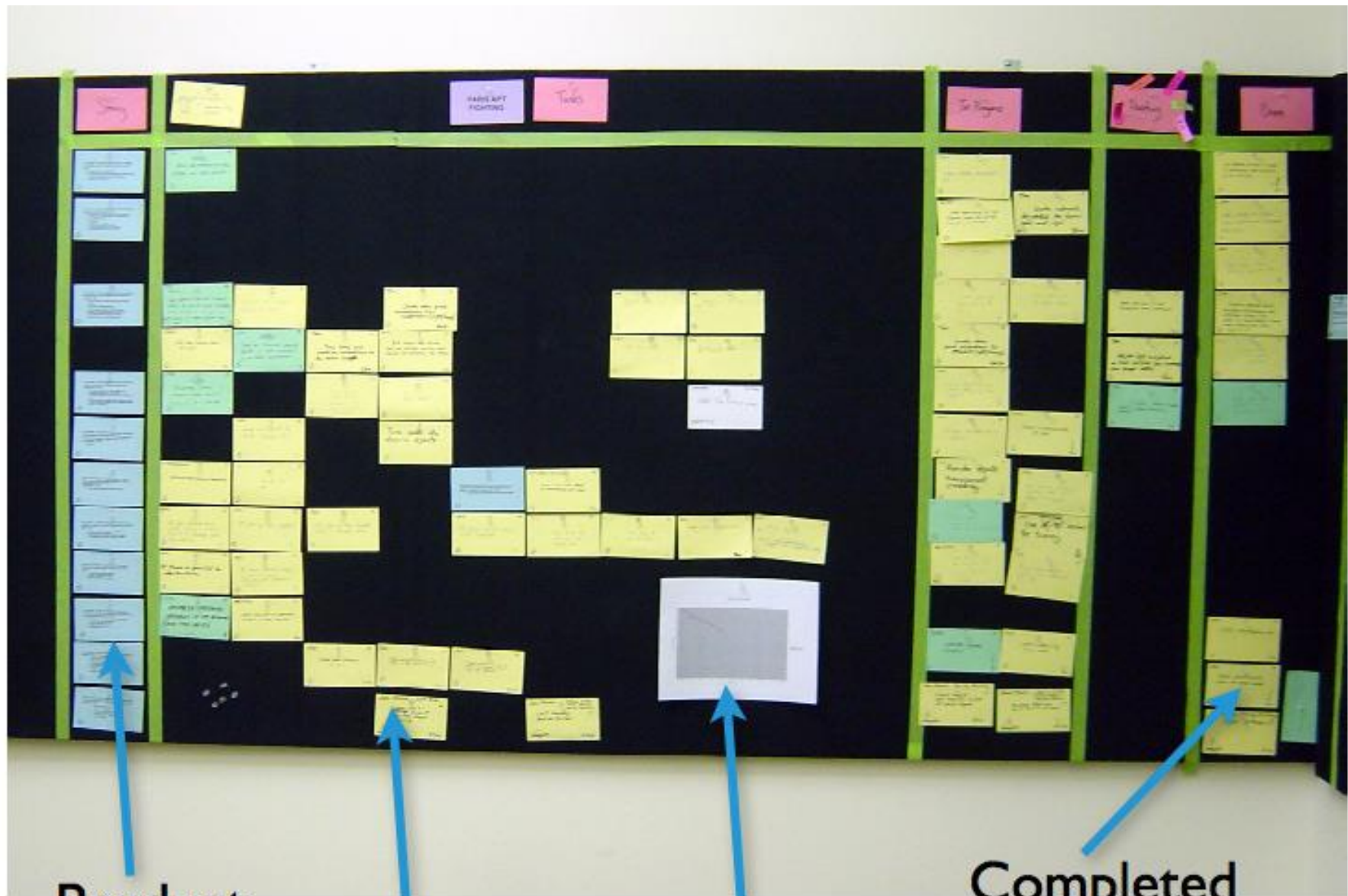
Display message here if not successful. (see confirmation scenarios over)

Advantages of Story Cards

- ▶ It keeps the story small and the description concise
- ▶ They can be displayed easily for whole team to look at and monitor
- ▶ They can be moved around easily on the display board as work is completed

Stories and task boards

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC Test the... SC Test the... SC 6



Product
backlog

Tasks
to do

Burndown
chart

Completed
tasks

Disadvantages of Story Cards

- ▶ If teams are not co-located the stories cannot be shared unless in electronic format
- ▶ If teams do not have a permanent workspace of their own it is impractical **(this is why we are not going to use story cards for your projects)**

User Roles (As a ...)

- ▶ It is unusual for there to be only one type of user
- ▶ The role may refer to a *logical* user rather than physical user
- ▶ In some cases the 'user' will appear to be a service or software component but it can usually be traced back to an actual user
- ▶ Stories should be written from different types of user perspective
- ▶ Users can vary by
 - ▶ What they use the software for
 - ▶ How they use the software
 - ▶ Familiarity with software/computers
 - ▶ Etc.

Acceptance Tests

- ▶ The story must be testable to allow developers to know when they are finished
- ▶ A common acceptance criteria template used is:

Given ... <some initial context (the givens)> ...

When ... <an event occurs> ...

Then ... <ensure some outcomes> ...

- ▶ Examples:

Given the username and password are correct

When the login button is clicked

Then the profile page will be displayed

Given the username is correct and password is incorrect

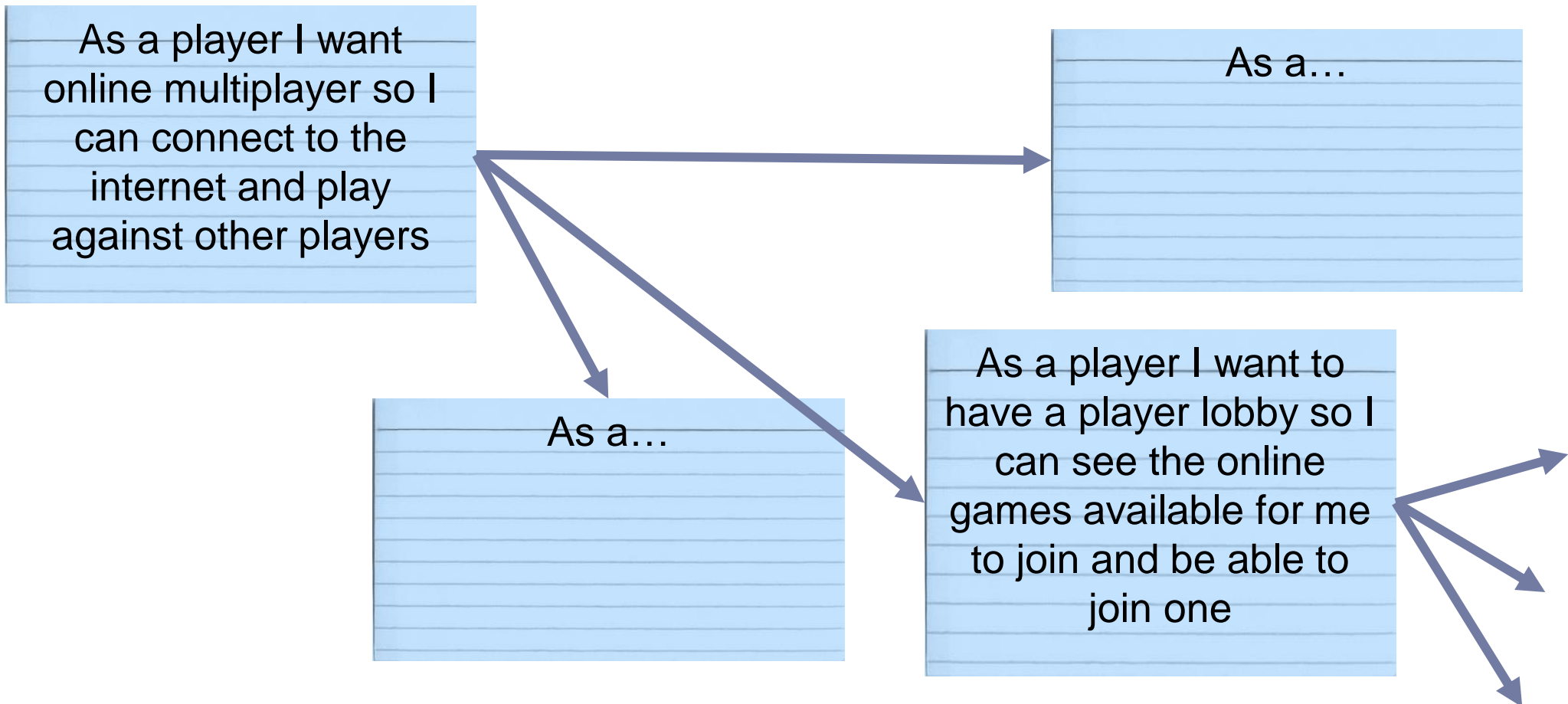
When the login button is clicked

Then the “wrong password” message should be displayed

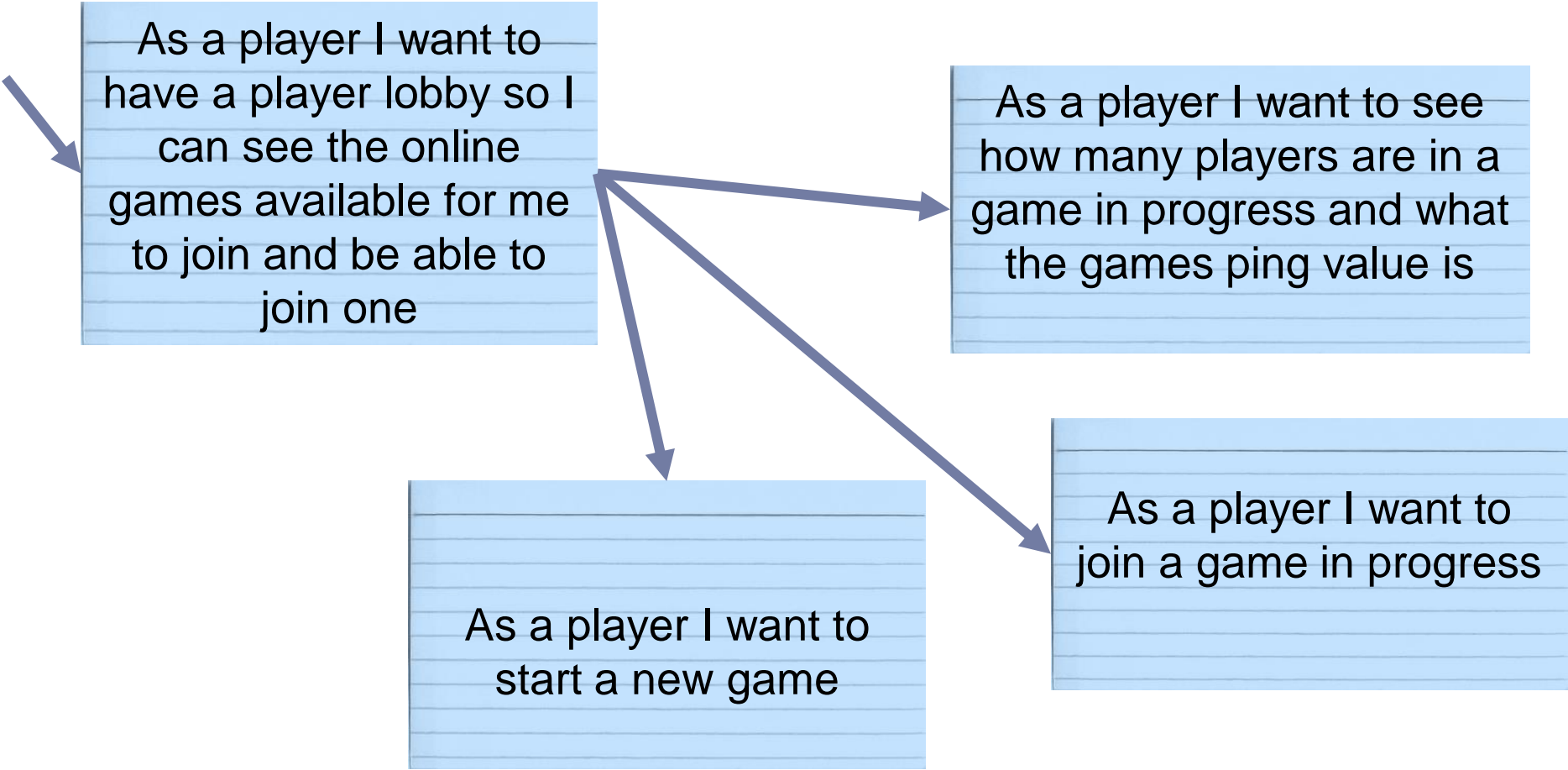
- ▶ They should be written and refined by product owner and developers following conversations

Developing a Product Backlog with User Stories

- ▶ User stories often start off being written in the form of **Epics** and then get refined into sets of smaller more manageable stories



Developing a Product Backlog with User Stories



```
graph TD; A[As a player I want to have a player lobby so I can see the online games available for me to join and be able to join one] --> B[As a player I want to see how many players are in a game in progress and what the games ping value is]; A --> C[As a player I want to start a new game]; A --> D[As a player I want to join a game in progress];
```

As a player I want to have a player lobby so I can see the online games available for me to join and be able to join one

As a player I want to see how many players are in a game in progress and what the games ping value is

As a player I want to start a new game

As a player I want to join a game in progress

- Keep refining until the stories are estimable – meaning you think they might be able to be done during **one sprint by one person**

Characteristics of good user stories

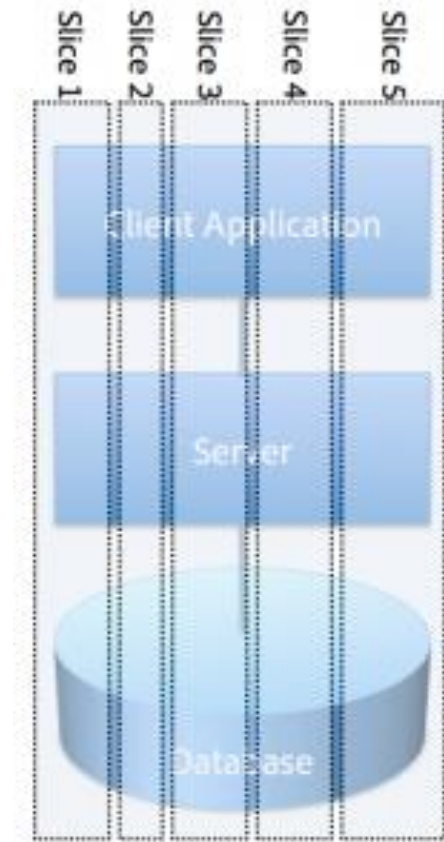
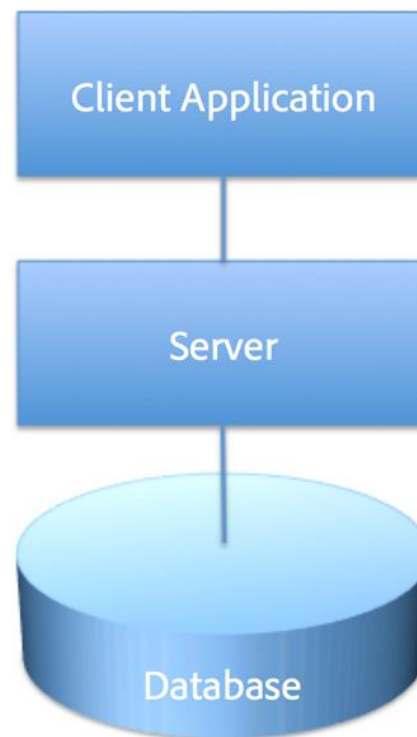
- ▶ Independent – otherwise difficult to prioritise
- ▶ Negotiable – not contracts, just reminders for discussion but the customer is the person who ‘knows’ exactly what it means, developer strives to understand it clearly
- ▶ Valuable – to the customer, otherwise why implement it?
- ▶ Estimable
 - ▶ Developers may not know enough about the domain
 - ▶ Developers may not know enough about the technology
 - ▶ The story is too large to see the ‘edges’
- ▶ Brief – stick to the recommended formats
- ▶ Testable – very easy to write un-testable requirements – take care to ensure you and product owner both agree what a successful outcome will look and act like

A Story Writing Workshop

- ▶ Includes developers, users, customers, others
- ▶ Brainstorm to generate stories
- ▶ Goal is to write as many stories as possible
- ▶ Some will be “implementation ready”
- ▶ Others will be “epics”
- ▶ No prioritization is done at this point
- ▶ Use questionnaires to gain further stories when appropriate
- ▶ Hold user interviews /observe users at work

Think Vertically

- ▶ User stories should be vertical slices of the system



Take Home Messages

- ▶ User Stories are a concise and efficient form of requirements for agile development
- ▶ Encourage verbal discussion with customer
- ▶ Are sometimes written on cards for display on 'swim lane' display boards
- ▶ They have a recommended format/template
- ▶ They are accompanied by Acceptance Tests
- ▶ They are user focused and written by product owner
- ▶ The developer is responsible for gaining understanding of stories
- ▶ Developers fill in gaps in testing where necessary and are responsible for general testing of logic and usability etc.