EBU6304 – Software Engineering

Requirements in Agile Development

- Topics
  - Requirements in Agile software development
    - User stories
    - Story cards
    - Product backlog
    - Prioritisation of stories
    - Estimating

User Stories

- In Agile process, user requirements are expressed as **user stories**.
  - helps shift the focus from writing about requirements to talking about them.
- These are written on cards, called **story cards**.
- All agile user stories include a written sentence or two and, more importantly, **a series of conversations** about the desired functionality.

User Stories

- The **customer** chooses the stories for inclusion in the next release based on their priorities and the schedule estimates.
- The **development team** break them down into implementation tasks. These tasks are the basis of schedule and cost estimates.

User Stories

- **User stories** are short, simple description of a feature
  They typically follow a simple template:

As a *<type of user>*,
I want *<some goal>*,
so that *<some reason>*.

User Stories

- **User stories** examples:

As a user,
I want to backup my entire hard drive
so that I won't lose any work.

As a Website administrator,
I want to be able to run checking reports every 15 minutes
so that I would have the chance to see when the application is approaching a threshold that will slow down the system.

?

User Stories

- **User stories** examples:

As a concerned mother,

I want to receive weather alerts for areas where my children live

so that I know when I need to worry and call them to make sure they are safe.

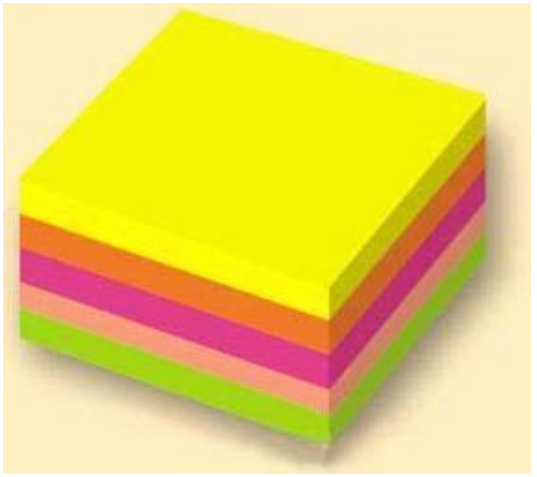
As a busy executive,

I want to be able to save favourites on my mobile weather application

so that I can choose from a finite drop-sown list to easily locate the weather in the destination I am travelling to.

Story cards

- User stories are often written on index cards or sticky notes.
- As such, they strongly shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.



Story wall

- User stories are often stored in a shoe box, and arranged on walls or tables to facilitate planning and discussion.

Project glossary

- It's important to capture the language of the business domain in a *project glossary*.
  - all terms, labels, names, etc. are clearly defined
  - everyone is using the *same* definition.
- The aim of the glossary is to define key terms and to resolve synonyms and homonyms.
- You are building a vocabulary that you can use to discuss the system with the stakeholders.

Glossary example

Glossary of Terms

Term	Abbreviation	Definition	Synonym: Allowed	Synonym: Not Allowed
Administrator		This actor represents an individual with an administrative function within the LCCS. Tasks include updating and deleting customer account information, validating and rejecting user requests, blocking and unblocking customer accounts, and backing up and archiving customer and payment data.		
Blue Badge		Offers parking concessions to people with mobility difficulties or severe problems with vision. A Blue Badge holder is eligible for a 100% Congestion Charge discount.		
Cart		Provides payment of charges in the form of shopping functionality to the customer. At the technical level, the cart is represented within the system by the Order object.		
Charge Certificate		A notice issued for non-payment of the Penalty Charge.		
Charging hours		The hours between 7:00am and 6:00pm, Monday to Friday, excluding weekends, public holidays, and the period from 25 December to 1 January inclusive.		
Congestion Charge	LCC	A fee for most vehicles travelling within those parts of London designated as the Congestion Charging Zone.	London Congestion Charge	

Epics

- User stories can be written at varying levels of detail.
- Large user stories are generally known as epics.
- Because an epic is generally too large for an agile team to complete in one iteration, it is split into multiple smaller user stories before it is worked on.

Epics break down example 1

Epic

As a user,

I want to backup my entire hard drive

so that I won't lose any work.

Break down story 1

As a power user,

I want to specify files or folders to backup based on file size, date created and date modified

So that I can manage the files better.

Break down story 2

As a user,

I want to indicate folders not to backup

So that my backup drive isn't filled up with things I don't need.

Epics break down example 2

Epic

As a busy executive,

I want to be able to save favourites on my mobile weather application

so that I can choose from a finite drop-sown list to easily locate the weather in the destination I am travelling to.

Break down story 1

As a busy executive,

I want to be able to save a search location to my list of favourites from my mobile device

So that I can reuse that search on future visits.

Break down story 2

As a busy executive,

I want to be able to name my saved searches from my mobile device

so that I know how to access each saved search in the future.

Epics break down example 2

Break down story 3

As a busy executive,

I want the name of my saved searches to default to the city name, unless I choose to manually override it

So that I can streamline saving my searches.

Break down story 4

As a busy executive,

I want my “saved favourites” option on the user interface to be presented on a mobile device near the “search location”

So that I have the option of starting a new search or using a saved one, and can minimise my keystrokes within the weather application.

Acceptance Criteria

- Acceptance Criteria is simply a high-level acceptance test that will be true after the agile user story is complete.
- Normally written on the back of the story card.
- It is a great way to ensure that a story is understood and to invite negotiation with the team about the business value that we are trying to create.

Acceptance Criteria examples

Story

As a vice president of marketing,

I want to select a holiday season to be used when reviewing the performance of past advertising campaigns

So that I can identify profitable ones.

Acceptance Criteria

- Make sure it works with major retail holidays: Christmas, Easter, Mother's Day, Father's Day, New Year's Day.
- Holiday seasons can be set from one holiday to the next (such as Christmas to New year's Day).
- Holiday seasons can be set to be a number of days prior to the holiday.

Acceptance Criteria examples

Story

As a busy executive,

I want the name of my saved searches to default to the city name, unless I choose to manually override it

So that I can streamline saving my searches.

Acceptance Criteria

- Verify that the correct city name is auto populated when the search found the location by *city* and the Save option is chosen from a mobile device.
- Verify that the correct city name is auto populated when the search found the location by *postcode* and the Save option is chosen from a mobile device.
- Verify that the default location name is saved to the database.
- Verify that the default location name is populated as a saved location in future visits.
- Verify that the location name can be manfully over written if the default location name is not desired.

Non-functional Requirements as User Stories

- Non-functional requirements (constraints) can also be handled as user stories. Examples:
  - As a customer, I want to be able to run your product on all versions of Windows from Windows 7 on, so that we can make use of older PCs.
  - 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% 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 Chinese, I want to have a Chinese version of the software.
- If you can't find a way to word the constraint, just write the constraint in whatever way feels natural.

Story card template (front)

Story name	**Story ID**
As a **I want** **So that**	
Priority **very high, high, medium, low, very low**	**Iteration number**
Date Started	**Date Finished**

The template is available to download from QMPLUS

Story card template (back)

Acceptance Criteria
Notes

The template is available to download from QMPLUS

Who write user stories?

- Anyone can write user stories.
- Over the course of a good agile project, you should expect to have user story examples written by each team member.
- Who writes a user story is far less important than who is involved in the discussions of it.

Not like this…

http://www.cs.uni.edu/~mccormic/humor.html

When are user stories written?

- User stories are written throughout the agile project.
- Usually a story-writing workshop is held near the start of the agile project.
  - Everyone on the team participates with the goal of creating a product backlog that fully describes the functionality to be added over the course of the project (or a three to six-month release cycle) within it.
- Epics will later be decomposed into smaller stories that fit more readily into a single iteration.
  - Additionally, new stories can be written and added to the product backlog at any time and by anyone.

Product backlog

- Product backlog, which is a prioritised list of the functionality to be developed in a product or service.
- User stories have emerged as the best and most popular form of product backlog items.
- Important: the written part of an agile user story ("As a …, I want …") is incomplete until the discussions about that story occur.
- It's often best to think of the written part as a pointer to the real requirement.

Product backlog (excel template)

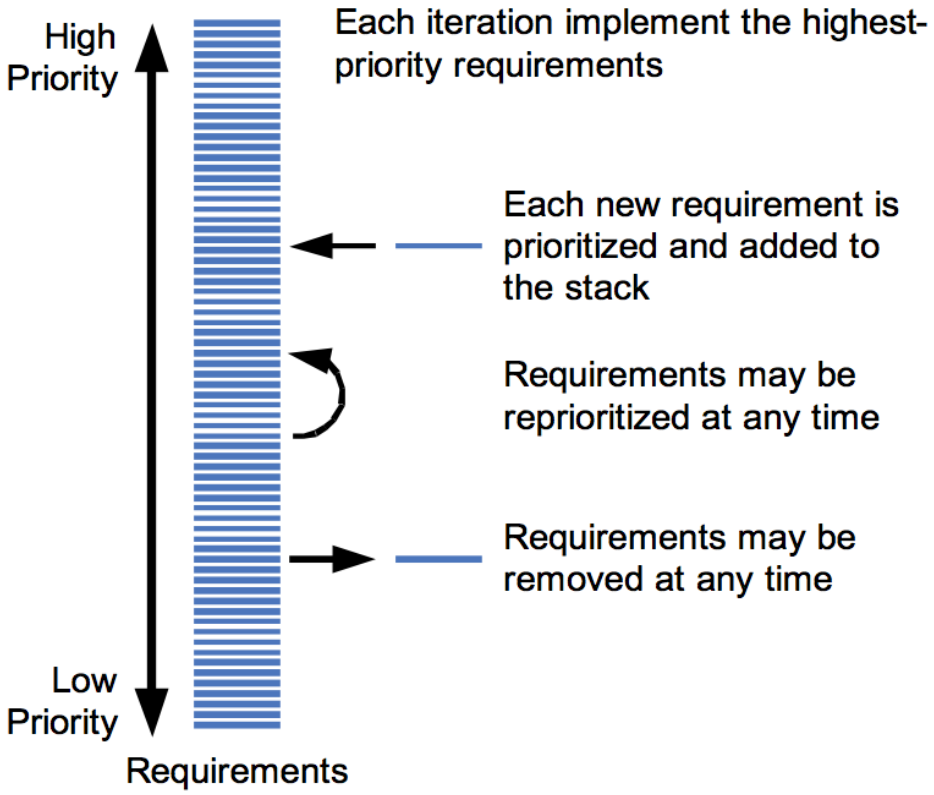
A spreadsheet is a simple good tool for writing and managing produce backlog. Other comprehensive tools are available, for example: Kanban, Trello, Jira, Axosoft…

Story ID	Story Name	Description	Priority	Iteration number	Acceptance Criteria	Notes	Date started	Date finished

The template is available to download from QMPLUS

Prioritisation of stories

- Product backlog user stories must be prioritised.
  - Based on business value
  - Value must also be supported by a positive return on investment (ROI)
  - Consideration of the risks



MoSCoW

- A method of prioritisation favoured by the DSDM (dynamic systems development method). Elements are:
  - Must have: features must be implemented, and if not, the system would not work.
  - Should have: features are important but can be omitted if time or resources constraints appear.
  - Could have: features enhance the system with greater functionality, but the timelines of their delivery is not critical.
  - Want to have: features serve only a limited group of users and do not drive the same amount of business value as the preceding items.

MoSCoW example

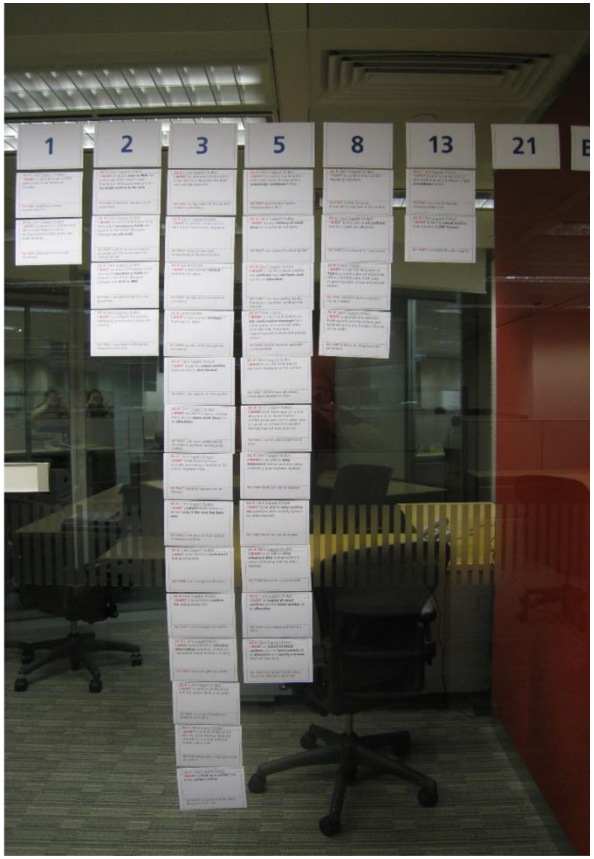
- Payment processing story:
  - Must have: ability to accept Visa and MasterCard.
  - Should have: add American Express.
  - Could have: add PayPal.
  - Want to have: add gift cards.

Estimating

- Estimates: how long the work is likely to take
- Some methods include:
  - Level of Effort or T-shirt sizing: small, medium, large, extra large…
  - Ideal Time: under ideal circumstance
  - Hours: actual clock hours
  - Story points: an arbitrary measure that allows the teams to understand the size of the effort

Estimating story points

- Story points:
  - Fibonacci Sequence: 1,2,3,5,8,13,21…
  - As the points get higher, the degree of uncertainty is increasing.
  - Abstract and negotiable
  - Everyone on the team agrees to the size of a story point, using relative sizing
    - E.g. adding a field to the database, agreed as a 5 story points



Good user story?

- INVEST
  - **I**ndependent
  - **N**egotiable
  - **V**aluable
  - **E**stimatable
  - **S**mall
  - **T**estable

Summary

- Requirements in Agile software development
  - User stories/Story cards
  - Product backlog
  - Prioritisation of stories
  - Estimating

References

- Chapter 4 – “Software Engineering” textbook by Ian Sommerville
- Chapters 2+3 – “Head First Object Oriented Analysis &Design” textbook by Brett McLaughlin et al
- User stories by Mike Cohn http://www.mountaingoatsoftware.com/agile/
- Introduction to Agile by Sondra Ashmore