

Lecture 2

Project management with Agile



UNIVERSITY OF
PLYMOUTH



Project Management

- Self-organising teams – no hierarchy
- Not expected to deliver it all
- Produce evidence in GitHub Repo
- Using an Agile approach



UNIVERSITY OF
PLYMOUTH

What is Agile?



UNIVERSITY OF
PLYMOUTH

Scrum as a form of Agile



Risks

- Be aware of what could go wrong
- “An uncertain event or condition that, if it occurs, has a positive or negative effect on a project’s objectives” PM-BOK
- Possible **future** problems, NOT current ones
- Boehm identifies top 10 development risks



Boehm's top 10 development risks

Risk	Risk reduction techniques
Personnel shortfalls	Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel
Unrealistic time and cost estimates	Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods
Developing the wrong software functions	Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals
Developing the wrong user interface	Prototyping; task analysis; user involvement



Boehm's top ten risk - continued

Gold plating	Requirements scrubbing, prototyping, design to cost
Late changes to requirements	Change control, incremental development
Shortfalls in externally supplied components	Benchmarking, inspections, formal specifications, contractual agreements, quality controls
Shortfalls in externally performed tasks	Quality assurance procedures, competitive design etc
Real time performance problems	Simulation, prototyping, tuning
Development technically too difficult	Technical analysis, cost-benefit analysis, prototyping , training



Risk : Agile

- Risk management is built into agile approach
- Greater all-round visibility for who is doing what reduces the risk
- Communication is essential
 - Leaving information out is as bad as misleading information
- Avoid large work items (see later)
 - The larger the requirements are, the harder they are to understand. So break them down into manageable chunks.
- Keep talking in the team!



Agile Principles

- PM is about applying agile principles across whole project
- Agile is about delivering value to the user in short increments
 - Minimum Viable Product (MVP)
- Not about waiting for a super system that does it all
- NOT about
 - Doing database first
 - Doing layout, styling, fonts and colours first
- Product Backlogs, Sprints, Reviews and Retrospectives are all part of these principles.



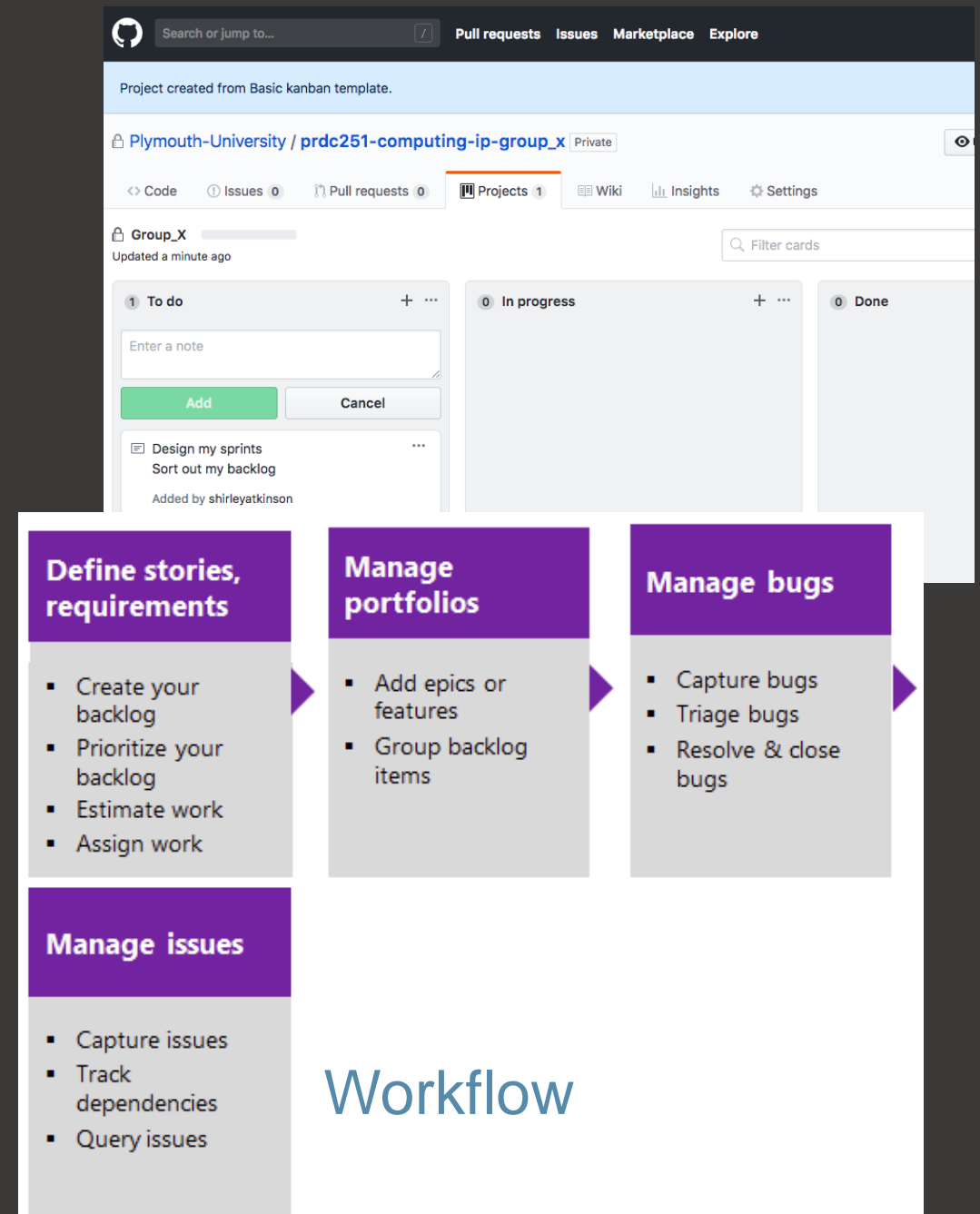
Work phases (aka Sprints)

- Work in 2 week **sprints** for delivery
 - 12 hours each per week
 - (Team number x 12 hours) x 2 weeks = x
- Start with highest value items first
 - MVP = what the client wants the most
- Present to your clients to ensure you are building the right thing
- As part of your documentation you should be showing how you put your work into sprints
- Sprints should be reviewed



Product Backlog

- Backlog is a list of items that need doing to create the system
- It is a roadmap, a plan for what you are going to do in the project
- A repository of everything that needs doing



Product Backlog

- Will need to ensure it is complete by the hand in date
 - Backlog will not be complete at beginning
 - Whole point of agile is that items are added to the backlog as you go along
- Refine the backlog with your client
 - Which bits are priorities – ask them



Epics

- If you cannot see how to immediately code up your user story, it is likely to an Epic
- Epic is a large value statement rather than a smaller user story
- They are NOT the same as User Stories



Decomposing Epics

- There is a way of breaking Epics down
- Flow – how do you flow through achieving the story
- Effort – how much time will it take for a developer to do it
- Entry – how does the data get in there
- Data – what happens to the data
- Business rules – what are the goals for the software
- Alternatives – can you break this down into alternative criteria
- Complexity – is there more value here
- Knowledge – does your team need more knowledge



Details & Estimates

- Agile uses relative sizing
- It is not meant to be precise but it gives a starting point
- Planning poker is a card game that helps get the best story estimates
 - It allows a conversation
 - Uses story points – not hours or days

Each backlog item must have an estimate



UNIVERSITY OF
PLYMOUTH

User stories

- SHOULD NOT focus on how to deliver the request but on what happens
- Customer value is focus of the story
 - It is in their language, not the technical
- Stories must be able to be estimated, or they are too big
 - Break it down
- User story ought to be delivered within the 2 week sprint – so it should be small
- User story should be testable – objectively
 - Testing is NOT – I ran it and it looked ok
 - Review theories on Unit testing – Test-Driven Development, Behavior-Driven Development



Common errors

- Each year students are told, focus on user stories, user stories are not “let’s create database first, create the interface next” etc
- Each year students ignore that advice
 - Each year the students wonder why they didn’t get good marks
 - Why they hit problems because they didn’t have anything working ahead of time
- They thought they could use Easter to catch up
 - Because they didn’t do anything in the 8 weeks before hand
 - 8 weeks work does not fit into 3!



Reviews

- At end of each sprint, you should review what you have done
 - As per Project Phases document
- It is a way of showing to your client progress made and to be able to gain feedback
- Look for resources on how to do a Sprint Review and demo – these are applied during seminar



Retrospective

- Done at end of a Sprint review
 - Is more about team
 - How did team work together?
 - What could be improved?
 - Actions to take forward
 - Record these and upload to repository
- [Agile Retrospectives Done Right - Agile Coach \(2019\) - YouTube](#)



Project closure

- Essentially a retrospective for the whole project
- What will you take forward?
- What did you deliver?



UNIVERSITY OF
PLYMOUTH

A wide-angle photograph of the University of Plymouth campus at sunset. The sky is a vibrant mix of purple, pink, and orange. In the background, modern university buildings with lit windows stand against the colorful sky. To the left, a historic stone building with a tall, pointed spire is visible. In the foreground, a dark body of water reflects the sky and buildings, with a small fountain spraying water on the right side. Bare trees and a fence line the left and middle ground.

Questions?

Short break



UNIVERSITY OF
PLYMOUTH