**Software Development:**

*This document is to go through each week and explain what it is, how it is done and an example.*

**Week 1: Introduction**

Lecture:

It was just an introduction to the unit, so no notes were needed to be made.

Lab:

Stakeholders are people or groups of people who have some specific interest in the system. Identifying stakeholders can be a tricky business, and missing stakeholders out early in the project can doom it to eventual failure.

Categories of system stakeholders:

* They might use the system.
* They might be paying for the system.
* They might regulate the system’s use.
* They might be affected by the system.

Read through the case study, identify the stakeholders, and create a list of them.

**Week 2: Requirements**

Lecture:

Requirements mainly come from the stakeholders.

Types of requirements:

* Functional Requirements
  + Specifies what the system needs to do.
  + Most common and most obvious.
  + E.g., display student’s classes in a timetable.
* Non-Functional Requirements
  + Difficult to define so it is difficult to determine if it is met.
  + E.g., reliability, speed.

Common techniques for finding out what the stakeholder want:

* Brainstorming
  + It is quicker but not interacting with stakeholder so won’t get an accurate picture of what is required.
* Interviews
  + Can ask questions for more detail which could then create a new requirement but is time consuming.
* Focus Groups or Workshops
* Reviewing Documents
  + Can see what data is being stored/used.
* Job Shadowing
  + Can find a requirement that is not obvious to others.
* Prototyping
* Surveys
  + Wide response but is limited as only get vague answers.

Fit Criteria:

Difficult to tell with non-functional requirements.

A fit Criterion defines a requirement’s explicit satisfaction threshold. E.g., request to web service must be completed in under 75ms.

Requirements may change due to:

* Stakeholder wishes or priorities may change.
* Developers might not have understood the system’s domain initially.
* Stakeholders might not have understood the developer’s plan.
* Things previously thought feasible/possible might turn out otherwise.

Requirements tracing:

This is an entire sub-discipline devoted to figuring out how to handle changes to the requirement.

* Making sure that the specification remains consistent.
* Identifying other requirements affected by a change.
* Identifying other stakeholders affected by a change.

Requirement Prioritisation:

A common method for this is MoSCow:

1. Must have
2. Should have
3. Could have
4. Won’t have

Lab:

Go through the list of stakeholders and identify which method/s is best to find their requirements. E.g., interviewing the stakeholder. Make a notes of which method/s are chosen for each.

Due to inability to converse with the stakeholders, the main method of finding requirements is brainstorming.

Go through each stakeholder and find their likely requirements. Make a note of this and if it is functional or non-functional.

For each requirement, a testing method is needed to check it fits the criteria. For functional this is usually an action performed and checking the behaviour of the system. For non-functional, may need to devise a measure, metric, or experiment.

For each requirement, order them in prioritisation of MoSCoW.