CITS4401/3301 Software Requirements and Design
Week 2: Eliciting requirements
Date: 15 March 2023

Reading: Pressman Chapter 5 (Understanding Requirements)

Q1. **Core concepts:** In your own words give a short explanation of each of the following concepts. Also, provide an example for each concept.
(a) Functional requirement
(b) Non-functional requirement
(c) Project scope
(d) Software scope
(e) Requirements elicitation

Solution:
(a) Functional requirement: a function that the system must be able to perform. Generally, functinal requirements describe system behaviour under specific conditions. Example: the system must allow administrators to create and change user's registration information.
(b) Non-functional requirement: a constraint on the system. It defines how the system should perform. Example: the system must protect users' personal data from unauthorised access.
(c) Project scope: The boundaries of the system to be built including the (business or systems) context it will operate in; what information it should produce and its function and performance. Project scope also involves the time, cost and resources available to do the project. Example: the UWA student LMS will interface to the Pheme system for authenticating users and managing their personal data.
(d) Software scope: What functions and properties the software will and will not have. Where project scope includes time and resources, software scope is focussed on the functional and non-functional requirements and the boundaries for interaction with other systems.
(e) Requirements elicitation: The process and result of discovering the requirements for a system. Example: interviews with stakeholders are one technique for RE.

Q2. Consider the following requirements for a mobile phone app for storing and managing photos.
R1: The system must be useable
R2: The system must provide visual feedback to the user within one second of issuing a command
R3: The availability of the system must be above 95 percent
R4: The user interface to the new system should be similar enough to the old system that users familiar with the old system can be easily trained to use the new system.

Answer the following questions about these requirements:
(a) Using the classifications for requirements from the lectures, assign a category to each of these requirements.
(b) Which of these requirements are verifiable? If not, suggest how it could be rewritten to be verifiable.

(c) Suggest at least one (new) requirement for this system which would conflict with the requirements above.
(d) Suggest a way to resolve that conflict.

Solution:
(a) All are non-functional requirements. R1 Usability, R2 performance ,R3 reliability (availability), R4 useability (training)
(b) Non of them are easily verifiable. Proposing a test environment for establishing these requirements would be one way to make them more verifiable. Take care: just adding detail to a requirement (eg 95% over 10 days of use) does not necessarily make it more verifiable.
(c) R5: The system must be attractive and intuitive so as to attract new users. This might conflict with R4 which wants a new version of an existing one. Why?
(d) Resolution: Find out why the clients want R4. Maybe it is low priority? Maybe it captures some other property such as "intuitive to use" instead. Don't just make an assumption and change the requirements - the conflict is an indication that the requirements need some more investigation. It is your job to work with the stakeholders to uncover and resolve such issues.

Q3. The remaining questions relate to a project for the University of Titipu. The University have decided to develop a web-based maintenance tracking and repair system (TUTRS). [Source: based on Pressman Ch 8 Question 8.3]

"Staff and students can log onto a website and report the location and severity of broken equipment or other maintenance issues. As issues are reported they are logged within a University maintenance database and assigned an identifying number, location, problem description and photograph, and repair urgency. Work order data are associated with each reported issue and include the problem, repair crew, number of people, equipment assigned, hours applied, status of repair, material used in the repair, and repair cost (computed from hours, number of people, material and equipment). Finally a damage file is created to hold information about the reported damage from the maintenance problem, including the reporting person, their contact details, type of damage and $ amount of damage. TUTRS is an online system. All queries are to be made interactively."

Stakeholders Identify three or four stakeholders for the TUTRS system. For each stakeholder you identify, explain their role in the system.

Solution: Incident reporters: Staff and students who will report situations that need attention from the maintenance staff. Scheduler: Responsible for prioritising, planning and assigning staff to situations. Likely responsible for costs too because they manage the maintenance budget too. Maintenance staff: Responsible for carrying out maintenance tasks Titipu HR/Admin: Owner of information about registered reporters and maintenance staff to be used by the TUTRS system. These stakeholders are responsible for e.g. passwords, login details, information about availability, skill sets and pay rates. They are also responsible for the security of this information.

Requirements Elicitation Methods Identify at least two different ways that you could elicit requirements for the TUTRS system. Justify why you have chosen that method and explain how you would carry out the elicitation process.

Solution: Interviews: with stakeholders. Identify their goals, and the main strengths and weaknesses of the existing reporting and repair system.

Observation: Observe the process that is currently used when maintenance is required.

Market research: Find and test existing products on the market to see if they can deliver some or all of the required functionality. Note this step is often not done in SW projects, resulting in expensive re-invention of the wheel!

Prototype: Make a simple prototype and then have run throughs with different stakeholders to check whether the prototype meets their needs. This is probably not so useful until you have a more detailed version of the required functionality. However, it could be a good solution if you already have a software system with similar functionality that you are going to modify for this application.

Requirements Elicitation In your groups, use role play to practice requirements elicitation for the TUTRS system. Use each of the elicitation approaches you identified in the previous question. For each requirement give a rationale: why is it needed and maybe its priority: how important it is.

Solution: See class discussions. Here are some ideas.
(a) Functional: A staff member or student enters details of a maintenance request into the system. Rationale: need a way to get info from the real world into the system so the maintenance can be managed. Priority: Must have.

(b)Functional: A team of maintenance people is assigned to work on a reported problem. Rationale: the main purpose of the system is to manage maintenance repairs to make sure they are done as quickly as possible and prioritised. Priority: Must have.

(c) Non-functional: Only authorised maintenance staff are able to be allocated to a task. Rationale: TUTRS needs HR information of the University in order to plan repairs. There may be safety or skills requirements for the people who can be chosen to do particular tasks. Priority: Must have (maybe should have).

(d) Non-functional: Reporting a maintenance problem should be quick and convenient (e.g. take no more time than the existing method). Rationale: If the system is not convenient to use then people will simply not report maintenance problems and that is a health and safety problem for the university. Priority: Must have.