Electronics and
Computer Science

University of
Southampton

Capturing Requirements

COMP6226: Software Modelling Tools and Techniques for Critical Systems

Dr A. Rezazadeh (Reza)
Email: ra3@ecs.soton.ac.uk or ar4k06@soton.ac.uk

October 24

Overview

- Software Development - From Where to Start?
- Problem Statement/Proposal
- From Problem Statement to the Requirements
- Types of Requirements
- The Requirements Engineering Process
- Attributes of Good Requirements

University of Southampton

Objectives

- Describe the basic concepts of software requirements engineering.
- Select and apply the appropriate requirements elicitation techniques to identify your system's requirements.
- Effectively analyse user/system requirements.
- Create a requirements specification to communicate the system requirements.
- Utilise various requirements validation techniques to critically evaluate your requirements to identify defects.

How is a new Software Project Initiated?

- Problem statement – what is it?
  - A problem statement is a clear and concise description of the issue(s) that need(s) to be addressed by a problem-solving team.
  - Issues- one or two sentences that describe the problems/specific issues that we are facing.
  - Vision/Goals - what does the world look like if we solve the problem?
  - Method/Objectives - the process that will get followed to solve the problem.
  - Scope – boundaries of the proposed solution/new system

University of Southampton

Problem statement – An Example

- Currently, passengers of airport x that have connecting flights in different terminals, are facing a chaotic situation leading to delay and missing flights due to the absence of an effective transport system between different terminals.
- The aim of this project is to build an efficient driverless train system, controlled by software, to minimise delays and achieve high level of safety by automatic monitoring of platforms, passenger boarding and departure.
- As this is a safety-critical system the control software should be validated/verified to a very high level of rigour.

How to Formulate Problem Statements?

- The 5 'W's - Who, What, Where, When and Why - is a great tool that helps
- ***Who*** - Who does the problem affect? Specific groups, organisations, customers, etc.
- ***What*** - What is the issue?
  - What are the boundaries of the problem, e.g. organisational, workflow, geographic, customer, segments, etc.
  - What is the impact of the issue
  - What will happen when it is fixed?
  - What would happen if we didn’t solve the problem?

How to Formulate Problem Statements? – Cont.

- ***When*** - When does the issue occur?
  - When does it need to be fixed?
- ***Where*** - Where is the issue occurring? Only in certain locations, processes, products, etc.
- ***Why*** - Why is it important that we fix the problem?
  - What impact does it have on the business or customer?
  - What impact does it have on all stakeholders, e.g., employees, suppliers, customers, shareholders, etc.
  - You might try to suggest a potential solution.

Example: writing a problem statement

- Write a proposal for a dental surgery (clinic) that has a simple website (with out-dated technology)
  - not always working properly
  - not supporting useful functionality such as patient booking, news dissemination, and user feedback.

An Example of Problem Statements– Dental Surgery

Problem statement: A dental clinic has a simple website with out-dated technology. The current website has limited usability, the database is not updated regularly, and the news/events details are not always synchronised to the server. These problems cause a high number of patient complaints, in the form of frequent phone calls and repetitive emails to staff, taking the precious time of both, the staff and patients.

Problem Statements– Dental Surgery (Cont.)

- **Goals:** The goals of this project is to build a new website with enhanced usability (using PHP and MySQL), by providing the following facilities:
  - an online booking system to improve the user experience for the patients, which send confirmation emails and/or SMS messages to remind patients of their appointment to prevent them from forgetting their appointments.
  - Putting details of the latest news and descriptions of provided services
  - A short bio of doctors and specialist nurses on the website, to provide patients with the ability to choose whom they want to treat their problems.

Problem Statements– Dental Surgery (Cont.)

- Patients can rate the doctors and specialist nurses after their treatment
- Both the doctors and specialist nurses have access to the system to view their bookings.
- System admin role is defined so that the clinic manager can add categories, such as doctors or specialist nurses, adding new staff name when they join the clinic and remove their name when they leave the clinic.

- **Scope:** Aspects like treatment records and payments are not handled by this system. However, the system may share information with these neighbouring systems, such as treatment system or payments system.

Identifying the Requirements from Problem Statement

- Requirements identification is the first step of any software development project.
  - Until the requirements of a client have been clearly identified, and verified, no other task (design, coding, testing) could begin.
- Requirements: Sommerville defines "requirement" [1] as a specification of what should be implemented.
  - Requirements specify how the target system should behave. It specifies what to do, but not how to do.
- A requirement is a stakeholder's expression of a need, wish, or desire regarding the system being built.

Requirements engineering

- Requirements engineering refers to the process of understanding what a customer expects from the system to be developed, and to document them in a standard and easily readable and understandable format.
  - This documentation will serve as reference for the subsequent design, implementation and verification of the system.
  - It is necessary and important that before we start planning, design and implementation of the software system for our client, we are clear about its requirements.

Product versus process requirements

Requirements can be said to relate to two fields:

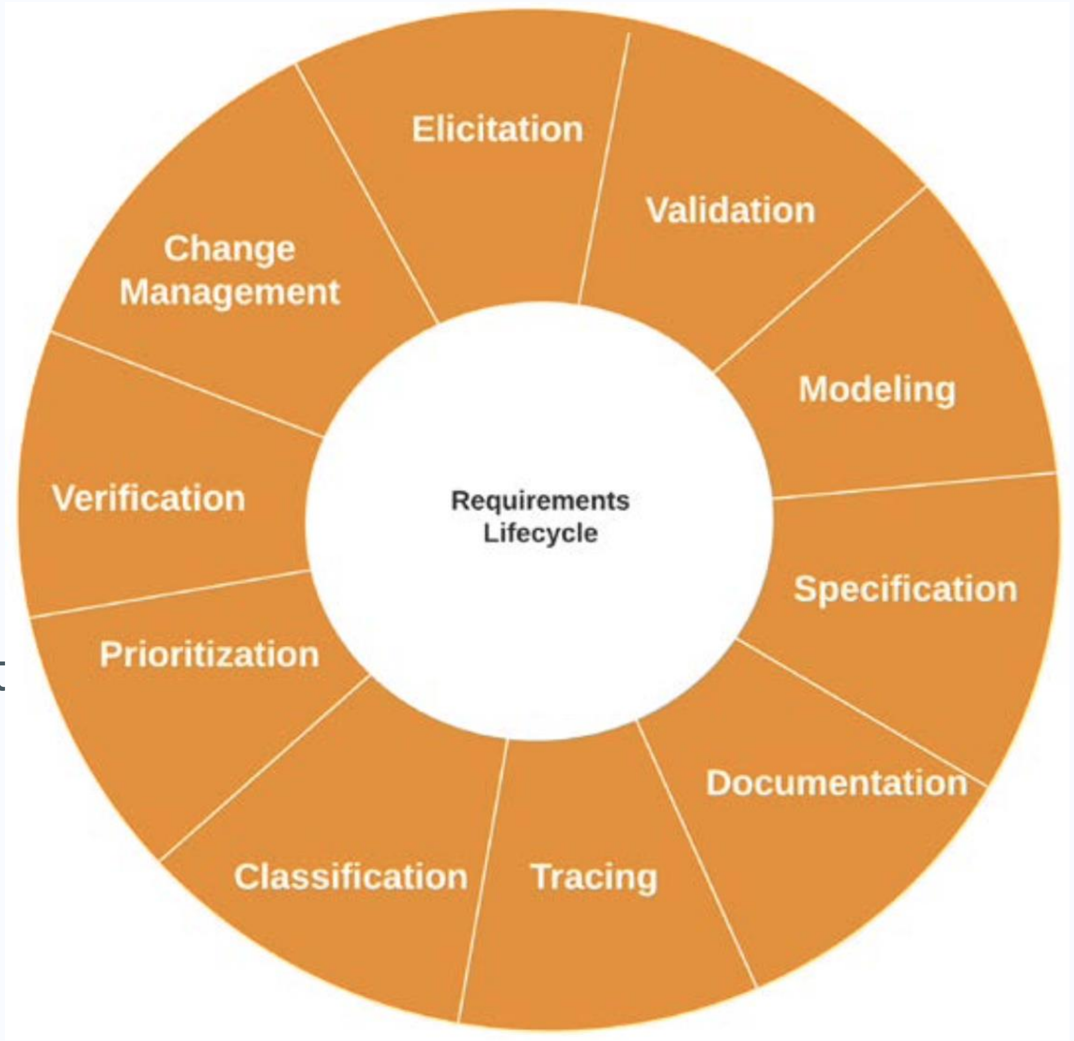
- Product requirements prescribe properties of a system or product.
- Process requirements prescribe activities to be performed by the developing organization. For instance, process requirements could specify the methodologies that must be followed, and constraints that the organization must obey.
- Product and process requirements are closely linked;

University of Southampton

The requirements life cycle

Requirements management, like many other software engineering disciplines, consists of several distinct stages:

- Elicitation: The collection of requirements, which can be done in many different ways.
- Validation: Ensuring that the requirements fulfill a realistic and useful business need.
- Modeling: Having a structured and consistent way to describe and store requirements.
- Specification: This is where we translate the requirements into concrete and clear system behaviors.



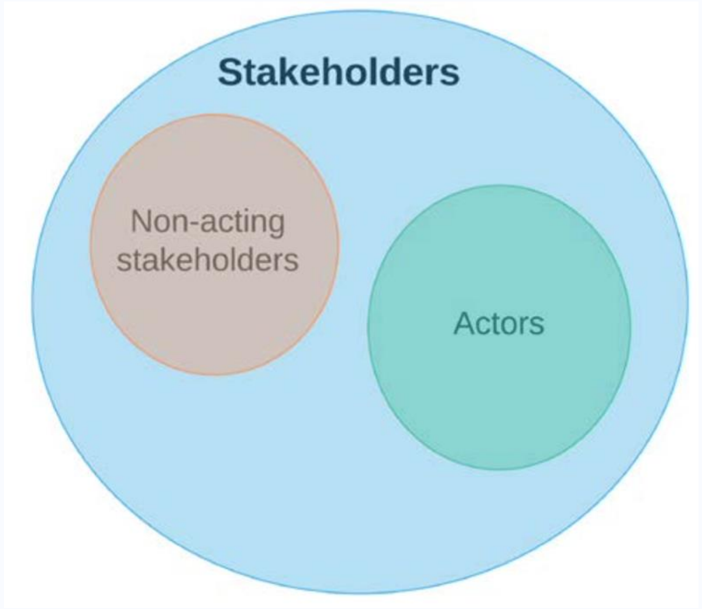
The requirements life cycle – Cont.

- **Documentation**: It should go without saying that requirements must be well described, detailed, and documented.
- **Tracing**: Knowing which requirements affect system behaviours and which behaviours are necessitated by specific requirements.
- **Classification**: Requirements can be classified according to the area of the system they affect, their complexity level, their risk level, and many other factors.
- **Prioritization**: This is usually done according to impact and precedence.
- **Verification**: Ensuring that our system functions in a way that fulfils the requirements.
- **Change Management**: Dealing with changes to requirements.

University of Southampton

Identifying stakeholders

- Requirements elicitation cannot be carried out without identifying stakeholders in the first palace.
- A stakeholder is someone, or something, that derives value, benefits from, or influences our system.
- There is a stakeholder sub-set that we call them Actors
  - These are stakeholders who interact with our system, either directly or indirectly.
- Another sub-set of stakeholders we call non-acting stakeholders.
  - These would usually be people like directors, business owners, enterprise architects, or senior managers.



Actors Classification

Actors may be further divided into two categories:

- **Primary actors** interact with the system directly through a **UI** or **API** in order to achieve a specific goal.
  - Primary actors always have a goal that they expect to reach by using the system.
- **Secondary actors** are actors that the system needs to be assisted by. This makes it possible to achieve the primary actor's goal.
  - Secondary actors may or may not have goals that they need to reach by using the system.

Actors – An Example

- Let's take the example of an online loan system.
- The consumer using the system expects to get a loan (specific goal).
- The system will provide the loan only if it receives a positive response from a credit reference agency, such as **Experian** or **Equifax**.
- The consumer is a primary actor as they interact directly with the system in order to accomplish a specific goal.
- The credit reference agency is a secondary actor as the system relies on it in order to achieve the primary actor's goal.

Identifying goals

- Our stakeholders will have their own goals and objectives for using or sponsoring the system we are building.
  - Understanding these goals is critical to the successful elicitation, analysis, and modeling of requirements.
- Stakeholder goals usually fall within two categories, depending on whether the stakeholder is a non-acting stakeholder or an actor.
- Actors have goals related to the domain in which they operate, for example, retail, education, and banking.
- Non-acting stakeholders are usually our system's business sponsors, such as directors and executives.
  - As such, their goals are business-related; they mainly sponsor our system because they want to expand or protect their business.

Domain goals

- Domain goals drive our system actors to use our system in order to increase their value within their domain.
  - For instance, if we're building a learning management system, many of its users would be aspiring to use our system to increase their knowledge or skills.
  - If we're building a marketplace system, most of our users would want to either make money by selling items or to acquire needed items by buying them.

Some examples of domain goals

- A student wants to get their coursework assessed by a teacher.
- A teacher wants to pass their knowledge on to a student.
- A teacher wants to ascertain whether the student has gained adequate knowledge.
- A seller wants to make a profit on an item they sell.
- A seller wants to get rid of an unwanted item.
- An actor may have multiple domain goals. In addition, value may be added to multiple actors via the same goals.

Business goals

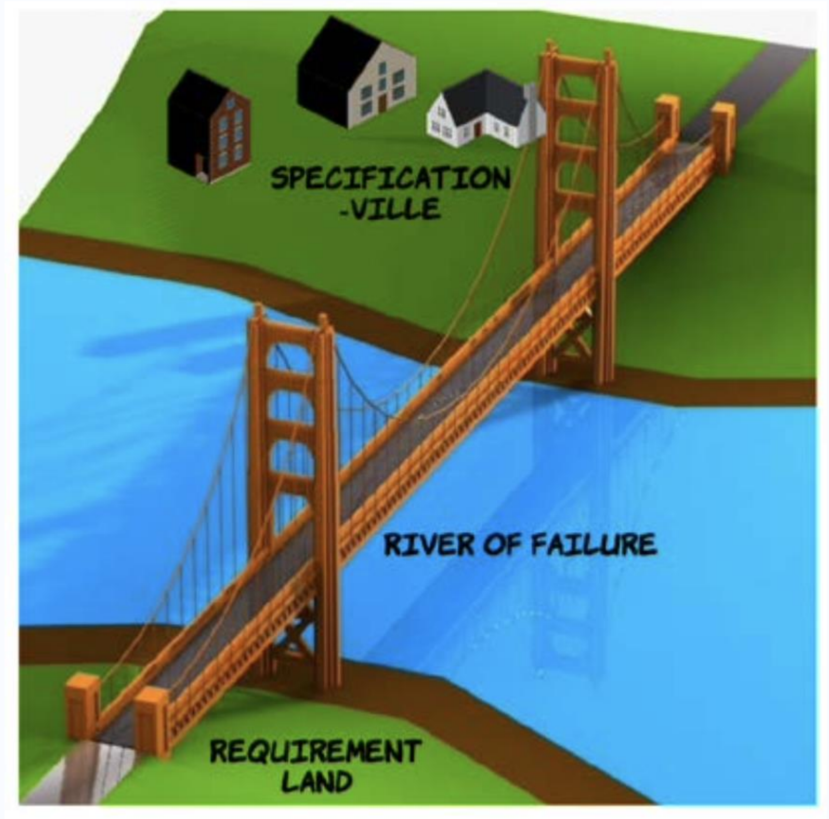
- These are goals that are mainly set by non-acting stakeholders, such as directors, VPs, and senior managers.
- They may approve or support the allocation of resources, and they may also champion the project to other members of senior management within the business.
- A business goal's motivation and expected outcome should ultimately fall under one of these categories :
  - Increasing revenue - Protecting revenue
  - Avoiding future costs - Reducing costs

Goals, requirements, and specifications

- Imagine, for a moment, that you're a visitor to the great city of Cardiff in Wales, United Kingdom, and you want to sample some of the famous Welsh cakes you have heard so much about – So, you take out your cell phone map app and you search for the nearest Welsh cake bakery. You then ask the app for directions to the bakery. The app gives you a choice of routes dependent on your mode of transportation, the time of the day and the local traffic conditions.
- Let's relate this to requirements management:
  - Your goal is to eat some Welsh cakes.
  - Getting to the bakery is your requirement.
  - The route you choose to follow is your specification.

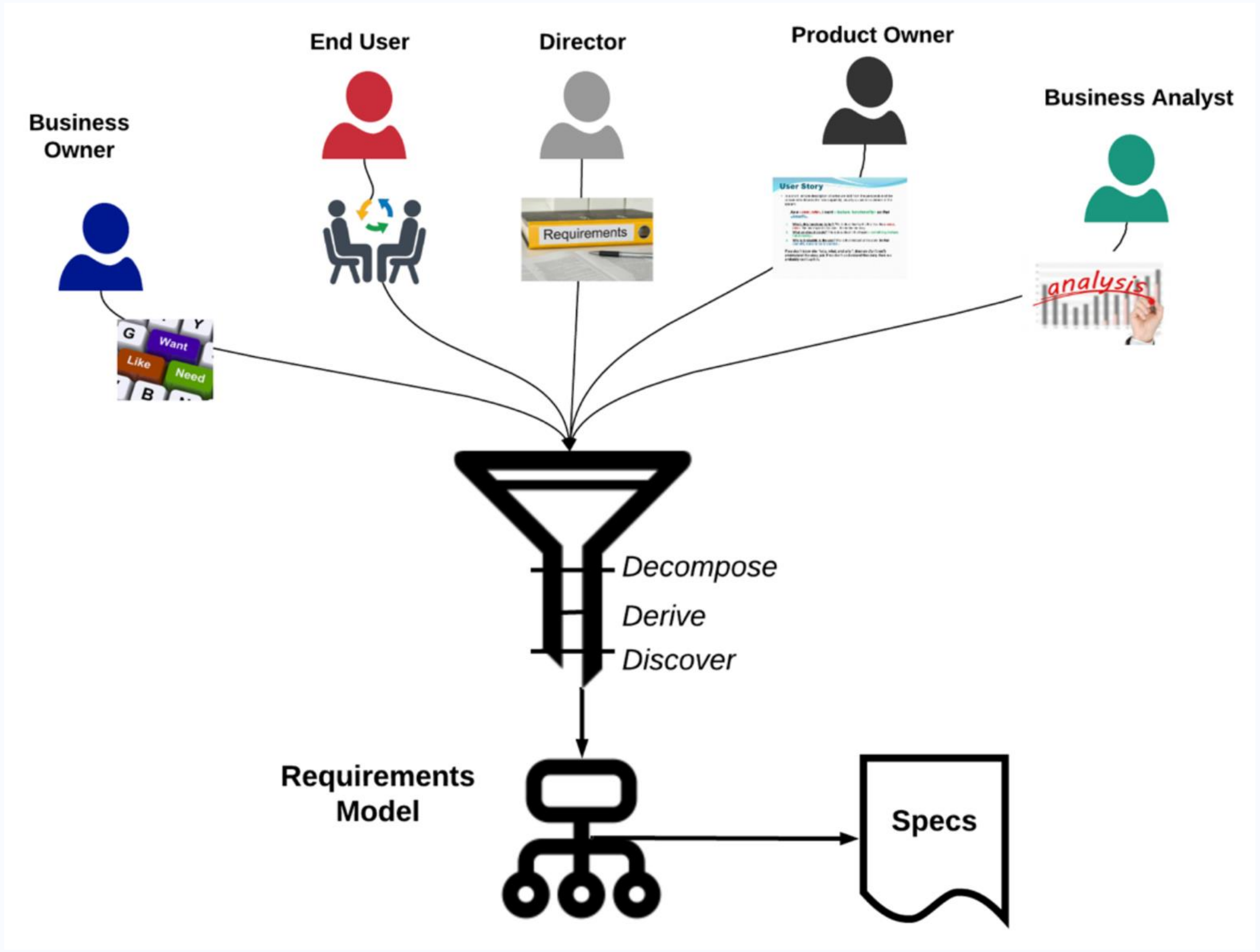
Crossing from requirements to specifications

- Turning requirements into specifications is one of the hardest parts of software engineering and is the part most software engineers get wrong most often.
- There are tools and metrologies around to assist us in getting this process right, such as behaviour-driven development (BDD) methodology



University of Southampton

The requirements funnel



Eliciting Requirements – What is Elicitation?

- Elicitation is the process of working with stakeholders to understand what they want to achieve through the project or to understand how the business is organised today.
- Elicitation involves bringing out the best ideas from people who see all the different aspects of the problem and includes documenting the results of what you find.
  - So, if you sit down and observe someone do their work and then you create a work-flow document that represents the understanding you achieved through that observation, you just completed some elicitation.
- Active Listening – Your Most Important Elicitation Skills
- Asking Good Questions

Eliciting Requirements

- Basics of elicitation
  - info collection from multiple resources
  - Dealing with bias, ambiguity, incompleteness, inconsistencies, and …
- A large collection of elicitation techniques
  - Background Reading
  - Hard data collection
  - Interviews
  - Questionnaires
  - Group Techniques
  - Participant Observation
  - Brainstorming
  - Focus Group
  - Interface Analysis
  - Prototyping

Difficulties of Elicitation

- Thin spread of domain knowledge
  - It is rarely available in an explicit form (i.e. not written down)
  - Distributed across many sources with conflicts between knowledge from different sources
- Tacit knowledge (The "say-do" problem)
  - People find it hard to describe knowledge they regularly use
- Limited Observability
  - The problem owners might be too busy coping with the current system
  - Presence of an observer may change the problem
- Bias
  - People may not be free to tell you what you need to know
  - People may not want to tell you what you need to know
  - The outcome will affect them, so they may try to influence you (hidden agendas)

Different Elicitation Techniques for Different Projects

- Different projects and stakeholder call for different requirement elicitation techniques.
- You never, in any project, use all these techniques at once.
- But you must used many of them selectively throughout the elicitation process across multiple projects.

- Think of your sets of techniques like a toolbox and as you become more sophisticated, your toolbox becomes bigger and more complex. And you have more tools that you can pull from to leverage in any given situation.

Factors to be Considered to Select Elicitation Techniques

- Nature of the information needed: depending on the type of information that needs to be gathered, certain elicitation techniques may be more appropriate than others.
  - For example, if you need to gather quantitative data, a survey or experiment may be more appropriate, while if you need to gather qualitative data, interviews or focus groups may be more effective.
- Target audience: the characteristics of the individuals or groups you are trying to elicit information from can also influence the choice of techniques.
- Level of detail needed: some elicitation techniques are more effective than others at gathering detailed information.
  - For example, interviews and observations can provide more in-depth information than surveys or questionnaires.

University of Southampton

Factors to be Considered to Select Elicitation Techniques

- Time and resource constraints: the resources available, including time and budget, can also influence the selection of elicitation techniques.
  - Some techniques require more time, expertise, and resources than others, and it's important to consider these factors when selecting techniques.
- Ethical considerations: ethical considerations, such as privacy, confidentiality, and informed consent, should also be taken into account when selecting elicitation techniques.
  - For example, if the information being gathered is sensitive, it may be necessary to use techniques that ensure participant anonymity and confidentiality.
- Context and purpose of the elicitation: the overall context and purpose of the elicitation should also be considered, including the goals of the project, the stakeholders involved, and the broader research or development context.

University of Southampton

Resources

- **Behaviour-driven development**
  - https://en.wikipedia.org/wiki/Behavior-driven_development
- **User story**
  - https://en.wikipedia.org/wiki/User_story
- **Requirements**
  - https://en.wikipedia.org/wiki/Requirement
- **List of software development philosophies**
  - https://en.wikipedia.org/wiki/List_of_software_development_philosophies
- **Software Engineering (10th Edition)** – 2016 by Ian Sommerville

YOUR QUESTIONS