University of Lancashire

CO3808 Honours Degree Project

Requirements Analysis and Specification

Dr Brendan Cassidy

Where opportunity creates success

This session should help you…

- Understand the Different Types of Requirements you may need to consider/specify in your project
- Explore Approaches to requirements analysis and specification
- Describe the role of Requirements Engineering
- Select an Approach appropriate for your specific project
- Have a plan for specifying requirements in next report deliverable

Managing your Supervisor

University of Lancashire

- There is a gap in summative submissions now until mid way through Semester 2
  - Don’t lose momentum, steady progress throughout the year is optimal
- Your supervisor is not just a marker
  - They are a resource
- To get the most out of your supervisor give some thought to your communication with them
  - Show them you have put some thought into your questions
  - Go to meetings with a clear objective, you can communicate this to them
  - Be prepared, show them things, give them something to talk about

Types of Requirement

University of Lancashire

User requirements

Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.

System requirements

A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

Types of Requirement

University of Lancashire

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Agile Methods and Requirements

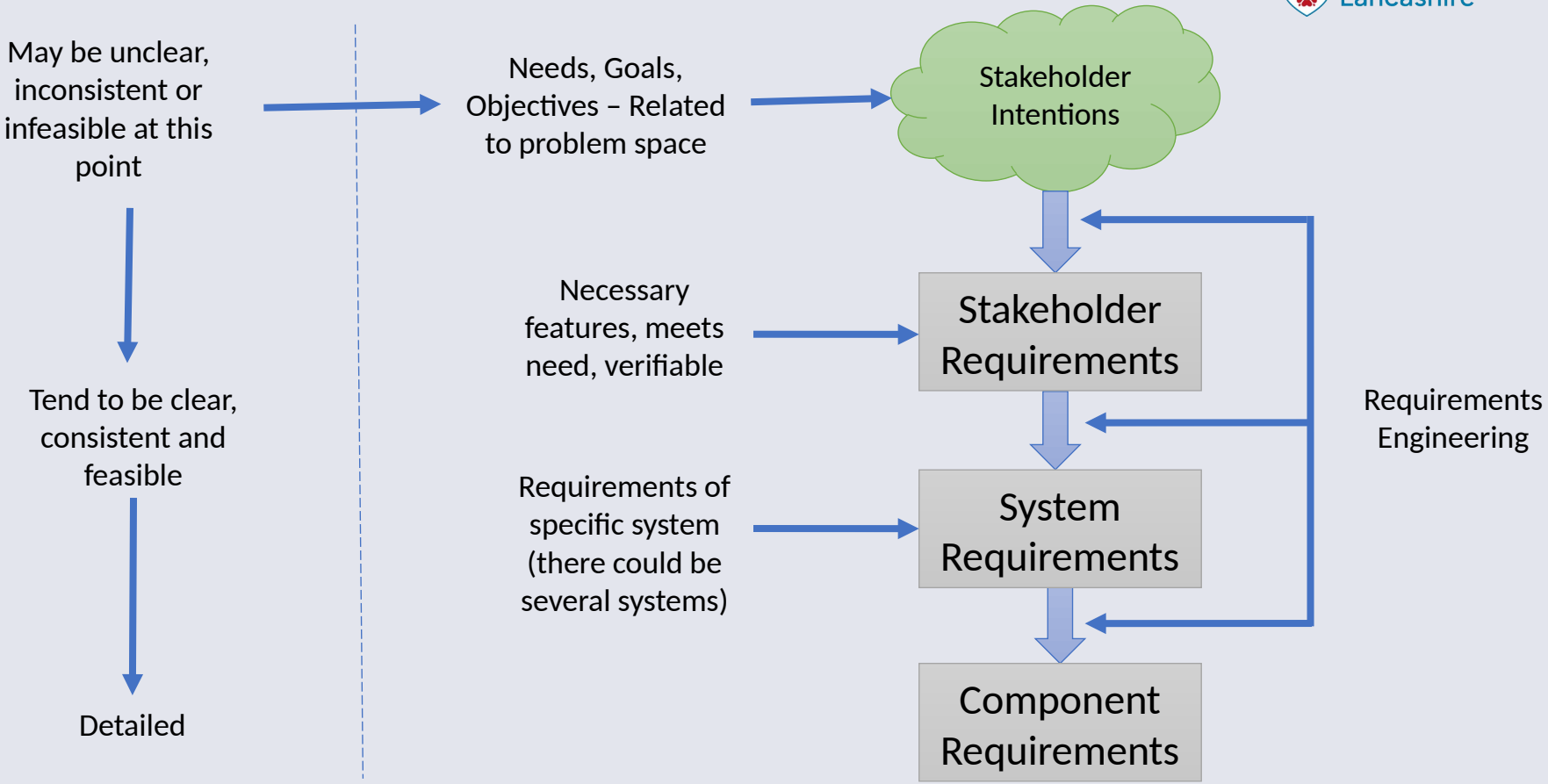
University of Lancashire

- Many agile methods argue that producing detailed system requirements is a waste of time as requirements are not well known or change so quickly
- A formal requirements document is therefore always out of date
- Agile methods usually use **incremental requirements engineering** and may express requirements as ‘user stories’
- This is practical for business systems but problematic for systems that may require pre-delivery analysis (e.g. critical systems) or systems developed by several teams

Requirements Engineering

University of Lancashire

- The process of establishing the services that a customer requires from a system and the constraints under which it operates and is developed.
- The system requirements are the descriptions of the system services and constraints that are generated during the requirements engineering process.



What do requirements cover?

University of Lancashire

- Functionality.
  - What the software is supposed to do

Functional Requirements

- External interfaces.
  - Interaction with people, hardware & software
- Performance.
  - Speed, availability, response, throughput,…
- Attributes
  - Portability, accuracy, maintainability, security,…

Non-Functional Requirements

Functional Requirements Example

~~A user shall be able to search the appointments lists for all clinics.~~

The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Problems arise when functional requirements are not precisely stated.

Ambiguous requirements may be interpreted in different ways by developers and users.

Consider the term ‘search’ in first requirement

User intention – search for a patient name across all appointments in all clinics;

Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

Non Functional Requirements

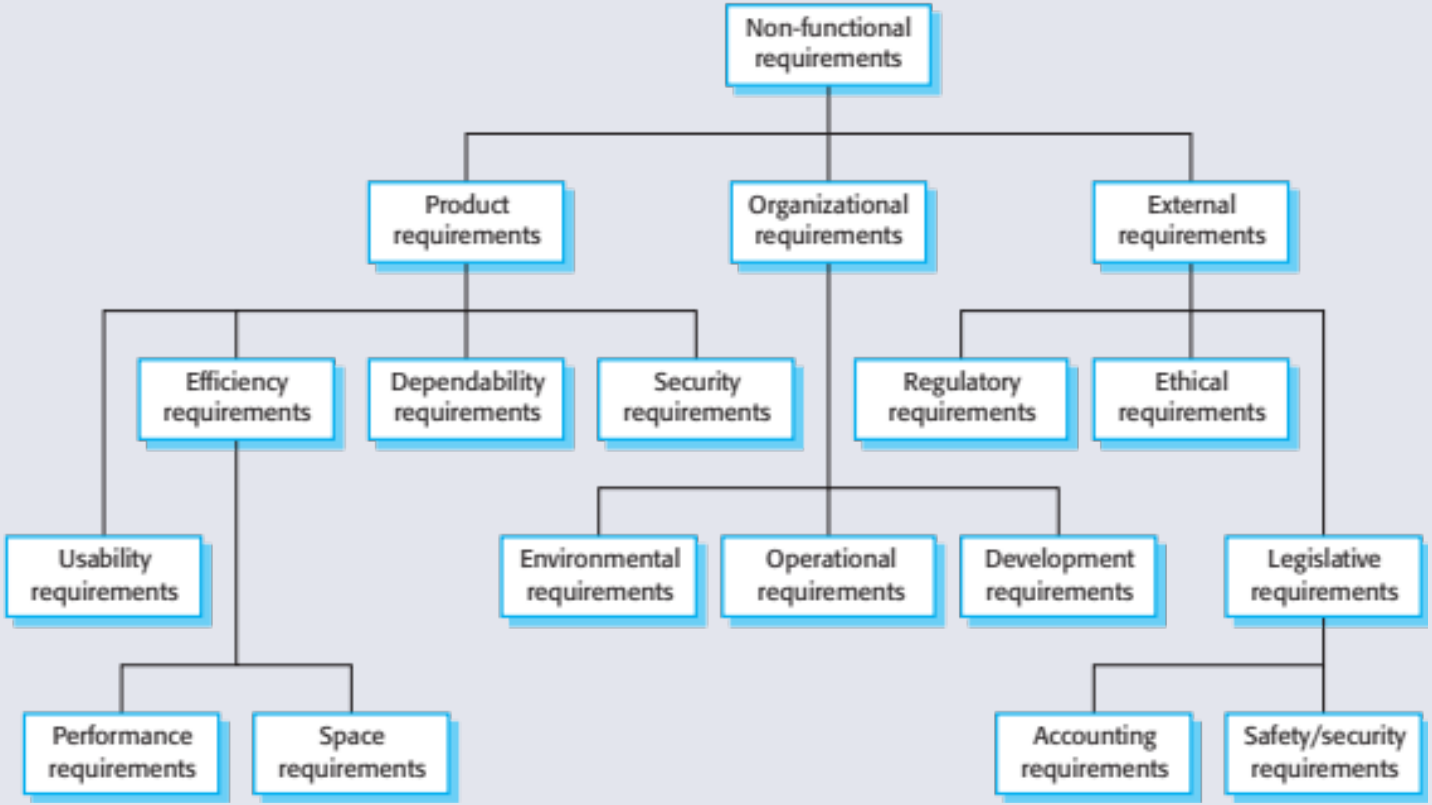
University of Lancashire

These define system properties and constraints e.g. reliability, response time and storage requirements.

Process requirements may also be specified mandating a particular IDE, programming language or development method.

Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.

Non functional Requirements Types



Demonstrating a Structured Approach

University of Lancashire

A good set of requirements will require analysis of the problem domain

- How did you arrive at your requirements?

UML Diagrams can help with analysis and specification

- Use Case Model: Diagram & Descriptions
  - Functional System Behaviour *as seen by the user*
- Activity Diagrams
  - Model Business Activities (processes)
    - Or ANYTHING that requires a specific process (E.g. Player Decision Making)

Use Cases

University of Lancashire

Use cases identify the actors in an interaction and describe the interaction itself.

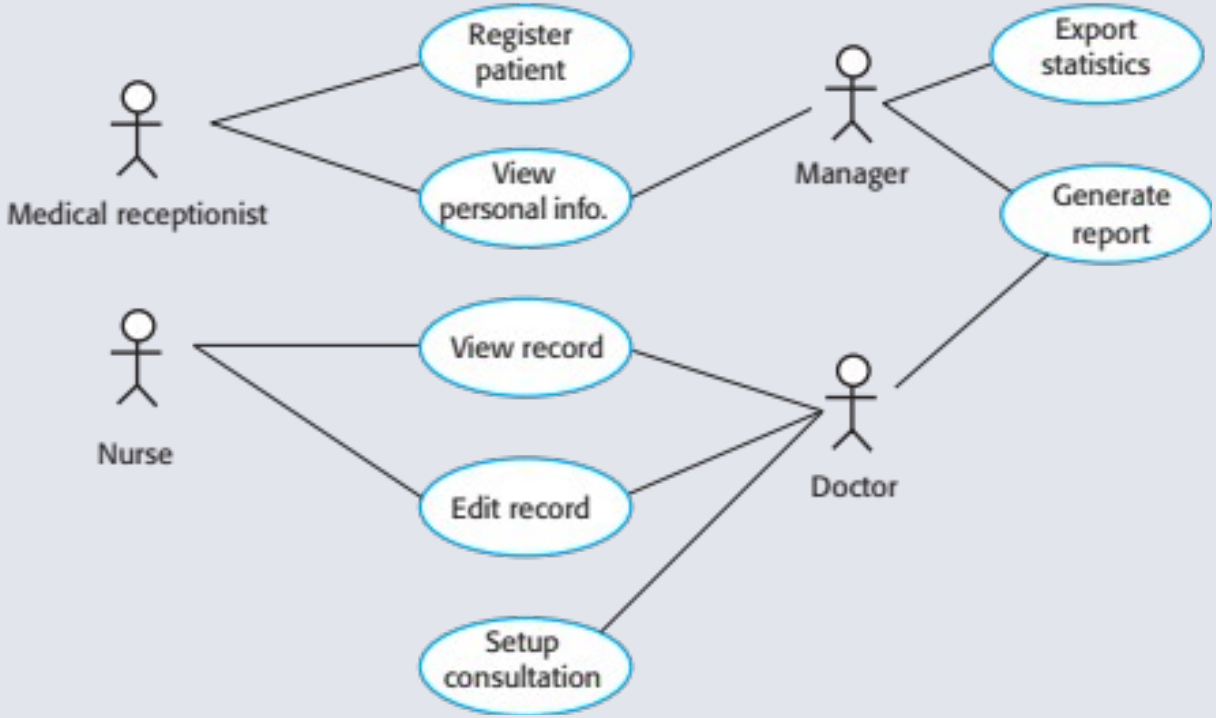
A set of use cases should describe all possible interactions with the system.

High-level graphical model supplemented by more detailed tabular description.

Use Case Diagram

University of Lancashire

Give a good idea of the functionality of a system from a USER PERSPECTIVE



Communicating Requirements

University of Lancashire

- Requirements should be stated concisely and clearly
- How do you know when that requirement has been met?
  - Is it measurable/testable? Makes your life easier in later sections of your report
  - Is it achievable?
- Summarise in your report and save any detail for an appendix
  - E.g. could include use case diagram in your report (doesn't affect word count) and associated use case descriptions in an appendix
- Practice communicating your functional and non-functional requirements in a supervision discussion
  - In addition to **subtly planting the seed with your supervisor that you are taking requirements gathering seriously**, it also helps check if your requirements are clear, appropriate and testable

User Stories

- Represent functionality that is valuable to user or owner of system
  - Not a full description
  - A note/marker that something is needed
  - In simple language: user / problem domain
    - Starts the process of developers understanding
- Detail obtained through conversation between user/client and developer
  - Possibly with prototyping
  - A story promises a future conversation

User Story Format

- **As a** <type of user>,
- **I want** <some function>
- **so that** <some reason>

- **As a** lecturer
- **I want** to be able to obtain a list of students
- **so that** it can be used for email, spreadsheets, …

Other formats can be used but this one explicitly identifies:

Who

What is needed

Why

User Stories

University of Lancashire

Capture a description of a software feature from a users perspective

Simplified description of a feature

Uses natural language

- That clients and stakeholders are able to understand
- Gives no information about how a feature is implemented
- Gives context, information about why a feature is needed

User Stories

- People naturally think in narratives/stories
- As people we have an intimate grasp of
  - Characters
  - Desires
  - Motivations
- First thing to think about in user stories is
  - Character/Role
    - Who is this task being done for
    - Who's perspective do we need to consider when
      - Designing a feature

User Stories

- Second thing to think about in user stories
  - What it is we want done (e.g. functionality)
  - Usually forms the main part of a user story
    - Only the middle of the process we should be following

- Third thing to think about in user stories
  - Motivation for wanting the functionality
    - Why does this character want this thing?
    - How is it going to serve this

User Stories

- What is missing from this user story?
  - Why does he want that functionality?
  - What purpose does it serve?
  - Is it just to keep entries in date order?
    - Or is it more serious?

Do the logs need to be unalterable to serve as some sort of audit functionality by Starfleet crime scene investigators?

Two very different implementations
Casual vs robust

User Stories

- Often Needs will change according to user roles

As a <type of user>,
I want a car
so that I can drive to work

As a Suburban commuter,
I want a car
so that I can drive to work

As a Highland Farmer,
I want a car
so that I can drive to work

Comparing User Stories to Use Case Models

University of Lancashire

- User stories – brief indication of a need/feature
  - Should indicate user type/role and reason for feature
    - Try not to be too generic with reason
  - to be expanded by discussion with stakeholders

- Use cases – detailed description of interaction between user and system

- Use cases - may be implemented directly and good starting point for writing tests

You will need to decide (and briefly justify in your report) what approach to requirements you are going to use.

Summary

University of Lancashire

Explored Types of Requirements	Explored Approaches to requirements analysis and specification	Compared some formal vs traditional approaches to specification
• Functional vs Non Functional	• Requirements Engineering Use Case Models, Activity Diagrams	• Use Case Model vs User Stories

The End

University of Lancashire