

COMP201 Software Engineering 1

Lecture 6 –

Security Requirements and The Requirements Engineering Process

Lecturer: T. Carroll

Email: Thomas.Carroll2@Liverpool.ac.uk

Office: Ashton G.14

See Vital for all notes



Recap

Recap Lecture 4

- Requirements set out what the system should do and define constraints on its operation and implementation
- Functional requirements set out services the system should provide
- Non-functional requirements constrain the system being developed or the development process
- User requirements are high-level statements of what the system should do, aimed at the user
- System requirements are more detailed statements of services the system should provide, and are not aimed at the user, but instead the customer and the developers

Recap Lecture 5

- User requirements should be written in natural language, tables and diagrams
- System requirements are intended to communicate the functions that the system should provide – more structured language needed to avoid ambiguity
- System requirements may be written in structured natural language, a PDL or in a formal language
- A software requirements document is an agreed statement of the system requirements



Today

Objectives

- Look at **security requirements** and **availability requirements**
- To describe the principal requirements engineering activities and their relationships
- To introduce techniques for **requirements elicitation and analysis**
- To describe **requirements validation** and the role of **requirements reviews**
- To discuss the role of **requirements management** in support of other requirements engineering processes



Security Requirements

Security

- Most modern systems have some security requirements
- Why?
- Because
 - Internet
 - Systems often control money
 - Systems nearly always contain data (much of it personal)
 - You are legally required often to keep your system secure
 - You could get sued

Security requirements of systems

- Broken down into 4 main issues
 - Confidentiality
 - Integrity
 - Authentication and Authorization
 - Non-repudiation
- One auxiliary issue
 - Availability (Performance security)

Confidentiality requirements

- Usually two main options
 - Encryption (hard security)
 - Permissions (soft security)
- Data must be kept secure
 - In storage (final or intermediary)
 - On the wire or wireless link
 - For as long as reasonably possible

Integrity Requirements

- Messages or data must not be modifiable without
 - Knowledge of the change
- Integrity approaches
 - CRC Checking (no good, easy to forge check value)
 - Hash value over data, similar problem to CRC
 - Hash value over data + secret value
 - Key distribution problem
 - Hash value encrypted using asymmetric cipher
 - Best approach to date

Authentication/Authorization

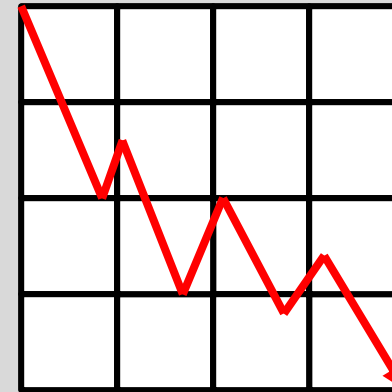
- Authentication
 - Who are you?
- Authorization
 - What are you allowed to do?
- Techniques
 - Usernames, Passwords, hardware (cards, dongles), Biometrics
- Often first point of attack

Non-repudiation issue

From: Bob
To: Broker

Please buy
lots of shares

Bob subsequently
denies sending the
email



Share Price

Bob



Non Repudiation in practice

- May require
 - Trusted broker, third party
 - Funding in Escrow
- Non repudiation is built on
 - Authentication
 - Integrity
 - Recording and time stamping
 - Broker style services

Security, logs and alerts

- Security is very dependent on knowledge of activity (audits and logs)
- Standard log (records all logins/logouts, database access requests)
- Failed login log (records all failed logins)
- Unusual activity log (high volume transactions on account)
- Alert log (failed logins for top level clearance users)
- Alerts
 - Unusual activity can be used to alert operators, suspend accounts etc.

Bell–LaPadula model

- All items given security clearance level
 - Top-Secret (4), Secret(3), Sensitive(2), Unclassified
- no read-up
 - A subject cannot read a document above their clearance level
 - If I am cleared to level 2, I cannot read a level 3 or 4 document
- no write-down
 - A document cannot be copied/included with another document with a lower security clearance
 - So if I want to add a top secret to a sensitive document the result will be a top secret document
 - If my classification is 2, I cannot produce an unclassified document
- Trusted subjects
 - Can write documents down
 - Must be shown trustworthy with regard to the security policy

Specifying Security

- Ideally kept as open as possible to allow for
 - Upgrading of encryption algorithms and protocols
- Security policy
 - Shredding documents
 - Secure disposal
 - Password reset protocols
 - Security training
 - Security audits
- Standards compliance
 - Payment Card Industry Data Security Standard



Availability Requirements

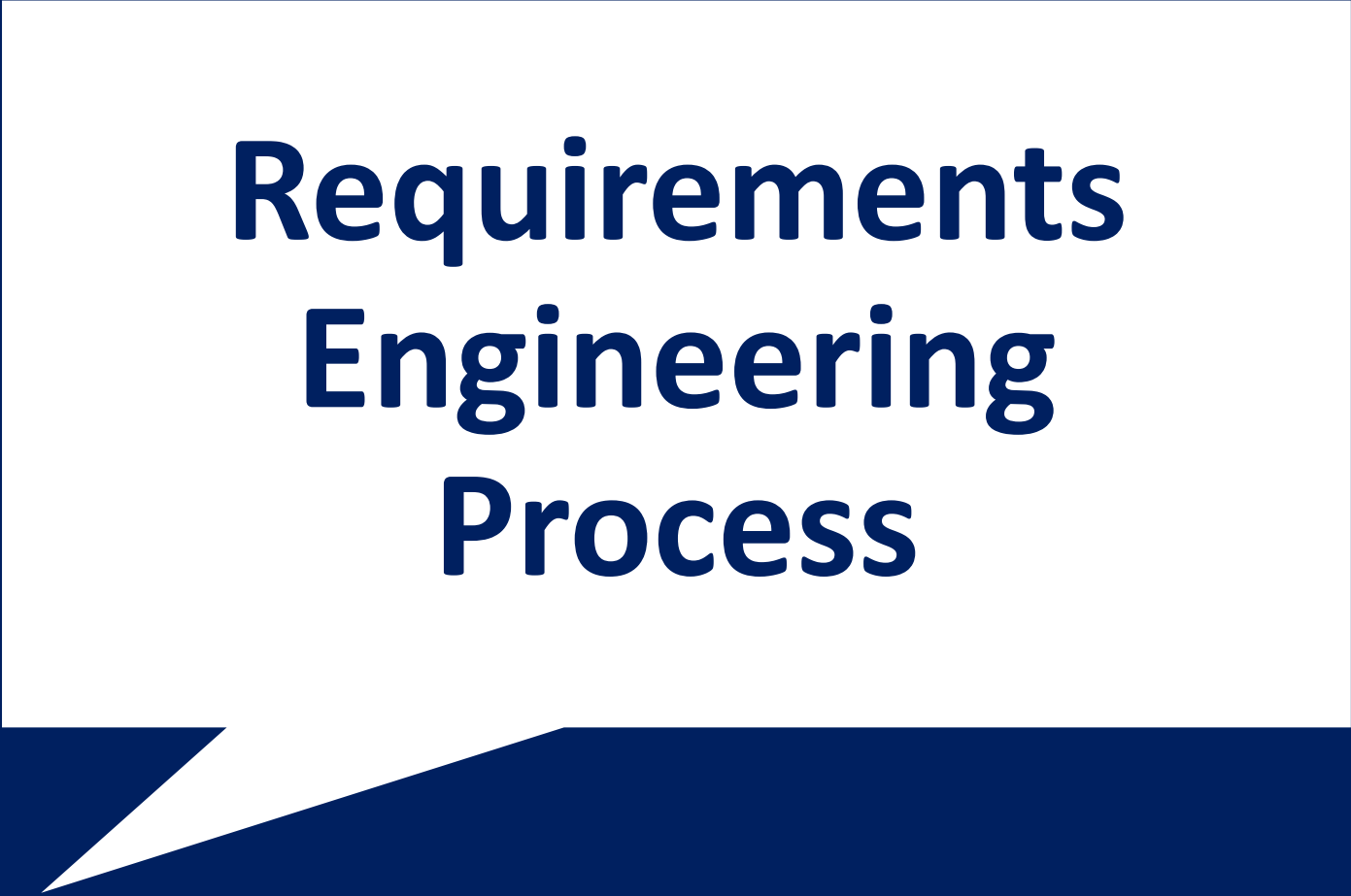
Availability requirements

- High availability of system
- Specifying in 9s terminology

Uptime	Uptime	Maximum Downtime per Year
Six nines	99.9999%	31.5 seconds
Five nines	99.999%	5 minutes 35 seconds
Four nines	99.99%	52 minutes 33 seconds
Three nines	99.9%	8 hours 46 minutes
Two nines	99.0%	87 hours 36 minutes
One nine	90.0%	36 days 12 hours

Availability in practise

- 9s terminology not always useful
- Imagine a computer system where:
 - Three 9s available but unavailability spread as 78 seconds per day
 - Or Five 9s available, failing once every 10 years for 50 minutes
- We should specify
 - Worst case scenarios
 - Worst case delay as well as down time
 - How the system can degrade gracefully



Requirements Engineering Process

Requirements Engineering Processes

- The processes used for requirements engineering vary widely depending on the application domain, the people involved and the organisation developing the requirements.
- However, there are a number of generic activities common to all processes which we look at today.
- The goal of this stage of the software engineering process is to help create and maintain a **system requirements document**.

Requirements Engineering Processes

1. Requirements elicitation;
 - What services do the end-users require of the system?
2. Requirements analysis;
 - How do we classify, prioritise and negotiate requirements?
3. Requirements validation;
 - Does the proposed system do what the users require?
4. Requirements management.
 - How do we manage the (sometimes inevitable) changes to the requirements document?

Requirements Engineering Process

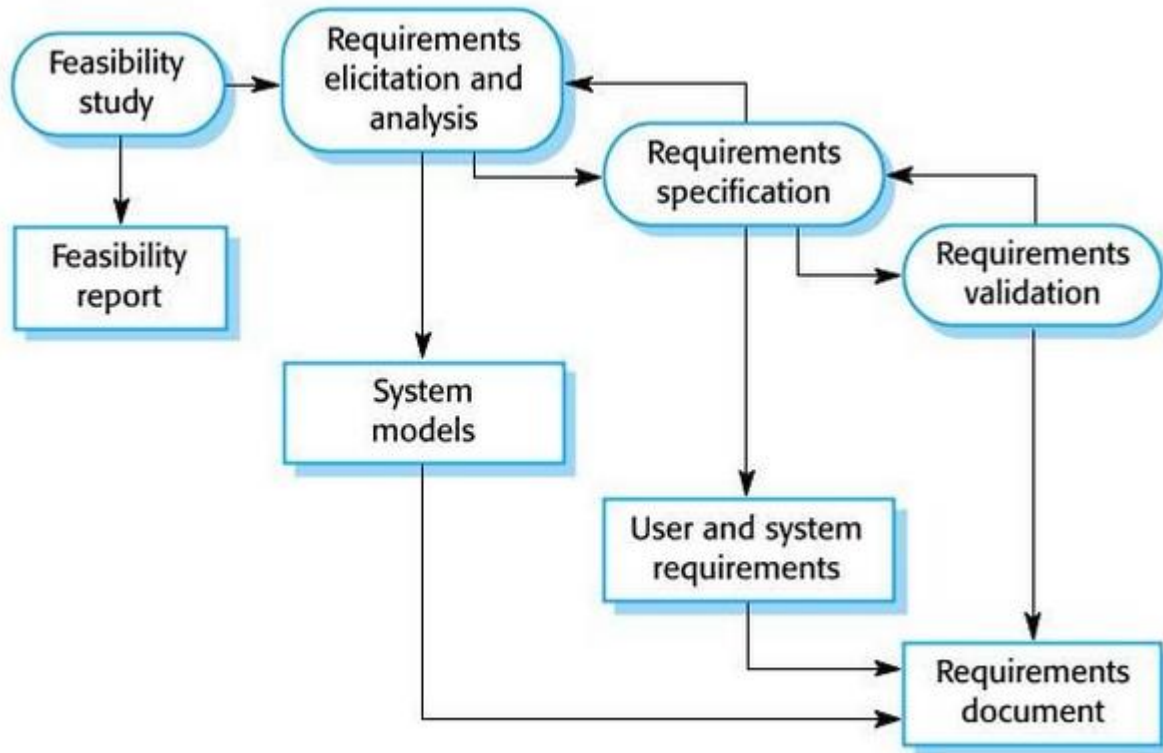


Figure 7.1 (Sommerville, 2007)

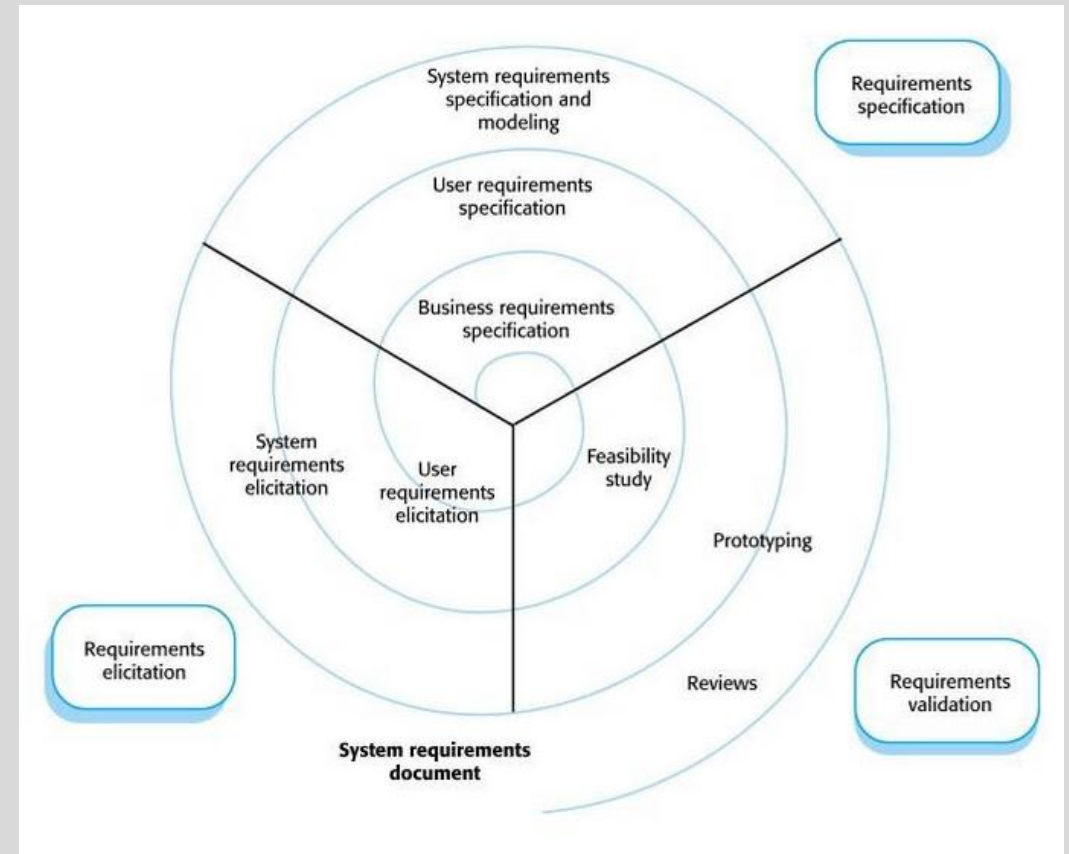


Figure 7.2 (Sommerville, 2007)

Example – Patient Records System

1. **(Elicitation)** Talk to patients, doctors, nurses, receptionists, managers to find out:
 1. Current system practise
 2. legal restrictions
 3. problems with current system
 4. needs for improvement
 5. **security issues**
 6. costs
2. **(Specification/Elicitation)** Develop draft documentation and review:
 1. Most important requirements
 2. Costings
 3. Timescale
 4. Hardware requirement
3. **(Validation)** Send requirements to end users; Present them with Q&A; Go back to step 1, discuss requirements again
4. **(Management)** Have annual **stakeholders meetings** to review requirements, review cost and feasibility of inevitable change to system

Feasibility Studies

- A **feasibility study** decides whether or not the proposed system is worthwhile.
- A short focused study that checks
 - If the system contributes to **organisational objectives**;
 - If the system can be engineered using **current technology** and **within budget**;
 - If the system can be **integrated with other systems** that are used.
 - Is there a **simpler way** of doing this (buy in software and customize)



Requirements Elicitation and Analysis

Elicitation and Analysis

- Involves technical staff working with customers to find out:
 - Application domain
 - Services that the system should provide
 - Operational constraints.
- May involve:
 - end-users
 - Managers
 - engineers involved in maintenance
 - domain experts
 - trade unions, etc.
- These are called **stakeholders**.



Acknowledgement:
https://farm9.staticflickr.com/8206/8267263835_11d235f22e_o_d.jpg

Problems of Requirements Analysis

- Stakeholders **don't know what they really want.**
- Stakeholders express requirements **in their own terms.**
- Different stakeholders may have **conflicting requirements**
 - Staff → easy of use
 - Management → highest security
 - Patients → change appointments easily
 - Management → plan staff resourcing, reduce costs
- **Organisational and political factors** may influence the system requirements (Data protection)
- The requirements **change during the analysis process;**
 - New stakeholders may emerge
 - The business environment may change.

Requirements Discovery

- **Gathering information** about the proposed and existing systems
- Distilling the user and system requirements from this information.
- Sources of information include
 - Documentation
 - System stakeholders
 - Specifications of similar systems

In the real world

- Requirements often come from
 - Copying /modifying the requirements of other systems
 - Copying and fixing the requirements of a legacy system
 - Looking at what competitors do and **improve on it**
- Prototyping
 - The initial requirements are often very thin
 - A lot of requirements are discovered by **prototyping**

Example - ATM Stakeholders

- Bank customers
- Representatives of other banks
- Bank managers
- Counter staff
- Database administrators
- Security managers
- Marketing department
- Hardware and software maintenance engineers
- Banking regulators

Viewpoints

- Structuring the requirements to represent the **perspectives** of different stakeholders.
- Stakeholders may be classified under different viewpoints.
- There is **no single correct way** to analyse system requirements.

Viewpoint Identification

- We may identify viewpoints using
 - Providers and receivers of system services;
 - Systems that interact directly with the system being specified;
 - Regulations and standards;
 - Sources of business and non-functional requirements.
 - Engineers who have to develop and maintain the system;
 - Marketing and other business viewpoints.

Interviewing

- In formal or informal interviewing, the RE team puts questions to stakeholders about the system that they use and the system to be developed.
- There are two types of interview
 - **Closed interviews** where a pre-defined set of questions are answered.
 - **Open interviews** where there is no pre-defined agenda and a range of issues are explored with stakeholders.
- Ideally, interviewers should be open-minded, willing to listen to stakeholders and should not have pre-conceived ideas.



Acknowledgement:
<https://pathtothepossible.files.wordpress.com/2011/10/patent-reexamination-interview.gif?w=545>

Ethnography

- Observing and analysing how people **actually work**.
- People do not have to explain or articulate their work.
- Social and organisational factors of importance may be observed.
- Ethnographic studies have shown that work is usually **richer and more complex** than suggested by simple system models.



Acknowledgement:
<http://dstudio.ubc.ca/files/2012/10/observation.jpg>

Focused Ethnography

- Developed in a project studying the air traffic control process
- Combines **ethnography** with **prototyping**
- Prototype development results in unanswered questions which focus the ethnographic analysis.
- The problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant.

Scope of Ethnography

- Requirements that are derived from the way that people **actually work**
- Not from the way in which process definitions *suggest* that they ought to work.
 - People may have “**short cuts**” or use their **previous knowledge** and **experience** to better perform their role which may not be evident.
- **Example:**
 - An air traffic controller may switch off a conflict alert alarm detecting flight intersections.
 - Their strategy is to ensure these planes are moved apart before problems arise and the alarms can distract them.

Scope of Ethnography

- Requirements that are derived from **cooperation and awareness** of other people's activities.
 - People **do not work in isolation** and may share information and use dialogue with colleagues to inform decisions.
- **Example:**
 - Air traffic controllers may use **awareness of colleagues work** to predict the number of aircraft entering their sector
 - Thus require some **visibility of adjacent sector**.

Requirements Validation

Requirements Review

- Validity.
 - Does the system provide the functions which best support the customer's needs?
- Consistency.
 - Are there any requirements conflicts?
- Completeness.
 - Are all functions required by the customer included?
- Realism.
 - Can the requirements be implemented given available budget and technology?
- Verifiability.
 - Can the requirements be checked?
 - This reduces the potential for disputes between customers and contractors and a set of tests should be possible.

Scenarios

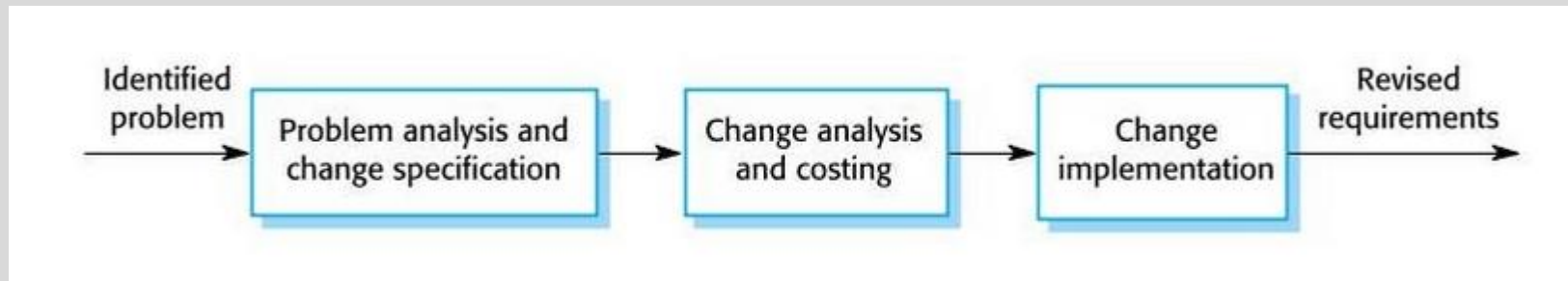
- Effectively **test cases** running in a given situation
- Example:
 - Try and withdraw cash with stolen credit card
 - Try and withdraw cash but machine has low cash stock
 - Withdraw cash with card number 3456123245677
 - Etc.
- Scenarios are very important:
 - They show the developer by example **what will happen given certain conditions**
 - They can be used as a **basis to test the software**
 - Make things very clear and **reduce ambiguity**



Requirements Management

Managing (the inevitable) Change in Requirements

- When a system is installed, new requirements often emerge
- Comes from better understanding of the user's needs
- **Enduring Requirements** are stable, derive from business core activity
- **Volatile Requirements** will change as the environment changes





Recap

Recap Lecture 6

- The **requirements engineering process** includes
 - feasibility study
 - requirements elicitation and analysis
 - requirements specification
 - requirements management.
- The **requirements elicitation and analysis** stage is iterative
 - involves domain understanding
 - requirements collection, classification, structuring, prioritisation and validation.
- Systems have **multiple stakeholders** with different requirements
- **Security** for most systems is a core service