

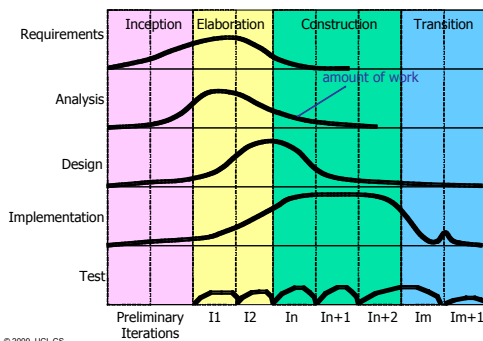
COMP2009 Software Engineering

Requirements Gathering

Requirements Analysis Definition

- Requirements analysis is the activity in which we are figuring out what we have to build.
 - Discover and reach agreement with customer on what the system has to do.
 - Create a high level specification of this understanding.
- NOTE: A system may not always have an identifiable customer who sets the requirements.
 - Some software is developed according to an organisation's perception of *market demand*.
 - For example: Office Suite software, shrink-wrap packages.
 - Nevertheless, requirements still must be specified in order to guide later stages of development.

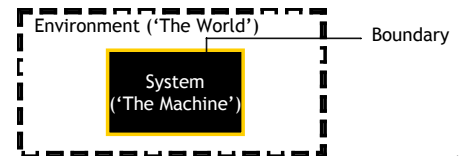
Requirements Analysis USDP Context



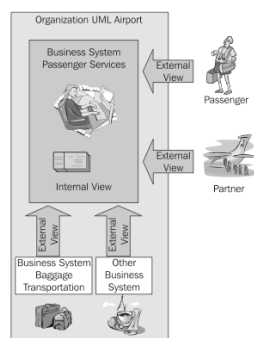
'The World and the Machine'

Michael Jackson

- The *environment* of a *system* is the set of elements (and their relevant properties) that are outside the system.
- A change in some element of the environment produces a change in the state of the system, or vice versa.
 - These effects at the *boundary* are the requirements.
 - When specifying the requirements, the machine should be treated as a *opaque box*.



Organisation Internal v. External View



System Boundary

- Setting boundaries is very important when analysing and designing a system.
 - Limits the 'problem space' to be considered.
- Example: Design of a new computer-controlled car.
 - Are the repair shops, refuelling stations and parts supply part of the system you are designing or not?
 - How much do they affect the design of the car?
 - Can you change them?
 - How much would changes in them affect your design?

Why Are Requirements Important?

- *'The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, & to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.'*
- Fred Brooks in 'The Mythical Man Month'

The Economics of Requirements - Boehm 1980

Relative cost to fix an error:

Phase in which found	Cost ratio
requirements	1
design	3-6
coding	10
development testing	15-40
acceptance testing	30-70
operation	40-1000

... and these figures are considered conservative!

The Starting Point

- Problem statement
 - 'You have been contracted to develop a computer system for a university library. The library currently uses a 1980s program, written in an obsolete language, for some simple bookkeeping tasks, and a card index for user browsing. You are asked to build an interactive system that handles both of these aspects.'
- Need to construct detailed requirements on what the application should do.
 - Determine boundary and interfaces.
 - Elaborate functions, features, behaviours.

Functional and Non-Functional Requirements

- Functional: What the system should do.
 - Example: The library system **shall** provide a facility for identifying a library user.
 - Example: The library system **shall** provide a reminder when a book is overdue.
- Non-functional: How the system is supposed to be.
 - Specify criteria used to judge operation of a system.
 - Also a constraint, quality attribute or QoS.
 - Example: The library system **shall** authenticate a library customer in five seconds or less.
 - Example: The library system **shall** communicate with borrowers using email.

Writing a Requirement

<id> The <system> shall <function>

unique identifier name of system keyword function to be performed

e.g. "32 The ATM system **shall** validate the PIN number."

- Aim for a uniform sentence structure.
- Each requirement should focus on a *single* distinct feature or behaviour.
 - Should not be too vague or abstract.
 - Or be too general or imprecise.
 - Or include implementation information.
 - e.g., names are stored in the Name table of the database.
- Maintain a consistent level of detail.

Representing Requirements

ID	Functional Requirements	Priority
Collection		
1	The system shall ...	M
2	The ...	
Borrowing		
3		
Browsing		
4		
Membership		
5		
User interface		
6		

- A spreadsheet is a good tool for writing and managing requirements.
 - Colouring is optional!

Requirements Prioritisation

- The 'MoSCoW' Approach
 - M**: Must Have
 - Mandatory requirements that are fundamental to the system.
 - S**: Should Have
 - Important requirements that could be omitted in early versions.
 - C**: Could Have
 - Requirements that can be omitted without reducing the value of the application significantly.
 - Depends on time, budget and resource availability.
 - W**: Won't Have (this time around)
 - Requirements that can wait.
- 'Bells and whistles', decorations.

Requirements Categorisation

- Can be tens, hundreds, thousands of requirements.
- Group related requirements together in categories.
 - For example:
 - Book management
 - Library user management
 - Admin
 - etc.
- All requirements should have an ID or number.
 - May want a hierarchical numbering system.

Other Requirement Information

- Can also include:
 - Who 'owns' the requirement?
 - Who to ask about it?
 - Who is responsible for checking the requirement is correct in the delivered system (acceptance testing)?
 - Detailed Requirement priority
 - Order in which Must Have, Could Have, etc. requirements should be added.

Functional Requirements List Example

ELECTION MAINTENANCE				
RQ16	The EVS shall email reminders to users who haven't yet voted 2 days before the election end date.	Functional	Election maintenance	Could Have
RQ17	The EVS shall display a list of candidates for each category and briefly outline their manifestos and store details in the election database.	Functional	Election maintenance	Must Have
RQ18	The EVS shall display a list of categories for each election and store details in the election database.	Functional	Election maintenance	Must Have
RQ19	The EVS shall support the insertion of candidate photographs.	Functional	Election maintenance	Could Have
RQ20	The EVS shall automatically start and stop an election based on a predefined date range.	Functional	Election maintenance	Should Have
RQ21	The EVS shall email an invitation to vote to all relevant voters once the election has been setup.	Functional	Election maintenance	Should Have

Example Format - Non Functional Requirements

ID	Non-Functional Requirements	Priority
Capacity		
1	The system shall ...	M
2	The ...	
Reliability		
3		
Performance		
4		
Security		
5		
Compliance		
6		

Categories of Non-Functional Requirements

- Externally imposed constraints, e.g.,
 - database system, hardware, development infrastructure/skills, legacy systems, legal issues, data protection, contracts
- Execution qualities, observable at runtime, e.g.,
 - response time, performance, security, robustness, throughput, usability
- Evolution qualities, embodied in the static structure of the software system, e.g.,
 - testability, maintainability, extensibility, scalability

Non-Functional Requirements Example

Req ID	Requirement	Category	Priority	Impact
RQ3	The EVS shall use an Internet browser as its user interface.	Non-Functional	General	Must Have
RQ4	The EVS shall support the ALL versions of Internet Explorer and Netscape browsers.	Non-Functional	General	Must Have
RQ5	The EVS shall be written using standard Java to run on different operating systems (e.g. Linux and Windows).	Non-Functional	General	Must Have

Project Glossary

- In any business domain there is always a certain amount of jargon. It's important to capture the language of the domain in a *project glossary*.
- Essential that all terms, labels, names, etc. are clearly defined
- AND
- everyone is using the *same* definition.
- The aim of the glossary is to define key terms and to resolve synonyms and homonyms.
- You are building a vocabulary that you can use to discuss the system with the stakeholders.

Glossary Example

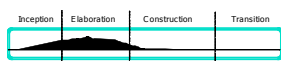
Glossary of Terms				
Term	Abbreviation	Definition	Synonym: Allowed	Synonym: Not Allowed
Administrator		This actor represents an individual with an administrative function within the LCCS. Tasks include updating and deleting customer account information, validating and rejecting user requests, blocking and unblocking customer accounts, and backing up and archiving customer and payment data.		
Blue Badge		Offers parking concessions to people with mobility difficulties or severe problems with vision. A Blue Badge holder is eligible for a 100% Congestion Charge discount.		
Cart		Provides payment of charges in the form of shopping functionality to the customer. At the technical level, the cart is represented within the system by the Order object.		
Charge Certificate		A notice issued for non-payment of the Penalty Charge.		
Charging hours		The hours between 7:00am and 6:00pm, Monday to Friday, excluding weekends, public holidays, and the period from 25 December to 1 January inclusive.		
Congestion Charge	LCC	A fee for most vehicles traveling within those parts of London designated as the Congestion Charging Zone.	London Congestion Charge	

Review

- Requirements need careful review sessions as part of the process.
- Check for:
 - Missing requirements
 - Duplicates
 - Mis-categorised
 - Priorities
 - Ambiguous or unclear
 - Unnecessary or out of scope
- Stakeholders should be involved.

USDP Requirements Workflow

- The goal is to produce a baseline description from which systems and software can be built, and against which the development process and its final outcome can be assessed.
 - Requirements document.
 - The Use Case Model (coming next).
- It is about eliciting and reaching agreements on stakeholder requirements.



Summary

- Overview of requirements specification.
- Functional and non-Functional Requirements.
- MoSCoW
- Categories
- Review
- Project Glossary