

Review

- *Requirements is critical for successful software*
- *Requirements* = **services** the system is expected to provide + **constraints** placed on the system.
- Two type requirements
 - Functional and non-functional requirements
- SRS: software requirements specification

Chapter 7

Requirements and Analysis Engineering Processes

Objectives

- 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

Topics covered

- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

Requirements engineering processes

- The processes used for RE 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
 - Feasibility study
 - Requirements elicitation;
 - Requirements analysis;
 - Requirements validation;
 - Requirements management.

Feasibility studies

For all new systems, the RE should start with a feasibility study:

- Goal: Feasibility report suggests continue or stop the system

Feasibility studies (continued)

- What: A short focused study that checks
 - If the system contributes to organisational **vision**;
 - If the system can be engineered using current **technology** and within **budget**;
 - If the system can be integrated with other systems that are used.

In fact, it is a significantly compressed analysis and design process. It should be completed in a short time.

Elicitation and analysis

- Requirements gathering, or discovering
- *How:* communication
technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints.
- *Stakeholders involved:*
end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc.

Stakeholder is any one or organization who either affects the system development or is affected by it

Problems of requirements elicitation/analysis

- 信息不对称，交流困难 : **The curse of knowledge** ¹
 - is a cognitive bias according to which better-informed people find it extremely difficult to think about problems from the perspective of lesser-informed people. The term was coined by Robin Hogarth
 - The Curse of Knowledge — The More You Know, the Worse You Become At Communicating That Knowledge

Problems of requirements elicitation/analysis

- 尼采说过：“人们无法理解他们没有经历过的事情”
 - 要用涉众（客户）熟悉的业务语言和他们交流
 - 但是软件开发人员往往对业务不熟悉，没有经历过
 - 业务建模的重要性

Problems of requirements elicitation/analysis

- 许多时候，开发人员把需求启发想得太容易。经常可以听到“采集需求”这样的表述，好像需求是蘑菇，乖乖地躺在森林里，开发人员需要时，就像采蘑菇的小姑娘一样，一个，两个，三个，四个.....把它们都采回来。哪有这么容易！需求不是蘑菇。开发人员要能够像猎人一样，用锐利的眼睛发现隐藏在丛林中的猎物；像侦探一样，用慎密的思维判断出伪装成好人的凶手。
- 需求的一个启发障碍是知识的诅咒（**Curse of Knowledge**），意思是：一旦知道某个东西，就很难想像不知道它会是什么样子。1990年，斯坦福大学研究生**Elizabeth Newton**做了一个著名的心理学实验：让敲击者在桌子上敲击最常见的歌曲，听众根据听到的节奏回答是什么歌曲，然后让敲击者估计听众答对了多少。120次的实验中，敲击者预测听众猜对的比率会大于50%，真实的结果是听众猜对的比率只有2.5%。因为听众听的是敲出来的声音，敲击者听的是大脑里已有的歌曲。
-

Problems of requirements elicitation/analysis

- 知识的诅咒在需求启发中体现为沟通的困难。开发人员懂得许多软件设计和实现的知识，这些知识会有意无意地引导开发人员从实现的角度看需求；涉众在领域里面工作多年，许多事情在他看来一目了然，很难用开发人员能理解的言语表达出来。
- 需求启发的另一个障碍是做和定义的不同。涉众会做一件事情，不代表他能够把这件事定义出来教给其他人。在足球领域，号称球王的贝利、马拉多纳执教并不成功，最近十年的世界最佳主教练穆里尼奥踢球水平却很一般。要克服需求启发中的障碍，需要回溯到第2章的陈述：
- 和涉众交流的形式应该采用视图，而不是模型
- 和涉众交流的内容应该聚焦涉众利益，而不是需求。
- 如果很好地理解了这两点，需求启发技能就可以上一个台阶。

Problems of requirements elicitation/analysis

- Stakeholders don't know what they really want. **IKIWISI effect: I'll know it when I see it.**
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- The requirements change during the process. New stakeholders may emerge and the business environment change.

Problems of requirements elicitation (continued)

Fred Brooks: The hardest single part of building a software system is deciding what to build. No part of the work so cripples the resulting system if done wrong. No other part of SE is more difficult rectify later.

No requirements analysis, equivalently
flash marriage.

Process activities of R elicitation

- Requirements discovery
 - Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.
- Requirements classification and organisation
 - Groups related requirements and organises them into coherent clusters.
- Prioritisation and negotiation
 - Prioritising requirements and resolving requirements conflicts.
- Requirements documentation
 - Requirements are documented and input into the next round of the iterative.

Requirements discovery

- The process of gathering information about the proposed and existing systems and distilling the **system** requirements from this information.
- Sources of information include documentation, system stakeholders and the specifications of similar systems.

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– for complete, conflict

Different stakeholders have different viewpoints:

- Viewpoints are a way of structuring the requirements to represent the perspectives of different stakeholders. Stakeholders may be classified under different viewpoints.
- This multi-perspective analysis is important as there is no single correct way to analyse system requirements.

Types of viewpoint

- Interactor viewpoints
 - People or other systems that interact directly with the system. In an ATM, the customer's and the account database are interactor VPs.
- Indirect viewpoints
 - Stakeholders who do not use the system themselves but who influence the requirements. In an ATM, management and security staff are indirect viewpoints.
- Domain viewpoints
 - Domain characteristics and constraints that influence the requirements. In an ATM, an example would be standards for inter-bank communications.

Interviewing --- communication

- 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.

Interviews in practice

- Normally a mix of closed and open-ended interviewing.
- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviews are not good for understanding domain requirements
 - Requirements engineers cannot understand specific domain terminology;
 - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating.

Effective interviewers

- Interviewers should be open-minded, willing to listen to stakeholders and should not have pre-conceived ideas about the requirements.
- They should prompt the interviewee with a question or a proposal and should not simply expect them to respond to a question such as ‘what do you want’.

Scenarios

- Scenarios are real-life examples of how a system can be used.
- They should include
 - A description of the starting situation;
 - A description of the normal flow of events;
 - A description of what can go wrong;
 - Information about other concurrent activities;
 - A description of the state when the scenario finishes.

Social and organisational factors

- Software systems are used in a social and organisational context. This can influence or even dominate the system requirements.
- Social and organisational factors are not a single viewpoint but are influences on all viewpoints.
- Good analysts must be sensitive to these factors but currently no systematic way to tackle their analysis.

Review

- requirement is critical
 - 软件利润 = 需求（收入） - 设计（成本）
- gathering requirement is difficult
 - Curse of knowledge
 - IKIWISI effect
 - I will know it when I see it
 - IQ, EQ, SQ

Analysis overview

- Why analysis?
- What is analysis?
- How do analysis?
- Requirements versus analysis
- Analysis versus design

Why analysis

Deeply understand the requirements by constructing models. You must understand a problem before attempting a solution. If you don't analyze, it's highly likely that you'll build a very elegant software solution that solves the wrong problem. The result is: wasted time and money, personal frustration, and unhappy customers.

What is analysis

- Refine and structure requirements
 - more precise understanding for developer
 - more maintainable
 - structures system, including architecture

What is analysis (continued)

- Provides a basis for agreement between the stakeholders and the developers
- Becomes the starting point for design and implementation

How do analysis?

It depends on thinking:

- PO = Procedure + Procedure + Data
 - SA
- OO = O + O + Messages
 - OOA

Requirements versus analysis

Analysis is based on requirements

- In requirements
 - Used language of user so disallowed use of formal models.

Requirements versus analysis (continued)

- In analysis
 - Language of developer
 - Class

Analysis in a word

The purpose of analysis is to understand the problem so that a correct design can be constructed.

A good analysis captures the essential features of the problem without introducing implementation artifacts that prematurely restrict design decisions.

Analysis versus Design

- Design refine and expand analysis model into design model, make it straightforward for implementation
- The analysis model is an abstraction, or generalization, of the design. It omits most of the details of how the system works and provides an overview of the system's functionality.

Analysis versus Design (continued)

- Analysis: refine and structure requirements
 - precise and detailed understanding of the requirements
- Design and implementation: shape the system that meets these requirements
 - technology choices
 - design alternatives
 - non-functional requirements
 - computer-speak details

Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants.
- Requirements error costs are high so validation is very important
 - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error.

Requirements checking

- **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?

Requirements validation techniques

- Requirements reviews
 - Systematic manual analysis of the requirements.
- Prototyping
 - Using an executable model of the system to check requirements.
- Test-case generation
 - Developing tests for requirements to check testability.

Requirements reviews

- Regular reviews should be held while the requirements definition is being formulated.
- Both client and contractor staff should be involved in reviews.
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.

Review checks

- **Verifiability**. Is the requirement realistically testable?
- **Comprehensibility**. Is the requirement properly understood?
- **Traceability**. Is the origin of the requirement clearly stated?
- **Adaptability**. Can the requirement be changed without a large impact on other requirements?

Requirements management

- The requirements for large software systems are always changing:
 - Requirements management is the process of managing changing requirements during the requirements engineering process and system development.

Requirements management (continued)

- Requirements are inevitably incomplete and inconsistent
 - The problem cannot be fully defined.
 - New requirements emerge during the process as business needs change and a better understanding of the system is developed;
 - Different viewpoints have different requirements and these are often contradictory.

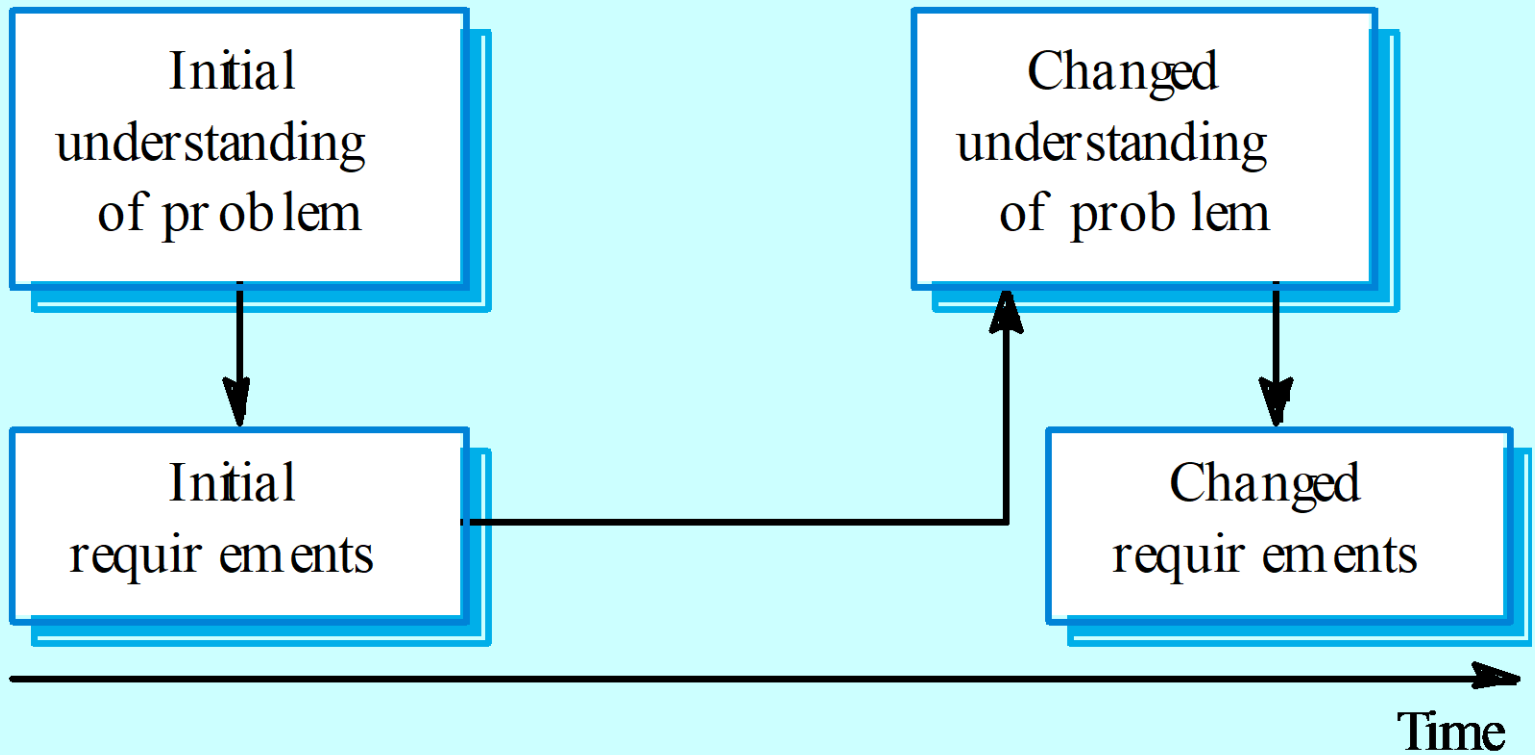
Requirements management (continued)

- Furthermore, once a system has been installed, new requirements inevitably emerge. Once end-users have experience of a system, they discover new needs and priorities:

Requirements change

- The priority of requirements from different viewpoints changes during the development process.
- System customers may specify requirements from a business perspective that conflict with end-user requirements.
- The business and technical environment of the system changes during its development.

Requirements evolution



Enduring and volatile requirements

From an evolution perspective:

- **Enduring requirements.** Stable requirements derived from the core activity of the customer organisation. E.g. a hospital will always have doctors, nurses, etc. May be derived from domain models
- **Volatile requirements.** Requirements which change during development or when the system is in use. In a hospital, requirements derived from health-care policy

Requirements management planning

- During the requirements engineering process, you have to plan:
 - Requirements identification
 - How requirements are individually identified;
 - A change management process
 - The process followed when analysing a requirements change;
 - Traceability policies
 - The amount of information about requirements relationships that is maintained;
 - CASE tool support
 - The tool support required to help manage requirements change;

Traceability

- Traceability is concerned with the relationships between requirements, their sources and the system design
- Source traceability
 - Links from requirements to stakeholders who proposed these requirements;
- Requirements traceability
 - Links between dependent requirements;
- Design traceability
 - Links from the requirements to the design;

A traceability matrix

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		D	R					
1.2			D			D		D
1.3	R			R				
2.1			R		D			D
2.2								D
2.3		R		D				
3.1								R
3.2							R	

They become unwieldy and expensive to maintain for large systems with many requirements

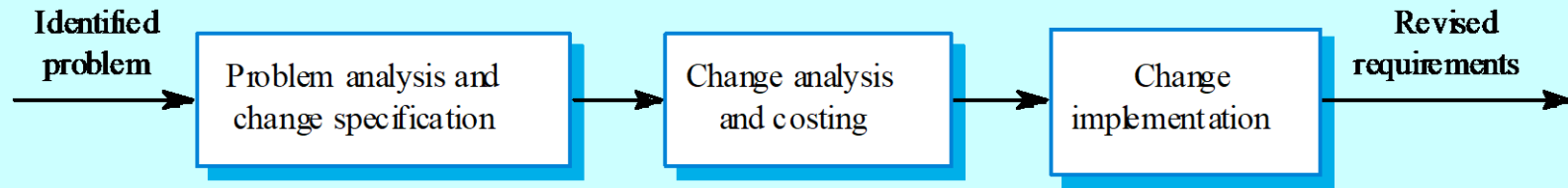
CASE tool support

- Requirements storage
 - Requirements should be managed in a secure, managed data store.
- Change management
 - The process of change management is a workflow process whose stages can be defined and information flow between these stages partially automated.
- Traceability management
 - Automated retrieval of the links between requirements.

Requirements change management

- Should apply to all proposed changes to the requirements.
- Principal stages
 - Problem analysis. Discuss requirements problem and propose change;
 - Change analysis and costing. Assess effects of change on other requirements;
 - Change implementation. Modify requirements document and other documents to reflect change.

Change management



Key points

- The requirements engineering process includes a feasibility study, requirements elicitation and analysis, requirements specification and requirements management.
- Requirements elicitation and analysis is iterative involving domain understanding, requirements collection, classification, structuring, prioritisation and validation.
- Systems have multiple stakeholders with different requirements.

Key points (continued)

- Requirements validation is concerned with checks for validity, consistency, completeness, realism and verifiability.
- Business changes inevitably lead to changing requirements.
- Requirements management includes planning and change management.