Requirements Engineering → Ingegnerizzazione dei requisiti

(from 10th Edition)

As proposed by the project sponsor

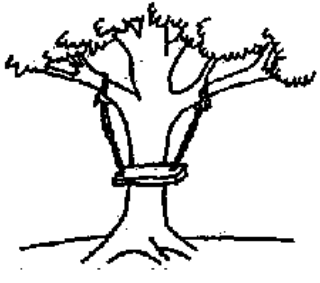
(proposta dal committente)

As specified in the project request

(specificata nei requisiti)

As designed by the senior analyst

(progettata dall'analista)

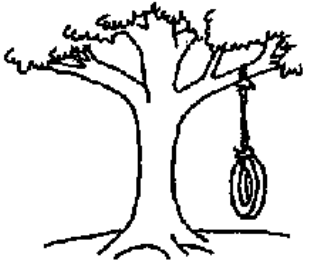


As produced by the programmers

(prodotta dai programmatori)

As installed at the user's site

(installata presso l'utente)



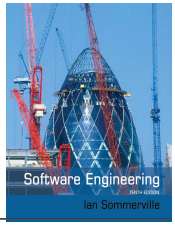
What the user wanted

(ciò che l'utente voleva)

Prima di tutto analizzeremo più a fondo il concetto di Requirement

e poi illustreremo il Processo di Ingegnerizzazione dei RequisitiSoftware Requirements

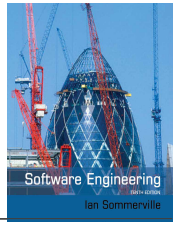
Requirements engineering (RE)



- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed
- The RE process produces the descriptions of the system services and constraints that are generated during the requirements engineering process
- What the customers (stakeholders) need is expressed in terms of their context, application domain, culture (these are the requirements), which then have to be 'specified' in more detail in terms of technical characteristics of the system to be developed (these are requirement specifications or simply sw specifications)

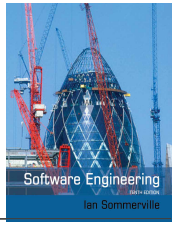
lo stakeholder ti parla nel suo linguaggio non tecnico e tu devi "tradurre" per i progettisti/programmatori

What is a requirement?



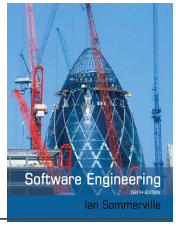
- Termine molto generale: è sempre utile capire le sfumature e i vari aspeti del significato e usarlo con precisione
- Its meaning may range from a high-level abstract statement of a required service or of a system constraint to a detailed mathematical functional specification
- This is inevitable as requirements may serve a dual function
  - May be the basis for a bid for a contract - therefore must be open to interpretation
  - May be the basis for the contract itself - therefore must be defined in detail
  - Both these statements may be called requirements

Requirements abstraction (Davis)



"If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organisation's needs. Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do. Both of these documents may be called the *requirements document* for the system."

Types of requirement

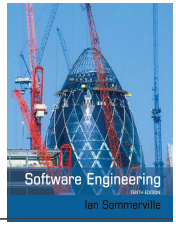


- T. 1 - User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers (**domain perspective**)
- T.2 - System requirements
  - A structured document setting out detailed descriptions of the system **services**. Can be written as a contract between client and contractor (**towards technical perspective**)

Aumentando il livello di dettaglio tecnico i requisiti diventano delle vere e proprie specifiche del sistema da costruire

- T.3 - Software specification (or Specification or System specification or Design specification)
  - A detailed software description which can serve as a basis for a design or implementation. Written for developers (software system , **technical perspective**)

Types of requirement



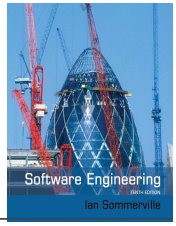
REQUISITI

- User requirements
  - Statements in natural language plus diagrams of th… the system provides and its operational constraints. Written for custo… **…main perspective)**
- System requirements
  - A structured document setting out detailed descriptions of the syste… …rvices. Can be written as a contract between client and contractor

SPECIFICHE

- Software specification (or Specification or System s… …on or Design specification)
  - A detailed software description which can serve … …s for a design or implementation. Written for developers (software … …n , **technical perspective**)

User and system requirements
Definitions and specifications



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 → sono date ai progettisti

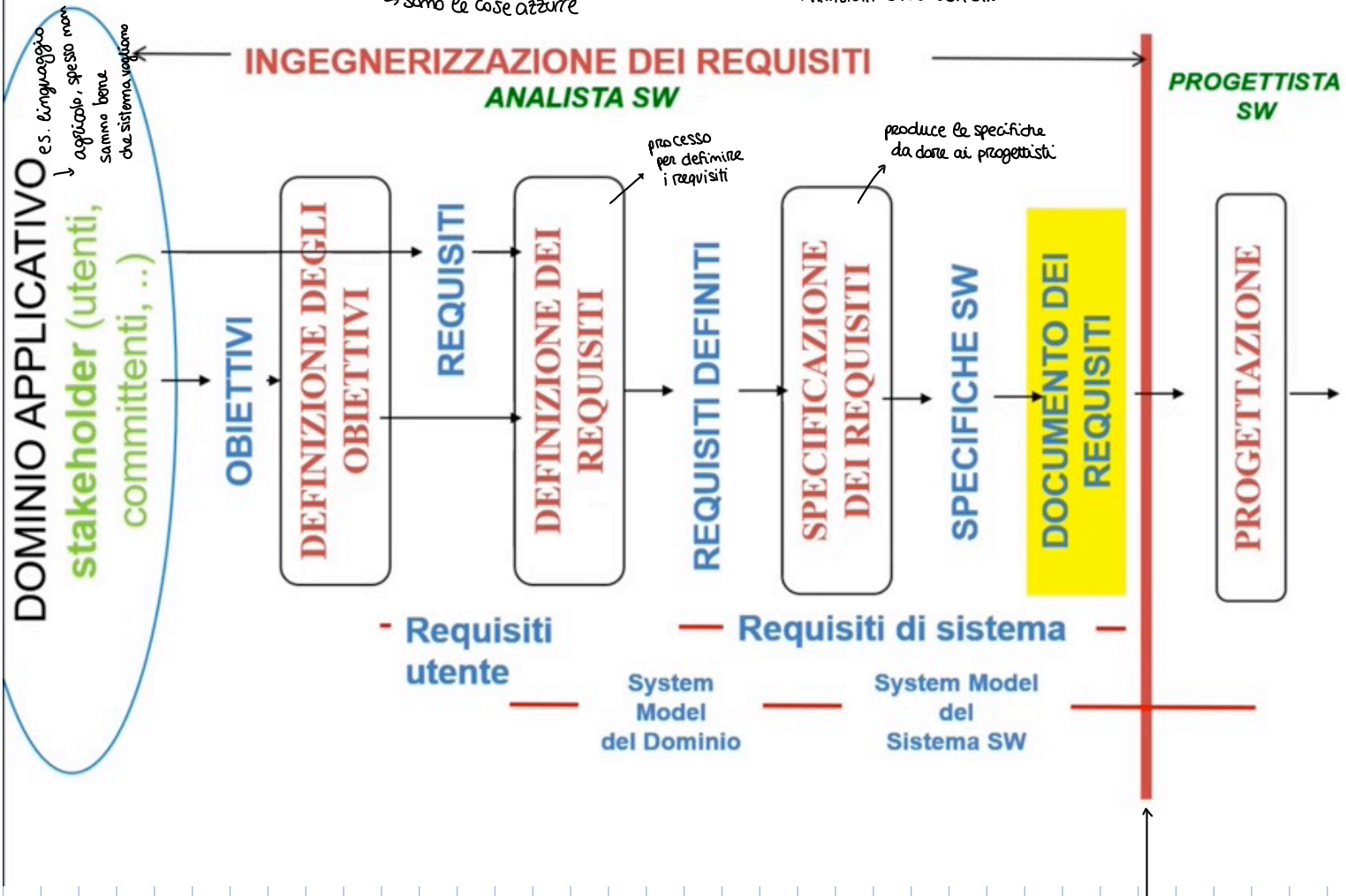
1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list 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.

Un visione complessiva del processo di trasformazione delle esigenze e degli obiettivi degli stakholder in 1. requisiti definiti e in 2. specifiche software

Processi/Input e Output coinvolti /Esecutori/Caratteristiche

↳ sono le cose azzurre

↳ ANALISTA e PROGGETTISTA



PROCESSO che passa dalle richieste dello stakeholder all'analisi dei requisiti (fino alla linea rossa)

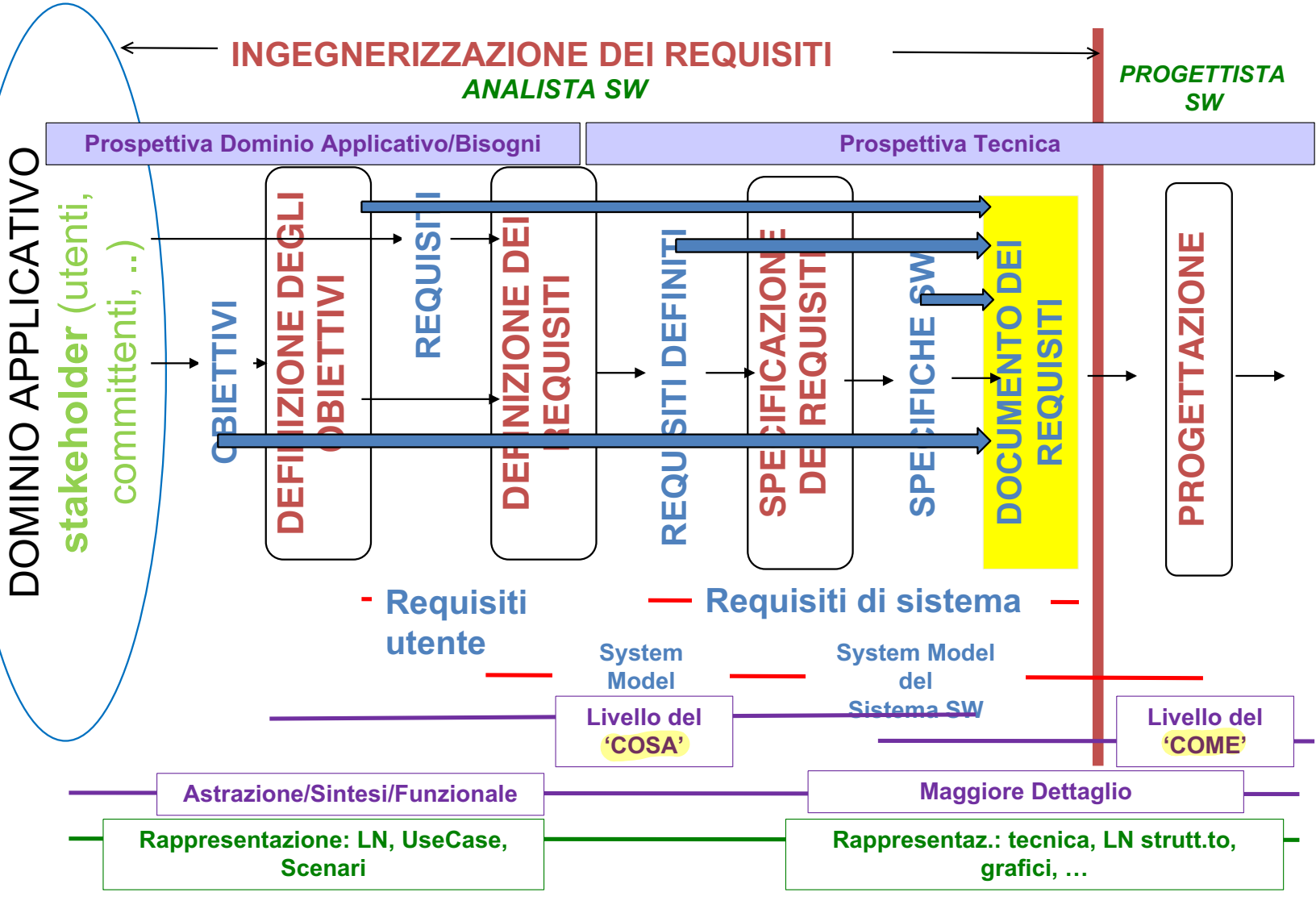
ogni passo ha dei ritorni indietro di verifica per evitare malintesi

SPESSO ANALISTA e PROGETTISTA coincidono

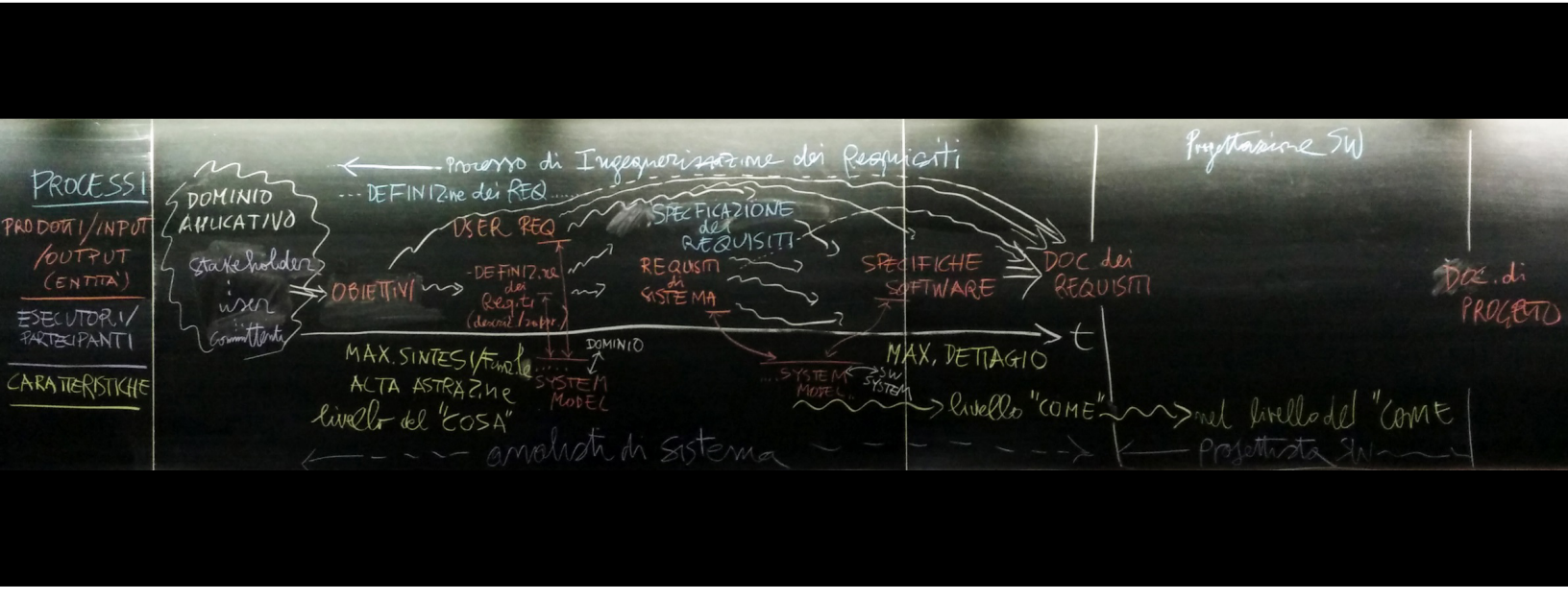
REQUISITI UTENTE → REQUISITI DI SISTEMA → SPECIFICHE → PROGETTAZIONE

Solitamente avvengono prima i REQUISITI e poi le SPECIFICHE, ma è possibile che delle volte da uno STAKEHOLDER arrivino le SPECIFICHE

Processi/Input e Output coinvolti /Esecutori/Caratteristiche

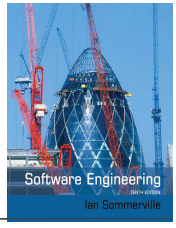


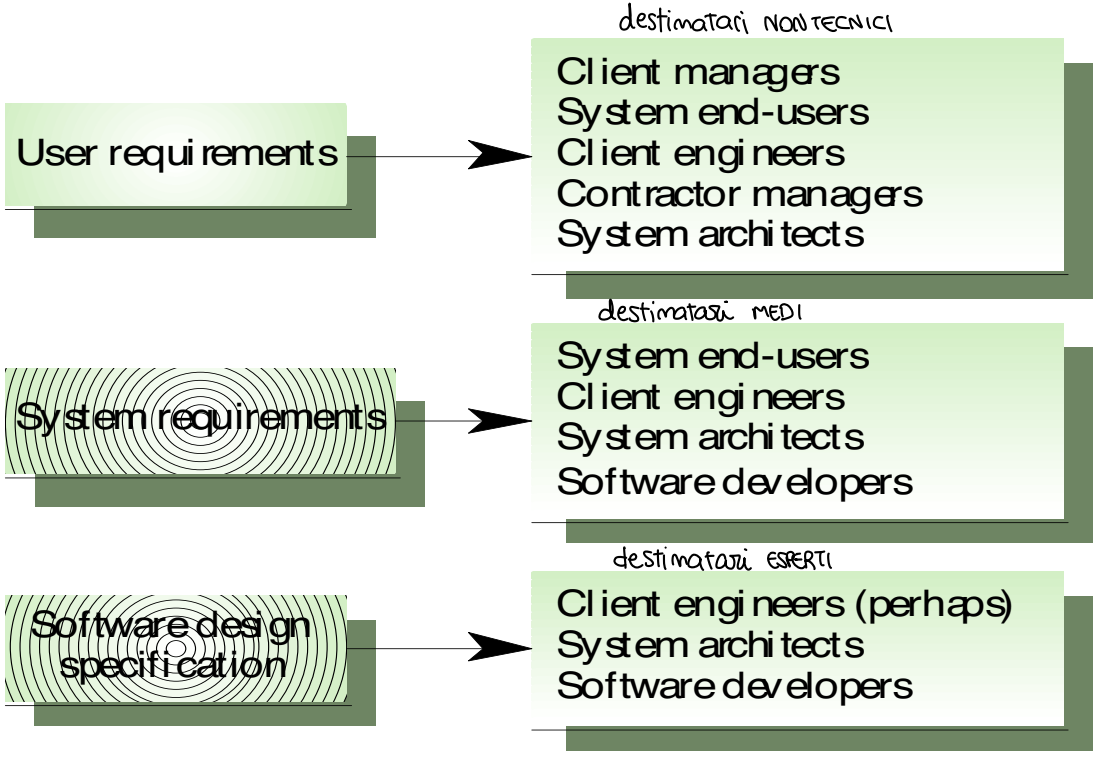
AA. 2015-2016


PROCESSI
PRODOTTI/INPUT /OUTPUT (ENTITA')
ESECUTORI/ PARTECIPANTI
CARATTERISTICHE
DOMINIO APLICATIVO
stakeholders
user
committenti
Percorso di Ingegnerizzazione dei Requisiti
DEFINIZione dei REQ
OBIETTIVI
USER REQ
DEFINIZione dei Requisiti (descrizione)
SPECIFICAZIONE dei REQUISITI
REQUISITI di SISTEMA
SPECIFICHE SOFTWARE
DOC dei REQUISITI
Progettazione SW
DOC. di PROGETTO
t
MAX. SINTESI/Funzionale
ALTA ASTRAZione
livello del "COSA"
DOMINIO
SYSTEM MODEL
MAX. DETTAGIO
SYSTEM MODEL
SW SYSTEM
livello "COME"
nel livello del "COME"
analista di sistema
progettista SW

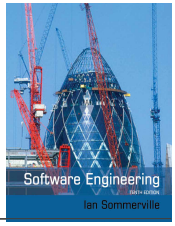
Requirements readers

↳ vari destinatari





System stakeholders

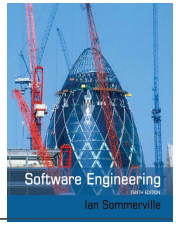


- Any person or organization who is affected by the system in some way and so who has a legitimate interest
- Stakeholders are the source of all the data and information which constitutes system goal, requirements (definition) and constraints

Stakeholder types

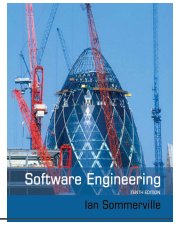
- End users
- System managers
- System owners
- External stakeholders

ESEMPIO: Stakeholders in the Mentcare system



- Patients whose information is recorded in the system.
- Doctors who are responsible for assessing and treating patients.
- Nurses who coordinate the consultations with doctors and administer some treatments.
- Medical receptionists who manage patients' appointments.
- IT staff who are responsible for installing and maintaining the system.

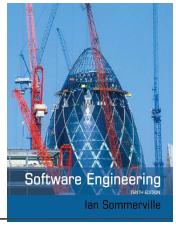
ESEMPIO Stakeholders in the Mentcare system



- A medical ethics manager who must ensure that the system meets current ethical guidelines for patient care.
- Health care managers who obtain management information from the system.
- Medical records staff who are responsible for ensuring that system information can be maintained and preserved, and that record keeping procedures have been properly implemented.

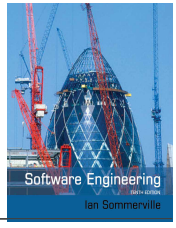
↳ il contrario del PLAN-DRIVEN

Agile methods and requirements



- Many agile methods argue that producing detailed system requirements is a waste of time as requirements change so quickly.
- The requirements document is therefore always out of date.
- Agile methods usually use incremental requirements engineering and may express requirements as 'user stories'.

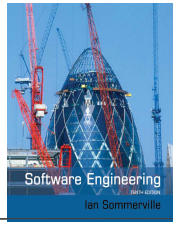
- This is practical for small business systems but problematic in several situations: for example for systems that require pre-delivery analysis (e.g. critical systems) or systems developed by several teams



Functional and non-functional requirements

Functional and Non-functional and Domain requirements

REQUISITI → FUNZIONALI / NON FUNZIONALI



FND.1 - Functional requirements

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

FND.2 - Non-functional requirements

- Constraints/caratteristiche/proprietà on the services or functions offered by the system such as timing constraints, constraints on memory usage, development process, standards, usabilità, etc.

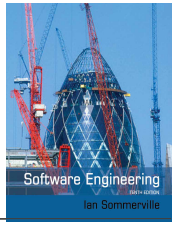
FND.3 - Domain requirements

↳ dominio dello stakeholder, es. dell'agricoltura. Le sue richieste vanno "tradotte"

- Requirements that come from the application domain of the system and that reflect characteristics of that domain

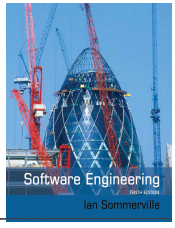
FND.1 - Functional requirements

=> descrizioni tecniche di COSA fa il SISTEMA : ovvero sono la traduzione tecnica delle richieste del cliente



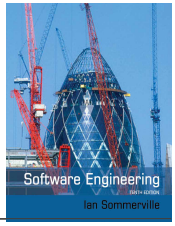
- Describe functionality or system services
- Depend on the type of software, expected users and the type of system where the software is used
- Functional user requirements may be high-level statements of what the system should do, but
- functional system requirements should describe the system services in detail (‘Cosa’ e non ‘Come’)

ESEMPIO: Mentcare system: functional requirements



1. A user shall be able to search the appointments lists for all clinics.
2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
3. Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

3C Rule: Criteria for Quality Control (validation) of Reqs. (Regola delle 3 C)



↳ misura il concetto di QUALITÀ

Correct

Each requirement should actually and accurately correspond to a need expressed by a stakeholder

Complete

They should include descriptions of all facilities required

Consistent

↱ nessun conflitto tra i requisiti

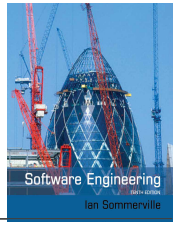
There should be no conflicts or contradictions in the descriptions of the system facilities

In practice, it is impossible to produce a complete and consistent requirements document

richiesta che se non sono rispettate posso buttare via tutto il sistema non può utilizzarlo perché ho stakeholder

FND.2 - Non-functional requirements

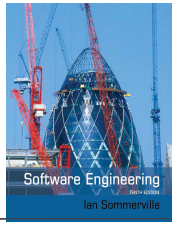
es. uso un DB che non è compatibile con i SW dello STAKEHOLDER



- Define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- Process requirements may also be specified mandating a particular CASE system, programming language or development method & tool, a standard, a protocol, ...
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless

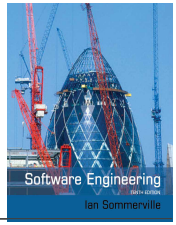
Non-functional classifications

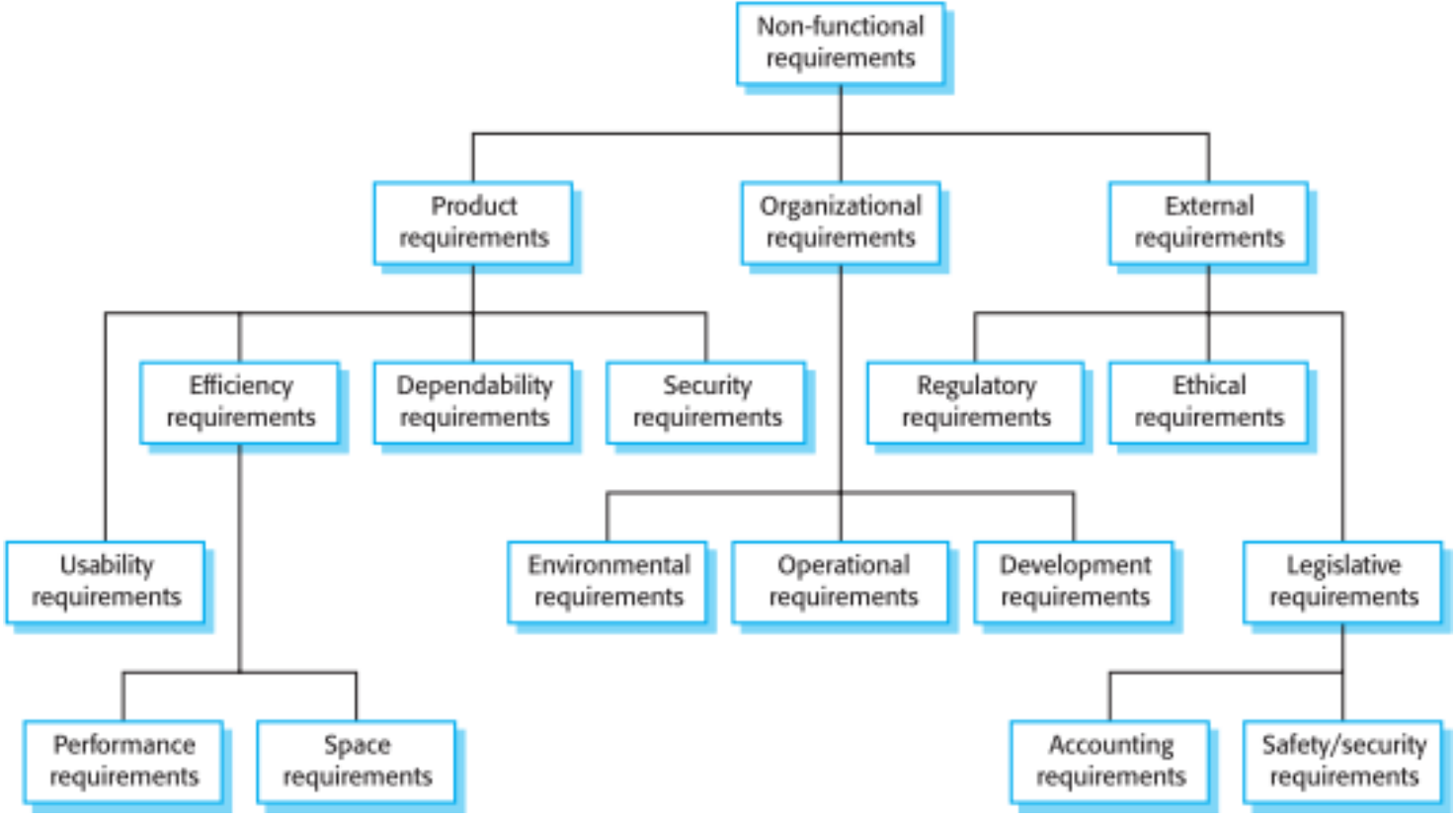
TIPI di NON FUNZIONALI:



- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements → es. fatto con un certo ambiente di sviluppo
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation and development tools requirements, programming languages, etc.
- External requirements → etica professionale
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Types of nonfunctional requirement

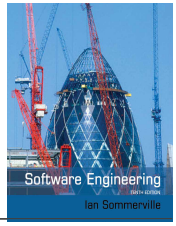




Non-functional requirements implementation

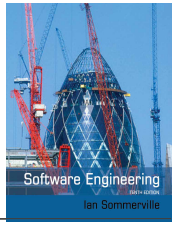
Non- Funct. Reqs. ⟶ Funct. Reqs.

↳ come dal NON funzionale emergono dei FUNZIONALI



- Non-functional requirements may cause as a consequence the introduction of related further functional requirements concerning the services that are required to manage/satisfy the non-functional requirement.
  - For example, privacy issues, GDPR, ecc..

Examples of nonfunctional requirements in the Mentcare system

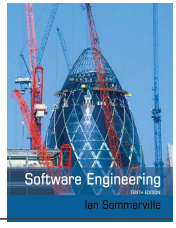


Product requirement
The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement
Users of the Mentcare system shall authenticate themselves using their health authority identity card.

External requirement
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

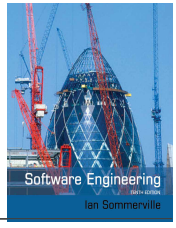
Goals (Obiettivi) and requirements



- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify. **Goals may be helpful to consider**.
- Gli obiettivi (goals) sono elementi importanti di per se nel SE process
- **Goal**
  - A general **intention** of the user such as '**ease of use**'

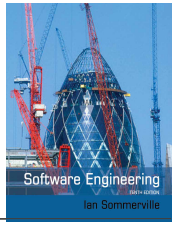
  IT SHOULD BE TRANSLATED INTO A:
- **Verifiable** non-functional requirement
  - A statement using some measure that can be objectively tested
- Goals are helpful to developers as they convey the intentions of the system users

Examples (of *usability* requirements)



- **A system goal**
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.
- **A verifiable non-functional requirement**
  - Experienced controllers shall be able to use all the system functions after a total of four hours training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use.

Metrics for specifying nonfunctional requirements



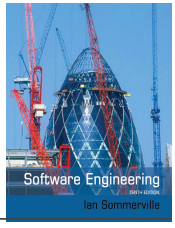
Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Recap

Obiettivi Req Specifiche

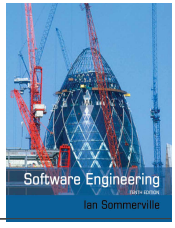
- Obiettivi:
  - **Formulazione sintetica e astratta degli scopi e intenzioni del cliente su: cosa vuol ottenere (funz.le o meno), che problema vuole risolvere, che situazione/prestazioni vuole raggiungere, le priorità, ecc.**
  - **Utili per comunicare, utili per analizzare req.&spec., utili per validare**
  - **Spesso poco verificabili**
- Requisiti: derivano dagli obiettivi, tradotti in modo verificabile e quindi di maggior dettaglio e precisione; prospettiva 'bisogni'
- Specifiche: ancor maggior dettaglio e precisione derivano dai Requisiti; prospettiva tecnica 'sistema sw'

FND.3 - Domain requirements



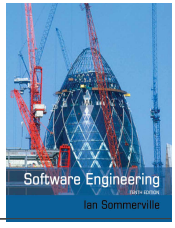
- Derived from the application domain and describe system characteristics and features that reflect the domain
- May be new functional requirements, constraints on existing requirements or define specific computations
- If domain requirements are not satisfied, the system may be unworkable

Esempi: Library system domain requirements

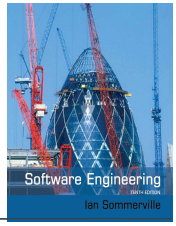


- There shall be a standard user interface to all databases which shall be based on the Z39.50 standard. (funct.)
- Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user's requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer. (constr. on exist. req.)

Domain requirements problems



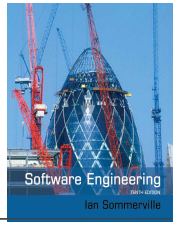
- Understandability, problem di vocabolario
  - Requirements are expressed in the language of the application domain
  - This is often not understood by software engineers developing the system
- Implicitness (vengono dati per scontati da parte degli stakeholder)
  - Domain specialists understand the area so well that they do not think of making the domain requirements explicit



Requirements engineering processes

- Processes used to discover, analyse, and validate system requirements -

Requirement analysis and software specification

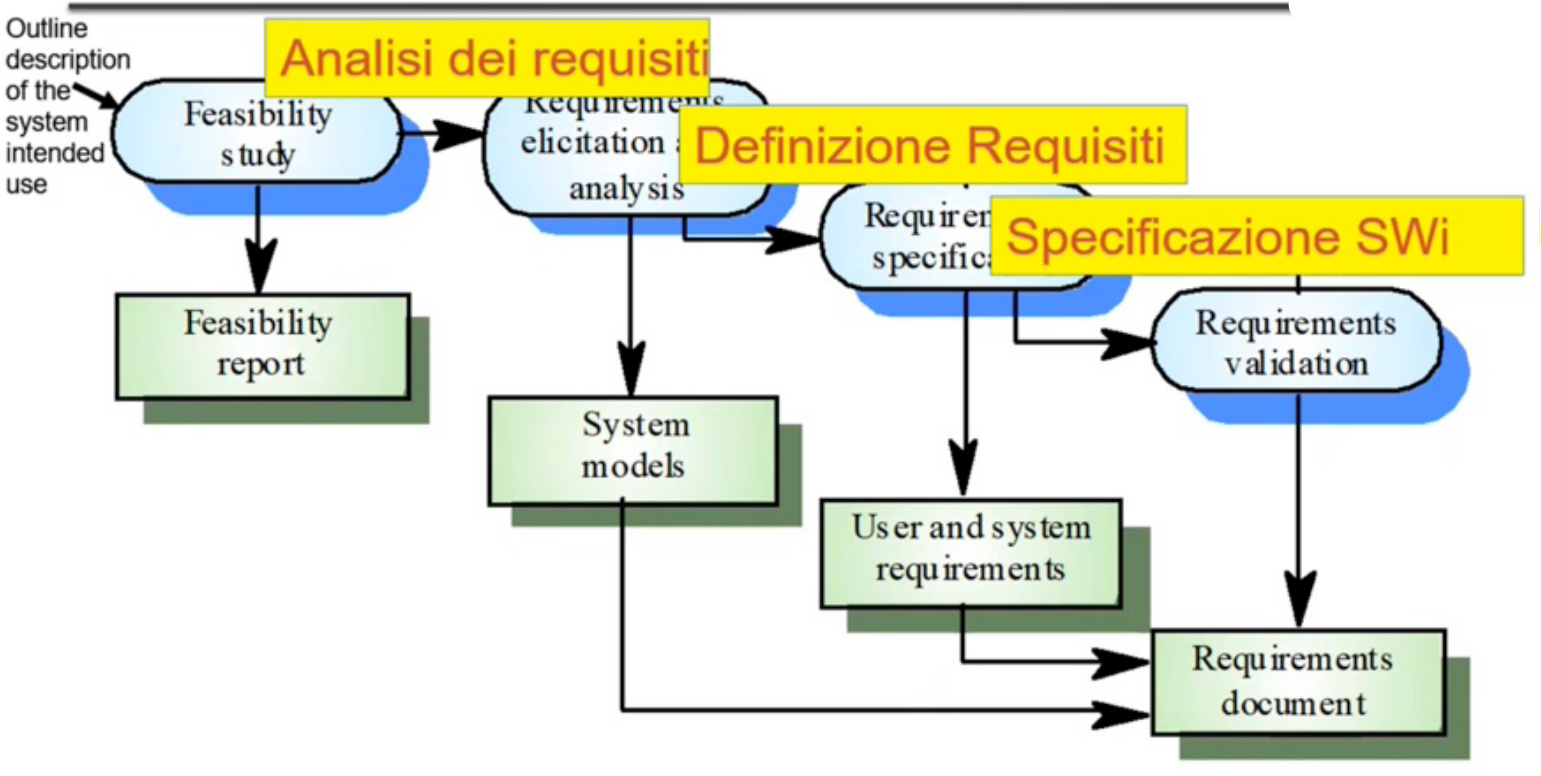


- The process of analysing needs and establishing what services are required and the constraints on the system's operation and development
- Si segue un preciso processo: Requirements engineering process
  - **Feasibility study**
  - **Requirements elicitation and analysis**
  - **Requirements specification**
  - **Requirements validation**

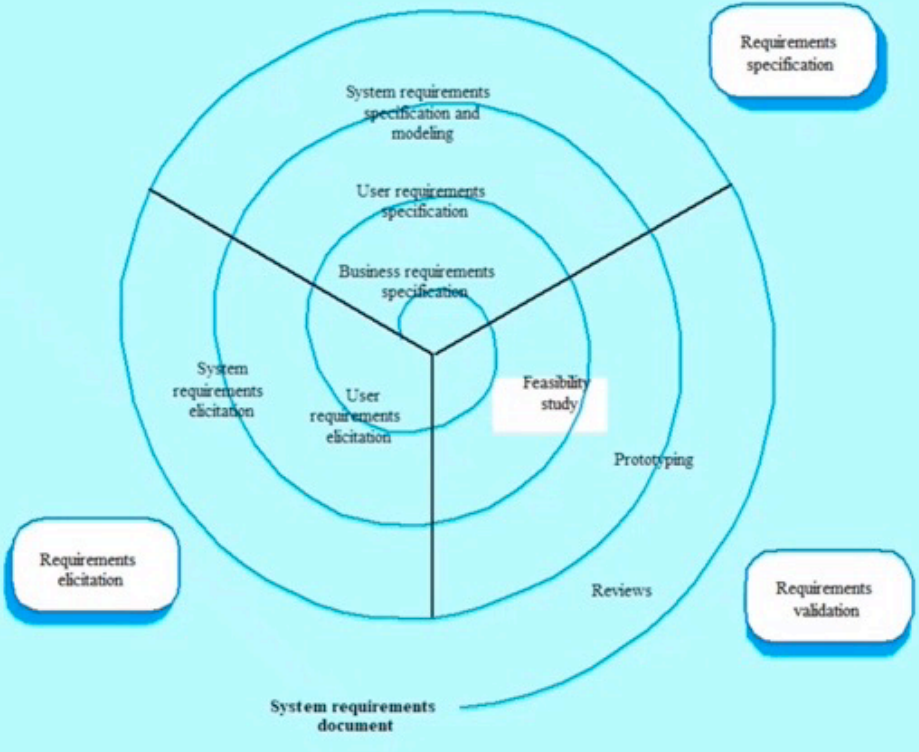
  e poi
  - Requirement managmenet

The ‘requirements engineering process’

process model → struttura il processo in sotto processi

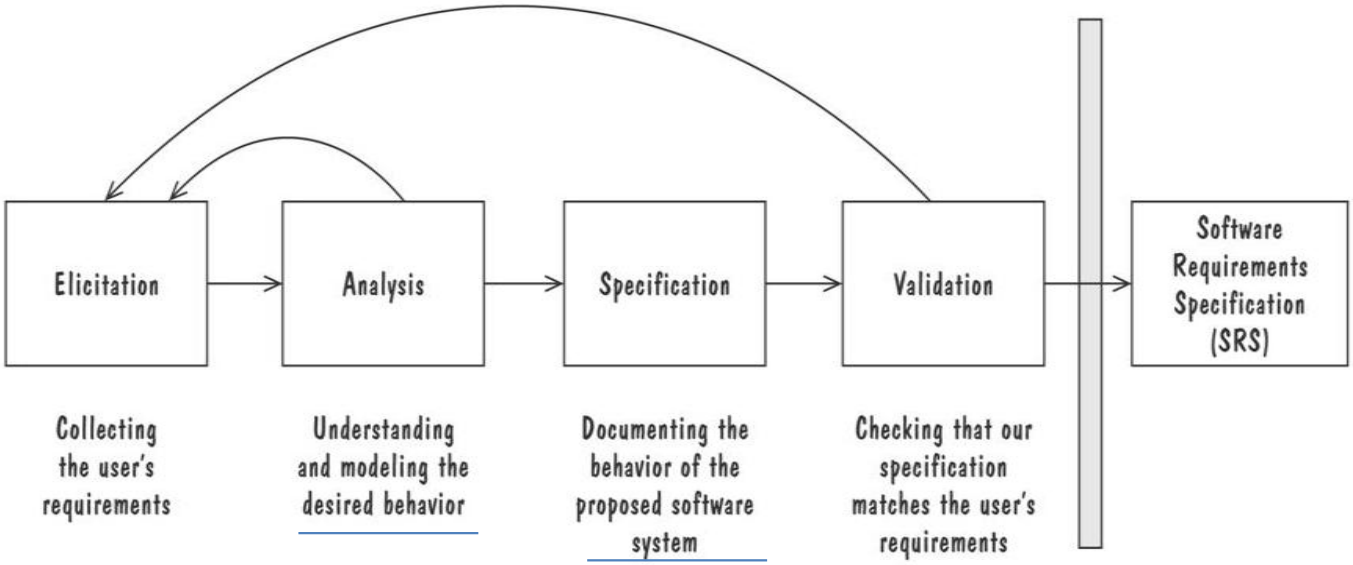


Un’altra rappresentazione, a spirale



Un'altra rappresentazione: The Requirements Process - Process for Capturing Requirements

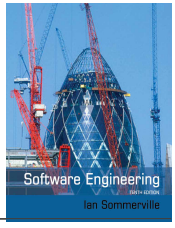
- Performed by the req. analyst or system analyst
- The final outcome is a Software Requirements Specification (SRS) document (Documento delle



(from Pfleeger and Atlee, © 2006 Pearson/Prentice Hall)

Feasibility study

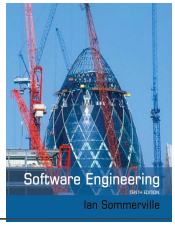
F.1 - Feasibility study (FATTIBILITÀ)



- A feasibility study decides whether or not the proposed system is worthwhile
- A short focused study that checks → chiedo il perché lo stakeholder vuole questo?
  a) If the system contributes to organisational objectives
  b) If the system can be **engineered** using current technology and within **budget**
  c) If the system can be integrated with other systems that are used
  d) Preliminary **economic** analysis

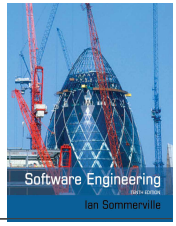
- Viene presentato ai decisori che decidono se continuare con le fasi successive del progetto

Feasibility study implementation

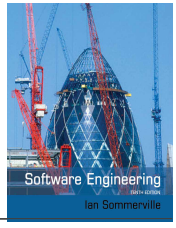


- Based on information assessment (capire what info is required), information collection&analysis and report writing. Duration: a few (2-6) weeks. (more – several months - in complex cases)
- Questions for people in the organisation
  - What are current process problems/criticalities?
  - How will the proposed system help?
  - What if the system wasn't implemented?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?

Feasibility study: results

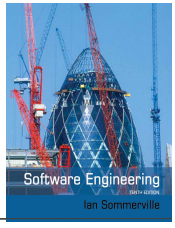


- Lo studio di Fattibilità viene presentato al decisore che ha la responsabilità di decidere in merito alla continuazione del progetti
- L'outcome di questa fase decisionale può essere:
  - Should the project continue or not?
  - Possible proposal of changes in scope, budget, schedule
  - Possible proposal of general req.s
- UNA buona idea:
  - Nel corso dello studio, esaminare più alternative, in presentare al decisore molteplici alternative.
  - Ad esempio, se i costi sono eccessivi, ipotizzare un Sistema (i) che non includa una parte che ha costo elevato, ovvero (ii) che non includa parti acessorie non urgenti, ecc., …



Requirements Elicitation &Analysis

F.2 - Elicitation and analysis

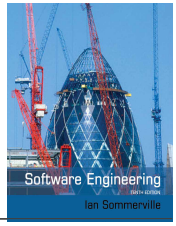


- Sometimes called requirements *discovery*

↳ spesso lo stakeholder non ha le idee chiare

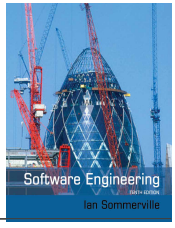
- Involves technical staff working with customers (stakeholders) to find out about the application domain, the services that the system should provide and the system's operational constraints
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called ***stakeholders,*** tutti coloro che hanno un interesse diretto o indiretto sui requisti del sistema sw da sviluppare.

Possible stakeholder (portatori di interesse)



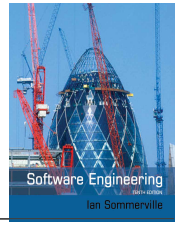
- **Clients**: pay for the software to be developed
- **Customers**: buy the software after it is developed
- **Users**: use the system
- **Domain experts**: familiar with the problem that the software must automate
- **Market Researchers**: conduct surveys to determine future trends and potential customers
- **Lawyers or auditors**: familiar with government, safety, or legal requirements
- **Software engineers** or other technology experts

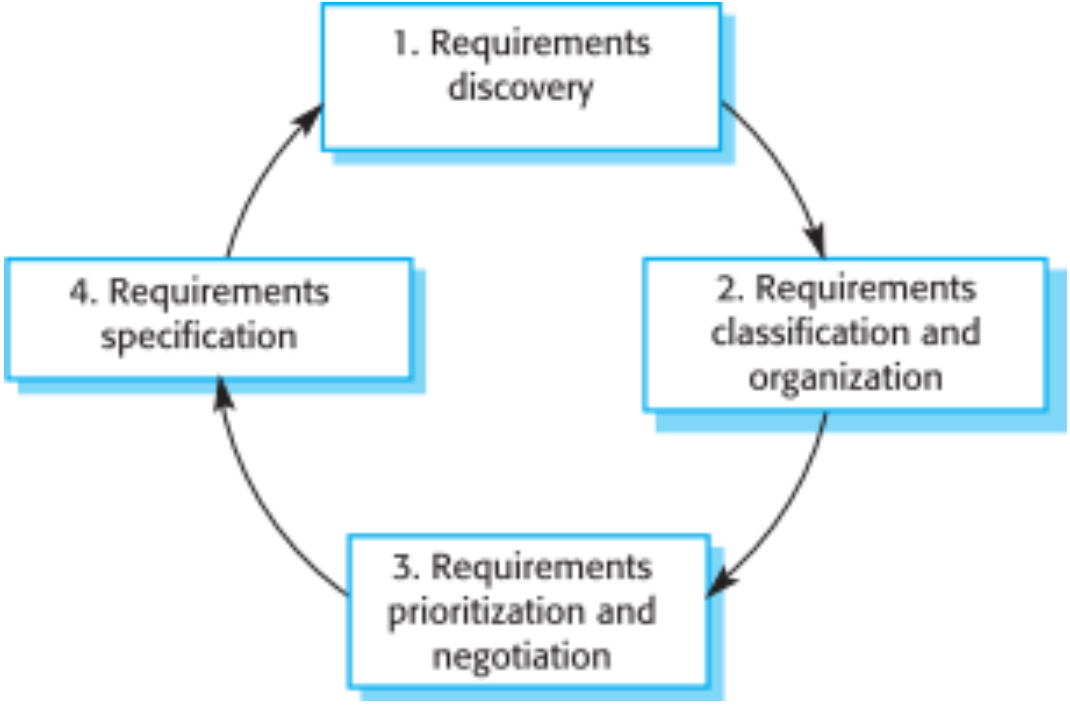
Problems of requirements analysis



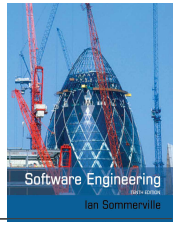
- Stakeholders don't know what they really want or they are not able to articulate what they want
- Stakeholders express requirements in their own terms
- Different stakeholders may have different views, resulting in different priorities and conflicting requirements
- Organisational and political factors may influence the system requirements
- The requirements change during the analysis process. Moreover, new stakeholders may emerge and the business environment changes → può succedere che spesso avvengano cambiamenti dei requisiti durante tutto il progetto

Iteration in the requirements analysis process (process model)



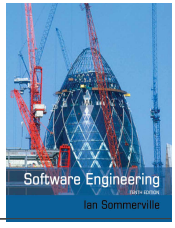


Process activities for Elicitation and Analysis



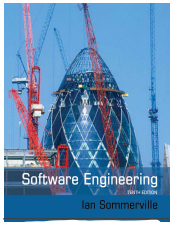
1. Requirements discovery
   - Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.
2. Requirements classification and organisation
   - Groups related requirements and organises them into coherent clusters.
3. Prioritisation and negotiation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

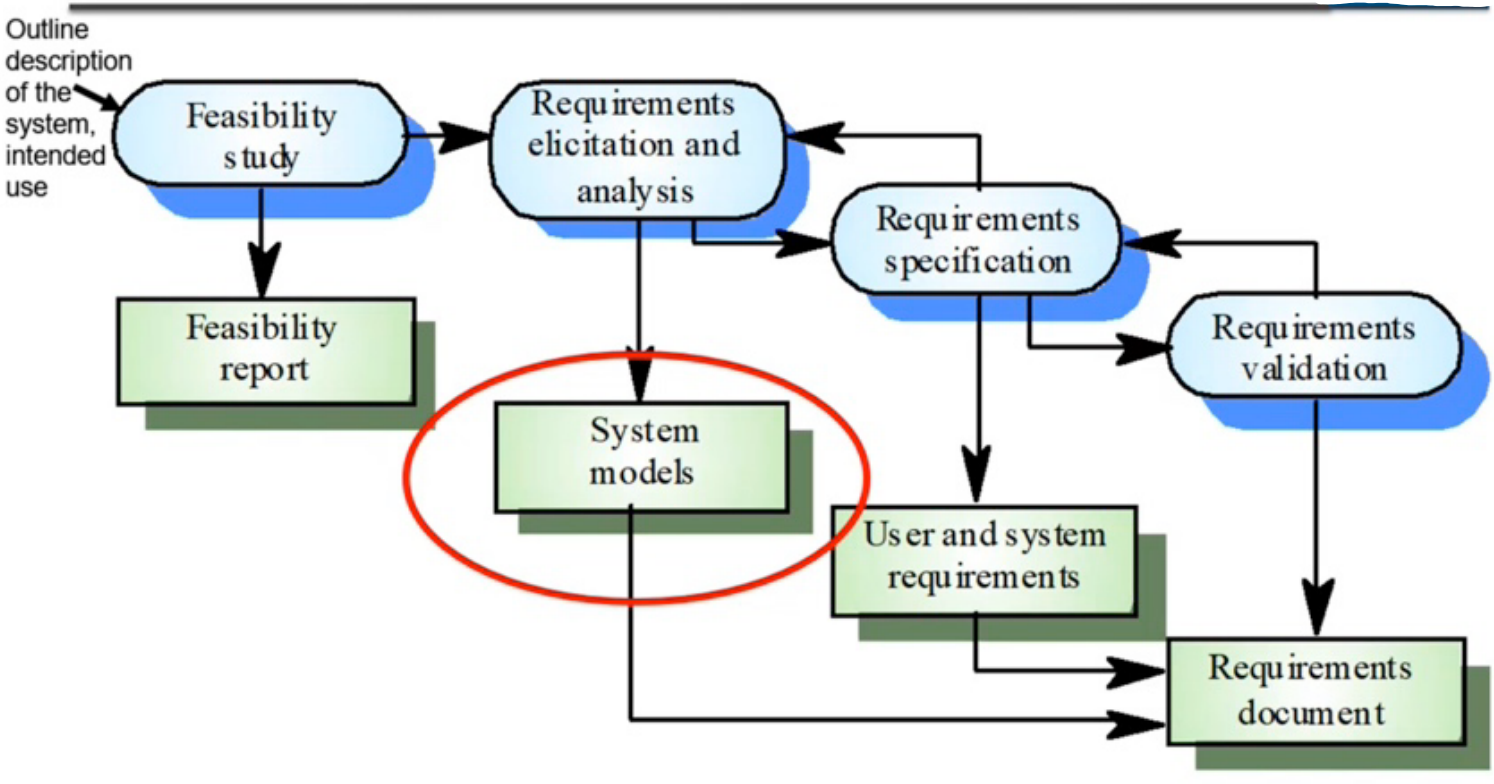
Requirements discovery



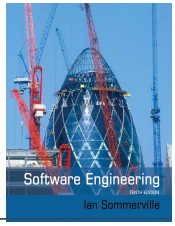
- The process of gathering information about the proposed and existing systems and *distilling* the user and system requirements from this information.
- **Sources of information** include documentation, system stakeholders and the specifications of similar systems.

The requirements engineering process





System models



- Different models may be produced during the requirements analysis activity
- 2 tipi di modelli:

  In fase iniziale di **definizione dei requisiti**

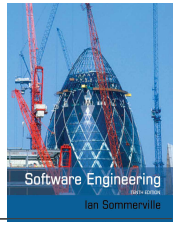
  - Modelli del dominio, e dei processi su cui si focalizza il progetto

  Poi, nella **specificazione sw**

  - Modelli del sistema che si sta specificando per soddifare I requisiti emersi
- Ne analizzeremo diversi nei capitol successivi

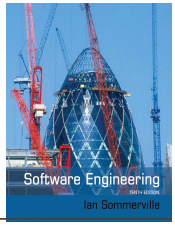
Several Tecniques for Eliciting Requirements

↳ estrapolare informazioni



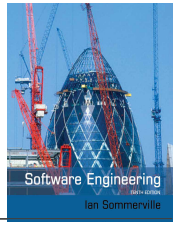
- Interviewing stake holders
- Reviewing available documentation
- Observing the current system (if one exists)
- Ethnography → studiare la realtà dello stakeholder ↳ dominio
- Apprenticing with users to learn about user's task in more details
- Interviewing users or stakeholders in groups
- Brainstorming and Focus Group with current and potential users
- User Stories, Scenario, Role Playing, Use Cases, Storyboarding, Prototyping

Interviewing



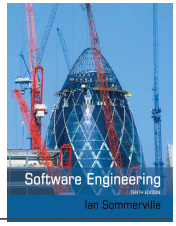
- In formal or informal interviewing, the RE team puts questions to stakeholders about the system that they use and the system to be developed.
- There are two types of interview
  - Closed interviews where a pre-defined set of questions are answered.
  - Open interviews where there is no pre-defined agenda and a range of issues are explored with stakeholders.

Interviews in practice



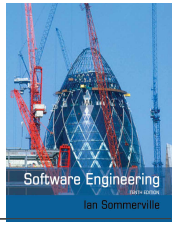
- Normally a mix of closed and open-ended interviewing.
- Three kinds of interview:
  1. 'Tutorial interview', informal, open question on general topics
  2. 'Structured interview'. More formal, open/closed questions about some topic already examined
  3. 'Focussed interview', more formal, open/closed questions about some very specific aspect, to be completed, to be better understood, to be corrected, conflicting, ..
- Tutorial interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- For understanding in detail specific requirements Structured/Focussed interviews are more suitable.
- Nota. Su tematiche organizzative/politiche (struttura del processo decisionale, chi decide cosa, …) può essere difficile acquisire informazioni tramite interviste

Problems with interviews



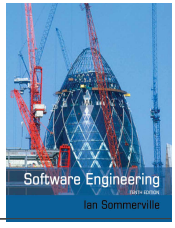
- Problemi di VOCABOLARIO: Application specialists may use language to describe their work that isn't easy for the requirements engineer to understand.
- Difficoltà con le CARATTERISTICHE del DOMINIO: Interviews are not good for understanding domain requirements → mi devo informare meglio io a parte sul dominio dello stakeholder perché nelle interview da delle cose del suo dominio scontate
  - 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.

Ethnography



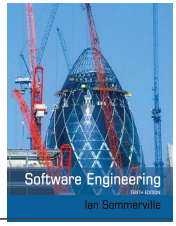
- Osservazione dei processi che interessano
- A social scientist spends a considerable time observing and analysing how people actually work
- People do not have to explain or articulate their work
- Social and organisational factors of importance may be observed
- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models

Scope of ethnography



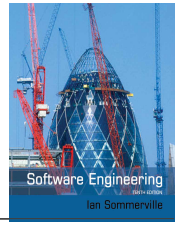
- Requirements that are derived from the way that people actually work rather than the way in which process definitions suggest that they ought to work
- Requirements that are derived from cooperation and awareness of other people's activities

User stories



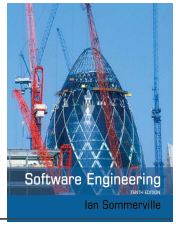
- Per alcuni stakeholder è molto più facile descrivere una situazione reale (attuale o futura), piuttosto che 'fornire requisiti'.
- Le storie sono descrizioni molto generali di una situazione/interazione.
- Ad esempio, di come il nuovo Sistema si situerebbe nel dominio e quali funzionalità generali potrebbe fornire.
- Quando da una storia si procede verso un maggior livello di dettaglio di solito si parla di Scenari
- L'approccio '**user stories**' è uno delle caratteristiche tipiche degli approcci **AGILE**.

Esempio di Storia (user story) Photo sharing in the classroom



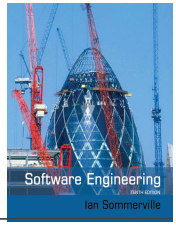
Jack is a primary school teacher in Ullapool (a village in northern Scotland). He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing. As part of this, pupils are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history resources site) to access newspaper archives and photographs. However, Jack also needs a photo sharing site as he wants pupils to take and comment on each others' photos and to upload scans of old photographs that they may have in their families.

Jack sends an email to a primary school teachers group, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account. He uses the iLearn setup service to add KidsTakePics to the services seen by the pupils in his class so that when they log in, they can immediately use the system to upload photos from their mobile devices and class computers.



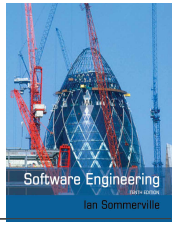
Requirements specification

F. 3 Requirements specification

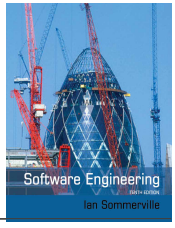


- The process of writing down the user and system requirements in a requirements document.
- User requirements have to be understandable by end-users and customers who do not have a technical background.
- System requirements are more detailed requirements and may include more technical information.
- The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

Requirements (cosa) and design (come)

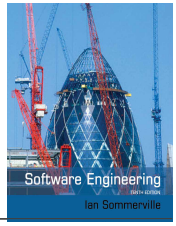


- In principle, requirements should state what the system should do and the design should describe how it does this.
- In practice, requirements and design are inseparable
  - A system architecture may be designed to structure the requirements;
  - The system may inter-operate with other systems that generate design requirements;
  - The use of a specific architecture to satisfy non-functional requirements may be a domain requirement.



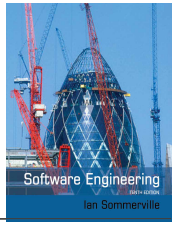
Come si rappresentano i requisiti?

Ways of writing a system requirements specification



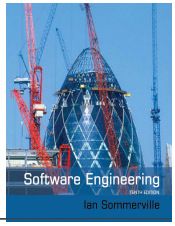
Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

Natural language specification



- Requirements are written as natural language sentences supplemented by diagrams and tables.
- Used for writing requirements because it is expressive, intuitive and universal. This means that the requirements can be understood by users and customers.

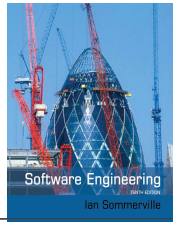
Problems with natural language in requirement definition



- Lack of clarity
  - Precision is difficult without making the document difficult to read
- Requirements confusion / mix
  - Functional and non-functional requirements tend to be mixed-up, general/conceptual and detailed requirements are mixed
- Requirements amalgamation → non mettere assieme cose diverse !!
  - Several different requirements may be expressed together

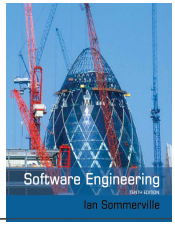
è un blocco lineare di informazioni

Example: Database requirement



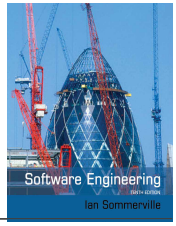
4.A.5 The database shall support the generation and control of configuration objects; that is, objects which are themselves groupings of other objects in the database. The configuration control facilities shall allow access to the objects in a version group by the use of an incomplete name. (**conceptual level** and **detail level mixed**).

Example: Editor grid requirement



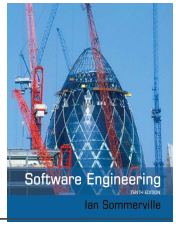
2.6 Grid facilities To assist in the positioning of entities on a diagram, the user may turn on a grid (**conceptual, functional**) in either centimetres
or inches (**non-functional**), via an option on the control panel. Initially, the grid is off. The grid may be turned on and off (**non-functional**) at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines (**dettagli, ma non completi**).

Example: Requirement problems



- The previous example on Database requirements include both conceptual and detailed information
  - Describes the concept of configuration control facilities
  - Includes the detail that objects may be accessed using an incomplete name
- The previous Grid requirement mixes three different kinds of requirement
  - Conceptual functional requirement (the need for a grid)
  - Non-functional requirement (grid units)
  - Non-functional UI requirement (grid switching)

Guidelines for writing requirements

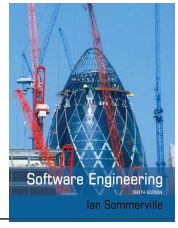


↳ per dire qualcosa mi esprimo sempre nello stesso modo

- Design a standard format and use it for all requirements
- Use language in a consistent(*) way. Use shall (deve) for mandatory requirements, should (dovrebbe) for desirable requirements
- Use text highlighting to identify key parts of the requirement
- Avoid the use of computer jargon

(*) Consistente: stessi termini per gli stessi concetti, non elidere parti del discorso se l'elisione può arrecare ambiguità, usare sempre lo stesso tipo di costruzioni delle frasi, usare sempre lo stesso stile tipografico, non introdurre differenze di stile se non in corrispondenza a differenze sottostanti più 'profonde', usare sempre termini il cui significato è chiaro e condiviso (cfr. data dictionary)…

Example requirements for the insulin pump software system



tra parentesi metti la motivaz della tua scelta, cambio di stile

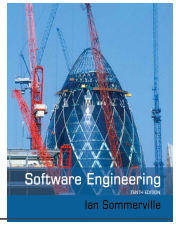
3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. *(Changes in blood sugar are relatively slow so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.)*

3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. *(A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.)*

appena inizia la fase di progettazione, teoricamente non si dovrebbe più interagire con gli stakeholder, dovrebbero essere chiare le idee e senza dubbi

Rappresentare le Specifiche

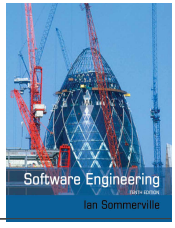
↳ diventano l'input per la fase di progettazione



- Il Linguaggio Naturale è il principale metodo di rappresentazione per i requisiti utente, ma le specifiche entrano maggiormente nei dattagli e nel livello tecnico
- SW Specifications are more detailed/tecnical of user requirements
- Serve as a basis for designing the system
- May be used as part of the system contract
- System requirements may be expressed using system models.
- Il linguaggio naturale viene sempre più affiancato da strumenti (notazioni) più sintetici, immediati, con semantica più definita (e quindi limitata), meno soggetta ad ambiguità.

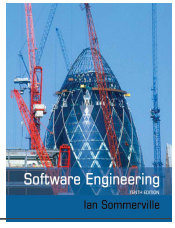
Problems with NL specification

→ natural language



- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult
- Over-flexibility → posso dire una cosa in più modi
  - The same thing may be said in a number of different ways in the specification
- Lack of modularisation, linearità del LN
  - NL structures are inadequate to structure system requirements

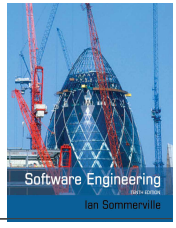
Alternatives



Overcoming limitations and criticalities of plain Natural Language

linguaggio naturale strutturato → lo uso per un requisito funzionale ha una struttura e rappresentazione più sistematica

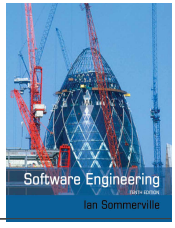
Structured language specifications



- A limited form of natural language may be used to express requirements
- The freedom of the requirements writer is limited by a predefined template for requirements.
- All requirements are written in a standard way.
- The terminology used in the description may be limited.
- The advantage is that the most of the expressiveness of natural language is maintained but a degree of uniformity is imposed on the specification.
- Often supported using a forms-based approach

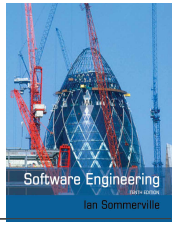
rappresentazione grafica

Form-based specifications



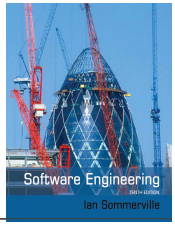
- Definition of the function or entity
- Description of inputs and where they come from
- Description of outputs and where they go to
- Indication of other entities required
- Pre and post conditions (if appropriate)
- The side effects (if any)
- Tracking information (from where the specification comes, where the specification is used)

Example: Form-based node specification



ECLIPSE/Workstation/Tools/DE/FS/3.5.1	
Function	Add node
Description	Adds a node to an existing design. The user selects the type of node, and its posi When added to the design, the node becomes the current selection. The user chooses the node positio moving the cursor to the area where the node is added.
Inputs	Node type, Node position, Design identifier.
Source	Node type and Node position are input by the user, Design identifier from the dat
Outputs	Design identifier.
Destination	The design database. The design is committed to the database on completion of operation.
Requires	Design graph rooted at input design identifier.
Pre-condition	The design is open and displayed on the user's screen.
Post-condition	The design is unchanged apart from addition of a node of the specified ty at the given position.
Side-effects	None
Definition: ECLIPSE/Workstation/Tools/DE/RD/3.5.1	

A structured specification of a requirement for an insulin pump/parte 1



Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: safe sugar level.

Description

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

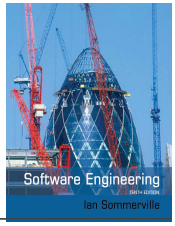
Inputs Current sugar reading (r2); the previous two readings (r0 and r1).

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose—the dose in insulin to be delivered.

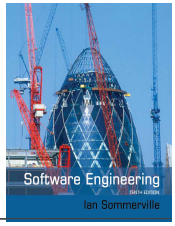
Destination Main control loop.

A structured specification of a requirement for an insulin pump/ parte 2



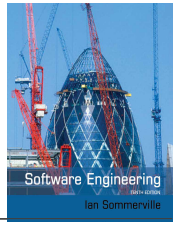
Action
CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.
Requirements
Two previous readings so that the rate of change of sugar level can be computed.
Pre-condition
The insulin reservoir contains at least the maximum allowed single dose of insulin.
Post-condition r0 is replaced by r1 then r1 is replaced by r2.
Side effects None.

Tabular specification



- Used to supplement natural language.
- Particularly useful when you have to define a number of possible alternative courses of action.
- For example, the insulin pump systems bases its computations on the rate of change of blood sugar level and the tabular specification explains how to calculate the insulin requirement for different scenarios.

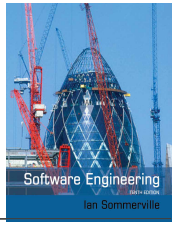
Tabular specification of computation for an insulin pump



Condition	Action
Sugar level falling (r2 < r1)	CompDose = 0
Sugar level stable (r2 = r1)	CompDose = 0
Sugar level increasing and rate of increase decreasing ((r2 – r1) < (r1 – r0))	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ((r2 – r1) ≥ (r1 – r0))	CompDose = round ((r2 – r1)/4) If rounded result = 0 then CompDose = MinimumDose

Graphical models

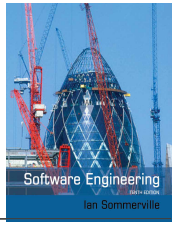
↳ modelli grafici



- Graphical models are most useful when you need to show how state changes or where you need to describe a sequence of actions or the structural composition properties.
- Different graphical models are explained later (DFD, Petri Nets, UML)

↳ reti di Petri: quando devo visualizzare molte attività che hanno dipendenze tra loro molto complicate
es. "questo processo può iniziare quando quello è terminato"

UML Use cases



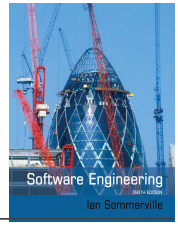
casi d'uso

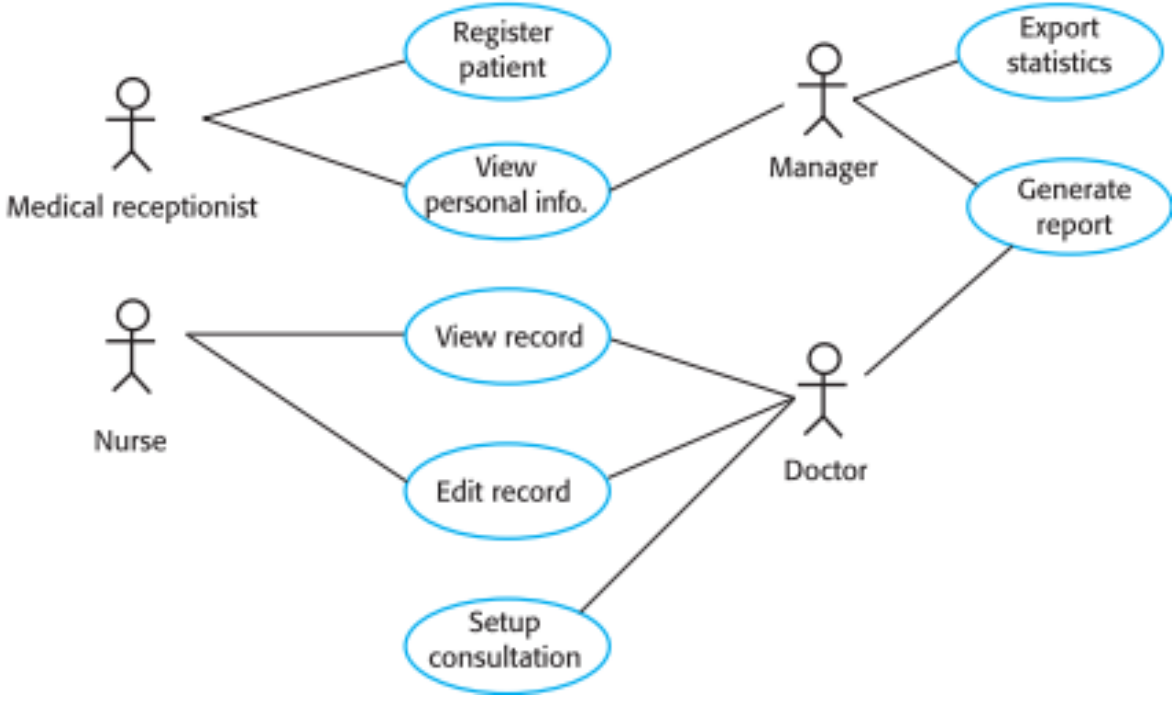
- Use-cases are a **scenario based technique** in the **UML** which identify the actors in an interaction and which describe the interaction itself
- A set of use cases should describe all possible interactions with the system
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system

Baruzzo

Use cases for the Mentcare system

sistema sanitario → uml del sistema ospedaliero

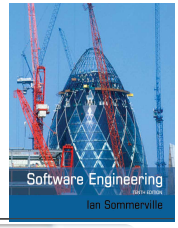


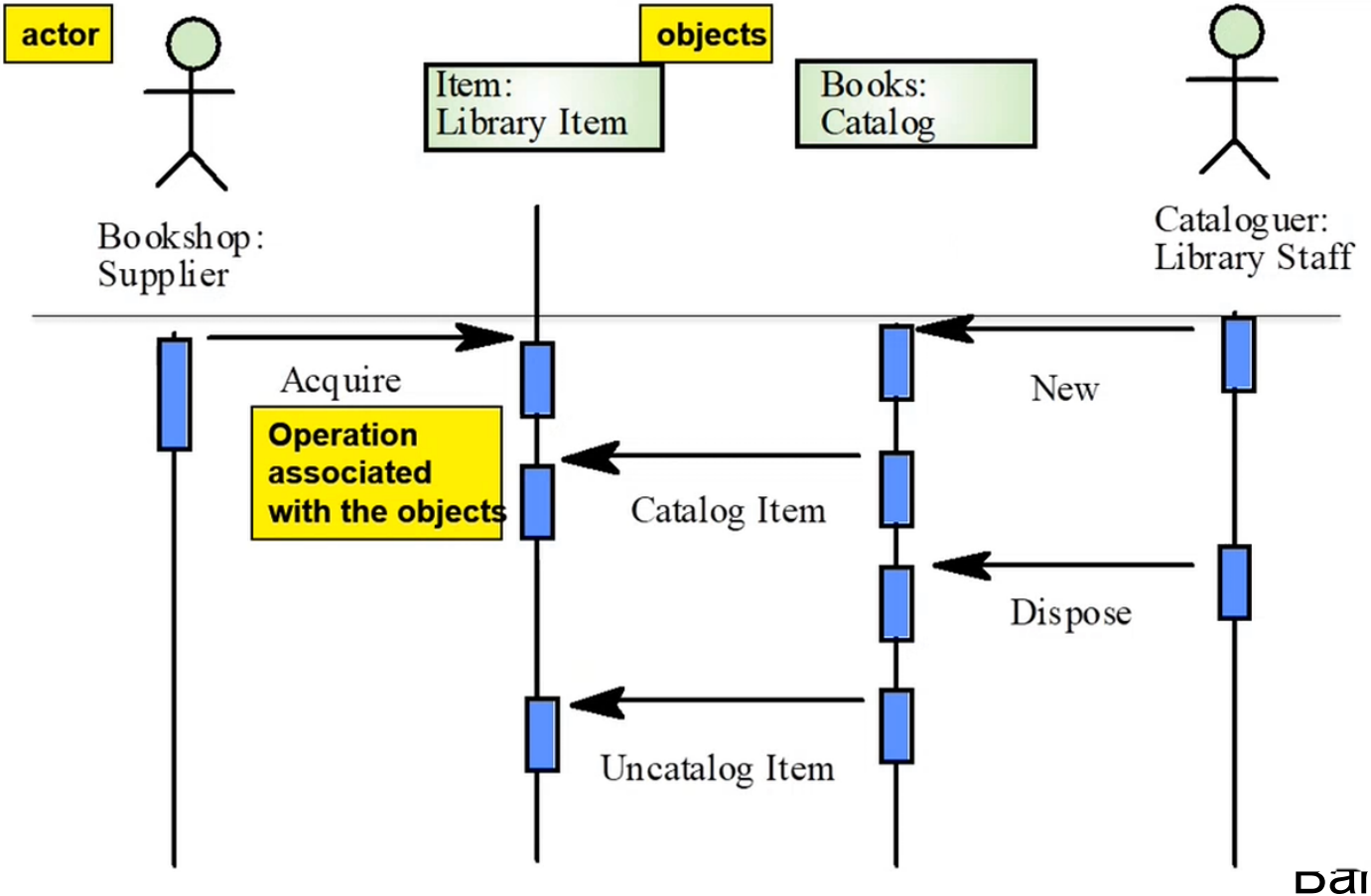


Baruzzo

UML Sequence diagrams: Catalogue management example

UML di uma biblioteca

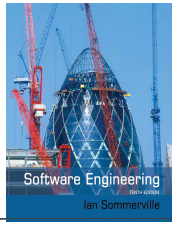




UN ALTRO POSSIBILE MODO di RAPPRESENTAZIONE:

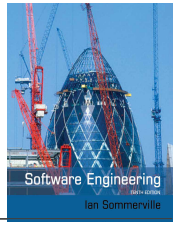
PDL - PseuDoLanguage-based requirements definition

↳ linguaggi di progettazione, solo per gente tecnica
non sono schemi grafici, servono a noi non per lo stakeholder



- Requirements may be defined operationally using a language like a programming language but with more flexibility of expression
- Most appropriate in two situations
  - Where an operation is specified as a sequence of actions and the order is important
  - When hardware and software interfaces have to be specified

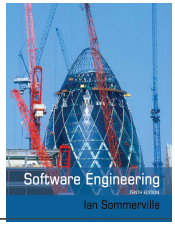
Example: Part of an ATM specification




```
class ATM {
        // declarations here
        public static void main (String args[]) throws InvalidCard {
                try {
                        thisCard.read () ;  // may throw InvalidCard exception
                        pin = KeyPad.readPin () ; attempts = 1 ;
                        while ( !thisCard.pin.equals (pin) & attempts < 4 )
                                {       pin = KeyPad.readPin () ;  attempts = attempts + 1 ;
                                }
                                if (!thisCard.pin.equals (pin))
                                        throw new InvalidCard ("Bad PIN");
                        thisBalance = thisCard.getBalance () ;
                        do {  Screen.prompt (" Please select a service ") ;
                                service = Screen.touchKey () ;
                                switch (service) {
                                        case Services.withdrawalWithReceipt:
                                                receiptRequired = true ;
```

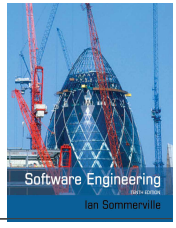
PDL disadvantages

↳ per gli stakeholders è difficile comprendere questo linguaggio



- PDL may not be sufficiently expressive to express the system functionality in an understandable way
- Notation is only understandable to technical people with programming language knowledge
- The requirement may be erronously taken as a design specification rather than a model to help understand the system => lo PSEUDOCODICE NON È UNA PRIMA VERSIONE del PROGRAMMA!!
  ↳ Siamo ancora nella fase di: "cosa deve fare il sistema?", il come verrà in fase di progettazione

Example: PDL interface description



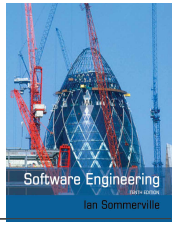
L'approccio PDL è molto utile anche per specificare l'interfaccia tra sottosistemi del sistema o con altri sistemi esterni.

```
interface PrintServer {

// defines an abstract printer server
// requires:          interface Printer, interface PrintDoc
// provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter

        void initialize ( Printer p ) ;
        void print ( Printer p, PrintDoc d ) ;
        void displayPrintQueue ( Printer p ) ;
        void cancelPrintJob (Printer p, PrintDoc d) ;
        void switchPrinter (Printer p1, Printer p2, PrintDoc d) ;
} //PrintServer
```

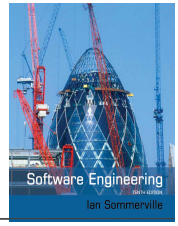
The requirements document (Documento di Specifica o Documento dei Requisiti)



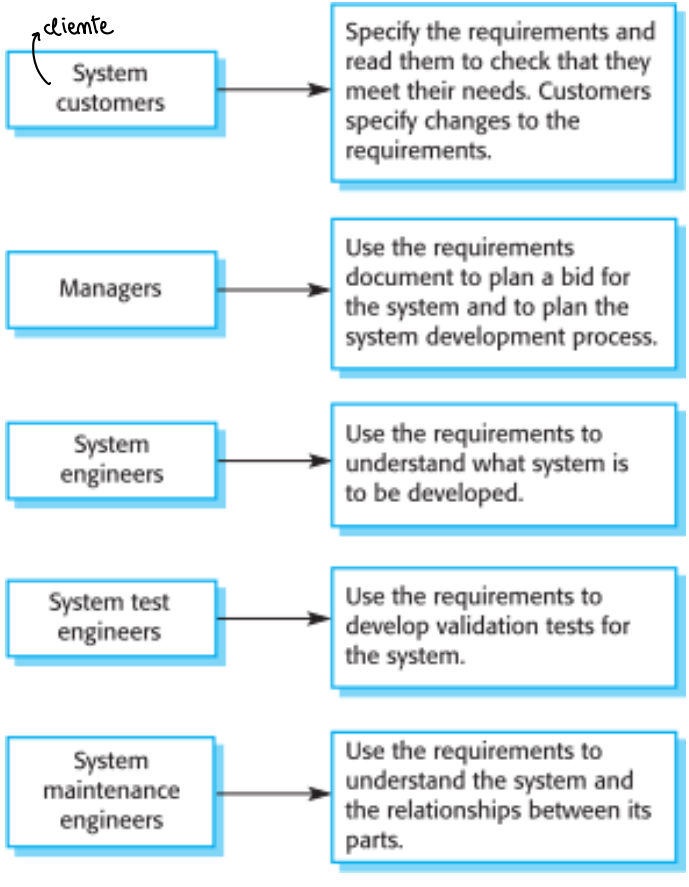
Tutto confluisce dentro al documento dei requisiti

- The requirements document is the official statement of what is required of the system developers
- Should include **both** a definition and a specification of requirements
- It is NOT a design (= progettazione) document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it

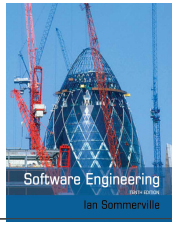
Users of a requirements document



A CHI VIENE RAPPRESENTATO il documento dei requisiti ?



Requirements document variability

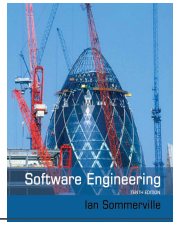


↳ il livello di dettaglio nei requisiti dipende dal tipo di sistema utilizzato (es. Agile o waterfall)

- Information in requirements document depends on type of system and the approach to development used.
- Systems developed incrementally will, typically, have less detail in the requirements document.
- Requirements documents standards have been designed e.g. IEEE standard. These are mostly applicable to the requirements for large systems engineering projects.

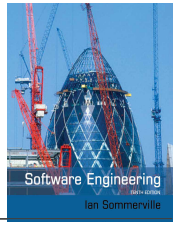
The structure of a requirements document

Una cosa importante da avere è il GLOSSARIO, che serve per risolvere le ambiguità del sistema naturale (terminologia dello stakeholder)



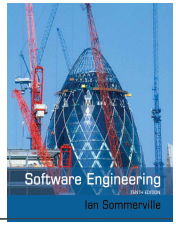
Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.

The structure of a requirements document



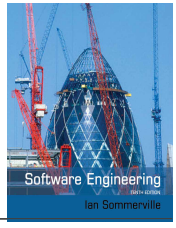
Chapter	Description
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution ↳ visione del futuro	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

Caratteristiche di un buon 'Requirements document'



- Specify external system behaviour
- Specify implementation constraints
- Easy to change
- Serve as reference tool for maintenance
- Record forethought about the life cycle of the system i.e. predict changes
- Characterise responses to unexpected events

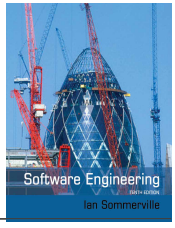
Anticipation of change
Non pensare solo al presente, ma cercare di proiettare l'analisi nel futuro



Requirements validation

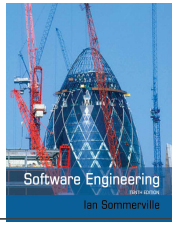
F. 4 Requirements validation

vedere se quello che propongo è veramente ciò che risolve i problemi, se va bene allo stakeholder
↳ conviene farlo prima di iniziare a sviluppare il sw → meglio correggere subito i dubbi



- 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

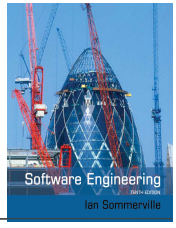
Paramentri per valutare i requisiti
Requirements checking



- Validity (corresponds to 3**C** - Correctness). Does the system provide the functions which best support the customer's needs?
- Consistency (3C). Are there any requirements conflicts?
- Completeness (3C). Are all functions required by the customer included?

altre caratteristiche che i REQUISITI devono rispettare

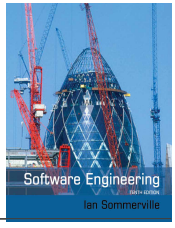
Requirements checking - 2



- Realism. Are requirements feasible? Can the requirements be implemented given available budget and technology?
  ↳ sono implementabili? sono impossibili?
- Verifiability. Can the requirements be checked? Are they realistically testable? → sono requisiti misurabili? es. tu mi dici che lo vuoi veloce, ok ma quanto? dammi una misura!
- Comprehensibility. Is the requirement properly understood?
- Traceability. Is the origin of the requirement clearly stated?
- Relevance
- Adaptability. Can the requirement be changed without a large impact on other requirements?

Requirements validation techniques

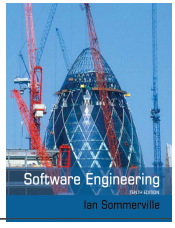
↳ come si fanno le validazioni?



- Requirements **reviews**: specific meeting for presenting, analysing and validating requirements
  - Systematic manual analysis of the requirements
- Prototyping → faccio un prototipo per decidere con lo stakeholder se va bene
  - Using an executable model of the system to check requirements.
- Test-case generation
  - Developing tests for requirements to check testability
- Automated consistency analysis
  - Checking the consistency of a structured requirements (formal) description

Requirements reviews

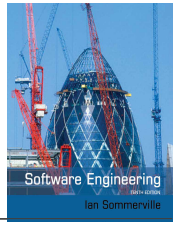
RIUNIONE in cui si esamina qualcosa



riunioni in cui si discute

- 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 communication between developers, customers, and users can resolve problems at an early stage

Review checks



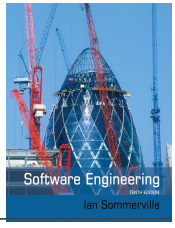
- …
- Verifiability
  - Is the requirement realistically testable?
- Comprehensibility
  - Is the requirement properly understood?
- Traceability
  - Is the origin of the requirement clearly stated?
- Adaptability
  - Can the requirement be changed without a large impact on other requirements?
- …

Un’ulteriore processo dedicato ai REQUISITI:

Requirement Management

F. 5 **Requirements management**

↳ gestire i requisiti che possono cambiare durante tutto il ciclo di vita

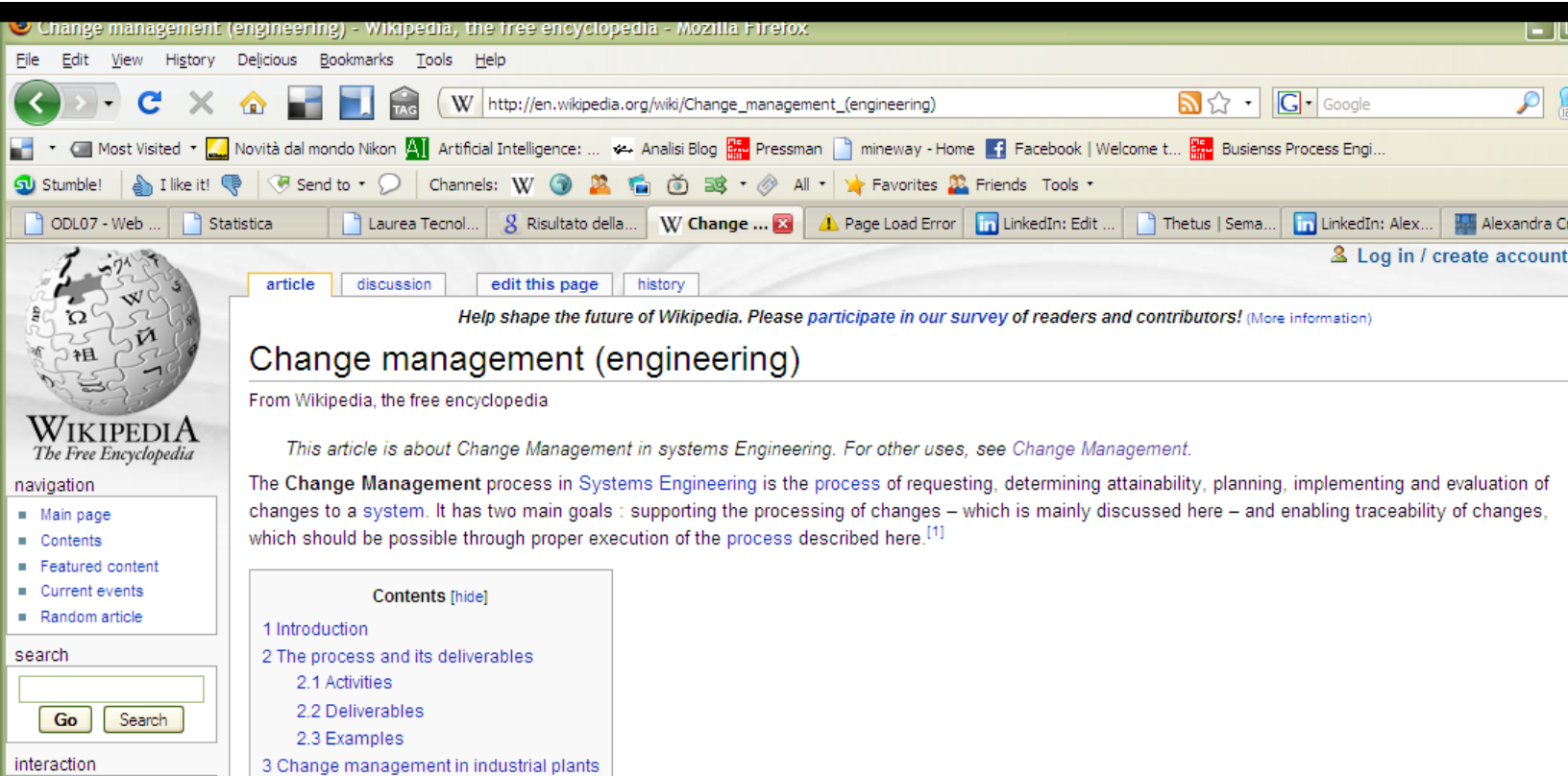


- Requirements management is the process of managing changing requirements during the requirements engineering process and entire system development and evolution!!!
- New requirements emerge as a system is being developed and after it has gone into use (Reqs. Change during SW Evolution!!)
- You need to keep track of individual requirements and maintain links between dependent requirements so that you can assess the impact of requirements changes. You need to establish a formal process for making change proposals and linking these to system requirements.

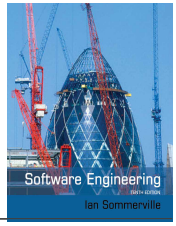
La ‘dimensione’ temporale

Evoluzione e Modifica dei Requisiti (Requirements Evolution & Change)

Change management

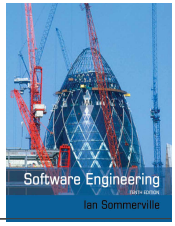


Changing requirements



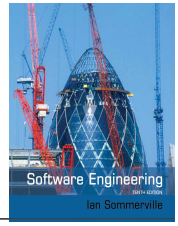
- The business and technical environment of the system always changes after installation.
  - New hardware may be introduced, it may be necessary to interface the system with other systems, business priorities may change (with consequent changes in the system support required), and new legislation and regulations may be introduced that the system must necessarily abide by.
- The people who pay for a system and the users of that system are rarely the same people and have different priorities.
  - System customers impose requirements because of organizational and budgetary constraints. These may conflict with end-user requirements and, after delivery, new features may have to be added for user support if the system is to meet its goals.

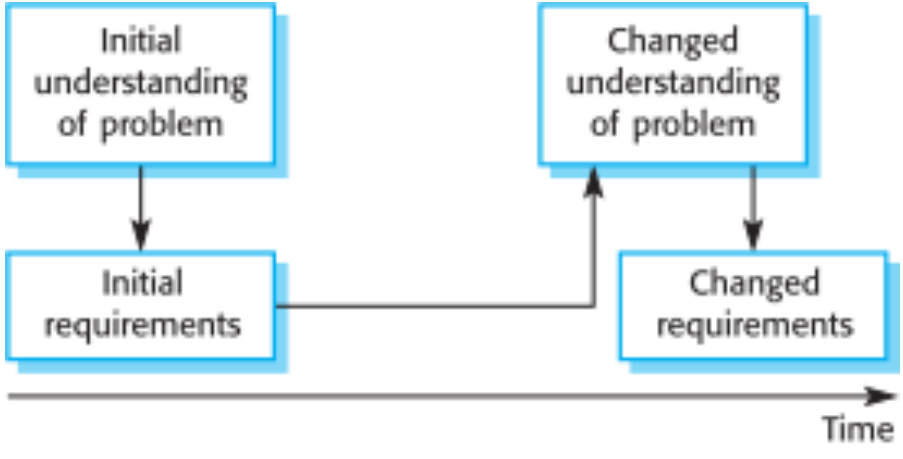
Changing requirements



- Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.
  - The final system requirements are inevitably a compromise between them and, with experience, it is often discovered that the balance of support given to different users has to be changed.

Requirements evolution





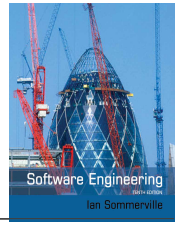
Categorie di requisiti dal punto di vista del cambiamento: Enduring and volatile requirements

Requisiti stabili e requisiti che cambiano:

- **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

↳ requisiti che cambiano spesso

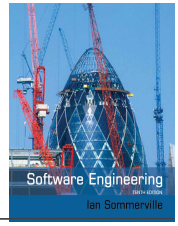
Classification of volatile requirements



Varie ragioni del cambiamento:

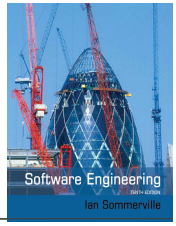
- Mutable requirements
  - Requirements that change due to the system's **environment**
- Emergent requirements
  - Requirements emerging as understanding of the entire **system** develops
- Consequential requirements
  - Requirements that result from the **introduction of the computer** system
- **Compatibility** requirements
  - Requirements that depending on **other** systems or organisational processes

‘Requirements management’ planning



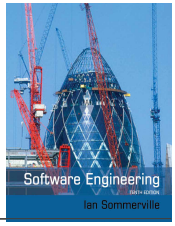
- During the requirements engineering process, you have to plan:
  - Requirements **identification**
    - How requirements are individually identified
  - A **change management** process
    - The process followed when analysing a requirements change
  - **Traceability** policies
    - The amount of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

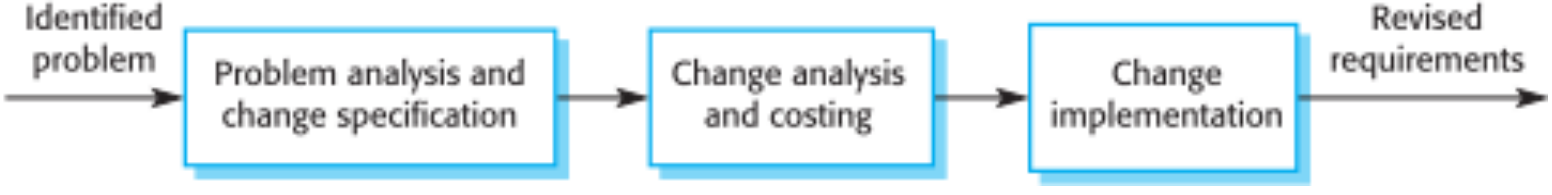
Requirements change management



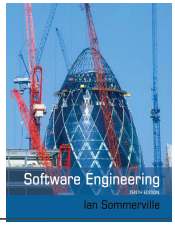
- Deciding if a requirements change should be accepted
  - *Problem analysis and change specification*
    - During this stage, the problem or the change proposal is analyzed to check that it is valid. This analysis is fed back to the change requestor who may respond with a more specific requirements change proposal, or decide to withdraw the request.
  - *Change analysis and costing*
    - The effect of the proposed change is assessed using traceability information and general knowledge of the system requirements. Once this analysis is completed, a decision is made whether or not to proceed with the requirements change.
  - Change implementation
    - The requirements document and, where necessary, the system design and implementation, are modified. Ideally, the document should be organized so that changes can be easily implemented.

Requirements change management





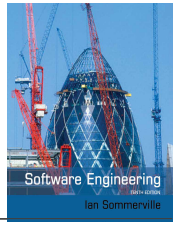
Traceability → TRACCIABILITA'



- Traceability is concerned with the relationships between requirements, their sources and the system design
- Source traceability – da dove si origina
  - Links from requirements to stakeholders who proposed these requirements (…rationale…)
- Requirements traceability – con quali requisiti
  - Links between dependent requirements
- Design traceability – verso quali aspetti dela progettazione
  - Links from the requirements to the design

come vederlo graficamente: dipendenze tra requisiti

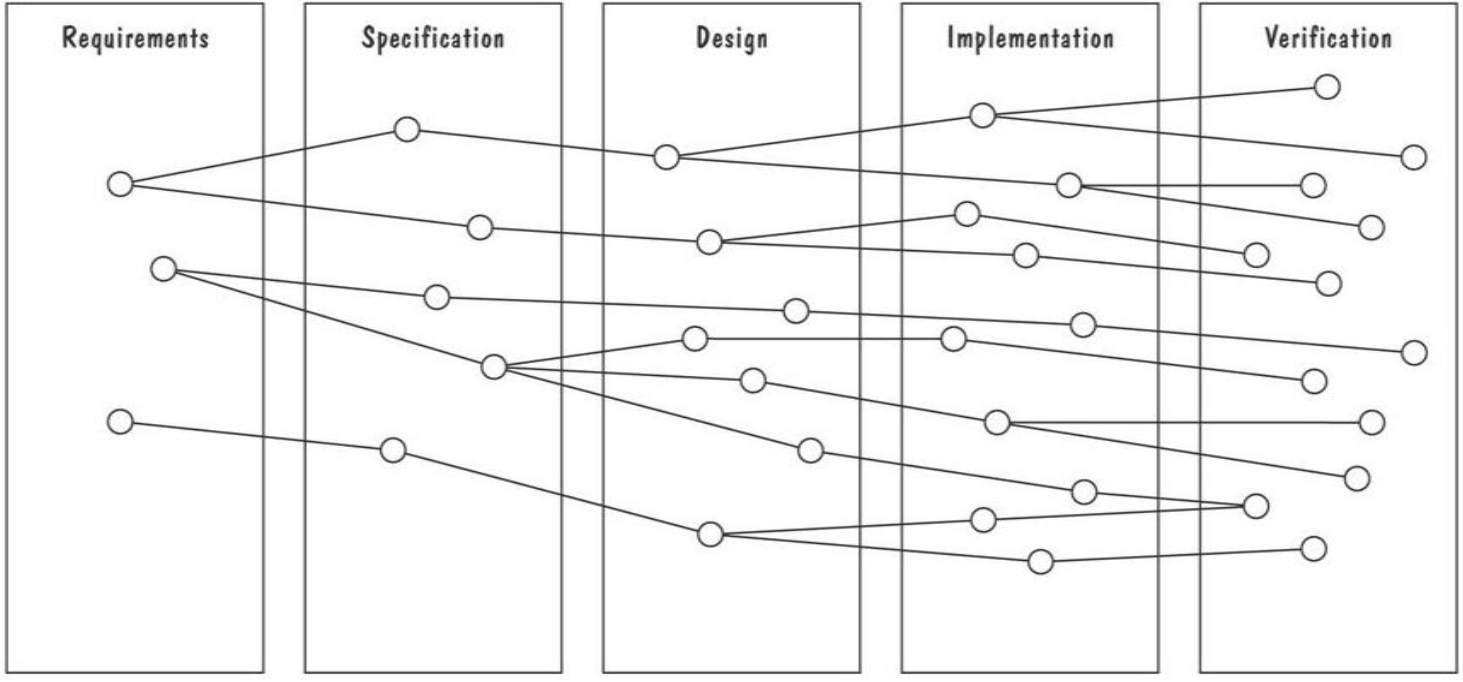
A traceability matrix



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	

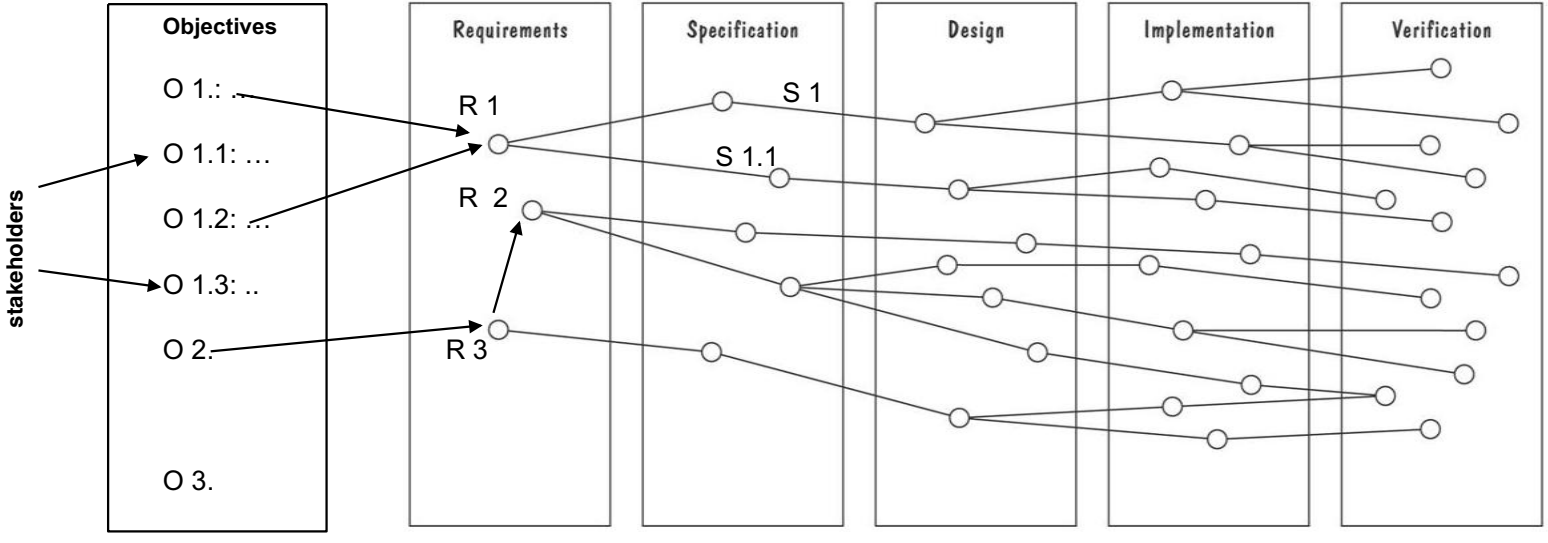
D – depends **R** - related

A simplified idea of tracebility structure



Recording the traceability structure end-to-end

capire le varie fasi e saperle identificare, e poi evidenziare le varie dipendenze



Traceability effects on documents intra-references

Set 5 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Obiettivi, servizi, requisiti utente, requisiti del Sistema, definizioni dei requisiti, specifiche, definizione degli obiettivi, definizione dei requisiti, specificazione dei requisiti, document dei requisiti, dominio applicativo, stakeholder, analista sw, progettista sw, prospettiva dominio applicativo e tecnica
- Requisiti funzionali e Non funzionali; domain requirement. Imprecisione/ambiguità dei requisiti; completezza, consistenza, correttezza, Regola delle 3 C. Requisiti non funzionali di prodotto, di processo ed esterni.
- Obiettivi e requisiti, verificabilità dei requisiti. Metriche per misurare i requisiti. Interazione e conflitti nei requisiti. Req. di dominio: non facilmente comprensibili e impliciti.
- Linguaggi di rappresentazione di Requisiti e Specifiche, con relative problemi: linguaggio naturale (chiarezza, confusione, amalgamazione di più requisiti); linguaggio strutturato, suggerimenti per l'uso del linguaggio.
- Problemi del LN per la specificazione: ambiguità, estrema flessibilità, linearità e mancanza di meccanismi di modularizzazione

Set 5 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Modi per strutturare il linguaggio: scrittura standardizzata, template, form, tabelle, modellli grafici, Use Case UML, sequence diagram, PDL pseudolinguaggi, specificazione in PDL delle interface
- Documento dei Requisiti (DdR), destinatari del document, caratteristiche (requisiti) del DdR, standard, struttura.

Set 5 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Attività generiche presenti nel processo di RE: elicitazione, analisi, specificazione, validazione, gestione; diverse prospettive nella rappresentazione del RE
- Studio di Fattibilità: obiettivi, modalità di realizzazione e risultati
- Elicitazione e analisi: partecipanti e ruoli, stakeholder; criticità; processo iterative di elicitazione e analisi, rappresentazione a spirale; attività del processo: discovery, classificazione e organizzazione, prioritizzazione e negoziazione, documentazione
- R. discovery: gathering+distilling; sorgenti; risutati: reqs.di dominio, altri reqs., system models ottenuti mediante diverse prospettive di analisi (partizionamento, classificazione e proiezione).
- Tecniche di elicitazione. Interviste. Storie. Use case. Sequence Digram. Etnografia (osservazione dei processi). Criticità dei fattori organizzativi/politici/sociali

Set 5 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Validazione dei Requisiti. Parametri da controllare per validare. Tecniche di validazione: review, review checks. Misurazione dei requisiti: modalità e profili di misura.
- Gestione dei requisiti. Motivazione del cambiamento. Categorie di requisiti dal punto di vista del cambiamento: duraturi e volatili. Varie tipologie di r. volatili. Pianificazione della gestione. Tracciabilità.