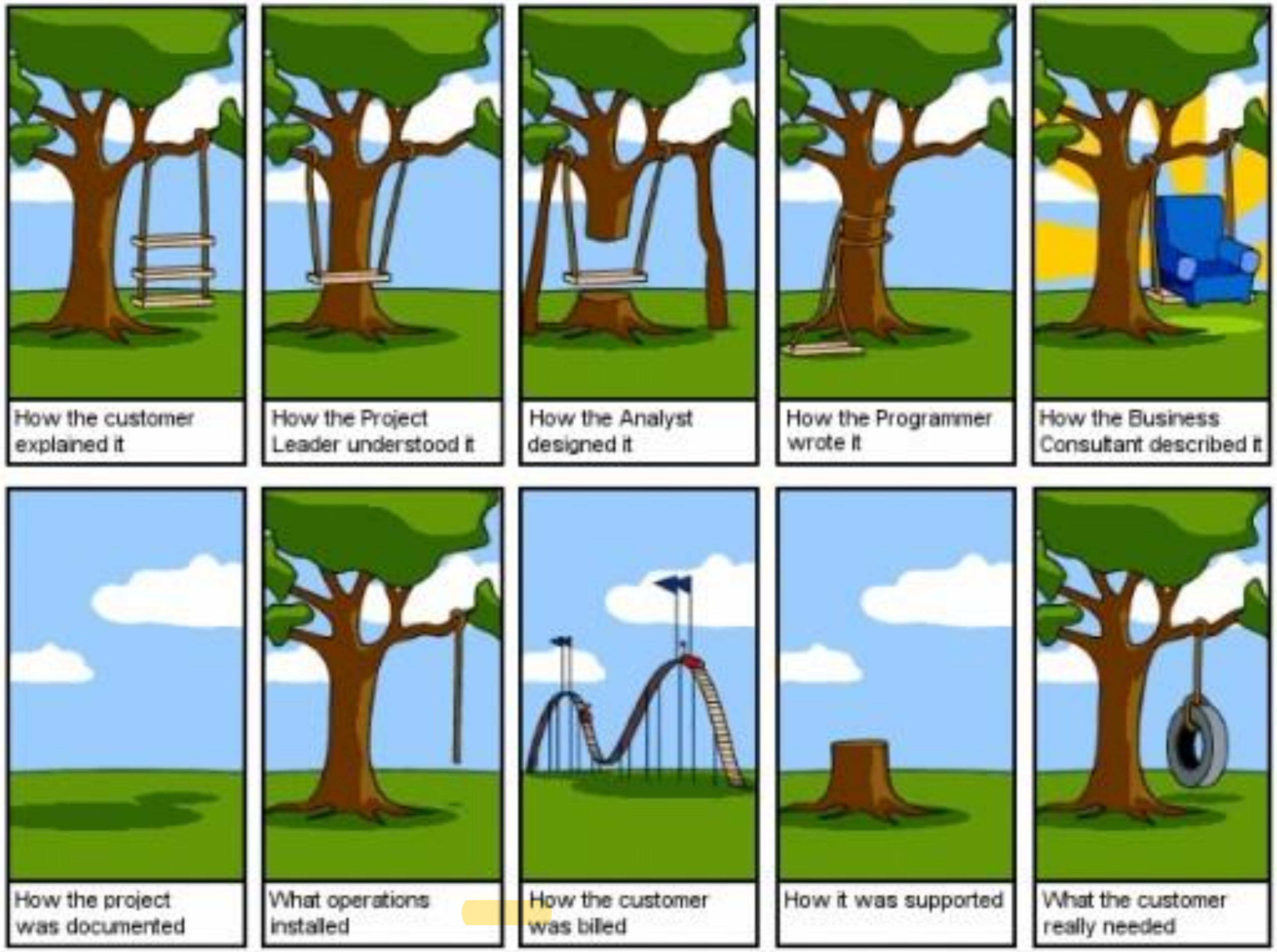
Requirements Engineering


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Topics covered

- Functional and non-functional requirements
- Requirements engineering processes

Requirements engineering

The process of establishing the **services** that a customer requires from a system and the **constraints** under which it operates and is developed.

The **system requirements are the descriptions** of the system **services** and **constraints** that are generated during the requirements engineering process.

What is a requirement?

It may range from a high-level abstract statement of a 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)

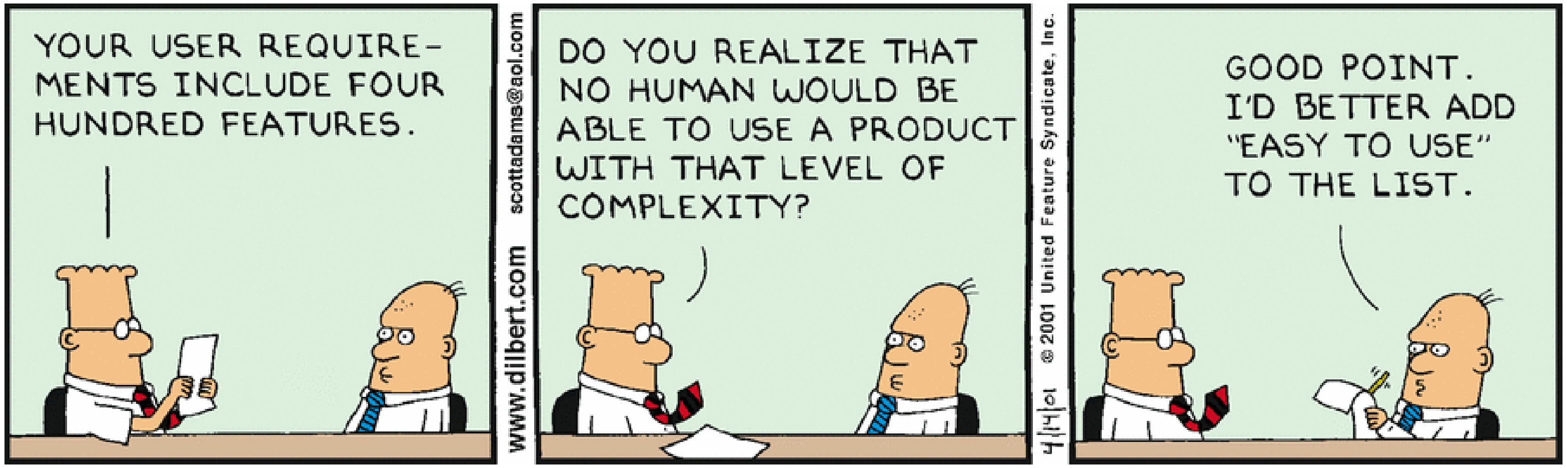
"If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization'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 requirements

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.
- System requirements
  - A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.


YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.
www.dilbert.com scottadams@aol.com
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
4/14/01 © 2001 United Feature Syndicate, Inc.
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.

User and system requirements

Mentcare is an information system that is intended for use in clinics

User requirements definition

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

System requirements specification

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

System stakeholders

- Any person or organization who is affected by the system in some way and so who has a legitimate interest
- Stakeholder types
  - End users
  - System managers
  - System owners
  - External stakeholders
  - Developers
  - …

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.

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.

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 business systems but problematic for systems that require pre-delivery analysis (e.g. critical systems) or systems developed by several teams.

Functional and non-functional requirements

THIS ISN'T WHAT I WANTED.
I KNOW.
Dilbert.com DilbertCartoonist@gmail.com
BUT GIVEN YOUR UNCLEAR EMAIL AND YOUR UNWILLINGNESS TO ANSWER FOLLOW-UP QUESTIONS, I DECIDED TO DO WHATEVER ENTERTAINED ME.
2-8-13 ©2013 Scott Adams, Inc. /Dist. by Universal Udick
DO WE HAVE A PROBLEM HERE?
NO, THIS TOTALLY WORKS FOR ME.

Functional, non-functional and domain requirements

✧ 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.
- May state what the system should not do.

✧ Non-functional requirements

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services.

✧ Domain requirements

- Info and constraints on the system from the domain of operation

Mentcare system: functional requirements

- ✧ R1. A user shall be able to search the appointments lists.
- ✧ R2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- ✧ R3. Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Requirements imprecision

- ✧ Problems arise when functional requirements are not precisely stated.
- ✧ Ambiguous requirements may be interpreted in different ways by developers and users.
- ✧ Consider the term 'search' in requirement R1. *A user shall be able to search the appointments lists.*
  - User intention – search for a patient name across all appointments in all clinics;
  - Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

Requirements completeness and consistency

In principle, requirements should be both complete and consistent.

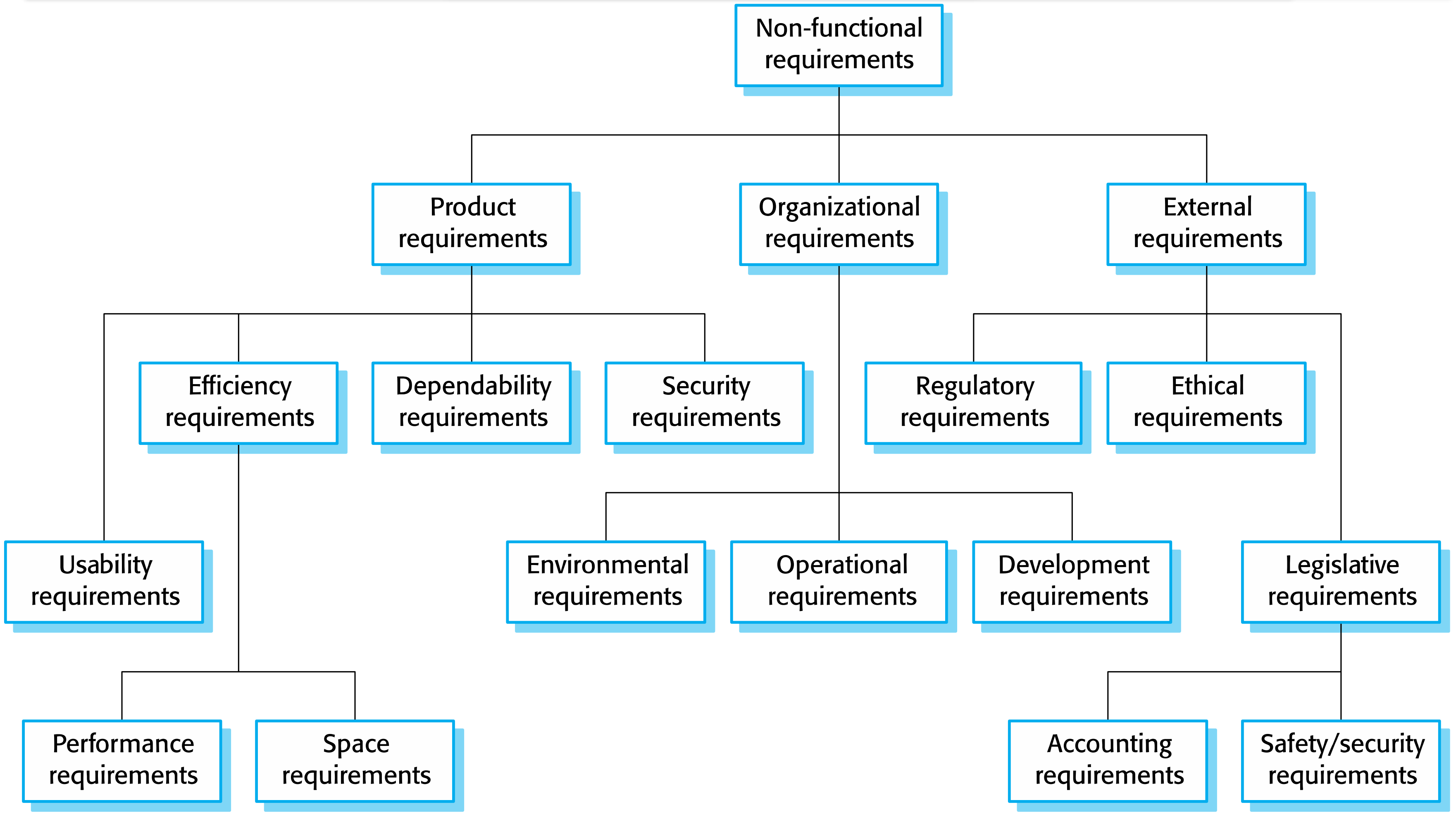
Complete	They should include descriptions of all facilities required.
Consistent	There should be no **conflicts or contradictions** in the descriptions of the system facilities.

In practice, because of system and environmental complexity, it is very **difficult to produce a complete and consistent requirements document**.

Non-functional requirements

- These 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 IDE, programming language or development method.
- Non-functional requirements may be more critical than functional requirements. **If these are not met, the system may be useless.**

Types of nonfunctional requirement



Non-functional requirements implementation

NFRs may affect the overall architecture of a system rather than the individual components.

For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.

A single NFR, (e.g. security req.), may affect a number of related functional requirements that define system services that are required.

It may also generate requirements that restrict existing requirements.

Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

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 and requirements

- NFRs may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- Goal
  - A general intention of the user such as ease of use.
- Verifiable NFR
  - A statement using some measure that can be objectively tested.
- Goals are helpful to developers as they convey the intentions of the system users.

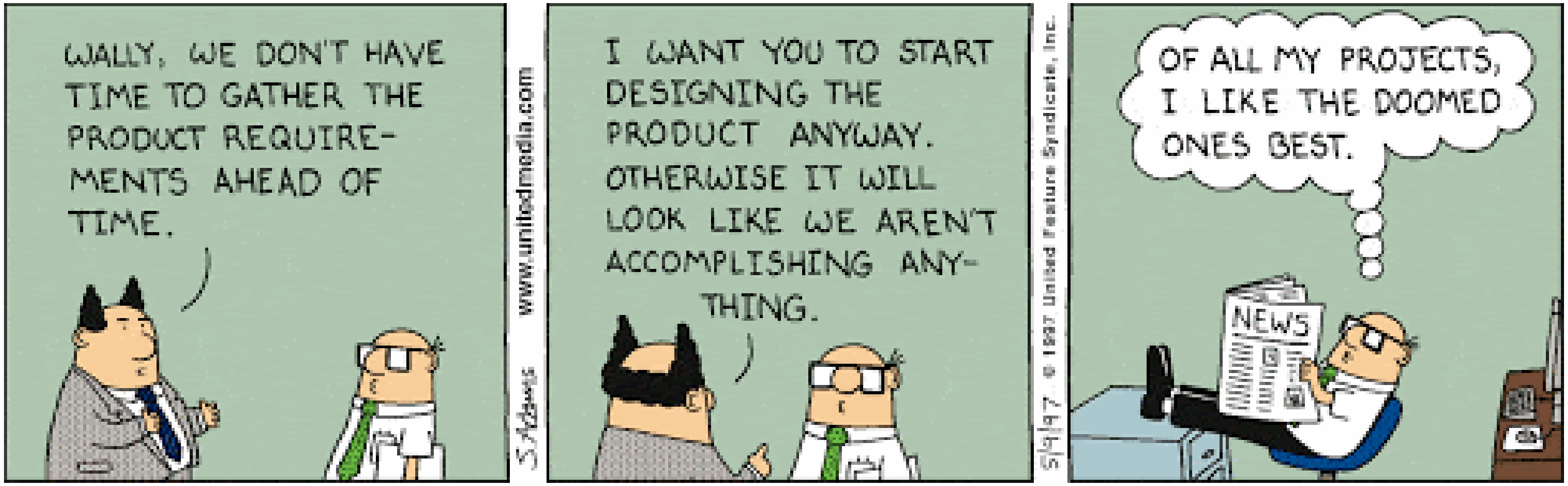
Usability requirements

- The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized. (Goal)
- Medical staff shall be able to use all the system functions after four hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use. (Testable non-functional requirement)

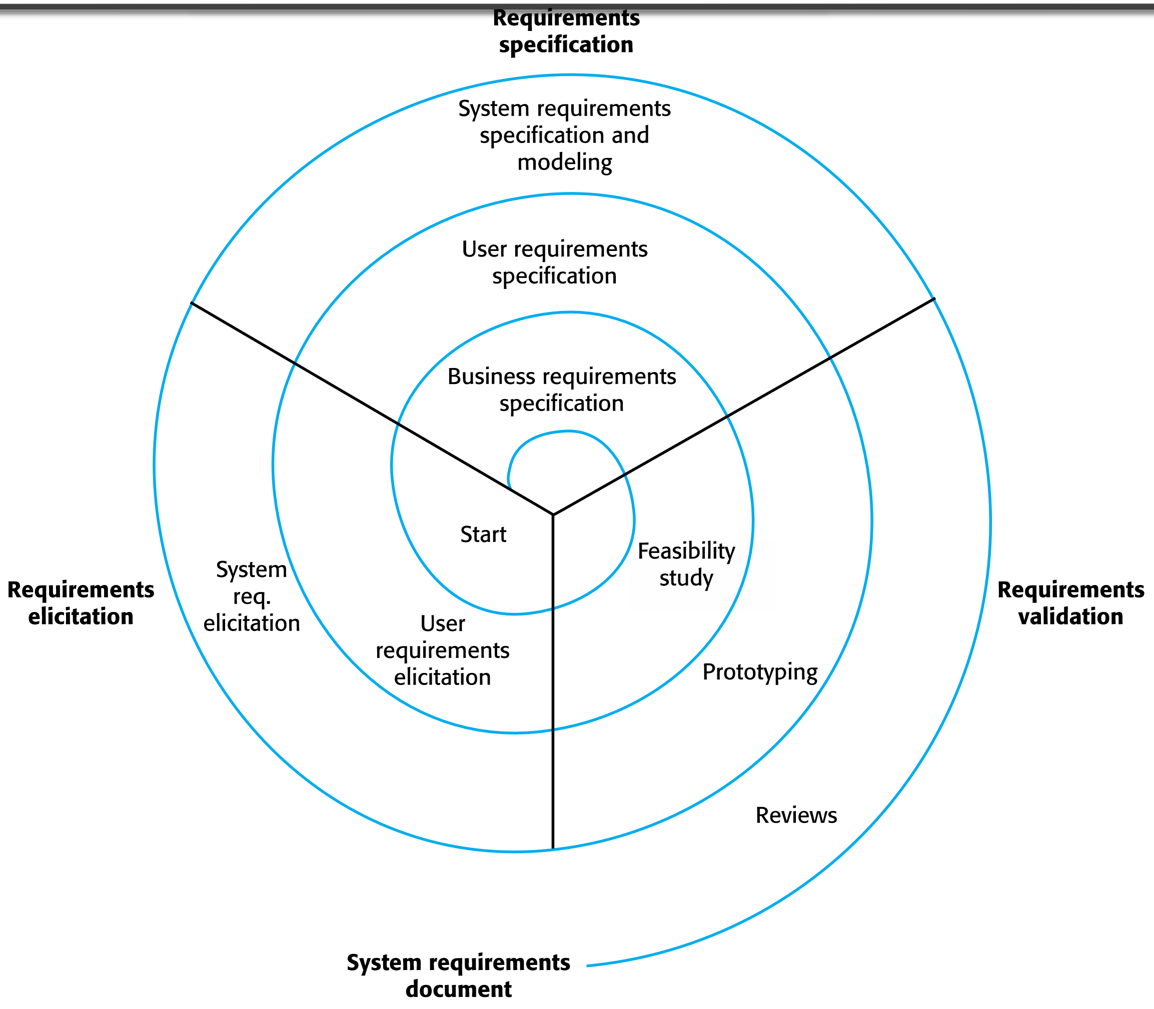
Metrics for specifying nonfunctional requirements

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
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

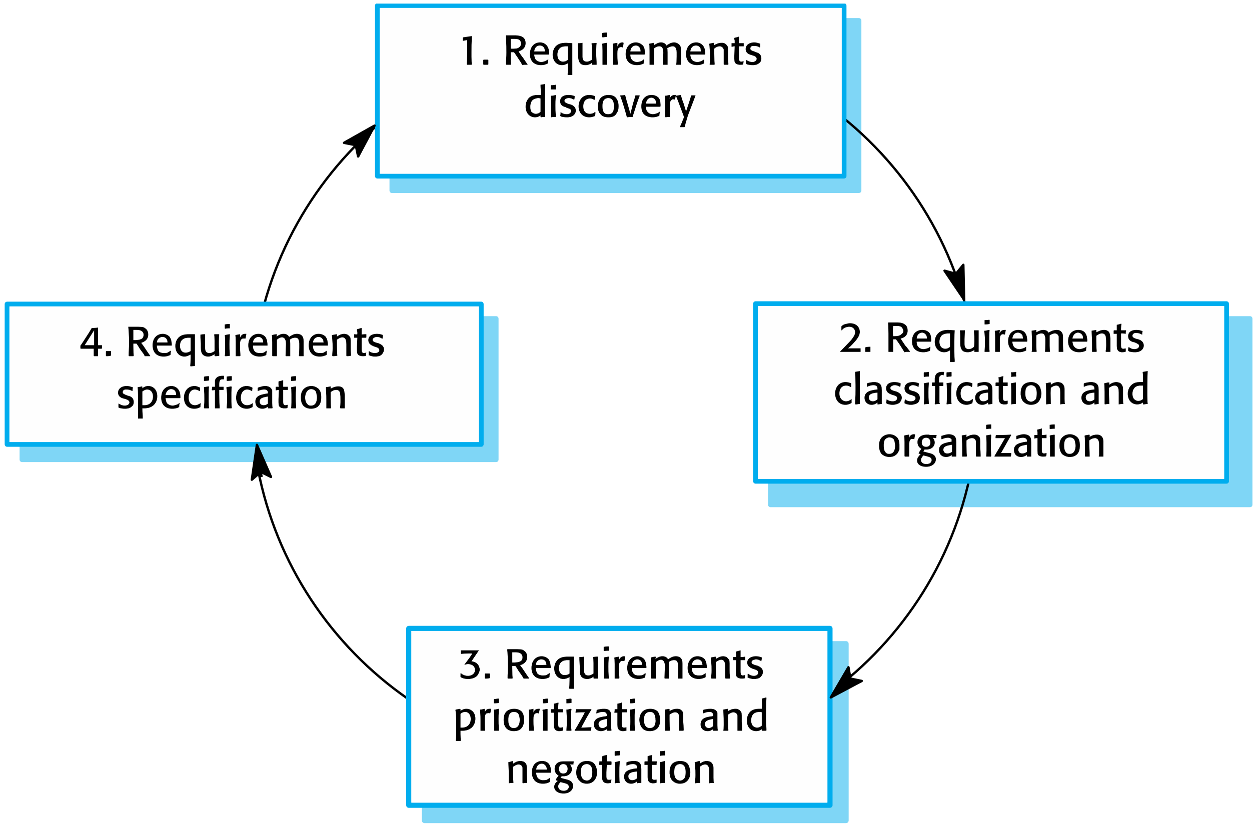
Requirements engineering processes


WALLY, WE DON'T HAVE TIME TO GATHER THE PRODUCT REQUIRE-MENTS AHEAD OF TIME.
www.unitedmedia.com
S. Adams
I WANT YOU TO START DESIGNING THE PRODUCT ANYWAY. OTHERWISE IT WILL LOOK LIKE WE AREN'T ACCOMPLISHING ANY-THING.
5/4/97 © 1997 United Feature Syndicate, Inc.
OF ALL MY PROJECTS, I LIKE THE DOOMED ONES BEST.
NEWS

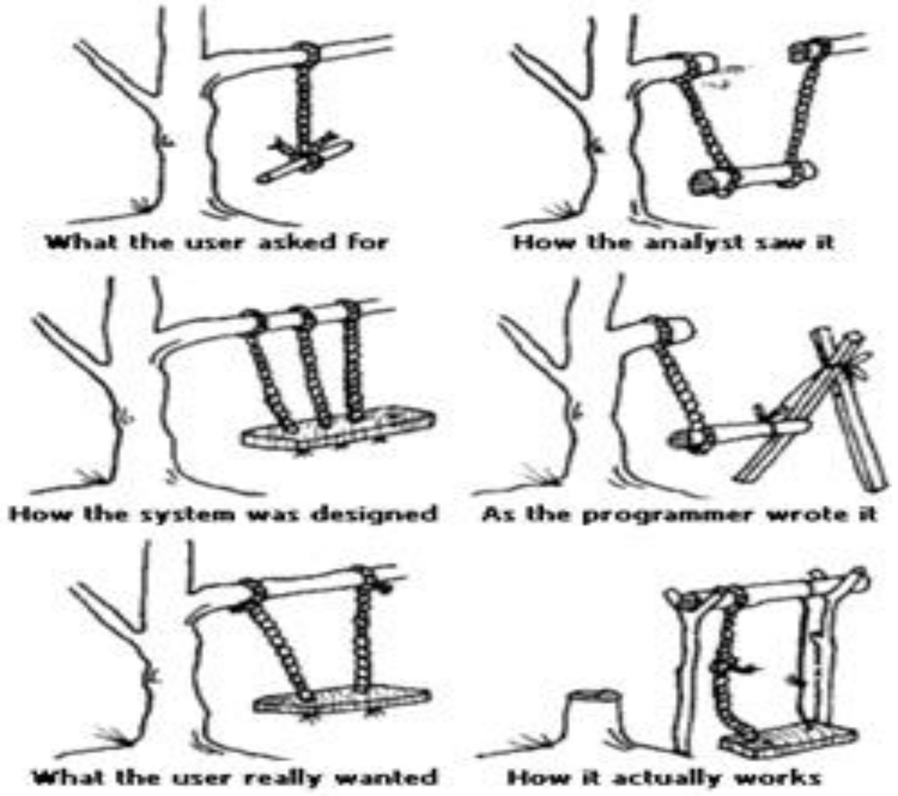
A spiral view of the requirements engineering process


Requirements specification
System requirements specification and modeling
User requirements specification
Business requirements specification
Start
Feasibility study
Prototyping
Reviews
Requirements validation
Requirements elicitation
System req. elicitation
User requirements elicitation
System requirements document

The requirements elicitation and analysis process


1. Requirements discovery
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements specification

PROBLEM


What the user asked for
How the analyst saw it
How the system was designed
As the programmer wrote it
What the user really wanted
How it actually works

Problems of requirements elicitation

- ✧ Stakeholders don't know what they really want.
- ✧ Stakeholders express requirements in their own terms.
- ✧ Different stakeholders may have conflicting requirements.
- ✧ Organisational and political factors may influence the system requirements.
- ✧ The requirements may change during the analysis process. New stakeholders may emerge and the business environment may change.

Requirements discovery

- The process of gathering information about the required and existing systems and distilling the user and system requirements from this information.
- Interaction is with system stakeholders from managers to external regulators.
- Systems normally have a range of stakeholders.

Requirements specification

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

Lack of clarity

- Precision is difficult without making the document difficult to read.

Requirements confusion

- Functional and non-functional requirements tend to be mixed-up.

Requirements amalgamation

- Several different requirements may be expressed together.

Example requirements for the insulin pump software system

Number reqs

Explaining reqs

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.)*

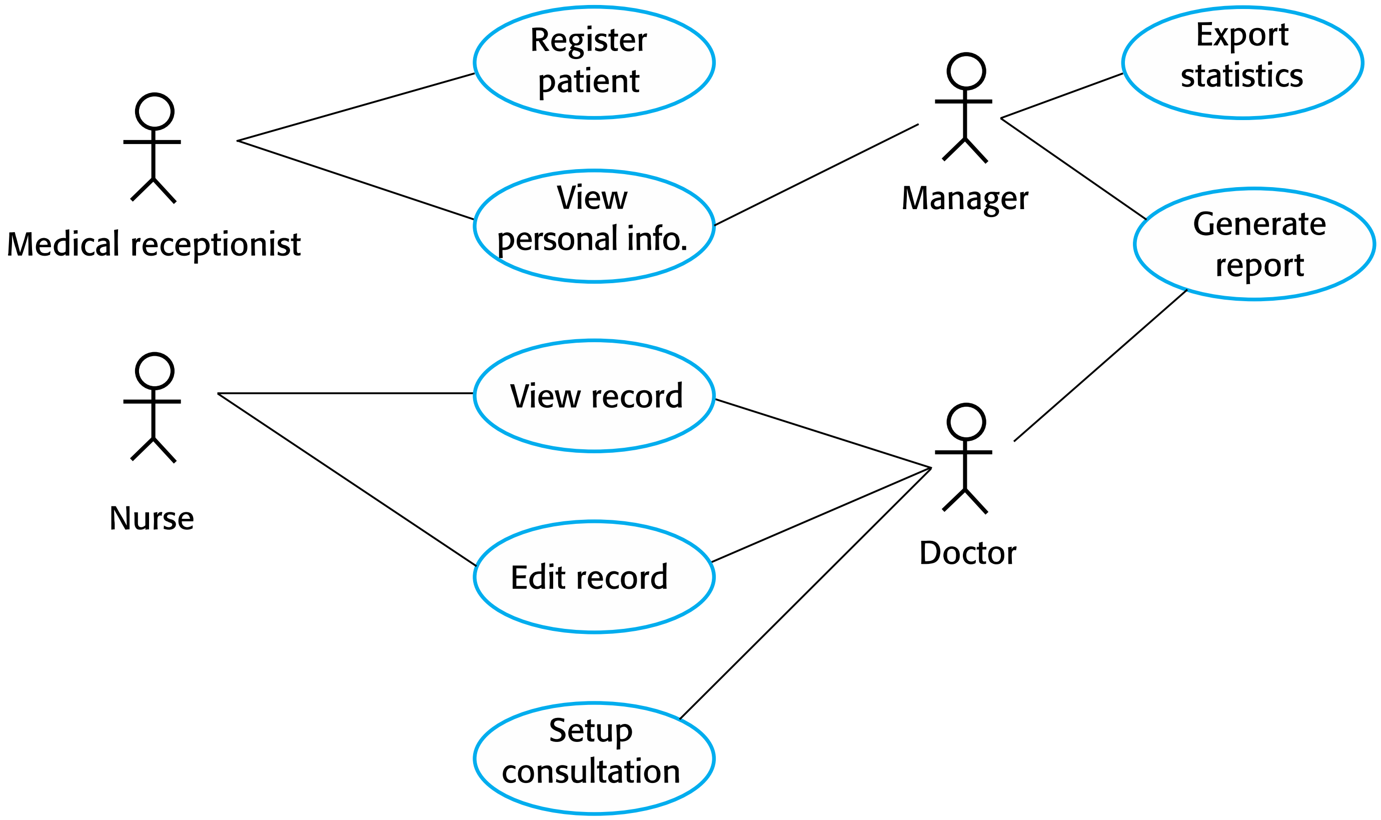
A structured specification of a requirement for an insulin pump

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.	
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.

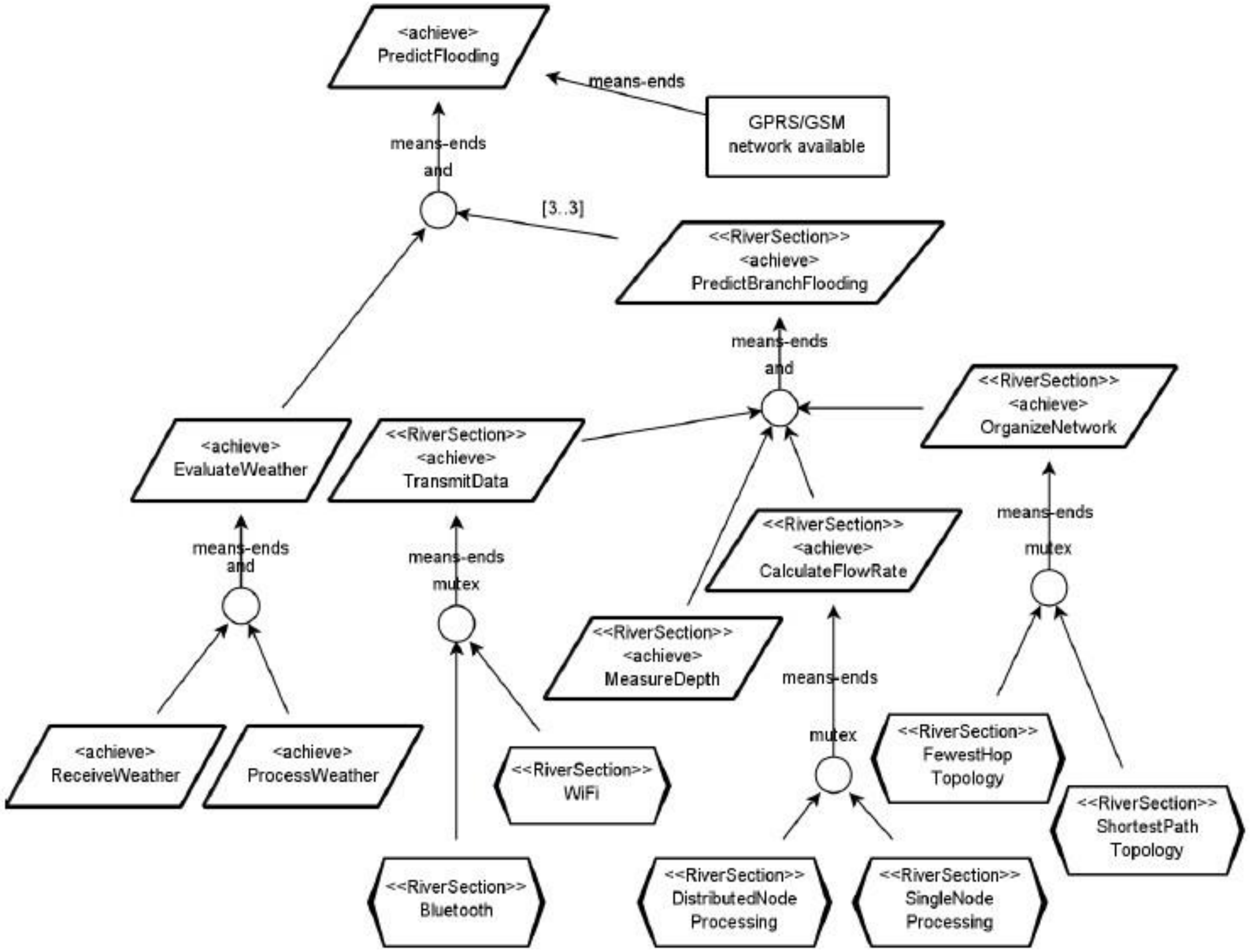
Tabular specification of computation for an insulin pump

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 notations: Use cases (Mentcare system)

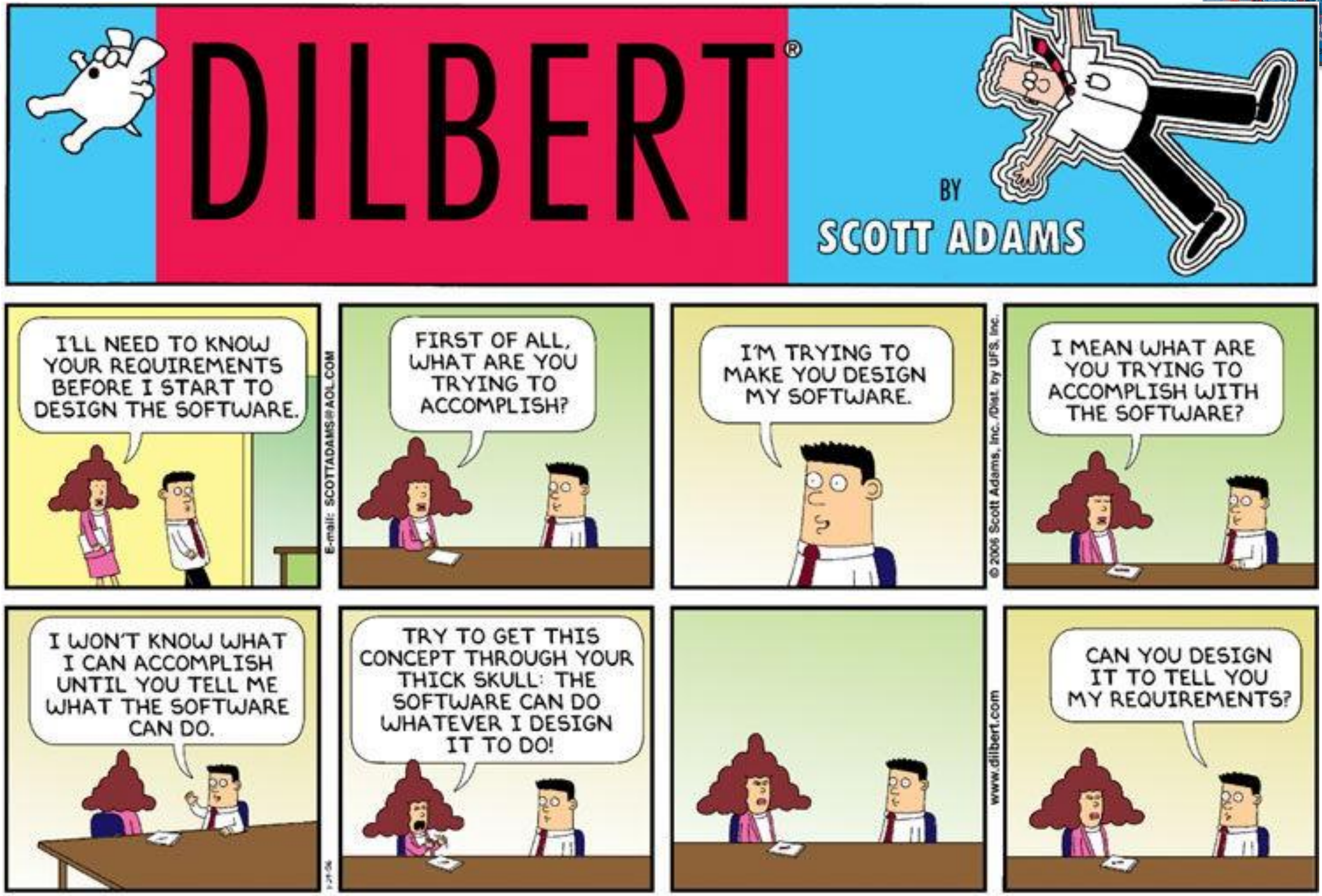


Goal models (Weather forecast)



<achieve> PredictFlooding
means-ends
GPRS/GSM network available
means-ends and
[3..3]
<<RiverSection>> <achieve> PredictBranchFlooding
means-ends and
<<RiverSection>> <achieve> OrganizeNetwork
<achieve> EvaluateWeather
<<RiverSection>> <achieve> TransmitData
<<RiverSection>> <achieve> CalculateFlowRate
means-ends mutex
means-ends and
means-ends mutex
<<RiverSection>> <achieve> MeasureDepth
means-ends mutex
<<RiverSection>> FewestHop Topology
<achieve> ReceiveWeather
<achieve> ProcessWeather
<<RiverSection>> WiFi
<<RiverSection>> ShortestPath Topology
<<RiverSection>> Bluetooth
<<RiverSection>> DistributedNode Processing
<<RiverSection>> SingleNode Processing

Requirements and design


DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc. /Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

Requirements and design

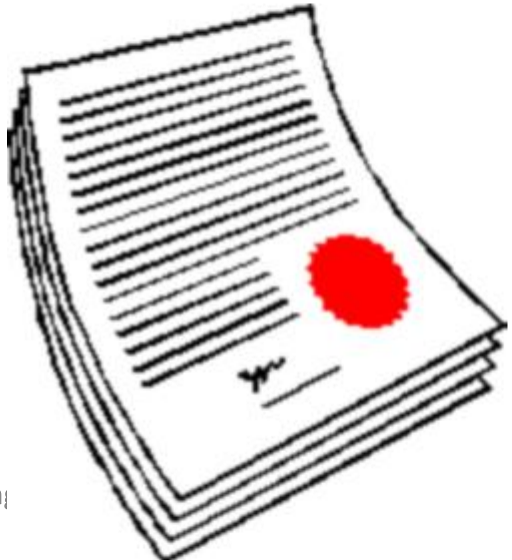
In principle, requirements should state what the system should do and the design should describe how it does this.

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;
- An architecture should satisfy NFRs.

The software requirements document

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



Requirements validation

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 or 200 times the cost of fixing an implementation error.

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.

Software prototyping

A prototype is an initial version of a system used to demonstrate concepts and try out design options.

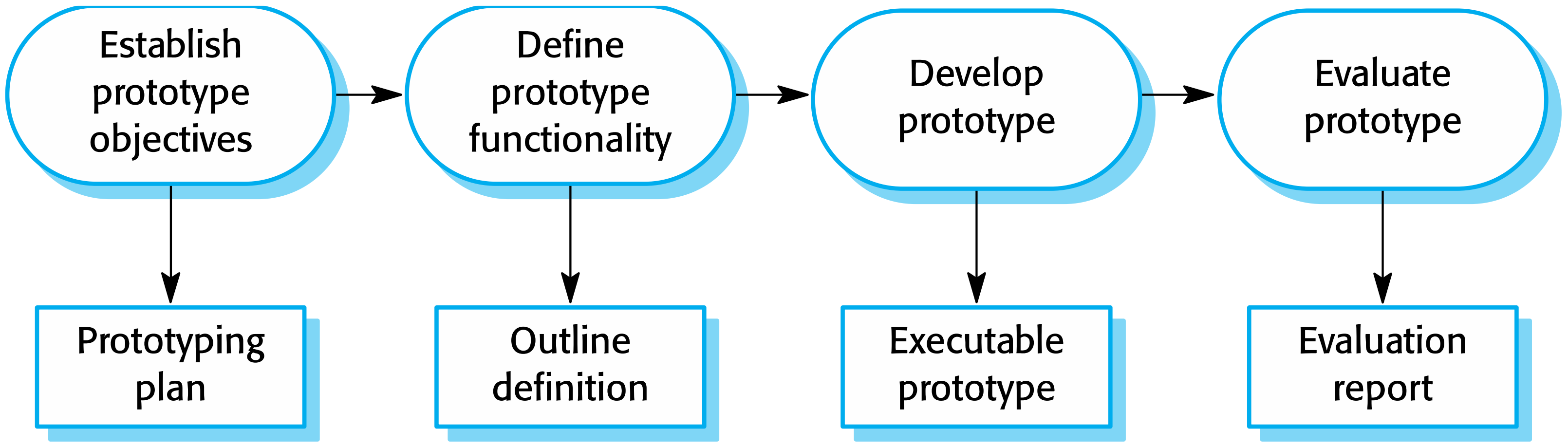
A prototype can be used in:

The requirements engineering process to help with requirements elicitation and validation;

In design processes to explore options and develop a UI design;

In the testing process to run back-to-back tests.

The process of prototype development



Requirements change

Changing requirements

DID I REMEMBER TO TELL YOU BEFORE YOU FINISHED THE CODING THAT THE USER'S SPECIFICATIONS CHANGED?
AAAIII-YIIII-YIIII-YIIII-YIIIIII!!!!!!
SO, NO-ISH?
BAM! BAM! BAM!
Dilbert.com DilbertCartoonist@gmail.com
5-16-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick.

✧ 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 management

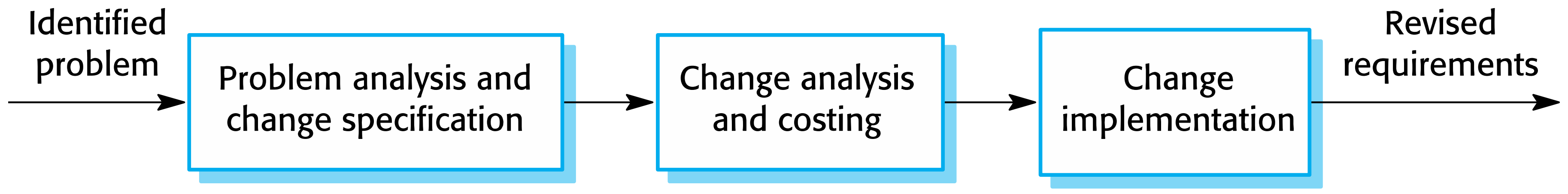
Requirements management is the process of managing changing requirements during the requirements engineering process and system development.

New requirements emerge as a system is being developed and after it has gone into use.

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.

Requirements change management


Identified problem
Problem analysis and change specification
Change analysis and costing
Change implementation
Revised requirements