

Analysis model

Requirements engineering

Davide Rossi

Dipartimento di Informatica – Scienze e Ingegneria
Università di Bologna



Abstractions

The software process can be seen as a sequence of activities lowering the degree of abstraction:

- Analysis
- Design
- Construction

Abstractions

The software process can be seen as a sequence of activities lowering the degree of abstraction:

- Mental model
- Analysis
- Design
- Construction

Abstractions

The software process can be seen as a sequence of activities lowering the degree of abstraction:

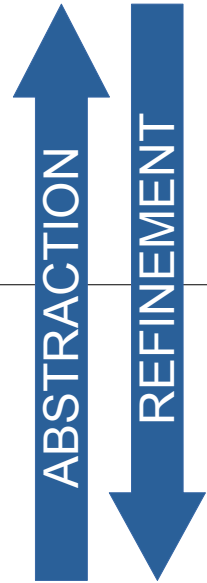
- Mental model
 - Analysis
-
- Design
 - Construction



Problem
space



Solution
space



Discussion

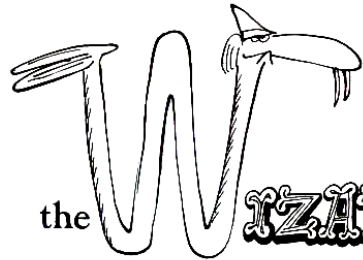
- How abstraction levels correlate to amount of information?
- What is really refinement?

Analysis model

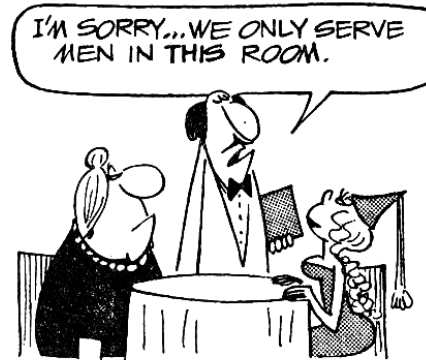
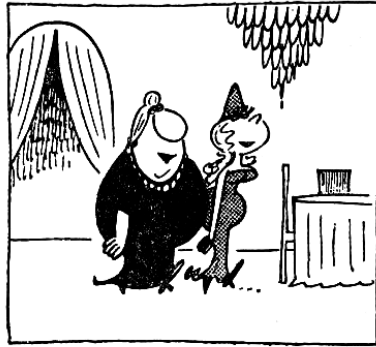
Objectives of analysis model

- **Understand** precisely what is required of the software
- **Communicate** this understanding to development team members and stakeholders
- **Define** a set of requirements that can be validated once the software is built

Natural language has quirks

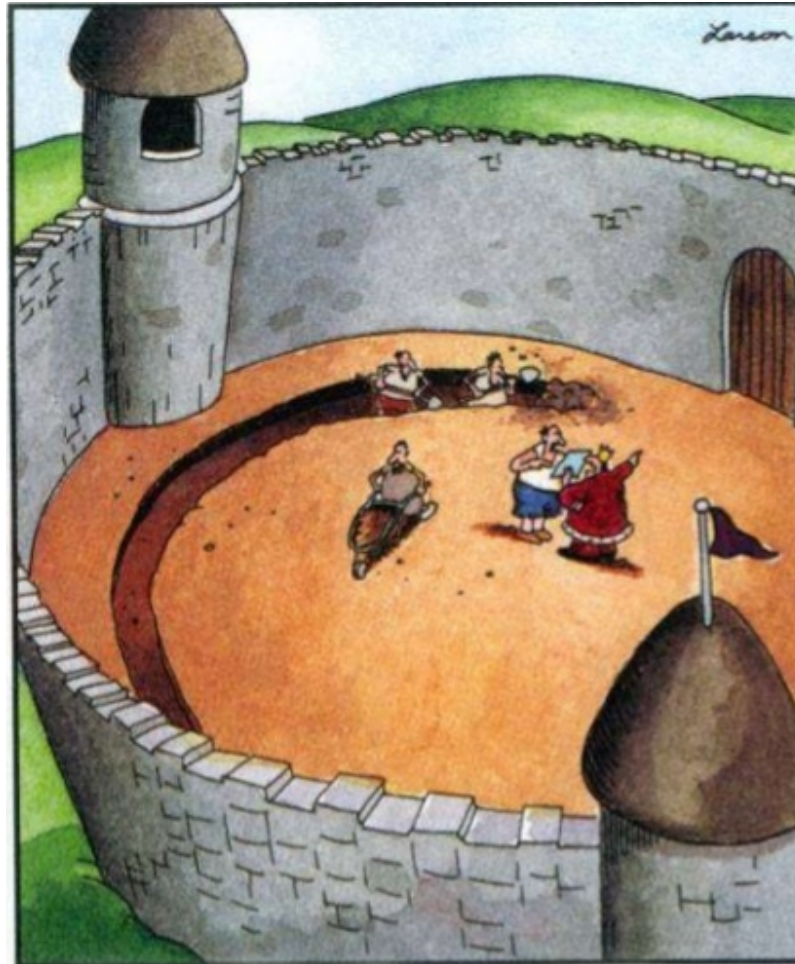


by Brant parker and Johnny hart



Analysis model artifacts

- Requirements document
- Use cases
- Domain model
- Supplementary Specification (captures and identifies other kinds of requirements, such as reports, documentation, packaging, supportability, licensing, ...)
- Glossary
- Business Rules (or Domain Rules)
- ...



Suddenly, a heated exchange took place between the King and the moat contractor, and hence, requirements management was born.

Requirements

- Software requirements are the needs and restrictions of a software product (see also IEEE 610.12-1990)
- Requirements engineering (RE) is the systematic handling of requirements.

Requirements

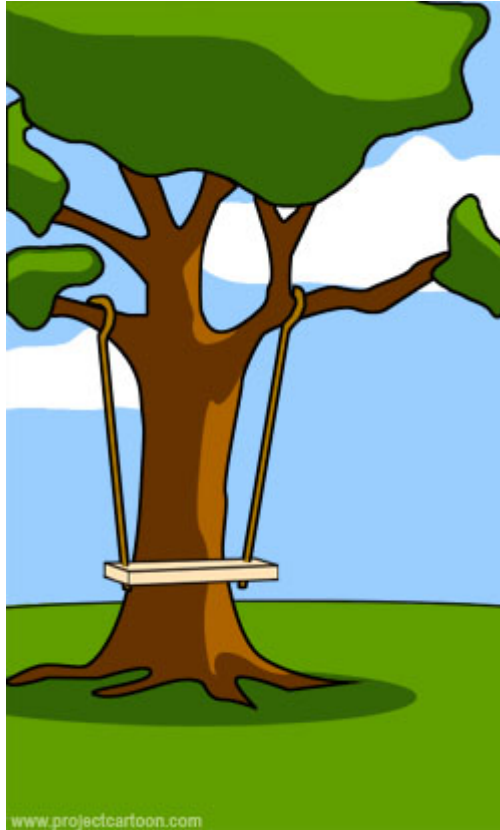
Software Requirements knowledge areas:

- Elicitation
- Analysis
- Specification
- Validation

How the customer explained it



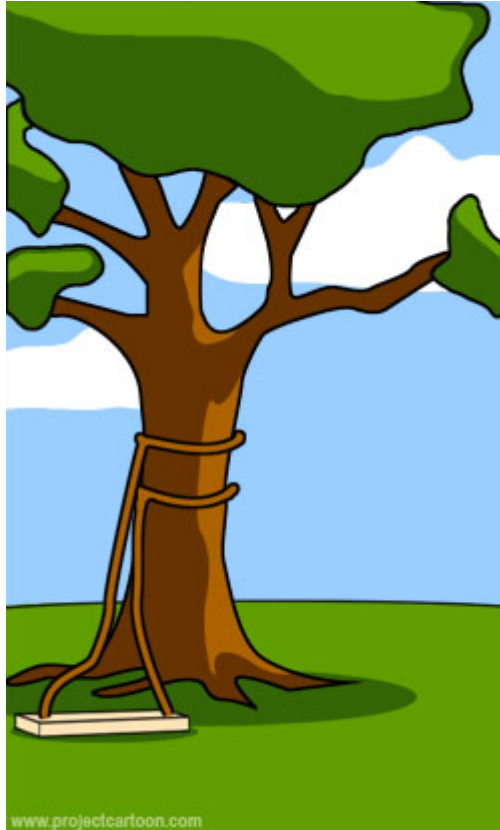
How the project leader understood it



How the analyst designed it



How the programmer wrote it



What the beta testers received



What the customer really needed



Formalize requirements

- There are several possible methods
- In structured development processes a good starting point is ISO/IEC/IEEE 29148:2011 - Systems and software engineering — Life cycle processes — Requirements engineering (replaces IEEE 830)

ISO/IEC/IEEE 29148:2011

Main documents:

- Stakeholder requirements specification document - **StRS**
- System requirements specification document - **SyRS**
- Software requirements specification document - **SRS**

Glossary

Stakeholder – those who share interest in the project.

At a minimum the end users and who is paying for it.

Stakeholder requirements

Requirements for a system that can provide the services needed by users and other stakeholders in a defined environment (**stakeholders' point of view**).

They express the intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is validated.

System requirements

A specification, **from the supplier's perspective**, of what characteristics, attributes, and functional and performance requirements the system is to possess, in order to satisfy the stakeholder requirements.

Software requirements

Only meaningful when the system is not just software.

ISO/IEC/IEEE 12207 describes system/software requirements.

Phrase a requirement

A requirement should use be as unambiguous as possible (controlled language).

It should be phrased as one or more sentences (one is better, if possible) containing a combination of: subject(s), action(s), object(s), value(s), condition(s), constraint(s).

Phrase a requirement

**[Condition] [Subject] [Action] [Object]
[Constraint]**

e.g.: When signal x is received **[Condition]**, the system **[Subject]** shall set **[Action]** the signal x received bit **[Object]** within 2 seconds **[Constraint]**.

Phrase a requirement

[Condition] [Action or Constraint] [Value]

e.g.: At sea state 1 **[Condition]**, the Radar System shall detect targets at ranges out to **[Action or Constraint]** 100 nautical miles **[Value]**.

Phrase a requirement

[Subject] [Action] [Value]

e.g.: The Invoice System **[Subject]**, shall display pending customer invoices **[Action]** in ascending order **[Value]** in which invoices are to be paid.

Discussion

- What makes a good requirement? And a requirement set?

Characteristics of a requirement

- Unambiguous
- Consistent
- Complete
- Singular
- Feasible
- Traceable
- Verifiable

Characteristics of a requirement set

- Complete
- Consistent
- Affordable
- Bounded

Requirements metadata

- Identification
- Stakeholder Priority
- Dependency
- Risk
- Source
- Rationale
- Difficulty
- Type

Types of requirements

- **Functional:** describe the interactions between the system and its environment, independently from the implementation.
- **Non-functional:** measurable/perceivable properties of the systems not directly related to functional aspects.

Discussion

Any example of perceivable aspects of a software system that are not functional?

Suggestion: think *how*, not *what*.

Functional vs nonfunctional

	Functional	Nonfunctional
Objective	Describe what the product does	Describe how the product works
End result	Define product features	Define product properties
Focus	Focus on user requirements	Focus on user expectations
Origin	Usually defined by user	Usually defined by developers of other tech experts
Testing	Before nonfunctional testing	After functional testing

Types of requirements

- **Functional requirements** are addressed by **system design**; they express what the system is supposed to do.
- **Non-functional requirements** are addressed by **system architecture**; they express how the system is supposed to be (or how it is supposed to do what it does).

FURPS+

- **Functional**: features, capabilities, security.
- **Usability**: human factors, help, documentation.
- **Reliability**: frequency of failure, recoverability, predictability.
- **Performance**: response times, throughput, accuracy, availability, resource usage.
- **Supportability**: adaptability, maintainability, internationalization, configurability.
- **+**: implementation, interface, operations, packaging, legal.

Requirements analysis process

The purpose of the Requirements Analysis Process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services.

Discussion

Elicitation is a way to share a representation of the stakeholder mental model about the system.
How can we do it?

Elicit stakeholder requirements

- Interviews, questionnaires
- Structured workshops with brainstorming
- Observation of environment or work patterns
- Technical documentation review
- Market analysis or competitive system assessment
- Simulations, prototyping, modelling
- ...

Constraints sources

- External or organization stakeholders
- External, interacting, or enabling systems.
- Activities from other life cycle phases and technical activities
- Measures of effectiveness and suitability that reflect overall acquirer/user satisfaction

Record the requirements

- Record the requirements in a form suitable for requirements management through the life cycle and beyond.
- Consideration should be given to using a requirements management tool, especially for more complex projects. This tool should have the capability to trace linkages between requirements to show relationships.