

Requirements Management

Requirements Management

The Requirements Engineering Process

It covers many tasks

- Elicitation
 - Expression and modelling
 - Requirements exchange
 - **Management**
 - **Verification and validation**
-

The Requirements Dilemma !!

- What is the best method for expressing requirements
- Formal or semi-formal ?

You said FORMAL

Nice looking formula

Example : Shakespeare (*the Gil Vicente of English Theater*)

To be or not to be , that is the question

Formalising ? ⇔ ? Nice looking Formulae

$2B \vee \neg 2B ?$ W. Shakespeare

Another example : a romantic invariant property

$\forall x \in M, \forall y \in Flove(x, y)(p1)$

$\forall x, \exists y | love(x, y) \rightarrow love(y, x)(p2)$

$pre\{p2\}$

$get_married(x, y) \vee live_together(x, y)$

$Post\{Get_marriage_experience(x, y)\}$

Classes of languages, models, tools, methodes, techniques

- Consider Software systems
 - Any language can be a requirement language
 - A programming language is a requirement language for the computer to execute what is required
 - Example : While {...} do
 - It is well specified and no ambiguity
 - **However in RE**
 - There many stakeholders
 - Different cultures (not necessarily computer scientists or familiar with programming languages)
 - Requirements are of many types
-

Classes

- Programming languages
 - Specific notation
 - General purpose Methods
 - Informal
 - Semi-formal : Used in industry
 - Formal : Developed by academia
 - Abstract : limited to specific issues (pure academia work)
 - Paradigms
 - Function oriented
 - Object
-

Review of basic related aspects (*seen in other courses*)

- Control structure (behaviour) : seq; //, if .. Else
 - Communication (shared data, synch, async, ..)
 - Abstraction
 - Encapsulation
 - Properties
 - Invariants
-

CRITERIA (*CMU-DoD_SEI Taxonomy*)

- Representation
→ Concepts and techniques described using the technique
 - Derivation
→ Methods to produce a specification from another
 - Validation-Verification
→ Properties that can be determined using the specification technique
-

REPRESENTATION

- Style
 - Concurrency
 - Communication
 - Non-Determinism
 - Fairness
 - Modularity
 - Time
 - Data
-

DERIVATION

- Transformation
→ Transformation rules from a specification technique to another (e.g multiformalism approach)
 - Elaboration
→ Same as above with a refinement process
 - Composition
→ Combination of various methods for a complex system
-

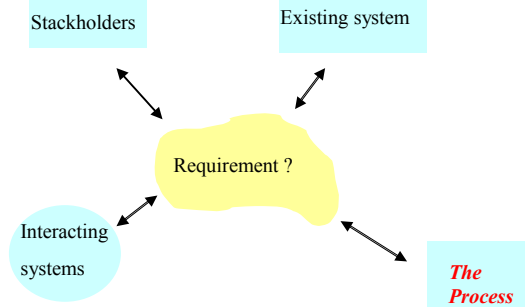
VERIFICATION-VALIDATION

- Equivalence
 - Consistency
 - Safety and liveness
-

An example of taxonomy

Criteria	Methods	VDM	RDP	Stateate	OMT	SART	LOTOS	SDL	Z	B	Estelle
Rigor		3	3	2	1	1	3	2	3	3	3
Data modeling		3	1	2	3	2	3	2	3	2	2
Function modeling		3	0	2	2	2	1	2	3	3	3
Control structures		2	3	3	2	2	3	2	0	2	3
TC expression		0	2	2	0	1	2	2	0	0	1
Exception handling		2	1	3	0	0	3	2	0	0	0
Verifiability		2	3	1	0	0	3	2	0	0	1
Validability		3	3	3	0	2	3	2	1	2	2
Modularity		2	1	3	3	2	2	2	2	2	1
Level of abstraction		3	1	3	2	2	2	2	3	3	1
Reusability		2	1	3	2	1	2	2	1	3	1
Implementability		2	2	2	2	1	2	3	1	2	3
Friendliness		1	2	3	3	3	1	3	2	1	2
Tool maturity		3	3	3	2	3	2	3	1	1	2
		0	3	3	2	1	3	3	1	1	2

The backbone is requirement Elicitation success



4 Dark Corners (see Zave and Jackson paper)

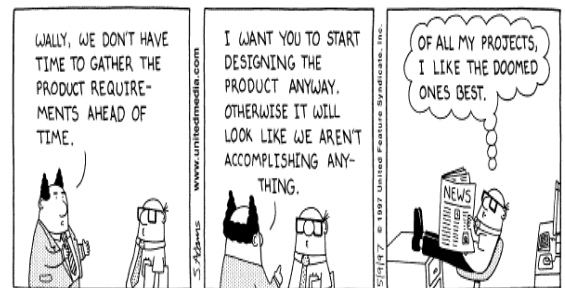
- Tendancy to concentrate only on the system to be built
- Lack of description of the environment
- Confusion in actions controlled by environment and those by the system
- Domain knowledge issue



Thinking about managing requirements :
It is not a tool problem only

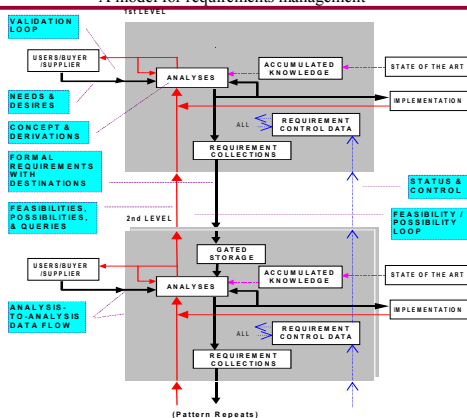


RE According To Dilbert



Copyright © 1997 United Feature Syndicate, Inc.
 Redistribution in whole or in part prohibited

A model for requirements management



Some figures

- Aeronautic : 10 to 15 000 requirements
- Elicitation duration : 4 to 10 months
- A need for a process management

→ **A need to manage complexity**

- * Increasing number size of requirements
- * Diverse types
- * Changing

	1. Tiny	2. Small	3. Medium	4. Large	5. Huge
Design professionals (number)	3	15	75	400	2,000
Duration (months)	12	24	36	48	60
Geographical area	Building	City	Multi-city	Country	World
States/modes (number)	3	30	300	2,000	10 000
States/modes (type)	Well defined	Deterministic		Stochastic	
Interface params (no.)	9	90	900	6,000	30 000
Physical environment of product	Indoor	In/outdoor 1 location	Outdoor global	Global	Space, Ocean
Requirements (no. of text pages)	15	150	500	2 000	50000
Requirements (approximate no)	150	1 500	5 000	20 000	500 000
Number of specs generated	1	3	10	200	1000
Examples	Electric kitchen mixer; low budget art film	Housing development; TV Set; grand opera	Night viewing system; highway system	Aircraft	Space vehicle; Battleship & armament

Why ?

- Requirements are basis for all successive steps
- It is the main reference for
 - Changing requirements
 - Adding a requirements
 - Changing a design
 - Improving the implementation (technology update)

What to manage ? What to do ?

- Managing requirements Is
 - Identify a requirement
 - Affect key attributes
 - Know all its history
 - Who expressed it
 - Validation test
 - Corresponding design
 - Corresponding implementation
 - Versions
 -

Some example through the lecture

- Control of a lift system : the lift has to bring people from a floor to a floor
- Refinement
 - The person calls the lift at any floor
 - The choice of destination is made when inside the lift
 - The door is maintained open for 30 sec unless closed or open by user (open min 10 sec)

Finish first the elicitation process

Requirements Attributes

- Characteristics : types, application
- Validation related links
- Traceability related links
- Stakeholder related links

Requirements characteristics

- Incose : see www.incose.org/rwg
- **Requirement Type**

The *requirement type* identifies the source and contractual applicability of a requirement.

* **Primary.** Those requirements (whatever their form contract provision, specification, database, market analysis, etc.) levied on a contractor/producer under force of contract.

The identification is unambiguous. If

– contractually applicable, or has the force of a contract as in a market analysis, and

– comes from a source external to program personnel, it's a primary requirement. An example would be "*The system must operate under temperature environment interval (-10, 50) °*".

Requirements characteristics (In cose/rwg)

- * **Derived.** Those requirements (whatever their source internal, supplier, team member, customer, etc.) that are generated apart from the primary requirements. The identification is unambiguous. If it is not a primary requirement, it is a derived requirement. An example would be "the card board temperature work in the interval (5,30)".
-

Req. Attribute : application

- **Requirement Application :** *The requirement application* identifies the object of a requirement, There are only two requirements applications:
 - **Product Parameter.** A product parameter is a requirement that applies to the product or service to be developed. The identification is unambiguous. If it directly acts on the product or service, it is a product parameter. Examples of such requirements would include "The external surfaces of all equipment shall be painted white", and "The operator training program shall result in a pass rate of not less than 95% for qualified candidates".
 - Product parameters have two primary subdivisions:
 - * **Qualitative** - This product parameter contains **no measurable requirement**. An example would be "The mixer shall produce a mixture of homogeneous appearance." A qualitative product parameter usually necessitates further analysis to determine suitable quantifiable criteria; i.e., usually necessitates the development of derived requirement(s), with the qualitative product parameter serving as the parent.
-

Req. Attribute : Application

- **Quantitative** - This product parameter contains a **measurable requirement**. An example would be "The mixer shall produce a mixture having a fineness granularity of XYZ within five minutes ", "a query should last less 5 sec". A quantitative product parameter may have children, but such children would be generated for the purpose of specifying particular approaches to meeting this measurable requirement.
-

Req. Attribute : Program parameter

- **Program Parameter.** A program parameter is a requirement that applies to the activities associated with enabling the creation of a product or service, such as conducting traders or holding design reviews. This also includes contract provisions which define the customer/contractor relationship; e.g., confidentiality, intellectual property rights, warranties, etc. In short, a program parameter defines activities related to the technical and administrative management of product development. The identification is unambiguous. If it does not directly act on the product or service, it is a program parameter. Examples of such requirements would include "The contractor shall develop a concept of operations " or "The contractor shall conduct internal design reviews every two weeks. "
-

Req. Attribute : Program parameter

- Program Parameters have three primary subdivisions:
 - **Task** - Identifies an analysis or other effort to be performed. Examples would include "Prepare a Systems Engineering Management Plan" or "Perform an analysis of the structural loads on the bridge pylons".
 - **Compliance Evaluation** - Identifies the methodology for measuring compliance with parameter. (There should be a verification program parameter associated with each product parameter.)
 - **Regulatory** - Identifies administrative elements such as "Deliverable data shall be furnished with unlimited rights to the Government".
-

Req. Attribute : Compliance levels

- **Requirement Compliance Level :** compliance levels:
 - **Mandatory.** Whether a primary or derived requirement, it *must be implemented*. Such requirements usually include a "shall" statement in their structure.
 - **Guidance.** Whether a primary or derived requirement, it is *desirable that it be implemented*. In general, failure to implement does not constitute noncompliance so long as it can be demonstrated that a reasonable degree of implementation was attempted.. An example would be "Use Mil-Std-499B as a guide in implementing the systems engineering process. "
 - **Information.** This unique characteristic is essential when requirements management systems (requirements databases) are used in lieu of hard copy source documents. By strict interpretation, these "requirements" are not actually requirements, but are *non-binding statements which significantly influence the context, meaning, and understanding of other requirements*. An example might be a reference to the customer's reasoning for specifying a particular approach or requirement.
-

Req. Attribute : Priority levels

- **Priority**

- This characteristic identifies the relative importance of a requirement in terms of implementation, particularly in establishing criteria for trade studies.

- In a cost reimbursable environment, a customer may use priorities to mandate that certain elements must be completed before a specified ceiling is reached.

- The priority characteristic may also be used for establishing the sequence in which specified design or test activities should occur. Unlike the other characteristics

- **the values of priority will be dependent on program and company needs.**

READING : See paper Ivy Hooks and Larry Fellows

Req. Attribute : Summary

- **Types**

- Primary
- Derived

- **Application**

- **Products parameters :**

- Qualitative: Functional, Process
- Quantitative: performance, procedural, Physical

- **Programs parameters :** *Task, compliance, evaluation*

- **Compliance :** Mandatory, guidance

- **Priority :** Specific to company needs

Traceability

- **Definitions**
 - **Requirements for traceability**
 - **Techniques**
 - **A generic model of traceability**
-

Traceability : Definition

- **Traceability (IEEE) :** the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another
 - **Traceability (ISO 8402) :** is the ability to trace the history, application or location of an entity by means of recorded identifications
-

Traceability : In General

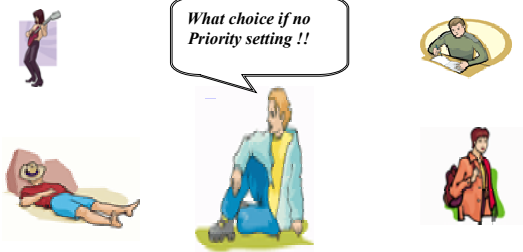
- **Consumer protection** is a cornerstone of our economic system. Therefore it is not surprising to see that *traceability* is an efficient method of exoneration from any dispute which may occur by proving that we work by the book.
 - **Traceability as a security factor :** Any defective products which present a serious risk to the user must be recalled without delay. Therefore it is absolutely necessary to have a distinctive mark on the products.
 - **Traceability as an investigator :** If a defective product has to be recalled, it is because it has passed through the checks that should have stopped it. Therefore we have to go back to the cause of the problem in order to find the solution.
 - **Traceability: an element of industrial politics :** Knowing what and how was done can be essential when responding to a client order. Traceability can lead to a greater knowledge of the company's capabilities, making it possible to meet an order in a shorter time and at a lower cost. Traceability can also be the point of initiation of the statistical methods of process control (SPC).
-

Traceability : In general (2)

- **Traceability: a stimulus for technical progress :** If you are content to accept the final result of a progress which conforms to specifications without question, there is a great risk of becoming complacent and losing one's motivation, resulting in a decreased competitiveness.
 - **Traceability: getting to know the customers :** *Traceability* also enables the collection of information concerning consumers and their spending habits. This in turn enables the customers to be categorised according to the marketing goals.
-

Prioritization and Categorization

- Can be linked to compliance level
- Help to discriminate in presence of constraints



Prioritization

- Parameters factors
 - Budget
 - Safety

Categorisation and classification

- Nature of requirements
 - Functional
 - Non functional
 - Security
 - Performance
 - etc ..
- Validation related
 - Consistency
 - Clarity

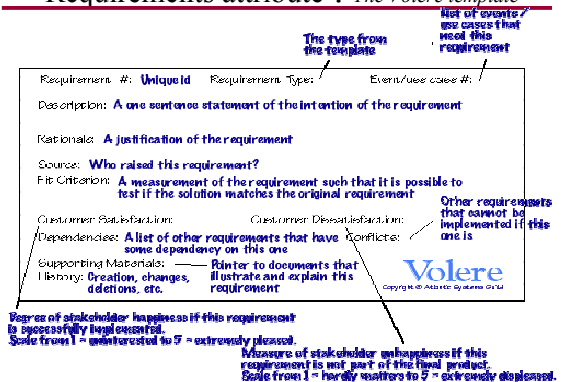
Categorisation

- Can be also structured through others key issue as
 - type of stackholder or derived
 - type of requirement (technical issue)
 - Technical
 - level of abstraction
 - technological basis
 - Non technical
 - Management
 - financial

Categorisation (see paper by A. Gabb)

- Organize requirements according different viewpoints
- Determine the requirements applicable to different aspects of development (safety, ...)
- A way to detect redundancy and consistency
- Planning and modelling (expression)

Requirements attribute : *The Volere template*



Requirements attribute : *The Volere template*

- **Requirement #** is the next unique requirement number
 - **Requirement Type** is the section number from the template for this type of requirement
 - *Requirement #: 128 Requirement Type: 9*
 - *We shall record the time when we are notified of a gritter truck breakdown*
 - A performance requirement comes from section 12, and the next unique number is 129.
 - *Requirement #: 129 Requirement Type: 12*
 - *Gritter truck drivers shall be informed of their schedule 30 minutes before leaving the depot*
-

Requirements attribute : *The Volere template*

- **Client:** The person or organisation for whom the product is being built, usually responsible for paying for the development of the product.
 - **Customer:** The person or organisation who will buy the product (note that the same person/organisation might play both the client, customer and sometimes user roles)
 - **Design or Systems Design:** Crafting a solution to fit the requirements.
 - **Developers:** The people who specify and build the product.
 - **Domain of Interest:** A subject matter area that has some relevance to the context of study.
 - **Non-Functional Requirement:** A property that the eventual product must have.
-

Requirements attribute : *The Volere template*

- **Examples** of stakeholders include:
 - Users (detailed in section 3)
 - Sponsor
 - Testers
 - Business Analysts
 - Technology Experts
 - System Designers
 - Marketing Experts
 - Legal Experts
 - Domain Experts
 - Usability Experts
 - Representatives of external associations
-

Requirements Metrics

- <http://systemsguild.com/GuildSite/Robs/apmeas.html>



Requirements Metrics

- Requirements, if they are to be at all useful, must be measurable
 - Requirements creep is one of the most common risks in software projects
 - To prevent requirements creep, it is vital that the requirements be measurable and that the requirements process include a measuring and testing activity to which all requirements are subject, without exception
 - See requirement validation course on properties validations and necessity of a requirement to validated.
-

CASE STUDIES

- CARE for A380 (see System engineering lecture)
 - Survivable systems : Software Engineering Institute
<http://www.sei.cmu.edu/publications/articles/ellison-survivable-systems/ellison-survivable-systems.html>
 - **Scientific Requirements and Scientific Commissioning for the SDSS (princeton Univ)**
http://tdserver1.fnal.gov/sdss/project/req_spec.htm
-

Survivable systems : concepts

- Survivability is defined as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.
- Unlike traditional security measures that require central control and administration, survivability addresses highly distributed, unbounded network environments with no central control or unified security policy.
- Survivability focuses on delivery of essential services and preservation of essential assets, even when systems are penetrated and compromised.
- As an emerging discipline, survivability builds on existing disciplines, including security, fault tolerance, and reliability, and introduces new concepts and principles

Survivable systems : Elicitation of Essential Service Scenarios

- Requirements for large-scale systems typically specify a substantial number and variety of services and their usage scenarios for all classes and types of users.
- These services can exhibit substantial variations in properties such as frequency of use, time constraints, and criticality to mission objectives.
- Some services, for example, stock buy and sell orders, are invoked minute-by-minute, and must satisfy strict time constraints.
- The continuous availability and timeliness of such services are usually essential to mission objectives.
- Other services, for example, production of quarterly investment reports, are less frequently invoked, and their use can often be postponed if necessary due to adverse conditions.

Survivable systems : Requirements

- Non-functional requirements** : They are specified in the areas of maintainability, extensibility, scalability and system distribution. In addition, there are implied availability requirements and there is an operational environment that also can specify system requirements.
- Availability requirements** : There is a set of requirements that have to do with viewing the data in the database on demand. Specifically, there is a requirement to view treatment plans on demand. There is also a requirement to view treatment plan history, but this is not considered to be an essential service for survivability purposes.

Survivable systems : Development of Intrusion Scenarios

To begin the process of modeling potential intrusion activity, the requirements were studied to determine potential *motive* that an intruder might have in using the proposed system.

Experience with Internet intrusions indicate that there are several categories of attackers that potentially would have interest in attacking the Sentinel system.

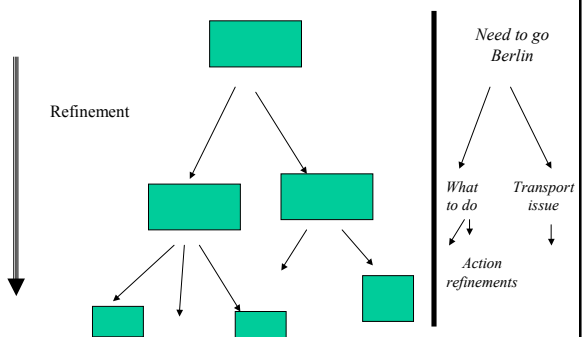
An analysis of security incidents provided the following list of attackers :

- Hackers
- Spies
- Terrorists
- Corporate Raiders
- Professional Criminals
- Vandals

Survivable systems : Medical database for patients

Hackers	<ul style="list-style-type: none"> Curious about medical records (especially on celebrities and public figures) Access as part of a larger sweep of networks regardless of application Badge of merit to access medical records
Corporate Raiders	<ul style="list-style-type: none"> Change of patient records to help a particular provider succeed or discredit another provider Control the resources provided to patients Change doctor recommendations to cut costs
Professional Criminals	<ul style="list-style-type: none"> Manipulate providers and patient care to commit fraud
Vandals	<ul style="list-style-type: none"> Destroy parts of the system to prevent access Maliciously modify records to hurt patients Make random changes

Derived Requirements



Tutorial_1 on requirements management

Give a general structure for a requirements list

Tutorial_2 on requirements management

Consider your RE project : give hints about managing requirements

Tutorial_3 on requirements management

If we consider the Req for a software system : indicate security types issues

Tutorial_4 on requirements management

Indicate requirements types for a web design

Tutorial_5 on requirements management: HCI

Where is the requirement ?

Statement resulting from ACM conference on RE and HCI, Jan 96

It seems surprising that there have been so few attempts to synthesise requirements engineering techniques for interface design and systems engineering. Both disciplines share a common concern to identify and analyse the needs of customers and clients. Both areas have recently identified a common set of problems which frustrate this task. For instance, in both HCI and Software Engineering there is an increasing concern that user requirements cannot be isolated from the external pressures of time and money. This works at two levels. Firstly, financial constraints impose pragmatic barriers to the development process. In Software Engineering this has led to the development of methodological techniques that can be used to gauge and monitor the early stages of the development cycle. These techniques are intended to control costs and support project management. Secondly, the problems of time and money can also impair the elicitation of user requirements from particular individuals within a company. Contextual pressures and the everyday stress of working in complex organisations can prevent designers from accurately gauging user requirements. In HCI this has led to the development of observation and contextual analysis techniques. A prime aim in this workshop was, therefore, to unify the project management skills of software engineering and the broad scope of requirements elicitation in HCI.

Tutorial_6 on requirements management

Develop Requirements management issues for air reservation system

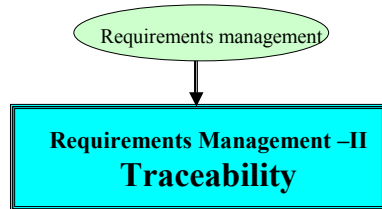
General summary

- Attributes
 - Categories
 - Priorities
-

Conclusions

- Managing requirements
 - Is not a tool pb
 - Parameters and attributes affected to each requirement
 - A process should be set up
 - Activities (traceability, validation, verification, planning, ..)
 - Associated method
-

Next lecture



Paper Reading and assignments

- Papers to read under req_management topics
 - Mandatory : req_cat_gabb.pdf, volere.pdf
 - Optional for further understanding (Others)
 - Assignment
- Apply the Volere template approach to the scheduling problem.
-