Six Sigma and the Application of Psychometric Techniques to Requirements Specifications

Dr Radouane Oudrhiri, Systonomy Limited

8th Annual CMMI Technology Conference and User Group
November 17-20, 2008 - Denver, Colorado

Systonomy background

Founded in April 1999, Systonomy is dedicated to the application of Six Sigma and DFSS to IT and Software Development from real-time and embedded systems to Management Information Systems (MIS) including the implementation and integration of COTS, EAI, ERP systems, CRM, Financial Systems etc.

Systonomy has devised a unique Six Sigma and DFSS framework for IT and Software Engineering that is at the forefront of current knowledge and is investing heavily in research into new methods. Our training has been designed from the ground up as an IT/Software Six Sigma and DFSS training programme and is not a superficial modification of manufacturing or transactional Six Sigma. Our adaptive approach offers our clients an innovative and low risk move from defensive strategies to those of growth.

Our Change Managers, Advisors, Engineers, Black Belts, Master Black Belts and Instructors are IT professionals first and statisticians second

Agenda

1. **Requirements Engineering – The Problem**
2. **The Six Sigma approach and philosophy**
    1. Towards an Experimental and Empirical approach to software process improvement
3. **Case Study – Application of Six Sigma to Requirements Specification**
    1. DMAIC approach
    2. Application of Psychometric Studies
    3. Results
4. **Conclusion**

Quality in Requirements Engineering
Some Research Results

Boehm (1981)

- late correction of requirements errors can cost up to 200 times more than if they were corrected during requirements phase!

Brooks (1987)

- “the hardest part of building a software system is deciding precisely what to build”

Standish (1995) - 8000 projects by 350 US companies

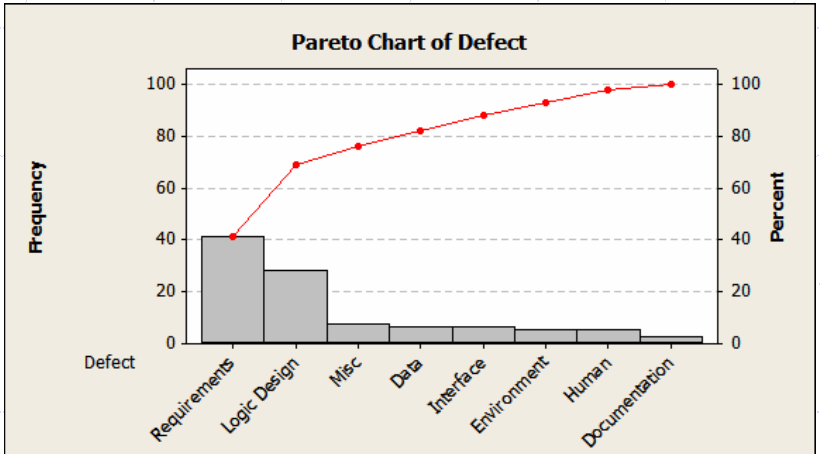
- Poor requirements processes and lack of end-user involvement, was identified as the main source of problems.

Caper Jones (1996)

- Requirements Engineering is deficient in more than 75 percent of all enterprises

ESI (1996) - 3800 organisations in 17 countries

- most (>50%) perceived problems in requirements specification and management.



Defect	Requirements	Logic Design	Misc	Data	Interface	Environment	Human	Documentation
Frequency	41	28	7	6	6	5	5	2
Percent	41.0	28.0	7.0	6.0	6.0	5.0	5.0	2.0
Cum %	41.0	69.0	76.0	82.0	88.0	93.0	98.0	100.0

Getting requirements right might be the single most important and difficult part of a software project.

Why Requirements Engineering is difficult

Many requirements are created, not found.

Inconsistency must be tolerated…for a while.

Little or no agreement on requirements

- Even the simple word "requirements" means different things to different people.

Requirements evolve during and after development

- Two things are known about requirements:

1. They will change!

2. They will be misunderstood!

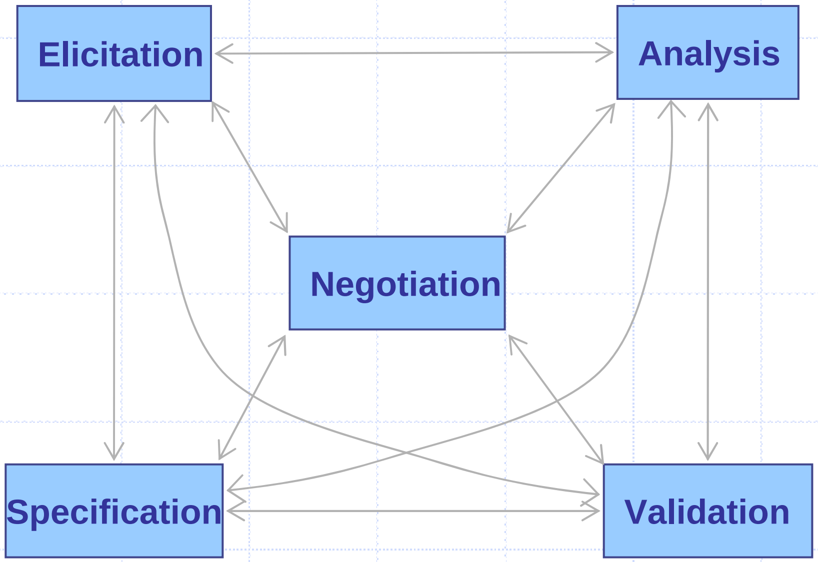
> "Our plan is to lead the public to new products rather than ask them what they want. The public does not know what is possible, we do"
>
> *Akia Morita, founder of Sony Corporation*

> *"The customer doesn't generally know what is needed and neither does anyone else! The initial requirements are therefore wrong and will change."*
>
> W. Humphrey

> "Users are tremendously un-self-aware . . . Software sucks because users demand it to."
>
> Myhrvold

Requirements Engineering – Life Cycle

*The **systematic** process of developing requirements through an **iterative co-operative** process of analysing the problem, documenting the resulting observations in a variety of representations formats, and checking the accuracy of the understanding gained.* [Pohl94]

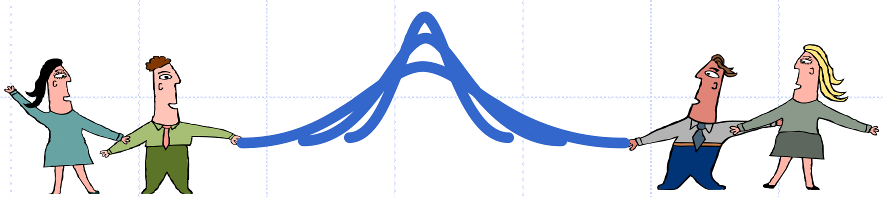


Joe Goguen [1994] says, *"It is not quite accurate to say that requirements are in the minds of clients; it would be more accurate to say that they are in the **social system** of the client organization. They have to be **invented**, not captured or elicited, and that invention has to be a cooperative venture involving the client, the users, and the developers. The difficulties are mainly social, political, and cultural, and not technical*."

Software engineering is a social discipline
Software in not visible…

Information and Software Systems Engineering is a social discipline

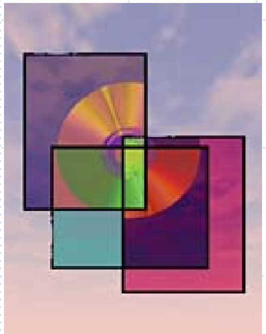
- There is no separation between the knowledge of how to develop products and the knowledge of how to organise development processes



- People are the "big issue" and the single most important source of variation in software development processes

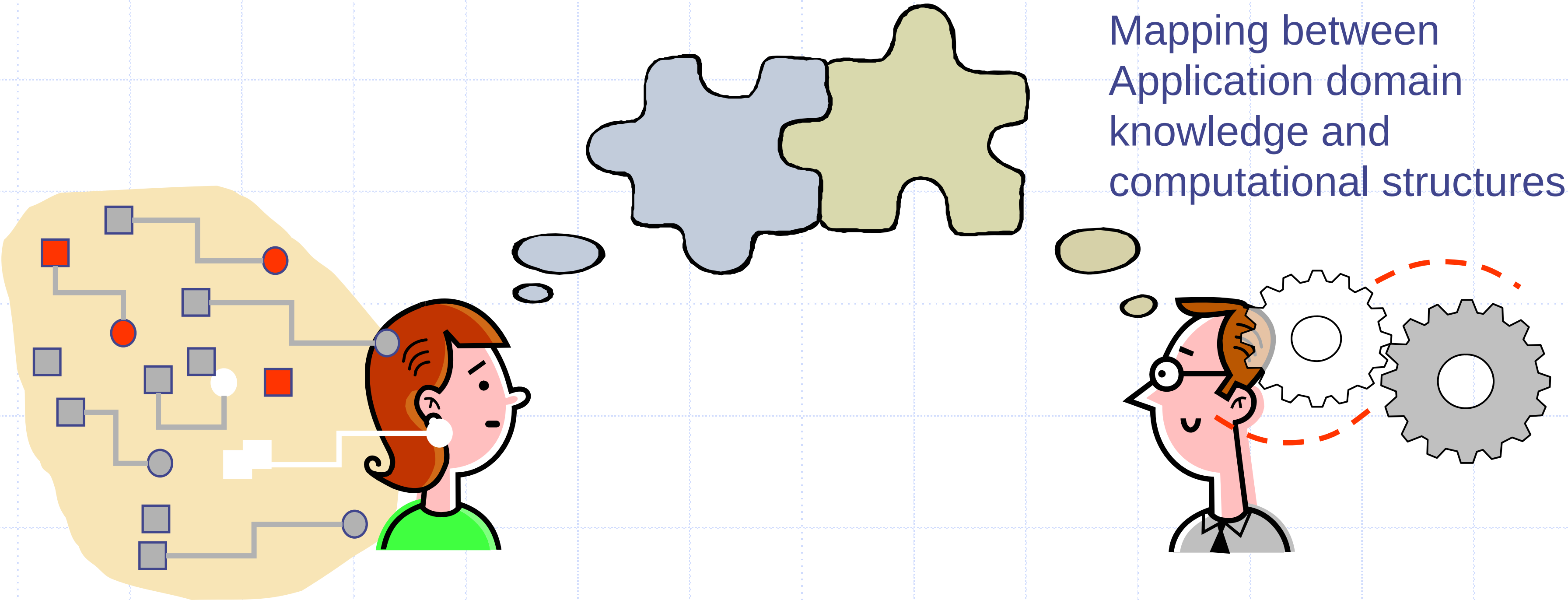
Software is not visible.

- We don't sense software but we sense its effect on people
- SW engineering is a particular engineering discipline where the work is mostly on models and rarely on real world objects.



- We can only create artefacts to make the software more visible during its development life cycle
- Artefacts, are pictures (models)
- There is always a translation process

Models and Modelling



Mapping between Application domain knowledge and computational structures

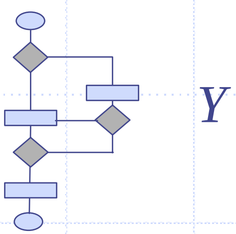
Through a game of representation and interpretation, Software Engineers communicate through shared models

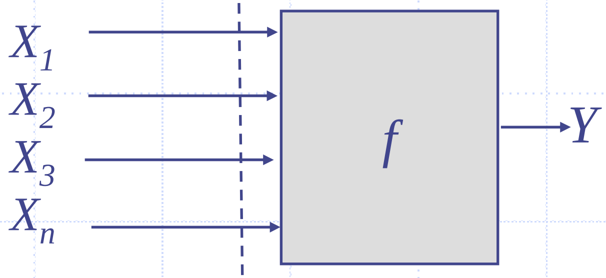
~~Variation~~ **Ambiguity** is the Enemy of Quality in software engineering

Six Sigma Methodology and principles

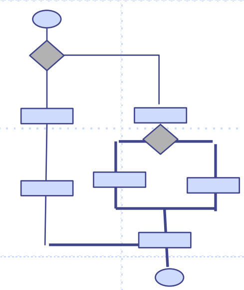
systonomy

DEFINE	MEASURE	ANALYSE	IMPROVE	CONTROL

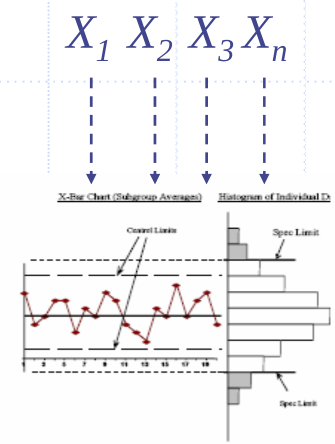




$Y=f(X_1, X_2,..,X_n)+\varepsilon$



$Y=f(X_1, X_2,..,X_n)+\varepsilon$

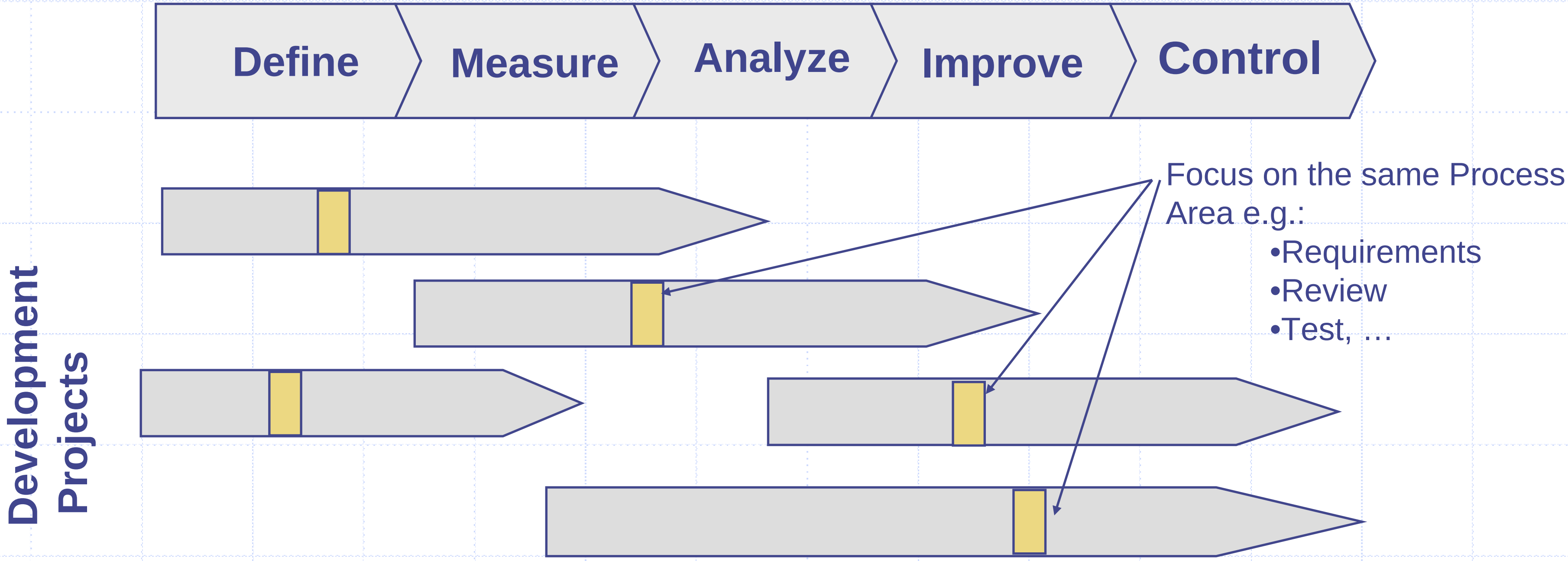


- **Define Problem qualitatively and quantitatively**
- **Determine current performance**
- **Establish Business Case**
- **Determine root cause of problem**
- **Pilot process improvement**
- **Control future performance**
- **Final Business Case Forecast**

Six Sigma is a pragmatic approach to Empirical and Experimental Business Process, Service and Product Improvement!

Experimental Projects

- The development projects are objects of studies and implementation for the Six Sigma Projects



Six Sigma is not just a theoretical concept, the observations and changes must be applied to real projects and improvement must be demonstrated. Six Sigma is not only about defining reference processes!

Quantitative Evaluation of Requirement Quality

Define | Measure | Analyze | Improve | Control

Problem Statement

The capability of the Requirements Engineering Process is low and not optimised. A large number of defects related to requirements are found late in the SLC. Ultimately this results in less last minute new requirements and changes, which affects the product quality, development costs and reworks as well as customer satisfaction and the sales cycle.
As a result requirements are vague, incomplete, not testable and are not prioritised. In addition, there is no technique or method available to measure quality of requirements objectively.

Goal of the Project

Establish KPI driven evaluation of software requirements
Establish learning cycles to ensure continuous improvement in the area of requirement engineering
Foster collaboration between all requirements

Business Case

Reduction of costs and rework effort, due to the lack of specification, in later development phases.
Reduction of defects due to inappropriate requirements
Reduction of artificial change requests

What makes this "6 Sigma" project?

- Strong focus on issues directly perceived by the customer
- Recurring problem
- Making requirements quality measurable

Requirements Engineering Process

- **Identify measurable attributes of interest for the Requirements Engineering Process**

Entry Conditions → **Requirement Definition Activities** (elicitation, analysis, negotiation..) → **Verification** (req. review) → **Exit Criteria**

-Standards
-Guidelines
-Templates
-Checklists
-Market Surveys
-Customer requests
-Change Requests

-Effort *(per phase)*
-Duration, cost

-Size of Document
-Methods
-# of statements
-Clarity / completeness
-Activity Yield (review)

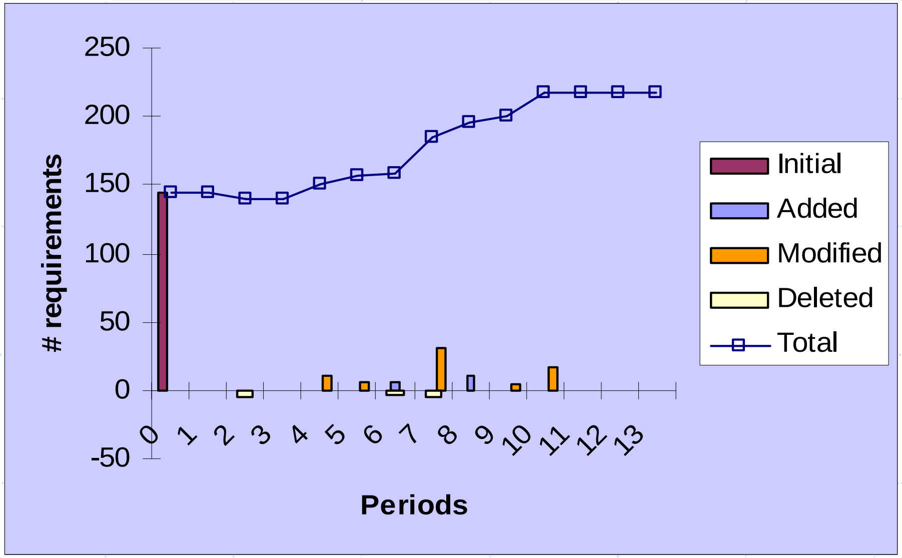
-Tested requirements
-Traceability
- Prioritisation
- Defect density (document)

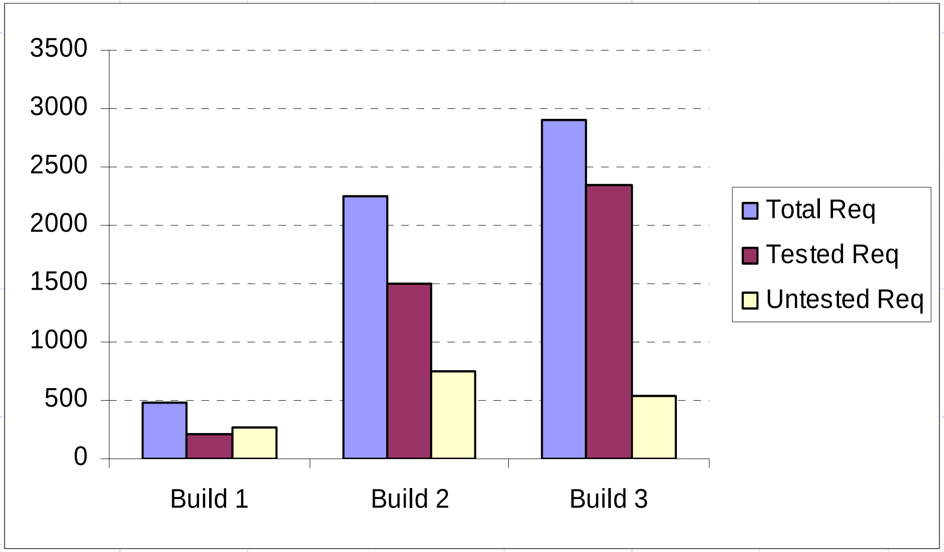
GQ(I)M approach can be used to identify these attributes

Requirements Engineering Process

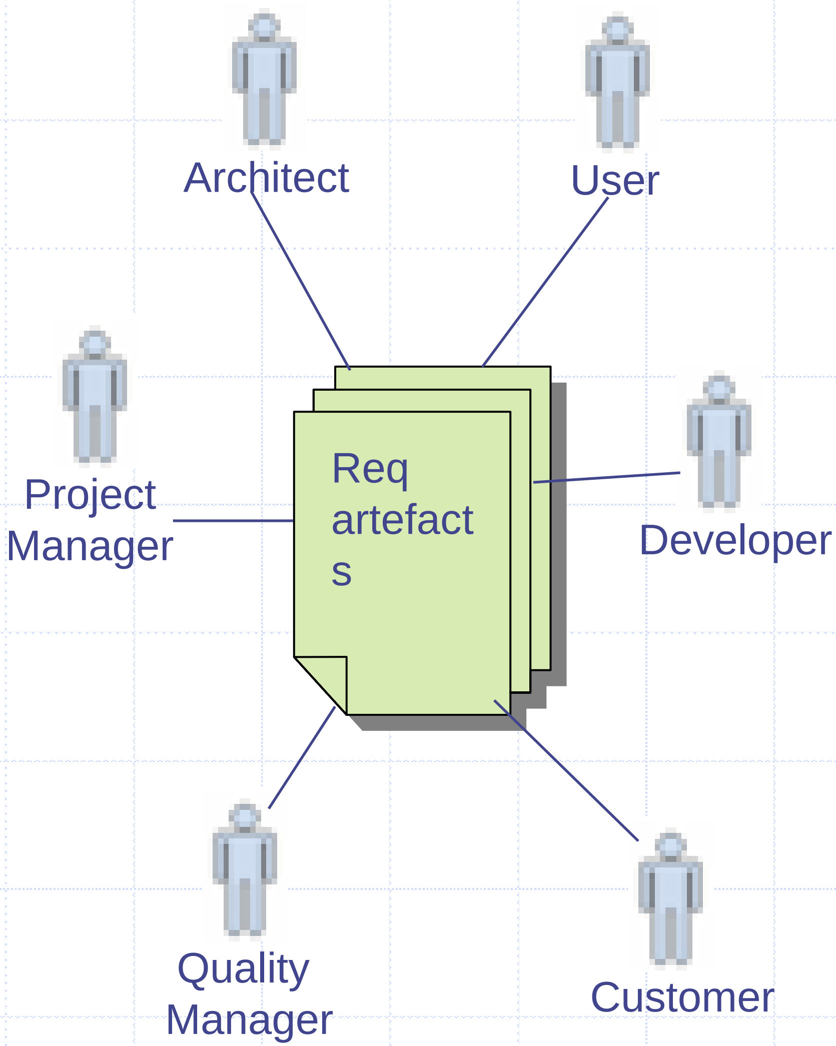
Define | **Measure** | Analyze | Improve | Control

- Some Questions that should be asked
  - Are our market studies complete?
  - Do customer change requests contain the information needed to carry out timely and effective changes?
  - How volatile are the requirements?
  - Are requirements categorised?
  - Are requirements prioritised?
  - How are the requirements documented?
  - Are the documents we produce readable?
  - Are documents concise and complete?
  - Is the terminology correct?
  - Are the documents we produce non ambiguous?
  - Is it possible to trace requirements back to customer needs?
  - Where defects related to requirements are found?
  - Where are the defects being introduced?
  - Are all requirements testable (verifiable)?





Stakeholders Perspective



- Different stakeholders, depending on their context, role, interest, etc, attach different ***values*** to the each requirements.

- This value system should be reflected in the requirements specification and evaluation.
  - Psychometric evaluation of the requirements.

  - Multidimensional evaluation

Quantitative Evaluation of Requirements Quality

Define | **Measure** | Analyze | Improve | Control

Quality attributes for requirements

- Completeness
- Un-ambiguity
- Conciseness
- Consistency
- Correctness
- Realizability
- …

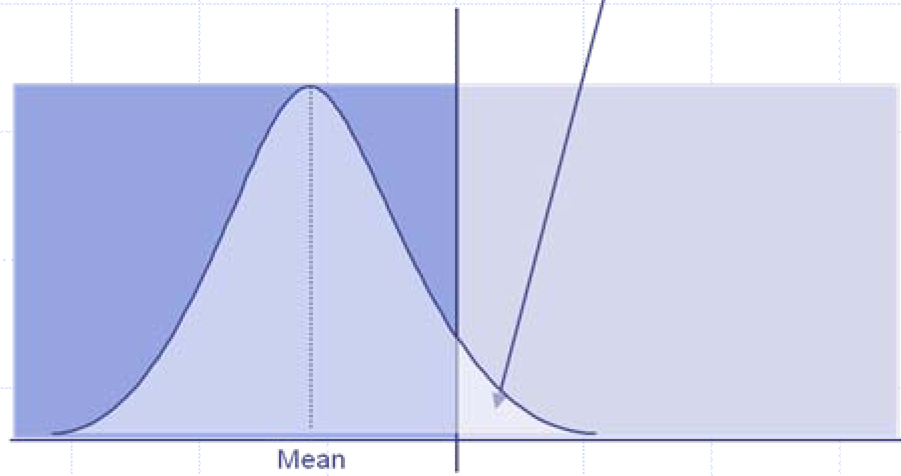
Subjective evaluation by stakeholders on a predefined scale from 1 (low) to 5 (excellent)

- # of data point per requirement = # of stakeholders x # of characteristics
- Aggregation to Requirements Quality Level and Consistency Index

Business Case

- Long term, external effects, hard to take back to requirements because of multiple influencing factors
- Short term, internal effects, possible to demonstrate direct effects

Requirement Quality Level



Concordance Index

i		1	2	3	4	5	6	7	8	9	10	11	12	x'_0
	eval.	4	5	5	5	5	4	5	5	5	4	5	4	
1	4	x	-1	-1	-1	-1	0	-1	-1	-1	0	-1	0	3
2	5		x	0	0	0	1	0	0	0	1	0	1	7
3	5			x	0	0	1	0	0	0	1	0	1	6
4	5				x	0	1	0	0	0	1	0	1	5
5	5					x	1	0	0	0	1	0	1	4
6	4						x	-1	-1	-1	0	-1	0	2
7	5							x	0	0	1	0	1	3
8	5								x	0	1	0	1	2
9	5									x	1	0	1	1
10	4										x	-1	0	1
11	5											x	1	0
12	4												x	0

Sum:	34
# of paires:	66

Measurement Operational Definition

Define | Measure | Analyze | Improve | Control

Example - Unambiguity

- Unambiguity (understandability) is possessed by a requirement if its purpose is clear and free of interpretation. Unambiguous description is supported by using defined key word (e.g. must, should, could) and by the usage of predefined terms.

- Questions and checklist were provided as well training

Evaluation Scale				
1	2	3	4	5
The purpose of the requirement is not clear.	…	Several interpretations of the requirement are possible.	…	You have a clear picture of the meaning.

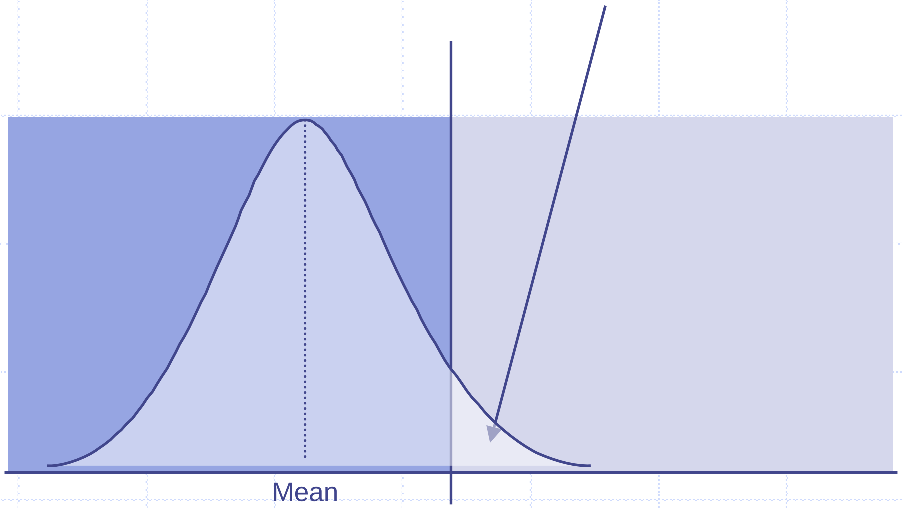
Quantitative Evaluation of Requirement Quality

Define | **Measure** | Analyze | Improve | Control

	Requirement Quality Level	Concordance	Quality Attribute 1	Quality Attribute 2	Quality Attribute 3	Quality Attribute 4	Quality Attribute 5
Requirement 1	57	35	3,60	2,80	2,80	3,80	4,67
Requirement 2	71	43	3,80	3,20	3,80	4,00	4,33
Requirement 3	28	33	1,50	3,25	2,00	3,00	2,50
Requirement x	42	24	2,20	2,60	3,00	3,80	2,33
Requirement N	62	42	3,60	3,80	3,80	3,60	3,00

Requirement Quality Level – Operational Definition

Represents the requirements that fulfill the targeted quality level.



Normsdist: Returns the standard normal cumulative distribution function.

Specification Limit: Defined Quality Target Level out of {2.5,3.5,4.5}

$$RQL = Normsdist\left[\frac{Mean - Spec\_Limit}{StDev}\right]$$

Six Sigma *Continuous Method can be applied*

- The **ordinal data** can be treated as **pseudo continuous data** —especially if the scale has at least 5.
- Prior meta-analysis and research studied user satisfaction data and found that users tended to rate systems/processes/services they preferred a 4 or above on 5 point scales). It would seem reasonable to use 4 as the target value for subjective requirements satisfaction. Our data supports 3.5 as a reasonable breaking point for helping set the goal for requirements satisfaction.
- While the specification limits of 4 (5-point scales) is a good guidepost for setting specification limits they should be used as starting points. Analysts should always investigate data for the specific domain that would either confirm these values as appropriate spec limits or specify slightly higher or lower values.

Concordance Index – Operational Definition

$$Concordance(m)\_Index = \frac{\sum_{i=1}^{n} X_0^i}{\left[\frac{n!}{k!(n-k)!}\right]}$$

Inspired by psychometric studies.

In this example, the analysis of Discordance is more than a Gage R&R, it provides an additional source of information.

i		1	2	3	4	5	6	7	8	9	10	11	12	X_0^i
	eval.	4	5	5	5	5	4	5	5	5	4	5	4	
1	4	x	-1	-1	-1	-1	0	-1	-1	-1	0	-1	0	3
2	5		x	0	0	0	1	0	0	0	1	0	1	7
3	5			x	0	0	1	0	0	0	1	0	1	6
4	5				x	0	1	0	0	0	1	0	1	5
5	5					x	1	0	0	0	1	0	1	4
6	4						x	-1	-1	-1	0	-1	0	2
7	5							x	0	0	1	0	1	3
8	5								x	0	1	0	1	2
9	5									x	1	0	1	1
10	4										x	-1	0	1
11	5											x	1	0
12	4												x	0

Sum:	**34**
# of paires:	**66**

Example:

m = 5; M={1,2,3,4,5}
n = 12
k = 2

Result:

Concordance(5) Index: = 34/66
= 51,51%

X_0^i - Number of equal votes between appraiser i and all the others

m - indicates the number of evaluation levels

n - Number of appraisers

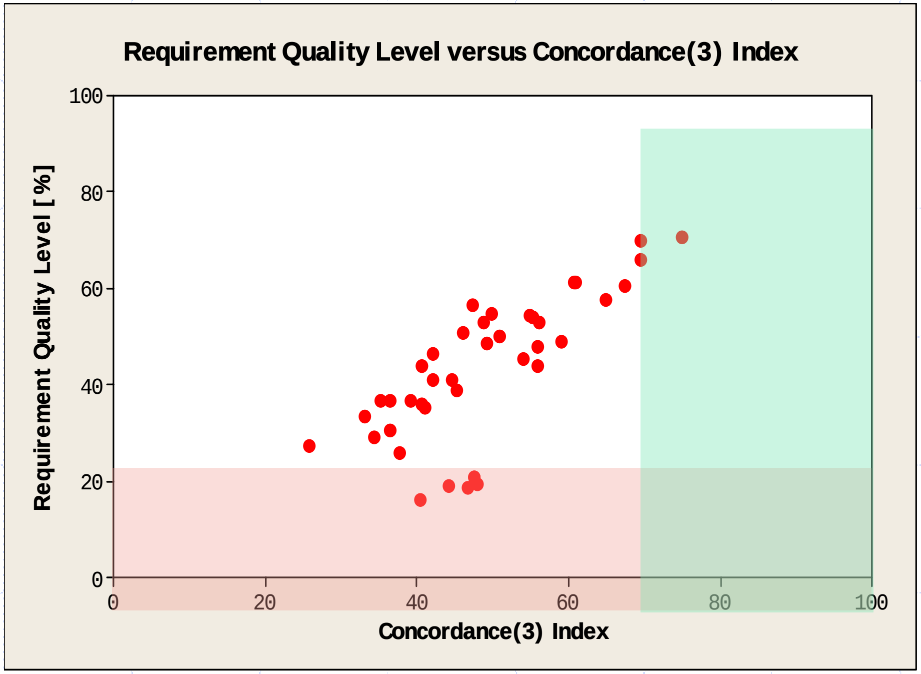
k - paired comparison of evaluations

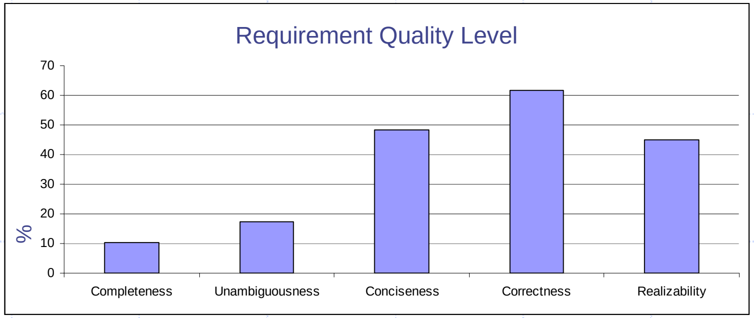
Quantitative Evaluation of Requirement Quality

Define | Measure | Analyze | Improve | Control

Analysis of evaluation results

- On requirement level
  - Possibility to identify weak points in the requirement portfolio
  - Identify requirements that doe not meet required quality level
- On quality attribute level
  - Possibility to identify general areas for improvements
  - Quality attributes can be identified that need systematic improvements
- On stakeholder level
  - Possibility to identify stakeholder groups with unrepresented needs
- All levels can be combined for more detailed analysis





Requirements Engineering Process

Define | Measure | Analyze | **Improve** | Control

Model the RE Process and identify key stakeholders

Introduce New Roles (Business / requirements Analyst)

(*) *Formal Training on RE*

Introduce Req. Management Tool (DOORS)

(*) *Clear measures related to Requirements*

Introduce multiple representation styles (*)

Clear definition of "Defect"

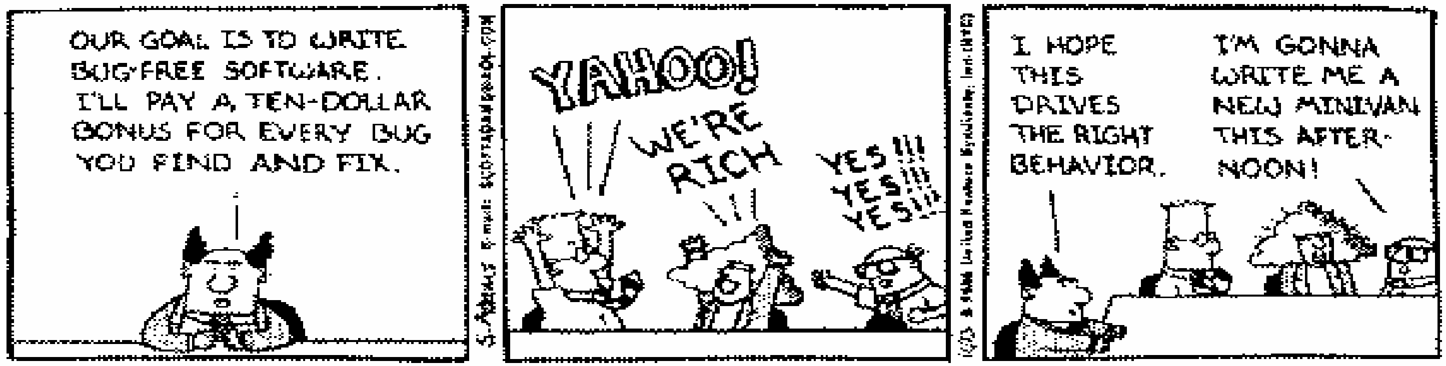
Formal Requirement Reviews and Inspections (*)

Customer involvement in Requirements Review

Improve the Requirements Engineering Process

Measurements drive behaviours...

Define | Measure | Analyze | **Improve** | Control



- Effort moves towards what is measured
- What get measured get done
- Use measurements to drive the right behaviour
- The process must generate its own measures

- Heisenberg's Uncertainty Principle
- DeMarco Theory of Cost Measurement Migration
- Be careful about the WYMIWYG syndrome

Use Measurements as an improvement enabler and catalyst

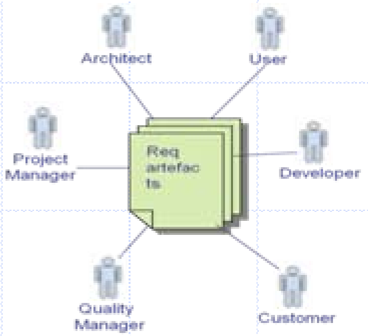
Pilot Six Sigma Project

Define | Measure | Analyze | **Improve** | Control

Pilot

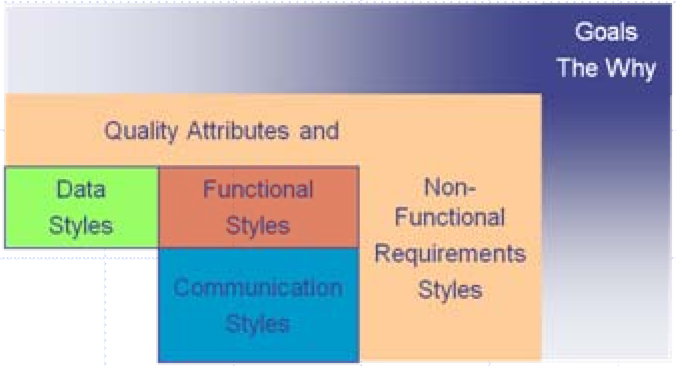
Initial Workshop

- Training on Requirements Engineering
- Introduction of Requirement Styles
- Incorporation of feedback of previous requirements definition cycles
- Agreement on requirements measures and defects



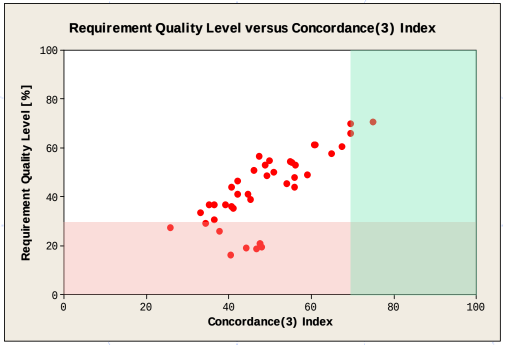
Execution

- Execution of the Requirements Definition Process using various styles
- Data Collection
- Evaluation of requirements using predefined checklist



Post-Mortem

- Analysis of data and measures
- Based on evaluation results gaining feedback (what went well, what needs to be improved)
- Re-work, Lessons Learned

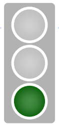


Requirement – Criteria (2)

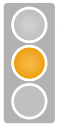
Define | Measure | Analyze | **Improve** | Control

Unambiguity

- Checklist
  - ❑ Are key verbs (must, should, could) used?
  - ❑ Is a glossary available?
  - ❑ Are all used terms unambiguous*?
  - ❑ Are key words and technical terms used consistently? (i.e. use of identical vs. synonymic terms to express identical issues)



- ❑ **Usage of well defined terms**
- ❑ **Usage of clear expressions**
- ❑ **Requirements should be described from the end user point of view**
- ❑ **Use common level language**
- ❑ **…**



- ❑ **Abstract/generic requirements**
- ❑ **Difficult to understand (terminology)**
- ❑ **Unclear terms and definitions**
- ❑ **Complex requirement structure (nested sentences)**
- ❑ **Requirement contains description of expected solution**
- ❑ **…**

* For details on ambiguous terms refer to backup slides

Quantitative Evaluation of Requirement Quality

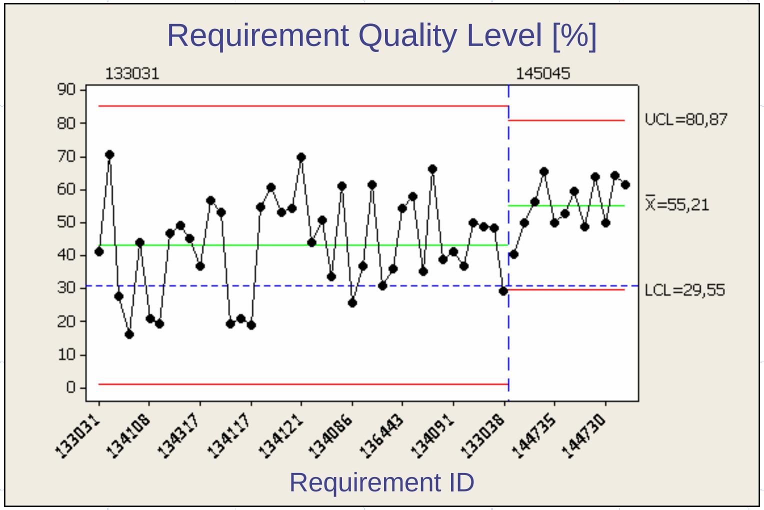
Define | Measure | Analyze | Improve | **Control**

The approach of quantitative evaluation of requirements quality has been integrated into the requirement definition process.

The requirement quality level can be used to control this process.

First projects already have shown: Significant improvements of software requirements quality can be reached.

- Better and broader understanding of development requests
- Smoother hand over of requirements to stakeholders (e.g. development)
- Finally improved product quality



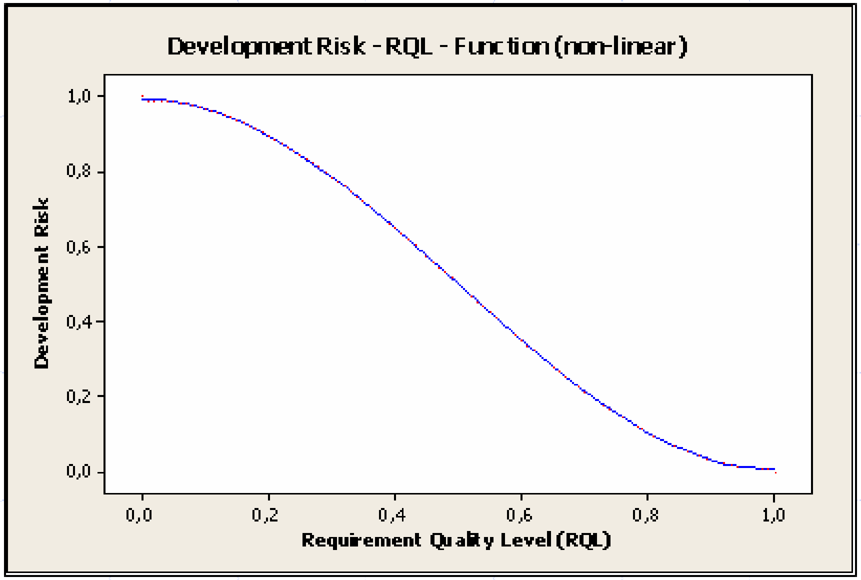
Conducting a second Control Phase for validating result and proving that defects related to requirements are reduced

Quantitative Evaluation of Requirement Quality

Define | Measure | Analyze | Improve | **Control**

Business Case Rational

- Perfect requirements (RQL= 1,0) will reduce the requirement related development risk to zero
- No requirement information (RQL = 0,0) leads to a maximum requirement related development risk
- The relationship between Development Risk Level and Requirements Quality Level can represented by an s-curve



Business Case = Development Risk Level * max. Dev. Risk Value * estimated Development Effort * Cost Rate

Conclusion

- **Requirements engineering is a very complex and sensitive topic**
- **Experimental and empirical approaches are more easily acceptable than prescriptive approaches**
- **Measurement drives behaviour... why not use it to drive the right one**
- **Subjective measurements can be very beneficial for value system**
- **A pragmatic step stone into value based software requirements engineering**
- **Software Engineering is a social Engineering Discipline**

= Science + Economics + Psychology

?

Thank you!

References

- [Pohl94] K. Pohl The Three Dimensions of Requirements Engineering: a framework and its application, Information Systems Vol 19, N 3, pp 243-258, 1994.
- [Pohl96] Pohl K. (1996) Process Centered Requirements Engineering, J. Wiley and Sons Ltd.
- [Ross77] D. T. Ross, K. E. Schoman., Structured analysis for requirements definition. IEEE Transactions on Software Engineering , vol. 3, N° 1, 6-15, 1977.
- [Rolland01] C. Rolland, N. Prakash "From Conceptual Modelling to Requirements Engineering", Special Issue of Annals of Software Engineering on Comparative Studies of Engineering Approaches for Software Engineering.
- [Rolland03] C. Rolland " Ingénierie des Besoins : L'Approche L'Ecritoire " , Techniques de L'Ingénieur, 2003.
- [Wilson et al.] Automated Quality Analysis of Natural Language Requirement Specifications http://satc.gsfc.nasa.gov/support/PNSQC_OCT96/png.html