REQUIREMENTS ENGINEERING

– THE ESSENTIALS

For all practical purposes:

requirements are µ-Contracts

(hence a full contract contains 1000nds of them)

There is one and easy way to capture the essence of Requirements, which from time beginning has been the sole purpose with requirements:

- Requirements are meant for contracts.
- Contracts deliver what is stated in its requirements.
- What is not in the contract, cannot be claimed for.

Once that sinks in, you will have a chance of survival in the voracious corporate jungle. You can sugar coat that with Agile manifesto, and try to work the contract over and over again, to get the requirements right, but at the end, the message still remains the same. *Contracts stand as their requirements are stated.*

Supply chain of requirements

Requirements, start their journey into your contract way outside your organization, and they go right through it and beyond. There is a contract long earlier in your path, which asked your client to deliver something for which they need your help.

Q: Do you deliver to someone?

A: For you to deliver whatever where-ever, there always exists an order, and an expectation for its complete coverage by your deliveries. That summarizes your end of the deal: contract, requirements, deliverables, coverage, completeness.

Q: Is someone delivering to you?

A: At least same level of scrutiny You will apply when You're asking someone to deliver something to You. Because what you ask for, will be what you will get – with best of intentions at least.

Requirements pose a three-fold challenge:

Get Right Requirements. First, you need to get all those requirements in place, which capture your or your client's needs. There cannot be enough emphasis on the methods and tools to widen up your perception of the problem domain, not least in trying to make you understand ***"why"*** your client needs your help in the first place. Increasing thus chances, you might discover something your client missed from the intended contract.

Get Requirements Right. Secondly, you need to check what those requirements actually ask for. Apart from the fuzzy world of Quantum Computers, binary values are still commonly considered mutually exclusive. It is either or, and not both at the same time. Logical inconsistency in requirements is equally damaging as capturing them only partially or misstating them.

REQUIREMENTS ENGINEERING

Get Requirements Right.
Get Right Requirements.

Deliver as required.
Receive as you required.

Requirements Engineering is indeed life of Dr Jekyll and Mr Hyde. Beautiful ideas and bright innovations disappear into what is a logical and legal mayhem.

Product liability.
Contract conditions.
Regulatory compliance.
Conformance to Standards. Evidence.
Cost. Value.

And at the end, everything is just a requirement.

> *"The man never feels the want of what it never occurs to him to ask for."*
>
> – Arthur Schopenhauer

You might want to underline that second 'never', the sole purpose for Requirements Engineering

The Reuse Company
Torplyckegränd 30
162 44 Vällingby, SWEDEN

Deliver as required – Receive as required

This means, you are bound to make interpretations depending what is your role and place in the overall "supply chain of requirements", progressing from one level of contracts to the next. In both, you need to assure, the contract and its requirements from both sides tell the same story. And that is a true challenge.

Contracts have two sides

A contract has two parties: the client and the supplier. It is a handshake between the two, and a balancing act on how the requirements in the contract are to be understood.

Change is everywhere

Real life wins always. For that reason, also your contracts and requirements need to consider the obvious:

	Customer does not care	Customer cares Measurable	Customer cares Unmeasurable
Observable to users	Requirement likely to change	Requirement (user interface)	Goal
Not observable to users	Implementation detail	Constraint	Goal

- **Change management**: prepare the way requirements – as part of contracts – can be coherently managed, to match changes in any part of the "supply chain for requirements".
- **Configuration management:** prepare correct requirements as configuration items with the correct attributes, and make them manageable as part of changes. Manage consistency in versions and variants of contracts and requirements.
- **Knowledge management:** manage large information structures, ontologies with rich variety of dependencies, where version and variants of these libraries of reusable knowledge need to be managed to efficiently reuse organisational knowledge.

Contact Us:

The Reuse Company

Christer Fröling
+46 (0) 72 232 24 63
christer.froling@reusecompany.com
eusecompany.com

AND THEN OF COURSE YOU NEED TO FORMULAT THEM

Surviving all this requires You to write requirements which do not fail you. They need to make sense as individual requirements, and they need to maintain coherence as a group of requirements, but also as a part of total body of information. They need to survive through the refinements and interpretations which your own organization puts them through. You need to Discover them, and you need also to Formulate, Review, Compare and most likely also to Trace them to their parents and siblings. For short, requirements need to be correct and for set of requirements you also add the need for consistency and completeness.

Tools?

Needless to say, to cope with all these moving parts, their versions, dependencies and how these are formulated, you really need to consider using help of a tool for both requirements management AND requirements development and formulation. They represent two equally important sides of efficient and effective Requirements Engineering capability and culture. And while you are thinking on your needs - Remember to embrace your inner Schopenhauer.