Alles hat zwei Seiten, auch das RE
Ein Streitgespräch unter Kollegen

Sven Krause

Product Developer

Business Analyst

Projekt- & Q-Mgt.

Beratung & Coaching

mail@svenkrause.ch
www.svenkrause.ch

Thomas Haas

Software Architekt

Software Engineer

Projekt- & Q-Mgt.

Beratung & Coaching

thomas.haas@infogem.ch
www.infogem.ch

Patrick Strasser
(Swisscom)

Alan M. Davis
Requirements Engineering, Both Sides of the Issues. 16th IEEE RE Conference, 2008

SAQ RE Arbeitsgruppe
(http://www.saq.ch/de/fachgruppen/informatik/arbeitsteams/requirements-engineering/)

Matthias Pohle
(Swisscom)

DANKE

Reinhard Stoiber
(Uni Zürich)

Samuel Fricker
(Uni Zürich, Fuchs Informatik)

Bernd Waldmann
(Phonak)

Frank H. Ritz
(Ritz Engineering)

Nicolas Fauser
(Swisscom)

Marcel Hebeisen
(Swisscom)

Ralf Fahney
(Fahney Anforderungsingenieurwesen)

What and How

What and How must be clearly separated

- Requirements should not define solutions (the HOW), but only the problem (the WHAT)
- If you state solutions in requirements, you restrict the choices the implementers can make
- Product Managers don't know about technical solutions, but they know about customer needs.
- Developers know about technical solutions and feasibility, let them write technical specifications.
- This is the well established separation of requirements (Lastenheft) and design specifications (Pflichtenheft)
- Matches organizational setup: business vs. technical

What and How depend on each other

- Both are interface documents for communication between different domains, e.g. Marketing – R&D, or Architect – Developer
- Design happens in successive refinements: from Needs to Concept to Requirements to Design Spec …
- WHAT/HOW depends on a reader's viewpoint: one person's WHAT is another person's HOW
- Lastenheft/Pflichtenheft are from the contracting business and have legal meaning, we are more interested in cooperative processes
- Mapping from needs to feasible solutions becomes transparent
- Business people like to think in terms of solutions, not problems – so let's talk about solutions in their language

What and How

What and How must be clearly separated

… may be appropriate for:

- purchaser ./. supplier setting
- green field projects

What and How depend on each other

… may be appropriate for:

- in-house implementers
- agile projects
- brown-field and maintenance projects

To document or not

Documented requirements are important

- Externalize knowledge, independent of people
- User needs and design decisions are important to know, and are hard to “reverse-engineered” from the product itself
- Requirements are needed by testers, technical writers, maintenance staff, …
- Basis for system transformation
- Re-use of components are simpler

We got the code, who needs documents

- Who has time to write and maintain all those documents?
- Comprehensive documentation is obsolete by the time it is finished
- Better feedback loop from developer to product owner
- Use the source, Luke

To document or not

Documented requirements are important

… may be appropriate for:

- long-lived products
- products or systems with multiple components, developed by different teams
- when developers and product owner are not co-located
- highly regulated environments

We got the code, who needs documents

… may be appropriate for:

- one-shot projects
- homogeneous software projects
- small teams

Clear cut or murky waters?

Requirements must be complete before design & implementation

- Project resources can be determined
- Project efforts can be determined
- Clear view of expected output
- serenity and stability
- Big picture for IT architecture

Requirements continue to evolve trough implementation

- Feedback loop for improvements
- High flexibility
- Make use of specific resources as needed
- Identified mistakes can be removed before shipping the product
- Priority driven
- Ignore less relevant Business requirements
- Profit from from lessons learned – add new requirements
- Smaller scope can be handled easier

Clear cut or murky waters?

Requirements must be complete before design & implementation

… may be appropriate for:

- Waterfall, incremental process models
- Supply /demand model
- Tendering procedures (WTO)
- Regulated environments
- inexperienced team
- Big bang

Requirements continue to evolve trough implementation

… may be appropriate for:

- Iterative approaches (agile)
- Interdisciplinary teams
- Product development (innovation)
- Staged roll-outs

Traceability

Traceability is essential

- How can you know you are building the right thing without linking design to requirements?
- No change management / impact analyses without it
- Helps with automatic testing, test coverage analysis, …
- Helps with reuse of design artifacts: modify requirements for new product → know what modules are affected
- Traceability is required by law (medical, automotive, …)

Traceability is not practical and too expensive

- Developers don't have time to create and maintain traces
- People aren't disciplined enough to maintain traces, and trace links are worthless if not 100% correct
- Requirements are in the problem domain, design artifacts are in the solution domain (Jackson) – there is no 1:1 mapping
- What is the purpose anyway? Impact analyses? Change management? Justification?
- Doesn't work for NFRs
- Difficult if you don't have integrated tool landscape / Application Lifecycle Management (ALM)

Traceability

Traceability is essential

… may be appropriate for:

- highly regulated environment
- many changes anticipated
- highly integrated Application Lifecycle Management (ALM)

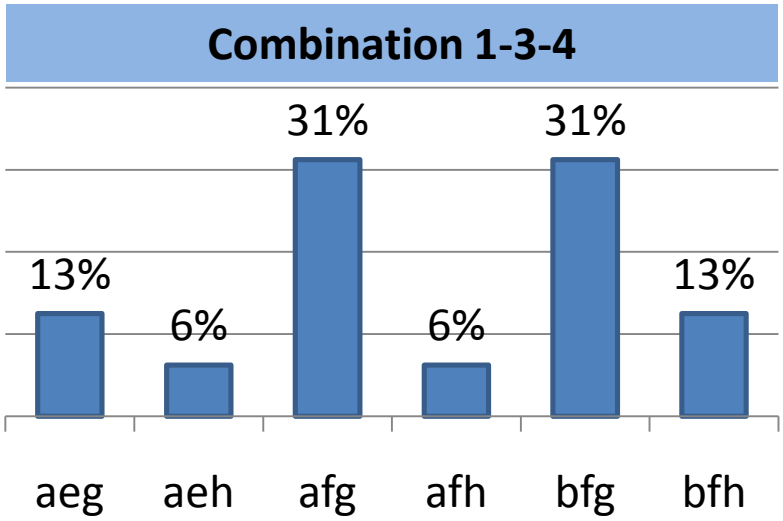
Traceability is not practical and too expensive

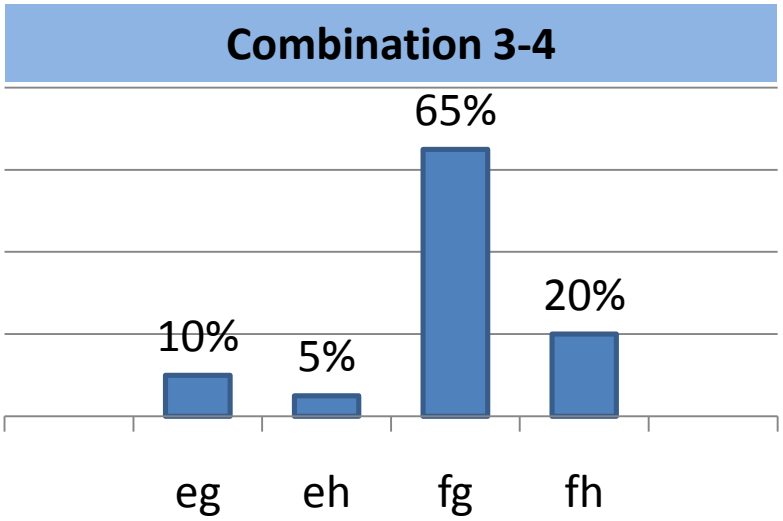
… may be appropriate for:

- one-shot products
- Prototypes
- not supported by tools and processes

Survey at REconf 2010 – 17 participants

1. What and How	a. What and How must be clearly separated	55%
	b. What and How depend on each other	45%
2. To document or not	c. Documented requirements are important	100%
	d. We got the code, who needs documents	0%
3. Clear Cut or Murky Water	e. Requirements must be complete at the start of the project	20%
	f. Requirements can be refined iterative during the project	80%
4. Traceability	g. Traceability is essential	75%
	h. Traceability is not practical and too expensive	25%





Evaluation – agile or not?

a. What and How must be clearly separated:
   Separation of requirements elicitation and solution design → waterfall approach likely
b. What and How depend on each other:
   No strict separation of requirements elicitation and solution design → interdisciplinary work likely, agile approach possible
c. Documented requirements are important:
   product of value and longer term use → any approach possible
d. We got the code, who needs documents:
   product of little value or short term use → agile approach most likely
e. Requirements must be complete at the start of the project:
   full scope defined upfront → agile approach unlikely; waterfall or incremental approach possible
f. Requirements can be refined iterative during the project:
   full scope not defined upfront → agile approach likely; incremental approach possible; pure waterfall highly unlikely
g. Traceability is essential:
   Many motivations → any approach possible
h. Traceability is not practical and too expensive:
   Iteration, interdisciplinary work requires some (transient at least) traceability → agile approach less likely

→ Mature agile implementation: bdfg

→ Pure Waterfall: aceg