Managing Requirements

In Real Life

Willem van den Biggelaar

Quality Assurance Consultant

PROCESS VISION

DOWN TO EARTH PROJECT IMPROVEMENTS

Abstract

Managing requirements in a project developing a complex product such as a wafer stepper or a X-ray system is a hell of a job. But even for smaller products such as a DVD player, the number of and changes to the requirements are huge.

Not only the product but also the project itself can be complex: distributed all over the world in several teams, products are being developed.

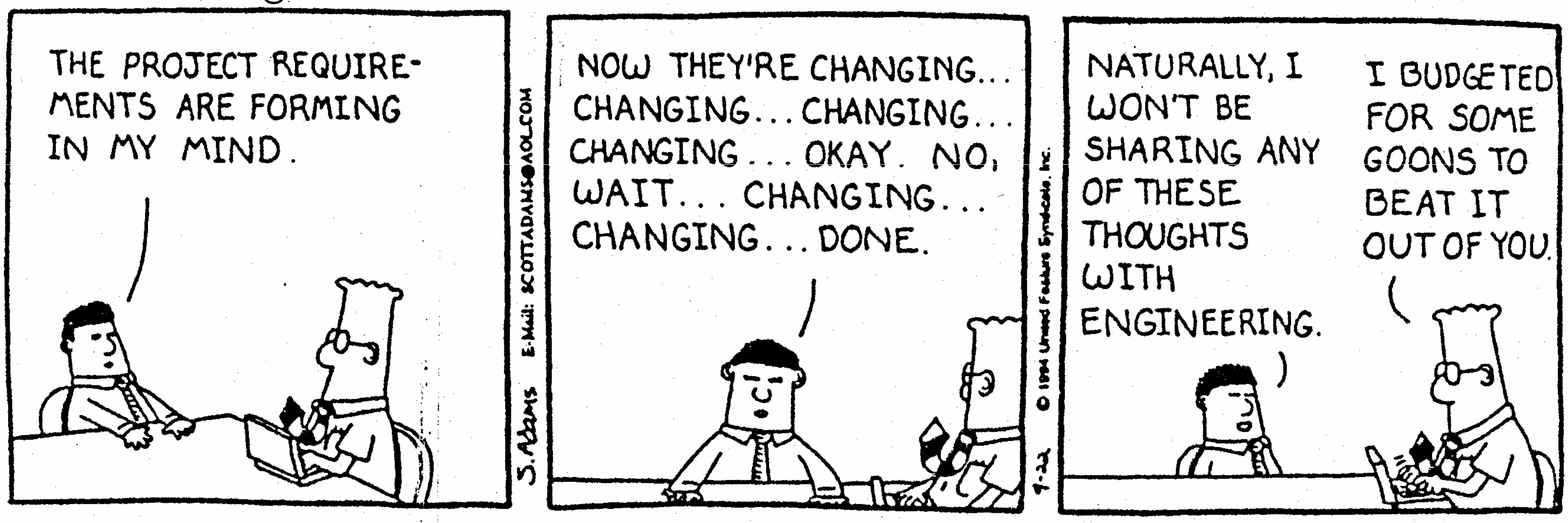
This paper will give a number of practical and pragmatic approaches on how these companies have handled above situations. It will not be the silver bullet but it will give several hints and tips to remember when you have to start developing a product and you are responsible for managing the requirements.

Requirements are……

Initially Fuzzy

Guaranteed to change

Difficult to Communicate


THE PROJECT REQUIREMENTS ARE FORMING IN MY MIND.
S. Adams E-Mail: SCOTTADAMS@AOL.COM
NOW THEY'RE CHANGING... CHANGING... CHANGING... CHANGING... OKAY. NO, WAIT... CHANGING... CHANGING... DONE.
9-22 © 1994 United Feature Syndicate, Inc.
NATURALLY, I WON'T BE SHARING ANY OF THESE THOUGHTS WITH ENGINEERING.
I BUDGETED FOR SOME GOONS TO BEAT IT OUT OF YOU.

Process Vision stands for

- Project consultancy: Quality Assurance officer
- Process Improvement coordination
- Process Assessments
- Training on
  - Requirements Management / Engineering
  - Inspections (Reviews)
  - Quality Assurance
  - Testing
  - Process Awareness in general
- “Down to Earth Project Improvements”
- For more info look at www.ProcessVision.nl

Who I am – Willem van den Biggelaar

- Bachelor degree Electronical Engineering (1985)
- 10 years embedded software
    - Developer / Designer/ Tester / Project Leader
    - Océ, Organon, Philips MS, Draeger
- 12 years Process Improvement / Quality Assurance
    - ASML, Philips CE, Philips MS, Centric TSolve
- Owner of company Process Vision

Why projects fail……Standish chaos report

➡	1. Incomplete requirements	13 %
➡	2. Lack of user involvement	12 %
	3. Lack of resources	11 %
➡	4. Unrealistic expectations	10 %
	5. Lack of executive support	9 %
➡	6. Changing requirements/specifications	9 %
	7. Lack of planning	8 %
➡	8. Didn’t need it anymore	7 %

Source: Standish group

In total 51% is about requirements

Why do we need requirements?

- It is documentation to align with customer on functionality and it forms the contract
- It is input for all other development phases
  - What should we build?
  - Against what should we test?
- Without it, it is very vague what is delivered at the end
  - Reverse requirements engineering will occur

Why do we need to manage requirements?

- Without it
  - Requirements are left ‘floating’
  - Changes are not or badly managed
  - The many changes and discussions occurring during development will not be documented
- Consequence
  - An inaccurate and unstable schedule

Some manager told me…….

“We are a CMM level 2 company,
so we have our requirements under
control !”

Advice:
Don’t start improving requirements
management without also improving
requirements engineering

Effect of product size on your project

		Effect on	
Change driver	Project Duration	Project Costs	Product Defects
Increase time pressure	↓	↑	↑
Increase Productivity	↓	↓	↔
Increase Product size	↑	↑	↑

30-50% of IT products have unused functionality!

Product size: minimise and prioritise

- First step - minimise product size
  - Discuss product strategy internally
  - Use field evaluation on released products as discussion input
- Next - prioritise requirements
- Prioritisation will be done anyway
  - Under time pressure at the end of the project
- So do it upfront at the start of the project:
  - “must have”
  - “should have”
  - “nice to have”
  - “won’t have”

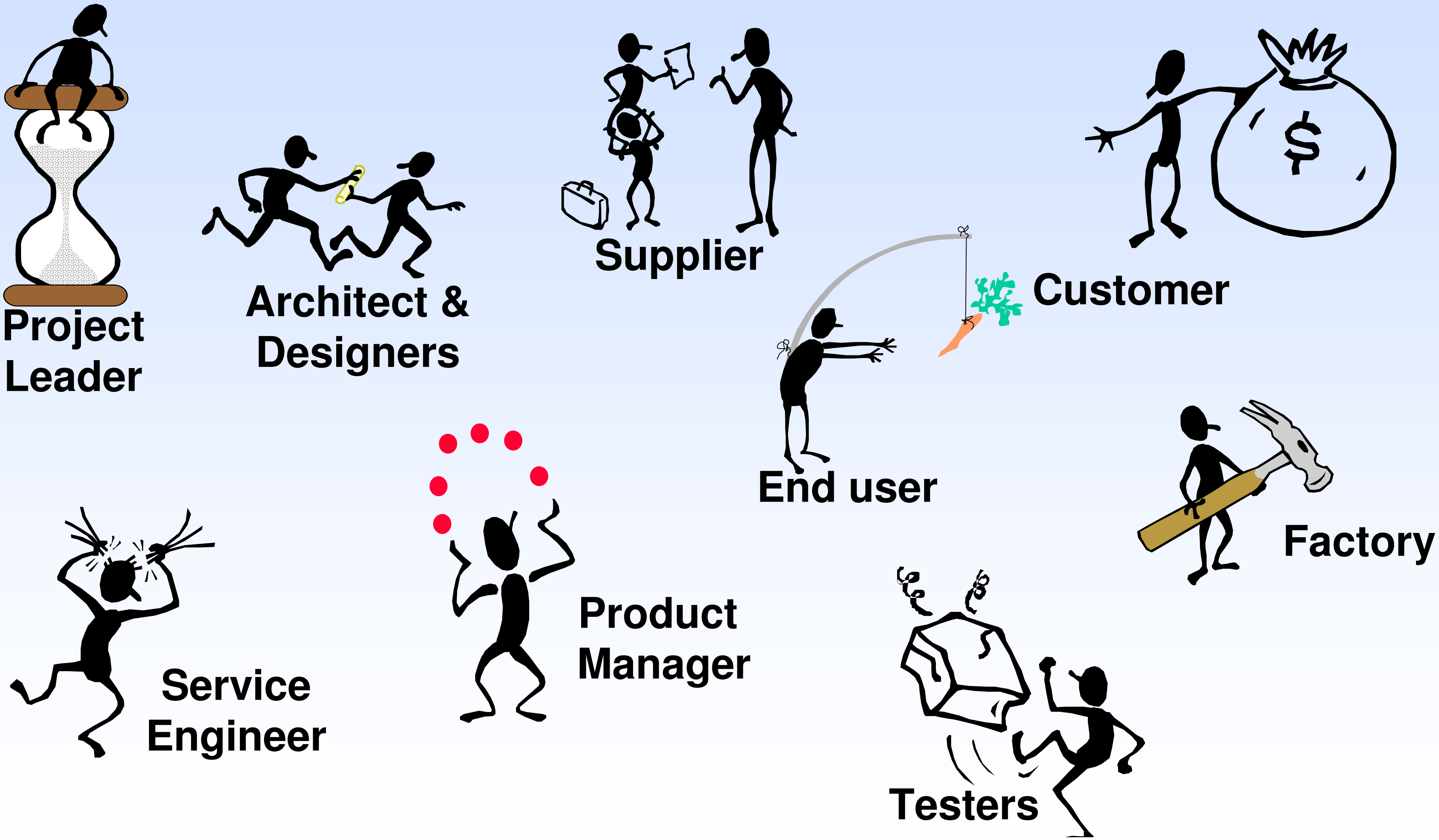
Elicitate (collect) requirements: problem 1

- Problem: authorisation takes too long
  - Lots of open issues
  - Customer has limited time
  - Development has already started with floating requirements

Tip: speed up authorisation

  - Define stakeholders
  - Organise brainstorm sessions with stakeholders.
    Use timeboxing: time leads level of detail
  - Cleary define the left overs in the document as “to be defined” or “to be confirmed”
  - Agree upon process for handling these open issues, e.g. by means of change requests.
  - Baseline and authorize so development & test team can start

Possible stakeholders



Project Leader

Architect & Designers

Supplier

Customer

End user

Factory

Service Engineer

Product Manager

Testers

Only 1 stakeholder has end decision - Customer

How to perform brainstorm sessions?

- Use a facilitator
  - Timekeeper
  - Everybody sticks to meeting rules
  - No content, process keeper
- Use time boxing
  - Time is leading with respect to the level of detail to be reached
- Define roles
  - Customer
  - Technical lead
  - Team
- Ask explicit commitment from everybody
- Immediately write down results
  - Partly brainstorm output
  - Partly already transformed to specifications

Quick results
True commitment (joint effort)

Elicitate requirements: problem 2

- Problem
  - Nobody is (feels) responsible for requirements (tree)

Tip: assign a requirements manager

  - Responsible for establishing the requirements
  - Responsible for deployment of requirement changes in the project
  - Not neccessarily content specialist

Elicitate requirements: problem 3

- Questions
  1. How detailed must a requirement be?
  2. Must all requirements be 100% complete?
- Answers
  1. Requirements must be to such a detail that the consumer (architect or designer) is able to start his design work
  2. To start development this is not needed.
     See also answer to point 1.
     Be sure to manage all open issues via CR's.
     At the end of the project, requirements must be 100% complete.

Changing requirements: problem 1

- No communication of changes towards test team
  - System testers execute tests on ‘old’ requirements

Tip: Improve this communication

  - System testers must be invited for review, if not, postpone.
  - Make test manager permanent member of CCB
    - He estimates impact on test lead time
    - He communicates changes towards test team

Changing requirements: problem 2

- No clear difference between change requests (CR) and problem reports (PR)
  - Everything is simply absorbed by the project as a PR

💡 Tip: define clearly what a CR and PR is

  - CR’s are negotiatable
  - PR’s must be solved

Changing requirements: problem 3

- Changes are handled by one person (the “CCB”)
  - Officially a CCB exists but in practice one person handles all change requests/problem reports

Tip: (Re-)establish CCB from start of project

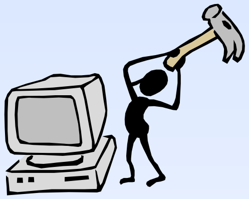
• Requirements Manager	chair
• Project Manager	project impact (time / budget)
• Customer representative	impact for user / product
• Architect	design impact
• Test Manager	test impact

Changing requirements: problem 4

- Senior Management forces project to solve urgent (non planned) customer problems with high priority
- 💡 Tip: let Senior Management sign charter
  - Analyse impact of the urgent problems: time, budget, quality
  - Perform risk analysis
  - Write down in charter and let senior management sign
  - Senior Management is now responsible if project goes down the drain due to these unplanned activities

Traceability problems

- You need traceability for
  - Impact analysis
  - Requirements coverage (Test team)
  - Regulatory (for instance FDA)
  - CMMI level 2 compliance



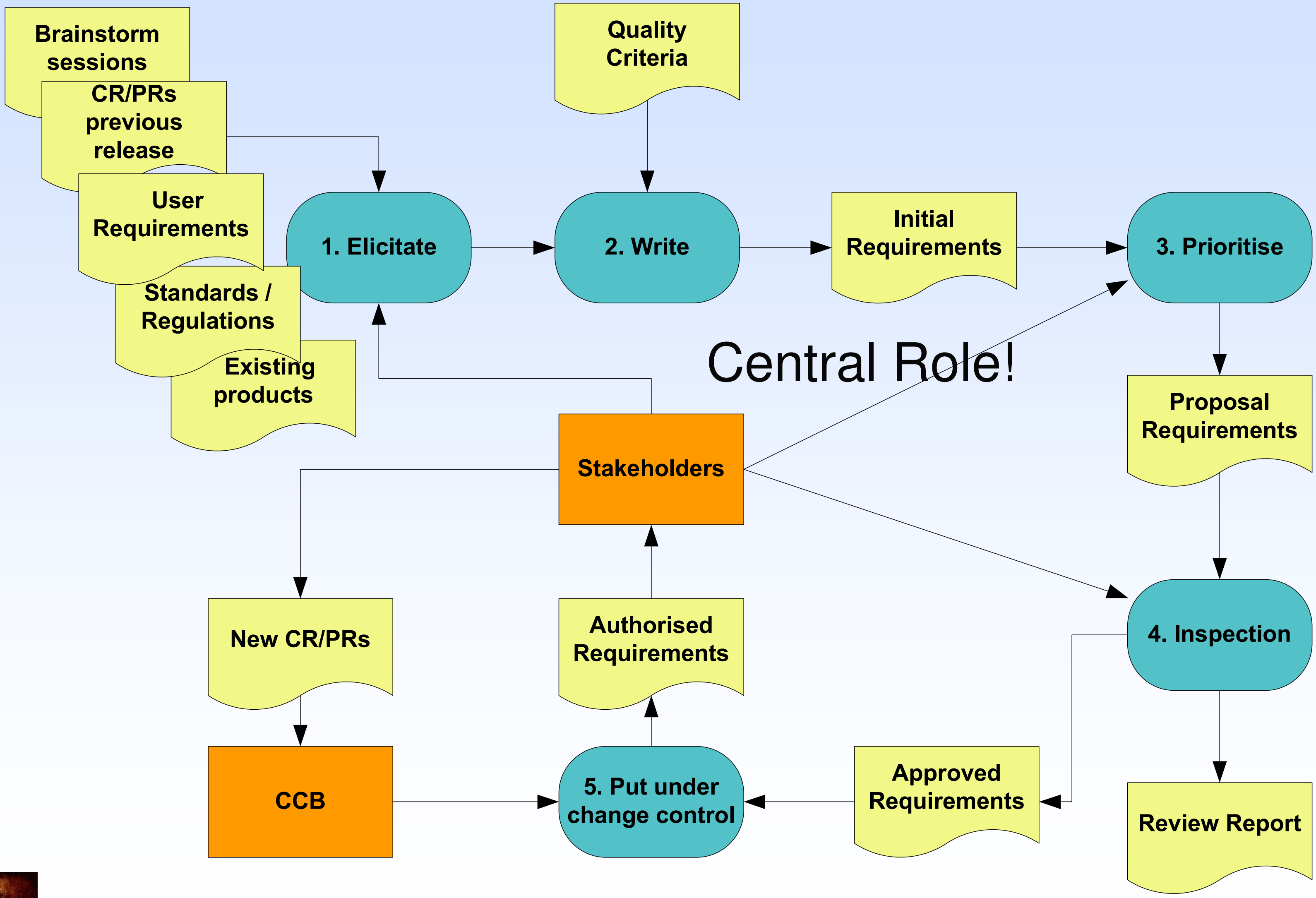
A fool with a tool is still a fool
Only making faster disaster

- Problems with traceability
  - You need uniquely identified requirements
  - For complex systems, a hell of a job
    - E.g. X-ray system contains 3000 requirement pages
  - Tooling is available, but asks for mature organisations
    - Immature organisations only make the mess bigger
    - No discipline to keep it all in the tool- offline editing (suppliers)

Traceability tips

- If you have no traceability, start simple
  - Start with system level only
  - Start on document level: make document tree
- If you have no uniquely identified requirements
  - Tag requirements in the text
  - Discussion about the tagging will follow, that's ok!
- If you use a tool like Doors or Requisite Pro
  - Define and deploy the process before using the tool
  - Everybody (including suppliers) must have easy and quick (remote can be a problem) access
  - Keep strict control: only editing via tool

The Requirements process



Central Role!

Quality criteria for requirements

1.	Correct	It is a flawless description
2.	Complete	It is stand-alone readable and understandable
3.	Consistent	It does not contradict with itself or other requirements
4.	Unambiguous	It has one interpretation only
5.	Necessary	It adds value to the product, no “gold plating”
6.	Feasible	It is possible to implement.
7.	Verifiable	It fits to measure and is checkable by a test case
8.	Prioritised	It has a implementation priority
9.	Traceable	It is uniquely within the product

Summary

- Assign requirements manager
- (Re)-Establish proper CCB (involve test manager)
- Minimise product size
- Use brainstorm sessions
- Prioritise requirements
- Start traceability simple
- Deploy the process before the tool
- Manage open issues (and changes)
- Continuously align changes with your stakeholders