

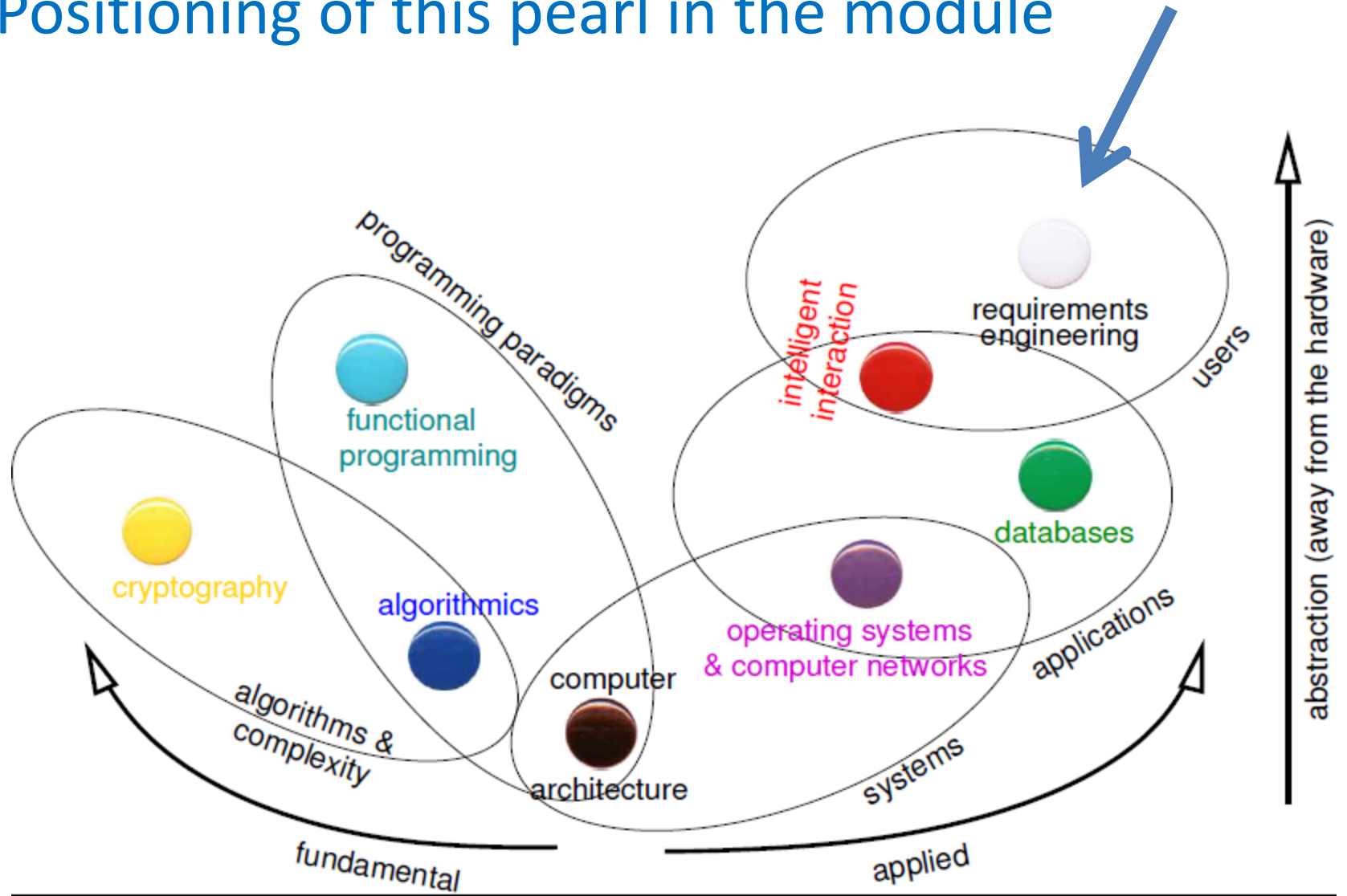
Pearl 111  
Requirements Engineering  
**Lecture 1: Introduction**

Klaas Sikkel

# What we will learn

- ***Project Management***
  - What is a project
  - What makes a project successful
- ***Requirements Engineering***
  - Clarifying the problem / the task
  - Gathering requirements for the product
  - Documenting these requirements
  - *Validating* (do we have the proper requirements?)

# Positioning of this pearl in the module



*Why this topic is very easy for some students and very difficult for others:*

- The focus is on *what a system is to be used for in the real world*
- This is necessarily somewhat imprecise, because the real world is not a formal system
- Requires no math, formulae, algorithms
- Requires a bit of common sense

# Contents of this lecture

1. Short intro, practical arrangements
2. Project management
3. Pitfalls and challenges for (IT) projects
4. Requirements Specification
  - What is it?
  - What do you use it for?
  - How do you get it?

# Practical arrangements

- Study Materials
  - Module guide Chapters 8 and 0 (Project)
  - Introduction to Project Management (+ appendix)
  - A Primer on Requirements Analysis (+ 2 appendices)
  - [Trello](#) (task board)
- Forming project groups:
  - Everybody is member of a project group (?)

# What we are going to do (1)

## ***Monday afternoon***

- Starting next week's Module Project  
*(the pearl assignment is the preparation phase of the module project!)*

...

## ***End of the week***

- You have submitted a requirements specification for the module project

# What we are going to do (2)

**Mon 8+9:** which problem (whose problem) are we going to solve, *what is really the problem?*

**Tue 1+2,3+4:** Exercises for RE techniques

– Lecture (with exercises), Lab session

**Tue 7:** make a planning for the rest of the week

**Tue, Wed, Thu(, Fri):** execute the planning

**Thu 8.45:** Pearl test

**Fri 13.45:** Math test

# How are we going to do it

## ***New?***

- For part of the week it is not prescribed what you have to do when  
*(You can plan – and manage – this yourself!)*

## ***New?***

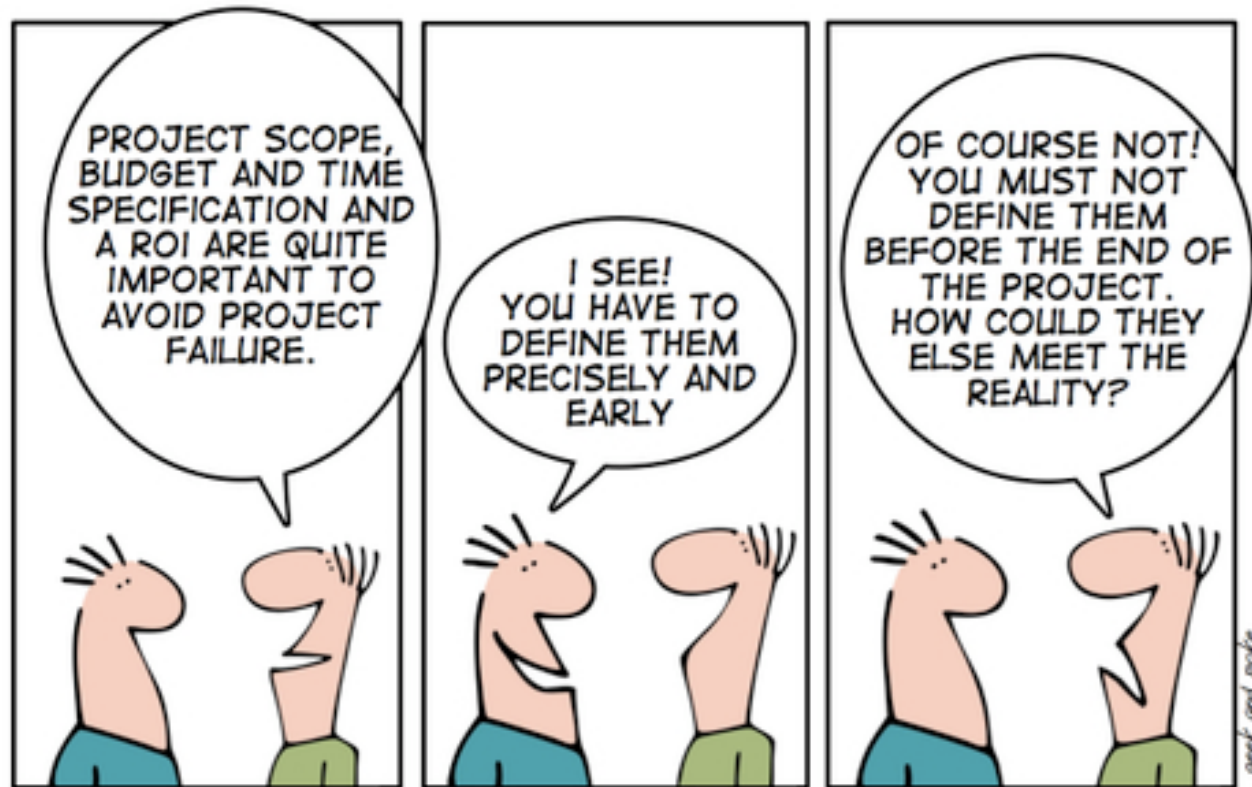
- There are study materials, but no homework assignments  
*(Find out for yourself what you have to read when!)*

# Important!

- Please sort this out today (8+9):

For the pearl assignment you are requested to interview a project stakeholder. How can you arrange that for this week (preferably Wed)?

## 2. Project management



# Characteristics of a project

- A project is **limited in time**
- A (good) project has a **clearly defined goal**

## *Project phases:*

(Preliminary work)

1. Starting phase
2. Preparatory phase
3. Execution phase
4. Closing phase

# Project work this week:

gathering requirements, writing a requirements spec

## Pearl Assignment:

- Starting phase
- Preparatory phase
- Execution phase
- **Closing phase:**

## Module Project:

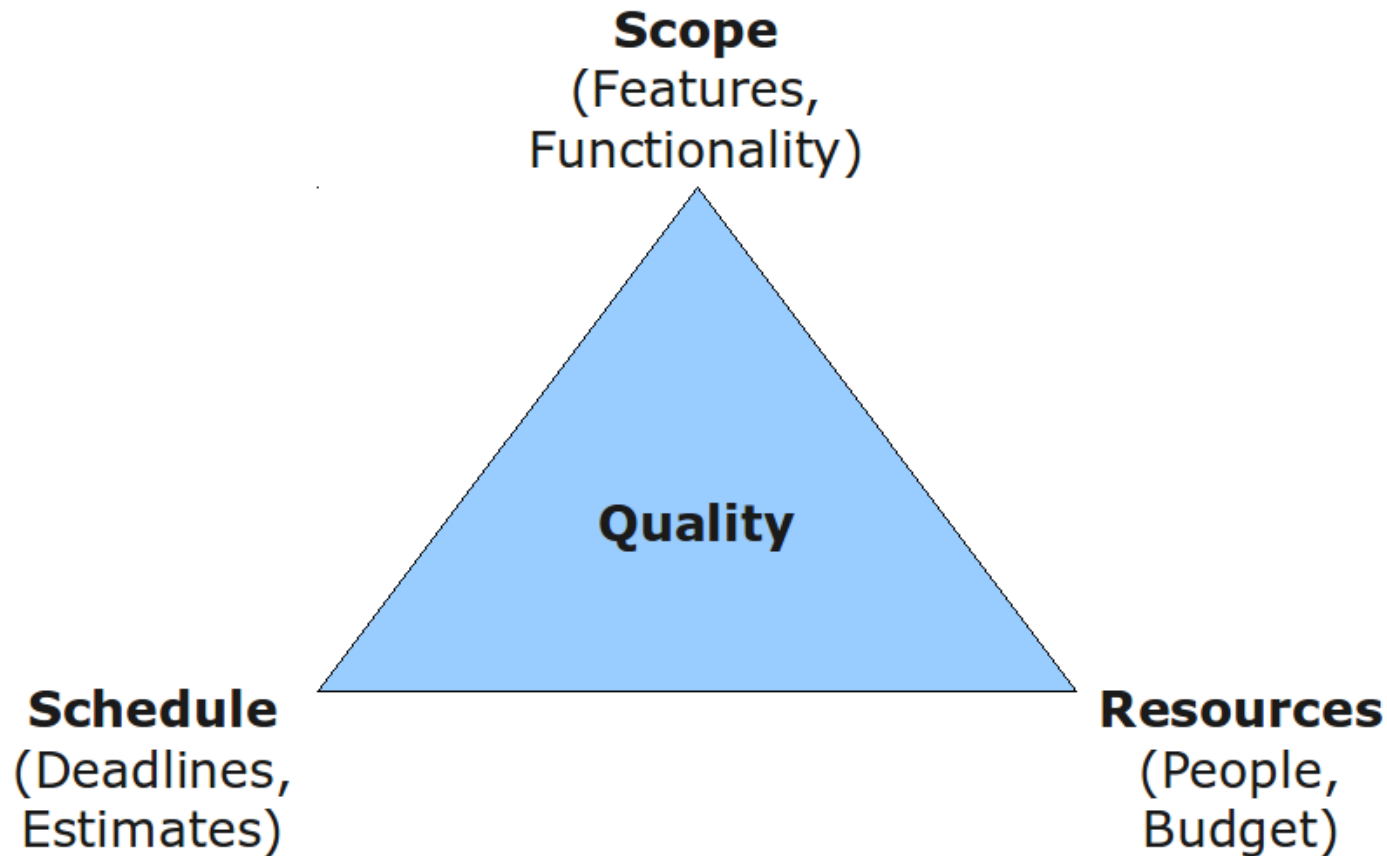
- Starting phase
  - Form project team
- **Preparatory phase:**

– A requirements specification for the Module Project is available (and has been submitted)

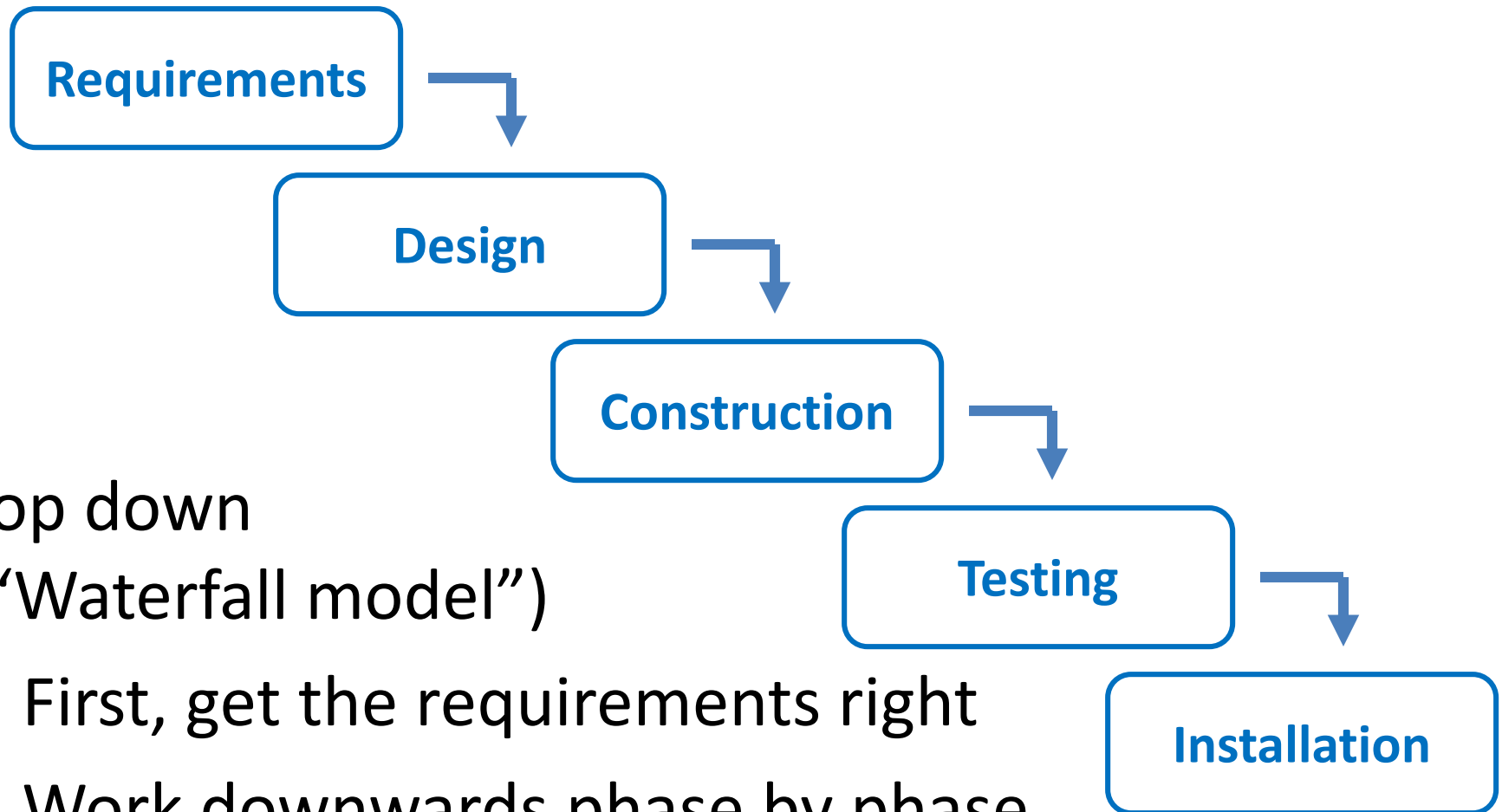
*The project 'Pearl Assignment' is a part (sub-project) of the project 'Module Project'*

# The “Iron Triangle”

- You can only control two of the three dimensions



# Software Development methods (1)



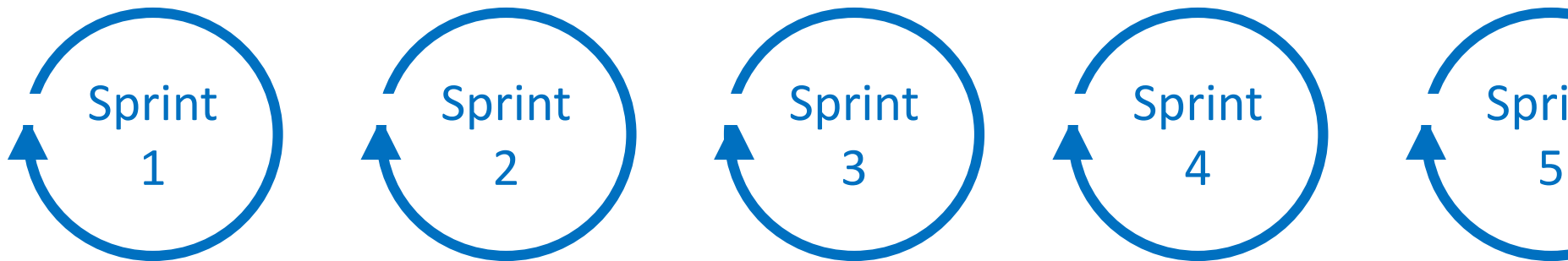
Top down  
("Waterfall model")

- First, get the requirements right
- Work downwards phase by phase

# Software Development methods (2)

## Agile (Scrum)

- Regular contacts with client
- Have a working system ready every 2–4 weeks



# Software Development methods

- For which types of project is a top-down approach preferable?
- For which types of project is an agile approach preferable?

# 3. Pitfalls and challenges for (IT) projects

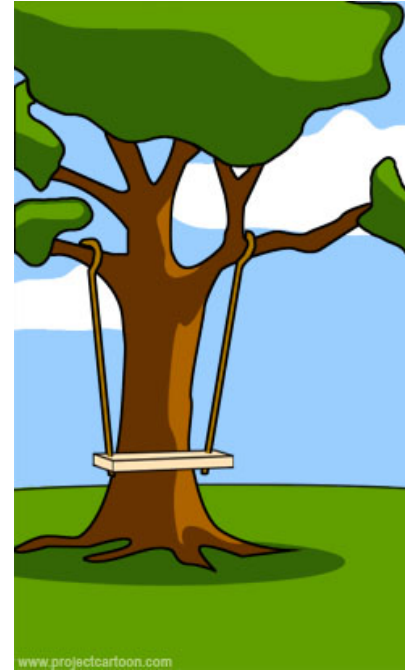




**What the customer wanted**



**What the customer asked for**



**What the analyst understood**



**What was delivered eventually**

# Success rate of IT-projects

<b><u>IT-projects (mostly USA)</u></b>	<b><u>2012</u></b>	<b><u>1994</u></b>
Successful	39 %	16 %
schedule/cost overrun	43 %	53 %
functional deficiencies		
complete failure	19 %	31 %

[Standish Group]

# Common causes for IT project failure

1. Unclear requirements  
(includes unclear goals, changing requirements)
2. Underestimation of project complexity  
(as well as underestimation of time and resources)
3. Poor project management  
(includes communication breakdowns at various levels)

# Example of an IT disaster in NL

- Project SPEER (Ministry of Defence):  
a single ERP system for all army units
  - Started 2005, stopped 2015
  - 450 M€ spent on IT development
  - 450 M€ spent on additional costs
  - Ambitions cannot be realized,  
but some fragments are usable (say politicians)  
*In cleartext: almost total failure*

# Parliamentary Enquiry in NL (2014)

## “Government bodies

- do not prepare projects well,
- have insufficient knowledge about IT and about IT tender processes,
- have no clear goals,
- keep changing the requirements, and
- do not realistically estimate time and costs.”

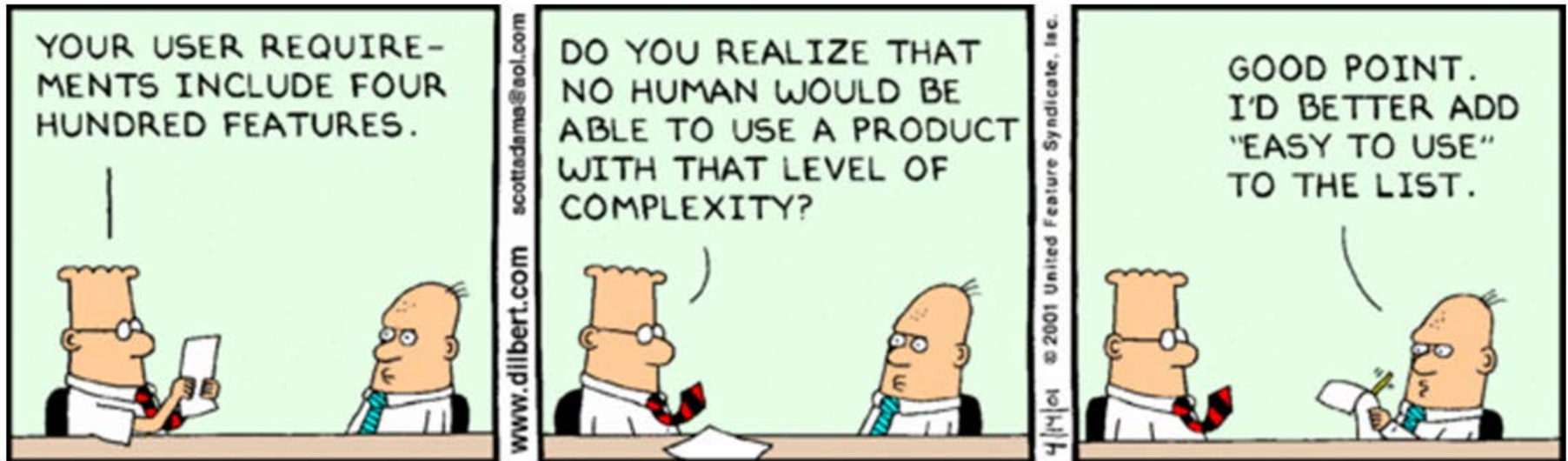
# Parliamentary enquiry in NL (2014)

	<u>All govt. Projects</u>	<u>Projects &gt; 10 M€</u>
Successful	30 %	7 %
Schedule/cost overrun functional deficiencies	46 %	57 %
Complete failure	24 %	36 %

# The not so bad news

- In 2014 government bodies in the Netherlands wasted 4.000 – 5.000 M€ on ICT, but in 2007 this was estimated at 6.000 M€ (from a smaller total budget)
- For other European countries, figures are comparable

# 4. Requirements specification



# Requirements analysis

is the part of system development  
in which people attempt to  
discover what is desired

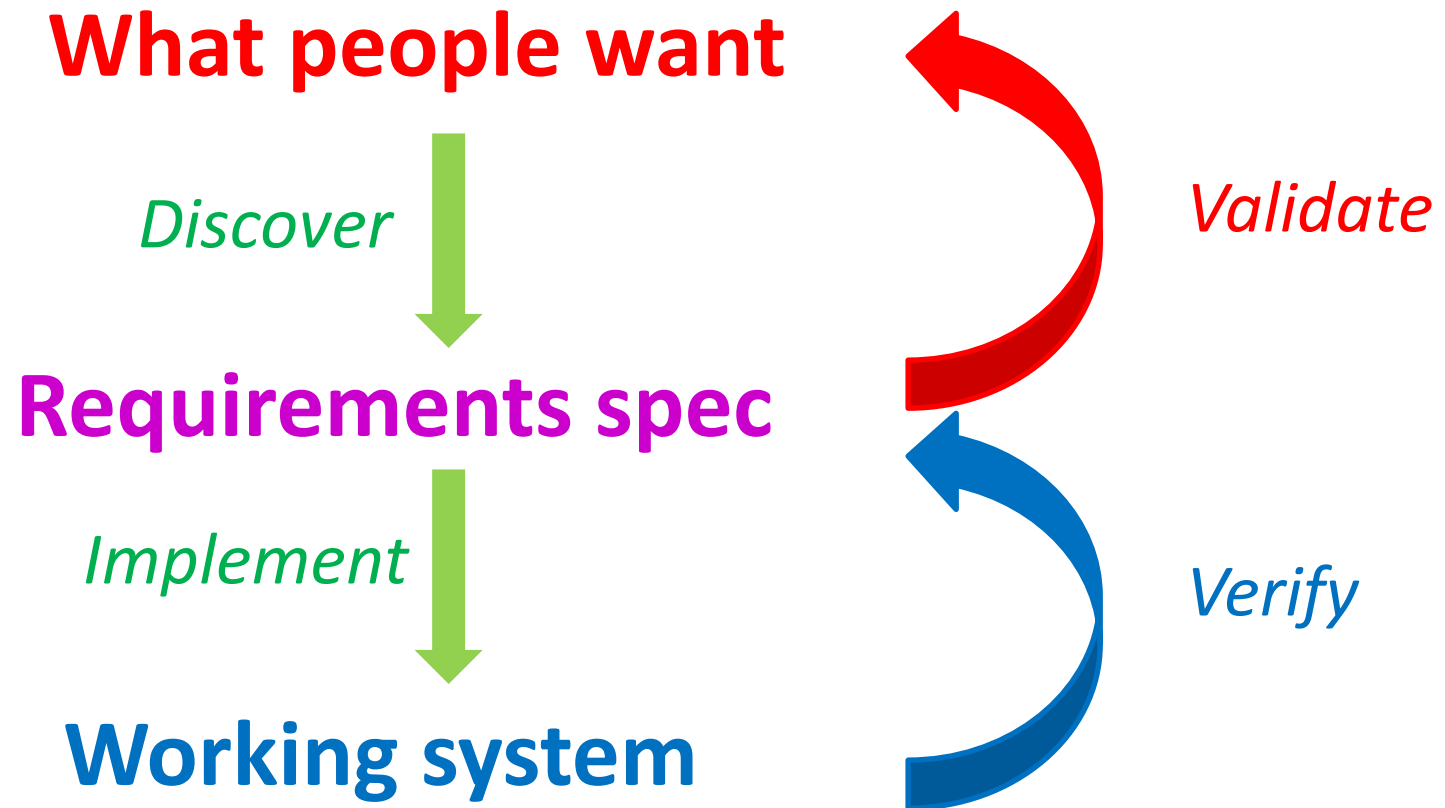
[Gause & Weinberg, 1989]

The process is more important than the product –  
(but documenting requirements is part of the process)

# Why is it difficult to make a good requirements specification?

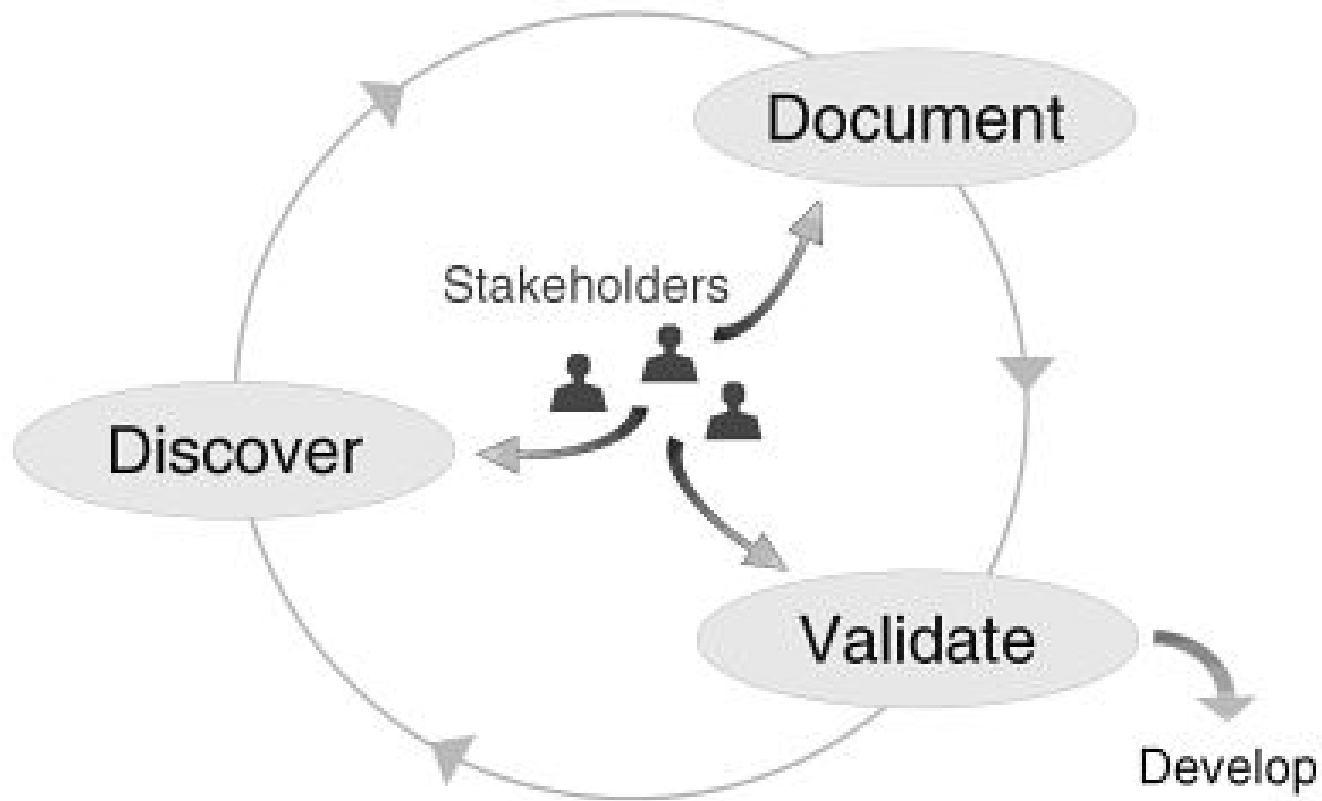
- It is not exactly clear what the system should do / what the stakeholders want  
and/or
- The objective is clear, but it hard to determine how to meet the technical challenges

# The roles of a requirements spec.



# A circular process

## The inquiry cycle [Alexander & Beus, 2009]



# A linear process

1. **Exploring** the situation

2. **Scoping** the project

3. **Discovering** requirements

4. **Documenting** requirements

5. **Validating** requirements

**Preliminary work**

**Preparation phase**

**Inquiry  
cycle**

**Execution  
phase**

**Delivering a requirements specification**

**Closing phase**

# A linear process

1. **Exploring** the situation
2. **Scoping** the project see Lecture 2
3. **Discovering** requirements
4. **Documenting** requirements see Lecture 2
5. **Validating** requirements

**Delivering a requirements specification** **Pearl assignment**

# Exploring the situation

- People tend to phrase problems in terms of solutions
- Try to grasp a situation, understand what the problems are, *before* you start looking for solutions

# Discovering requirements

- There are different kinds of requirements
- There are different methods to discover requirements  
(see *A Primer on Requirements Analysis*)

*Methods to discover requirements this week:*

- Brainstorm
- Interview [will be taught in later modules]

# Validating requirements

- In an interview, the steps *validating* and *discovering* may get mixed up
  - The interviewee can confirm – or reject – the requirements you came up with
  - The interviewee can add requirements that you did not think of
  - The interviewee can indicate which requirements are most important, which are less important

# Do you know all your group mates?

- *If yes*: please leave the lecture hall and start the project
- *If no*: please stay and meet them
- *If you still need a place in a group*: please stay and contact me immediately

# Success!