Introduction,
the Waterfall Model,
and Unified Process

Development of Large Systems

Fall 2017

Agenda

- Who are we
- Paradigms and more
- Waterfall disciplines
- Unified Process I
- Use cases

Anders Kalhauge

- aka@cphbusiness.dk
- 21 72 44 11
- 20 years experience as IT consultant in the private sector
- 17 years teaching computer science for students and private companies
- Master in Computer Science from DTU

Rolf-Helge Pfeiffer

Jens Egholm Pedersen

Paragigms

- Predictive
  - Waterfall
- Iterative (and incremental)
  - Unified Process
- Adoptive
  - Scrum and XP

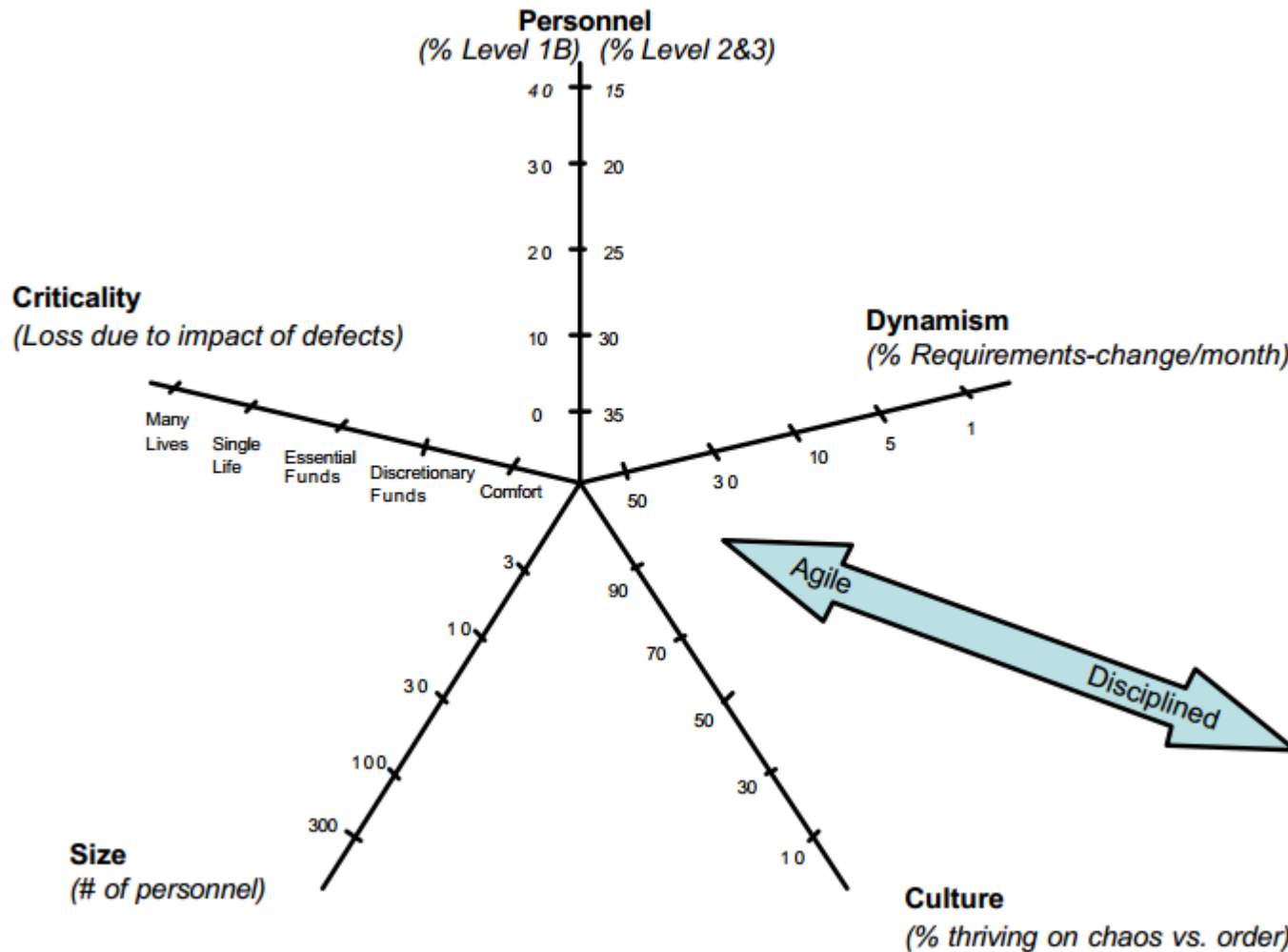**Disciplined**

Agile

Decision Theory

- Economic man
  - Knows everything
  - Always makes the perfect decisions
- Administrative man
  - Knows a lot
  - Makes qualified decisions
- Muddling through
  - Knows little
  - Can determine whether decisions go in the right direction.
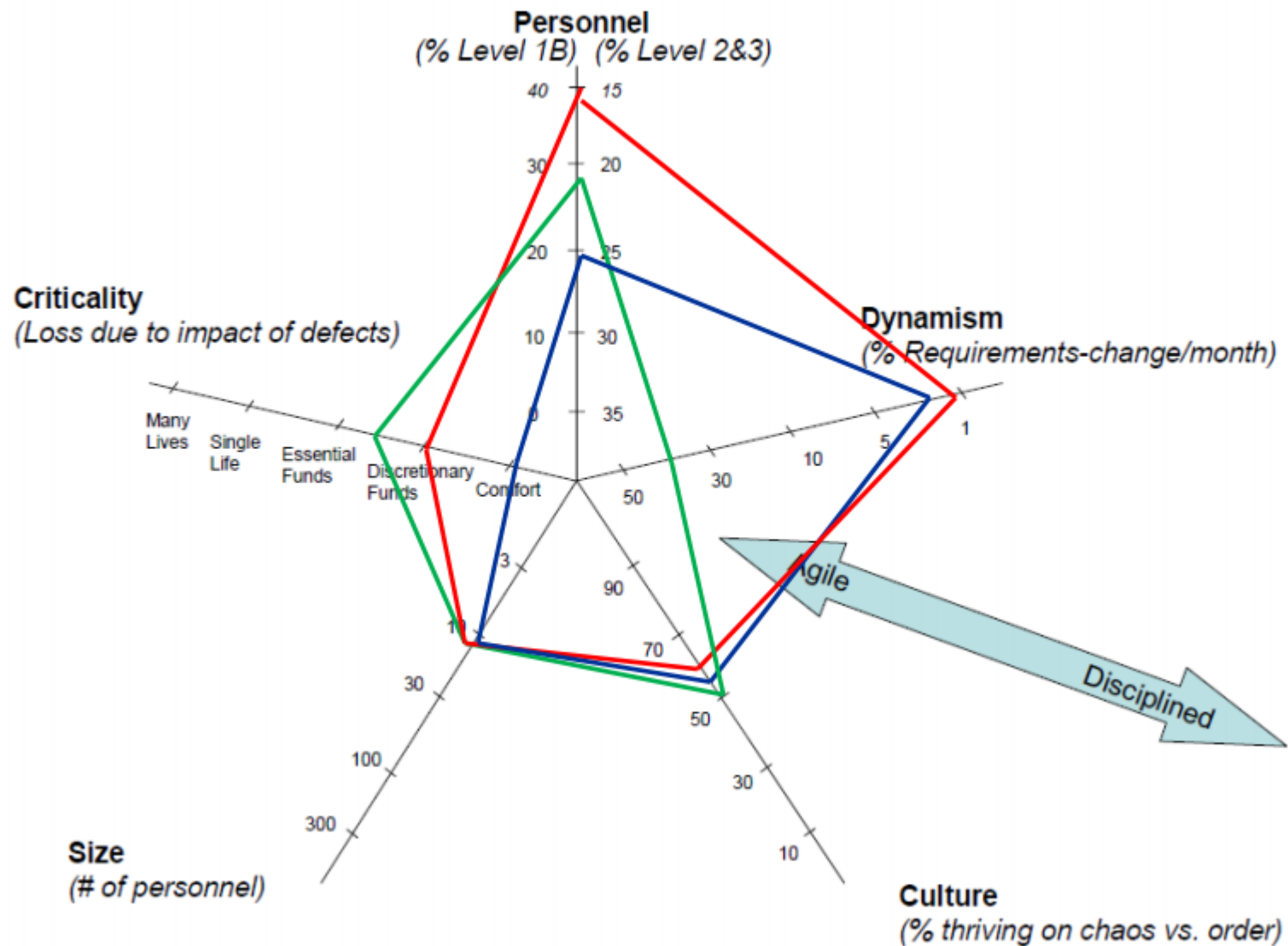
Uncertainty

- Routine
  - Known problem and known solution
- Problem solving
  - Known problem but Unknown solution
- Problem definition
  - Unknown problem and unknown solution
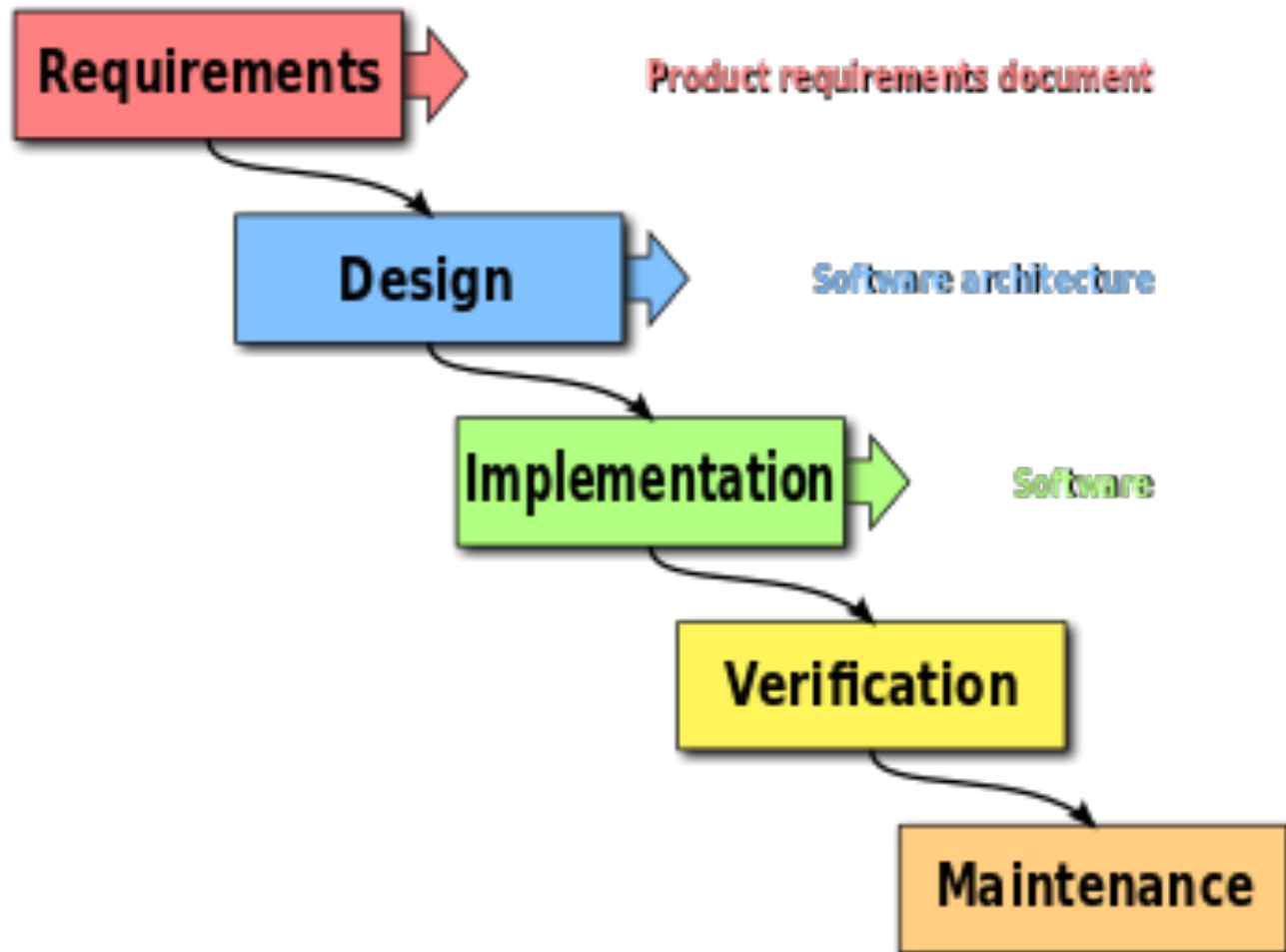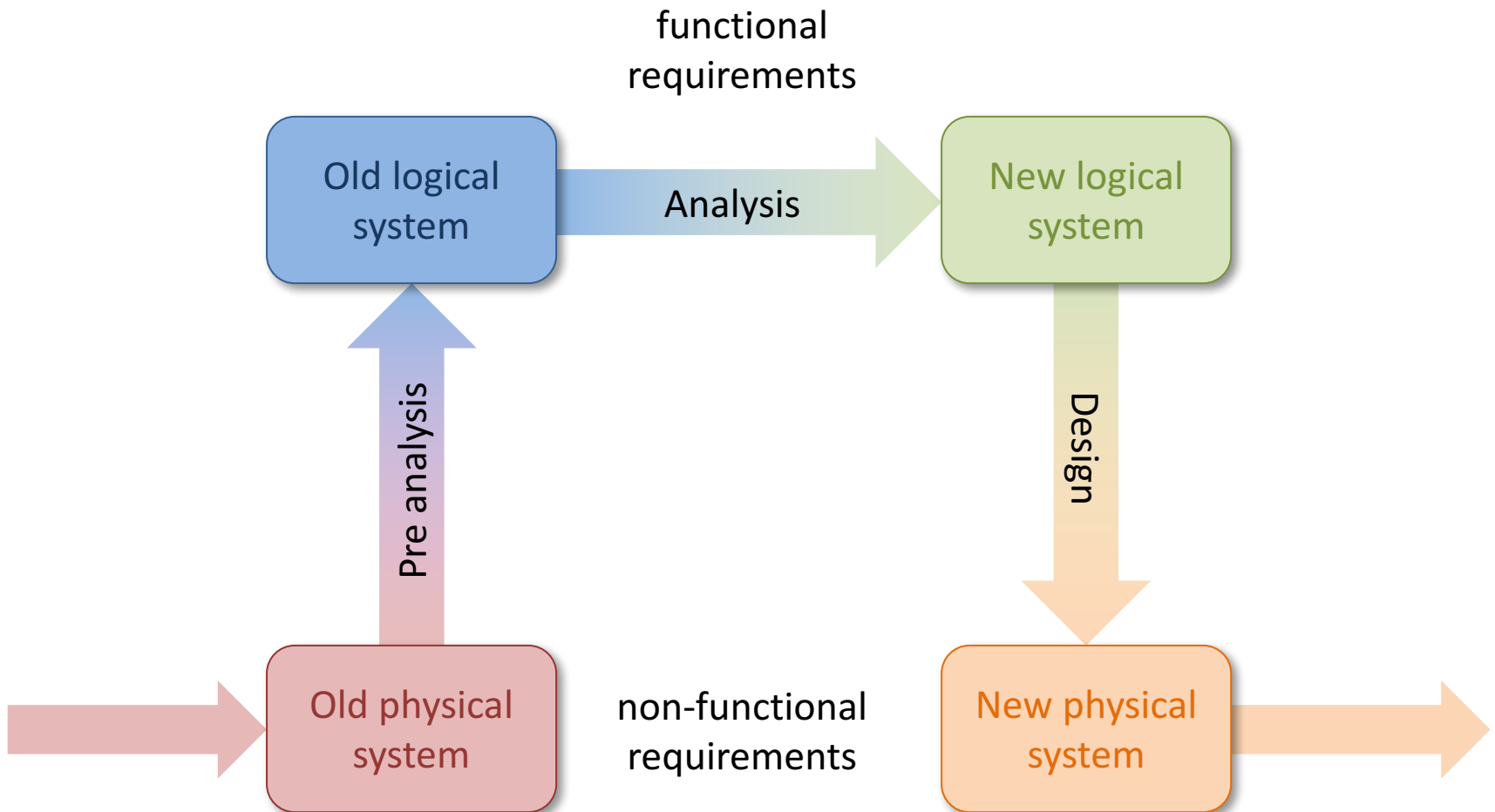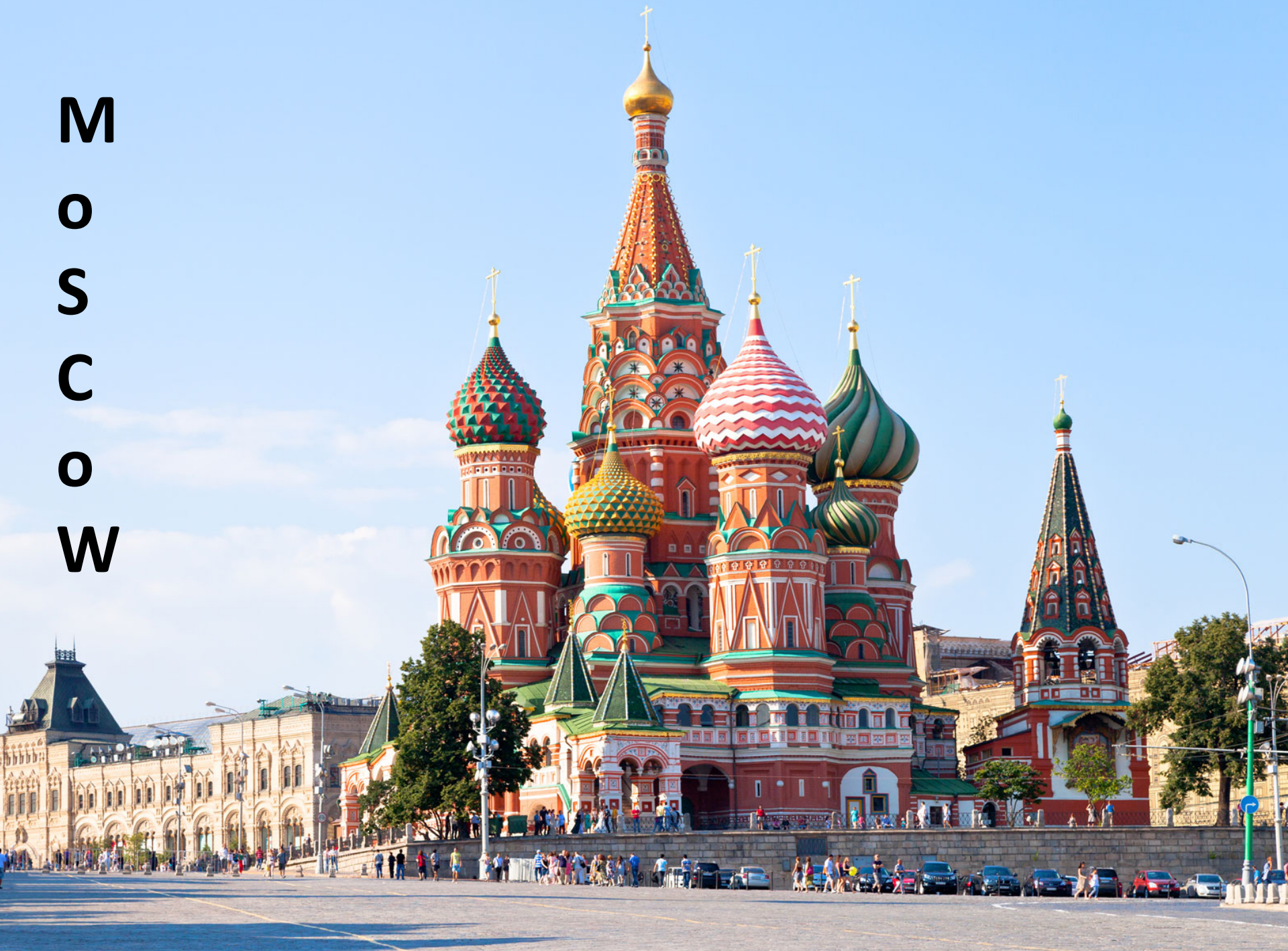
Boehm's spider web

Example

Personnel
(% Level 1B) (% Level 2&3)

Criticality
(Loss due to impact of defects)

Dynamism
(% Requirements-change/month)

Size
(# of personnel)

Culture
(% thriving on chaos vs. order)

Agile

Disciplined

Many Lives   Single Life   Essential Funds   Discretionary Funds   Comfort

Waterfall Model

Requirements

Moscow

Must have
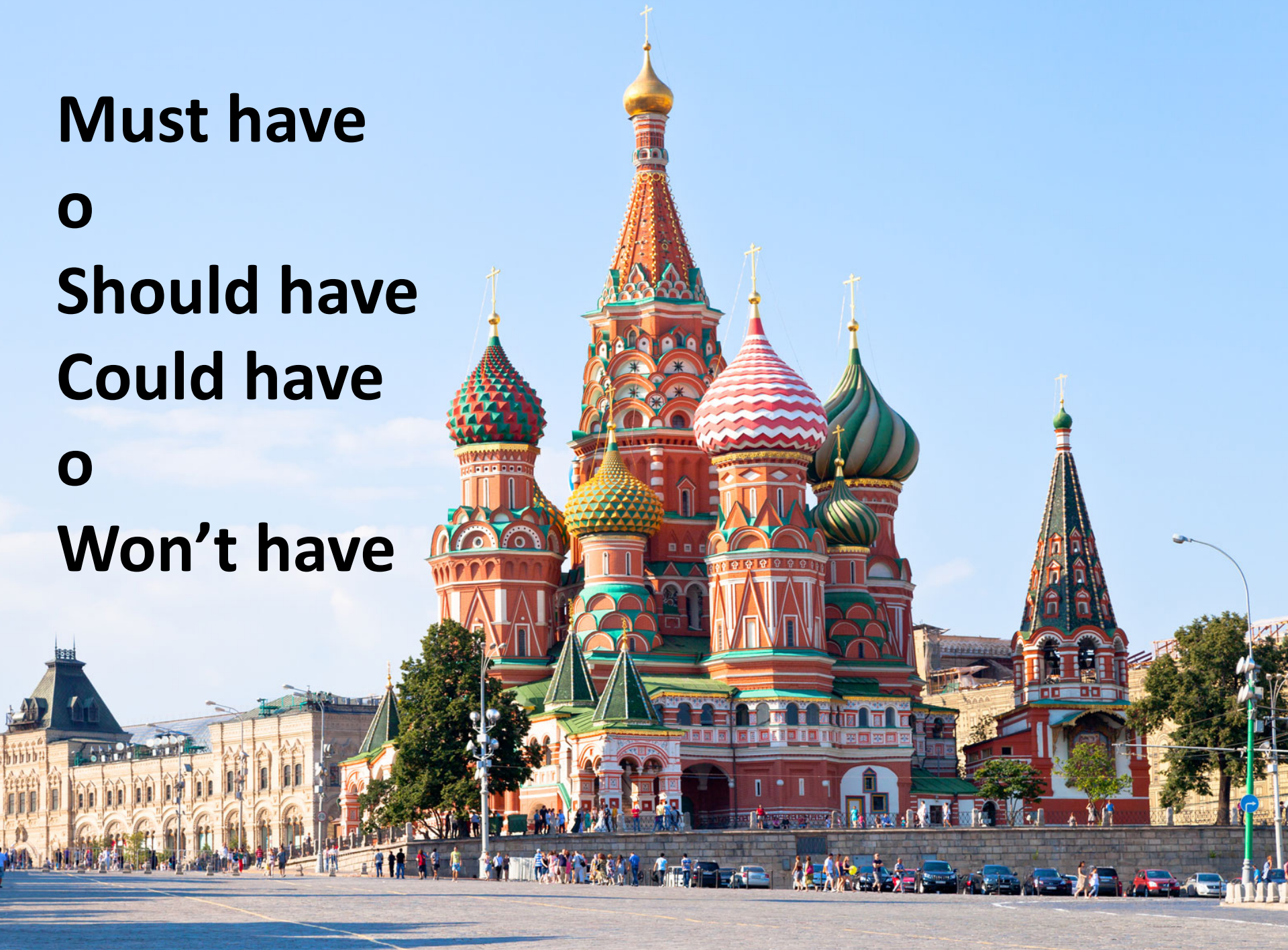
o

Should have

Could have

o

Won't have

Requirements - risks

- Requirements are about **what** is required

- Functional requirements
- Nonfunctional requirements
- The right system
- The right requirements
- Incomplete requirements
- Wrong requirements
- Missing requirements

- Business Modelling and LEAN Value Stream Analysis may be used to develop the right system
- When do we estimate and submit the tender

Analysis & Design

- The analysis helps understanding the required system and the production of necessary documentation

- Design is about **how** the requirements may be fulfilled

- A requirement/design matrix can be used to manage that all requirements are met and that all design is based on a requirement
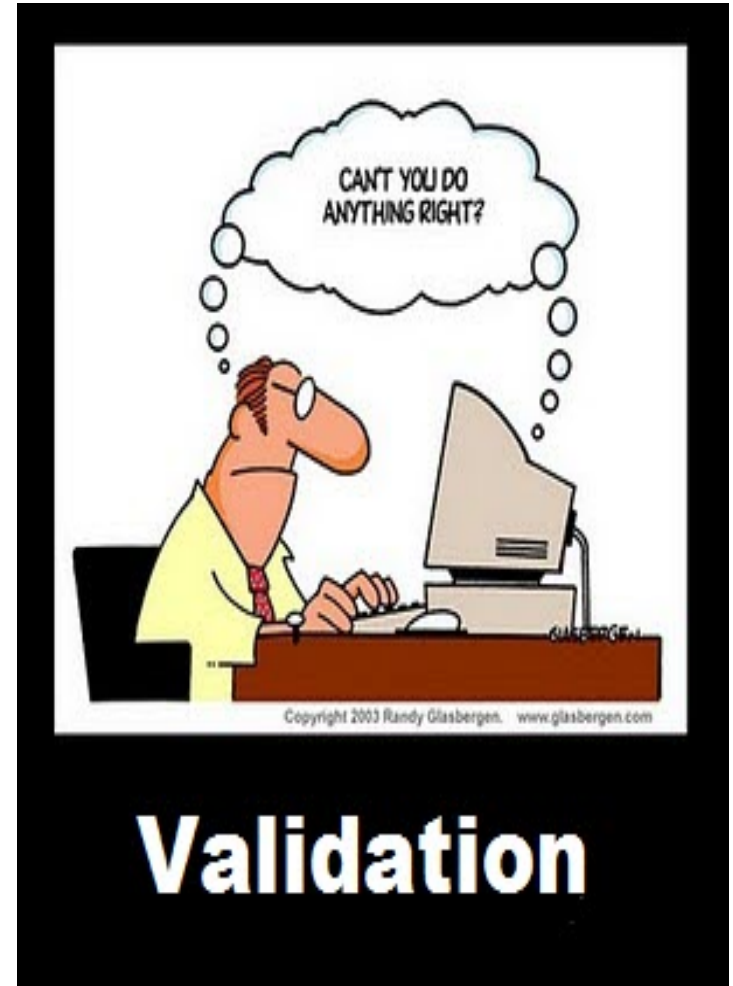
Implementation

- Is the architecture in place

- Is the code standard in place

- Is the proper use of design patterns in place

- Are the test cases in place

- Then code ☺

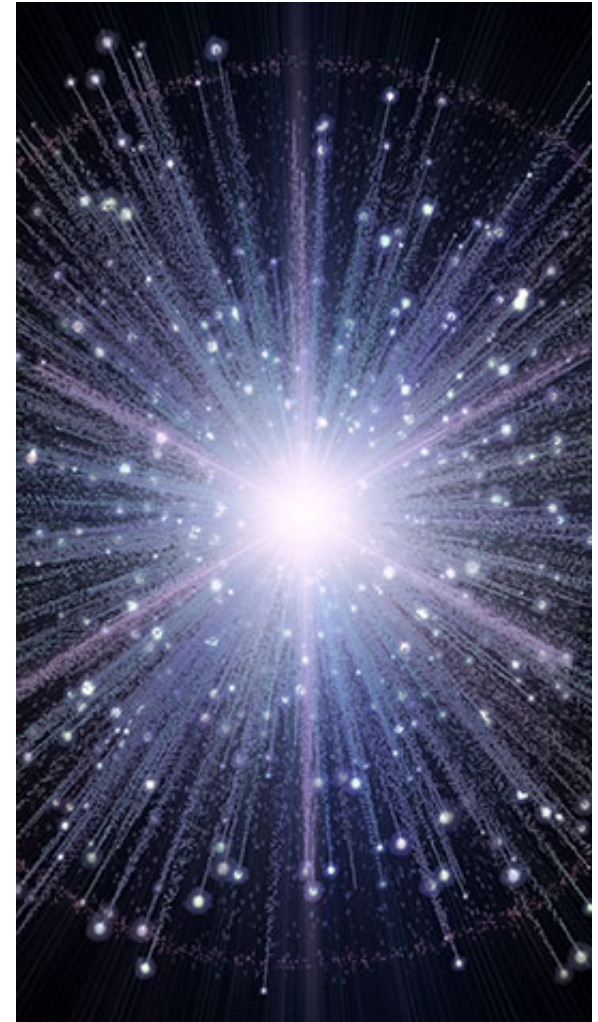- Are development and test environments in place

Test and Verification

- Do the test cases cover all scenarios and requirements

- Can all test cases be run without errors

- Are all the system requirements fulfilled

- Can the system be used

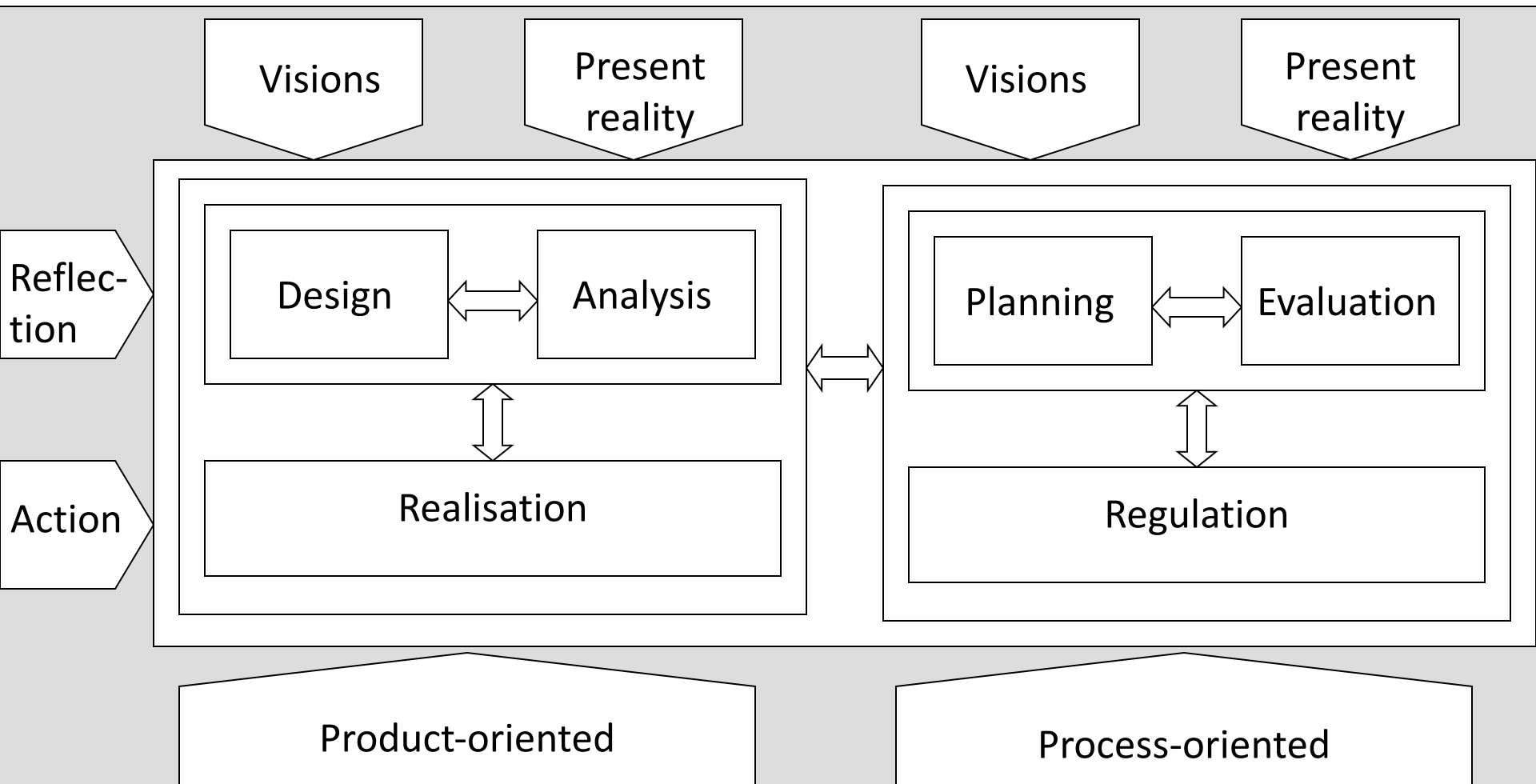- Did we cater for all activities (and not just 40%)

Waterfall disadvantages

- One Big Bang delivery
- Perhaps after very long time
- New or changed requirements will lead to considerable rework
- New technologies are not considered
- User's needs or requirements may have changed when the product is delivered

Where does Waterfall fit?

Agile Manifesto

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

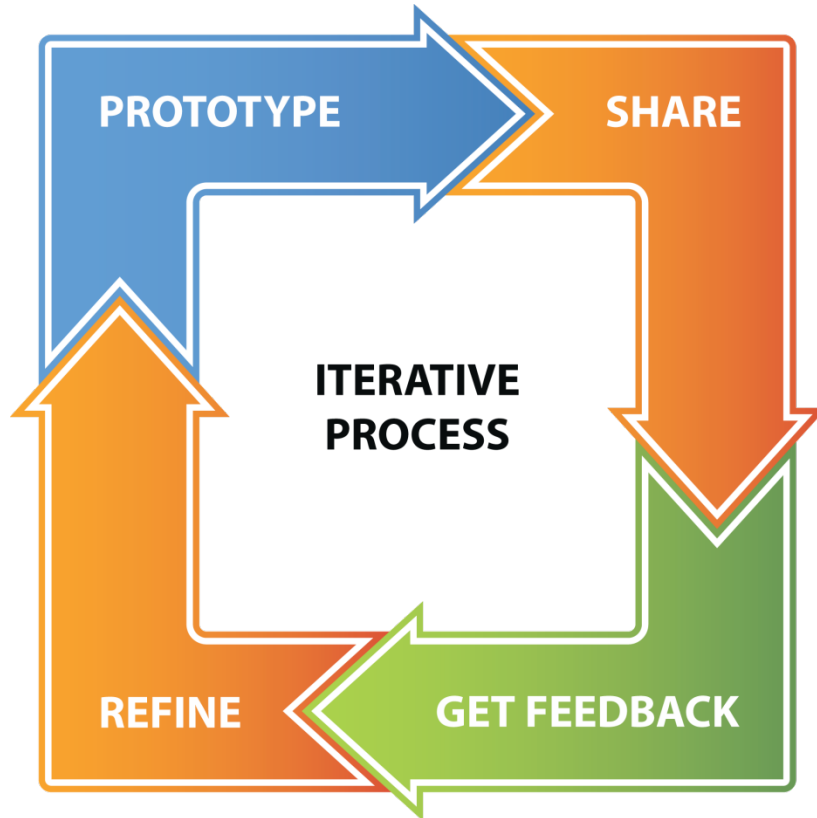**Individuals and interactions** over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Iterative and Incremental

Plan > Do > Check > Act