THE SMARTER WAY

Be Smart!
or
What they don’t teach you about software at school

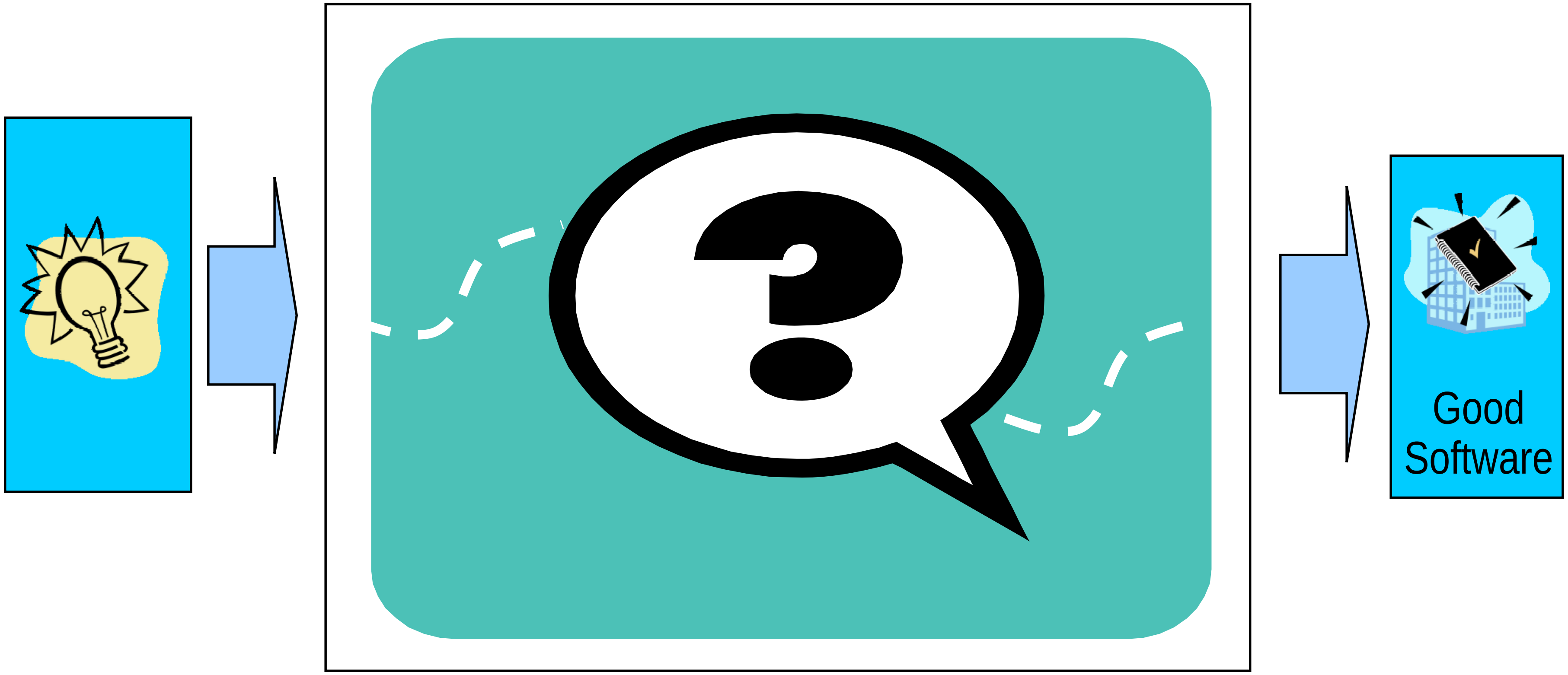
Ivar Jacobson

with

Ian Spence, Pan Wei Ng and Kurt Bittner

www.ivarjacobson.com

Our goal



Better, Faster, Cheaper and Happier

What it takes

What they don’t teach you about software at school ☺

IVAR JACOBSON INTERNATIONAL

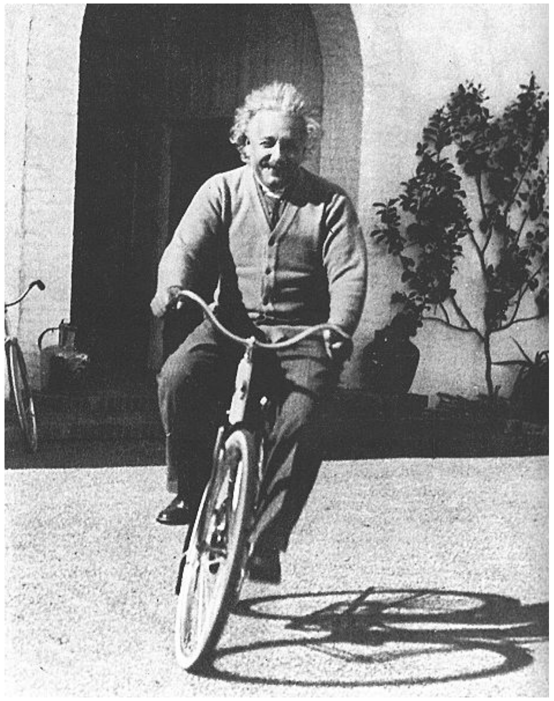
Agenda

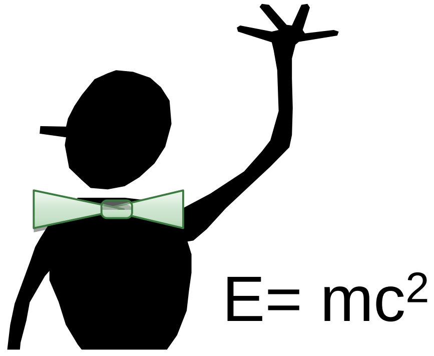
1. What does Smart mean?
2. Smart Cases – Recognize it when you see it
3. How do you become Smart
4. What does Smart really mean?

What does *Smart* mean?

Things should be done
as simple as possible – but no simpler

- *Albert Einstein*



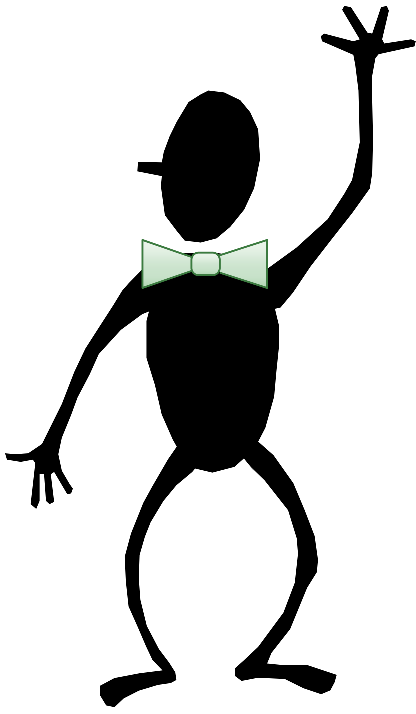


-

IVAR JACOBSON INTERNATIONAL

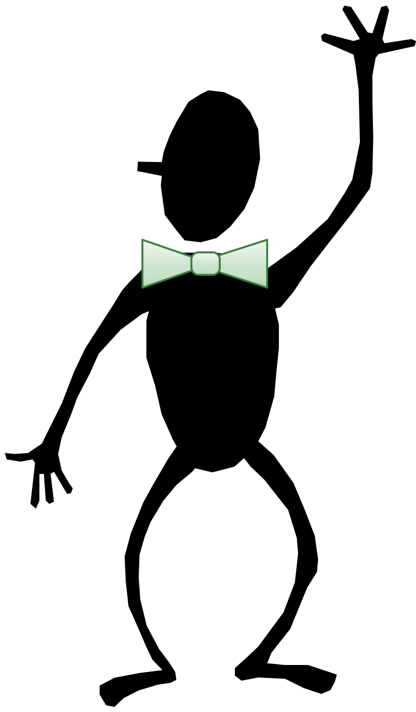
Smart and Intelligent?

- Being Smart is not the same thing as being intelligent
  - You can be intelligent without being smart, and
  - You can be very smart without being very intelligent


Mr Smart

Smart and Agile?

- Being Smart is an *evolution* of Being Agile
  – Agile means being flexible and adaptable.
  – Agile provide simple/lightweight starting points
  – But being smart is knowing when to go beyond agile
    - Knowing when to follow the rules and when to break them
    - Knowing when to be consistent and when to change
    - Knowing when to grow and when to shrink


Mr Smart

Smart = Agile ++

IVAR JACOBSON INTERNATIONAL

Agenda

1. What does Smart mean?
2. Smart Cases – Recognize it when yo
    1. People
    2. Teams
    3. Projects
    4. Requirements
    5. Architecture
    6. Modeling
    7. Test
    8. Documentation
    9. Process
    10. Knowledge
    11. Outsourcing
    12. Tools
3. How do you become Smart
4. What does Smart really mean?

What they don't teach you about software at school

☺

IVAR JACOBSON INTERNATIONAL

www.ivarblog.com

Not smart with People

Some companies view process and tools as more important than people

This is unsmart!

A fool with a tool is still a fool but a **dangerous** fool

IVAR JACOBSON INTERNATIONAL

Smart with People

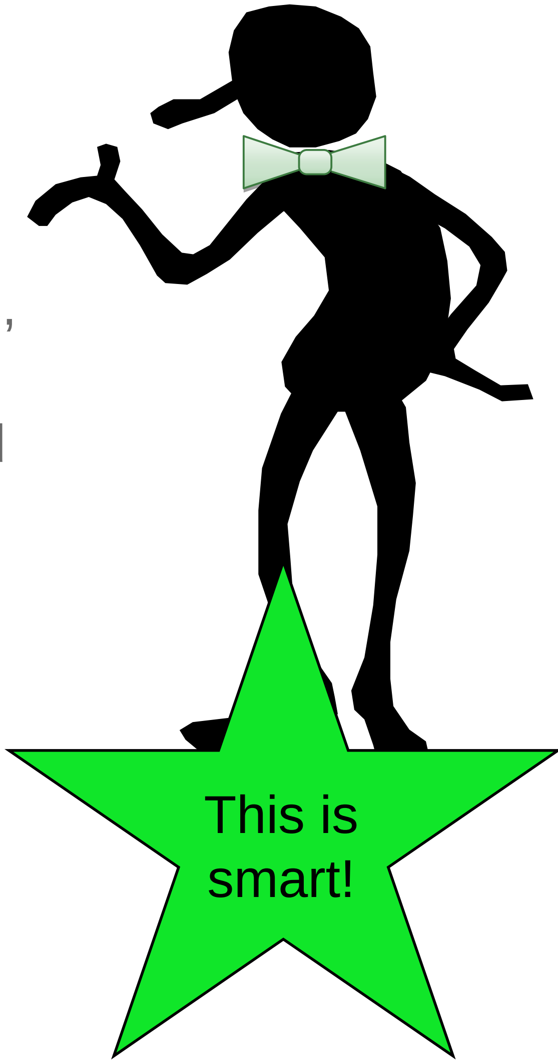
Case study: Ericsson AXE – the largest commercial success story ever in Sweden

–We had no tools and no defined process

–Despite this, we developed components, use cases, and a modeling language now part of UML

–This could only have been done with people – good people

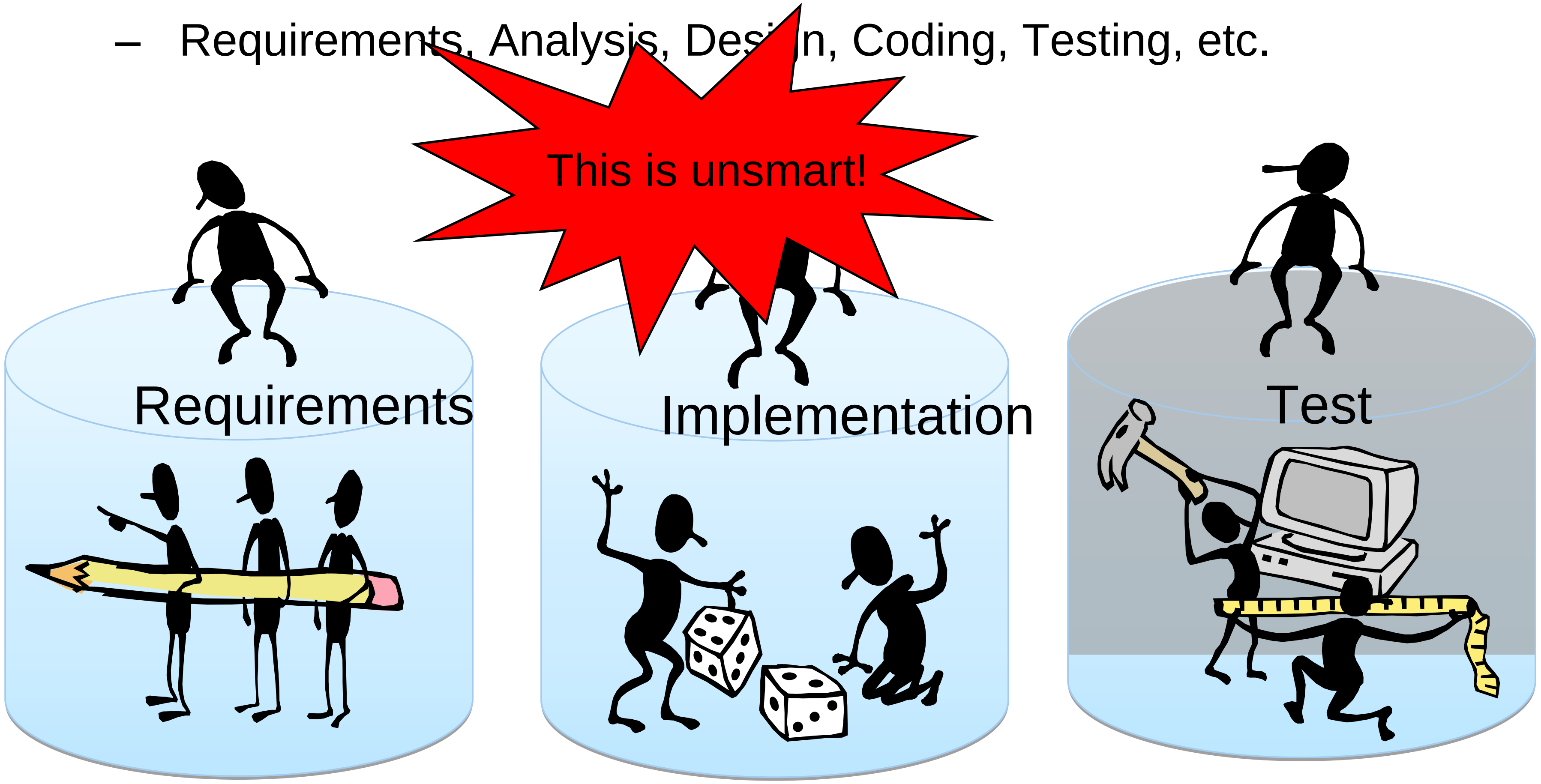
Software is developed by people, not by process and tools.



IVAR JACOBSON INTERNATIONAL

Not smart with Teams

- Many software projects involve 20+ people
- Often organized into stove-pipe groups:
  - Requirements, Analysis, Design, Coding, Testing, etc.

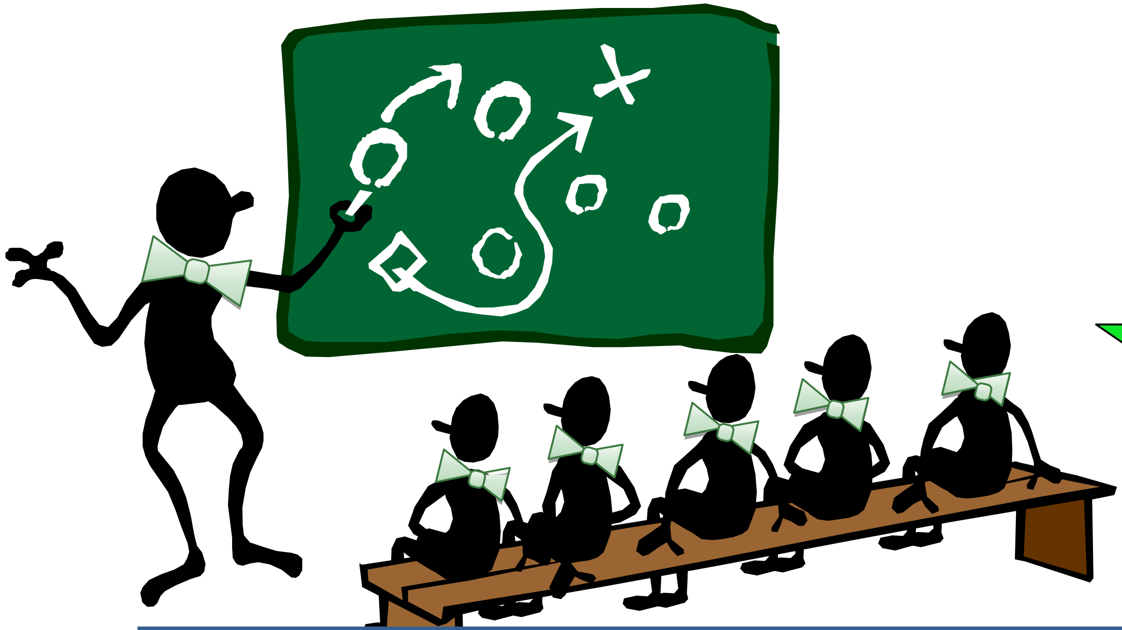


Smart with Teams

- Teams are cross-functional

  Including analysts, developers, testers etc…

- Ideal size of the team is less than 10 people



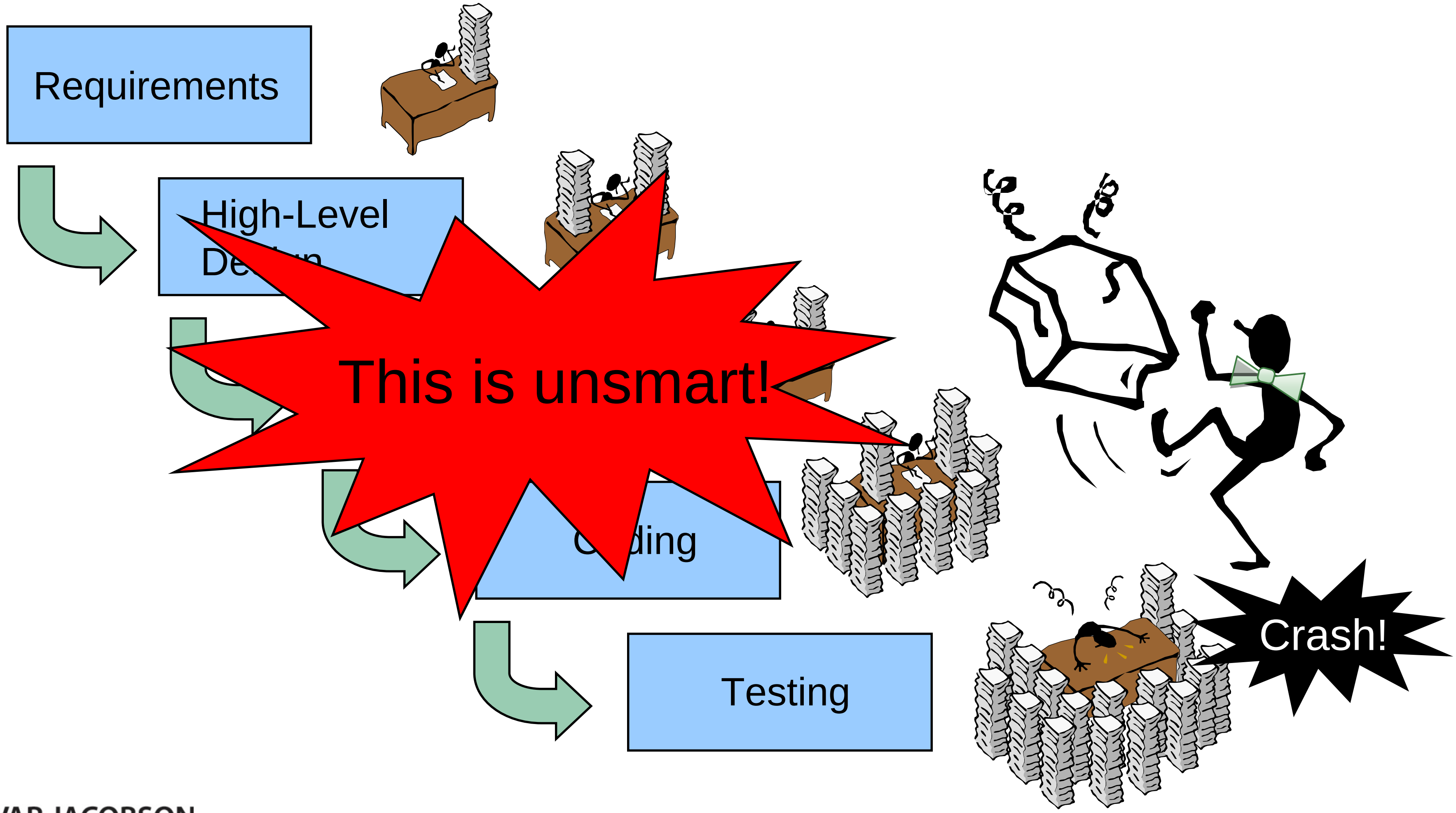
This is smart!

A software team is like a sport team with all needed competencies to win.

IVAR JACOBSON INTERNATIONAL

Not smart with Projects

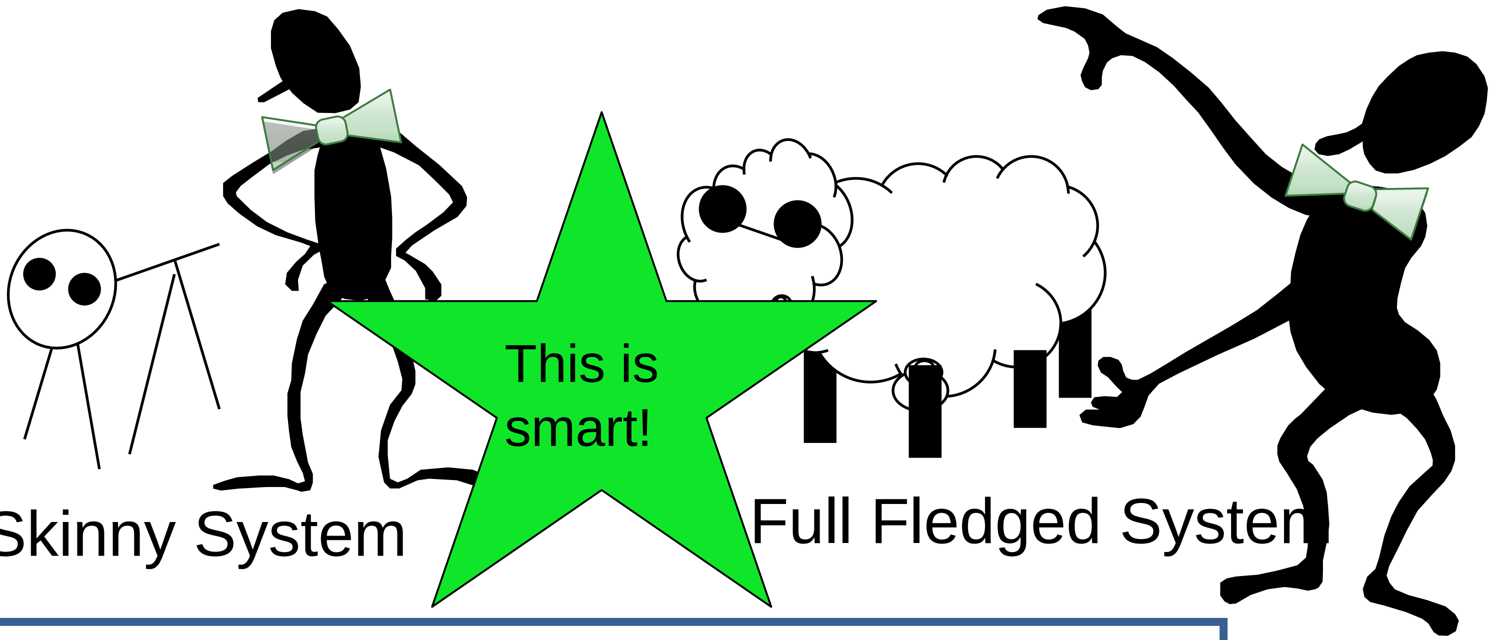
- Most companies still follow the waterfall approach



IVAR JACOBSON INTERNATIONAL

Smart with Projects

- Build a skinny system to demonstrate that you have eliminated all critical risks
- Add more capabilities on top of that skinny system

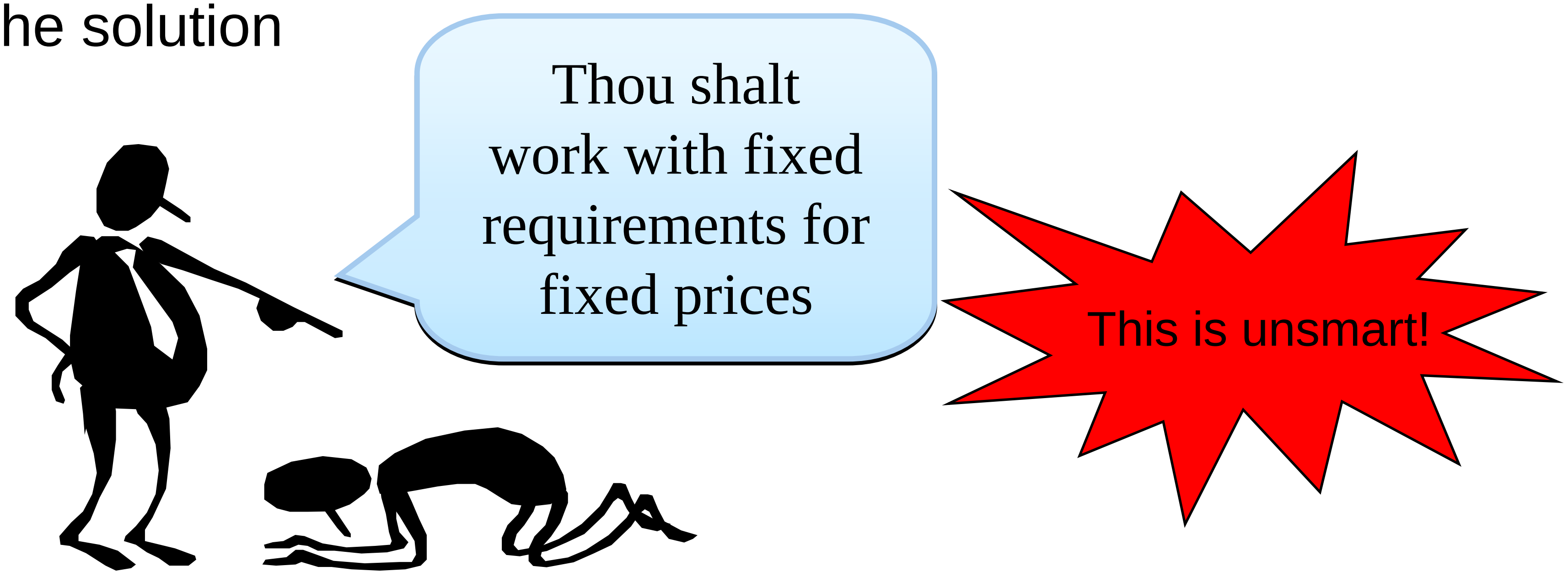


Think big, build in many steps

IVAR JACOBSON INTERNATIONAL

Not smart with Requirements

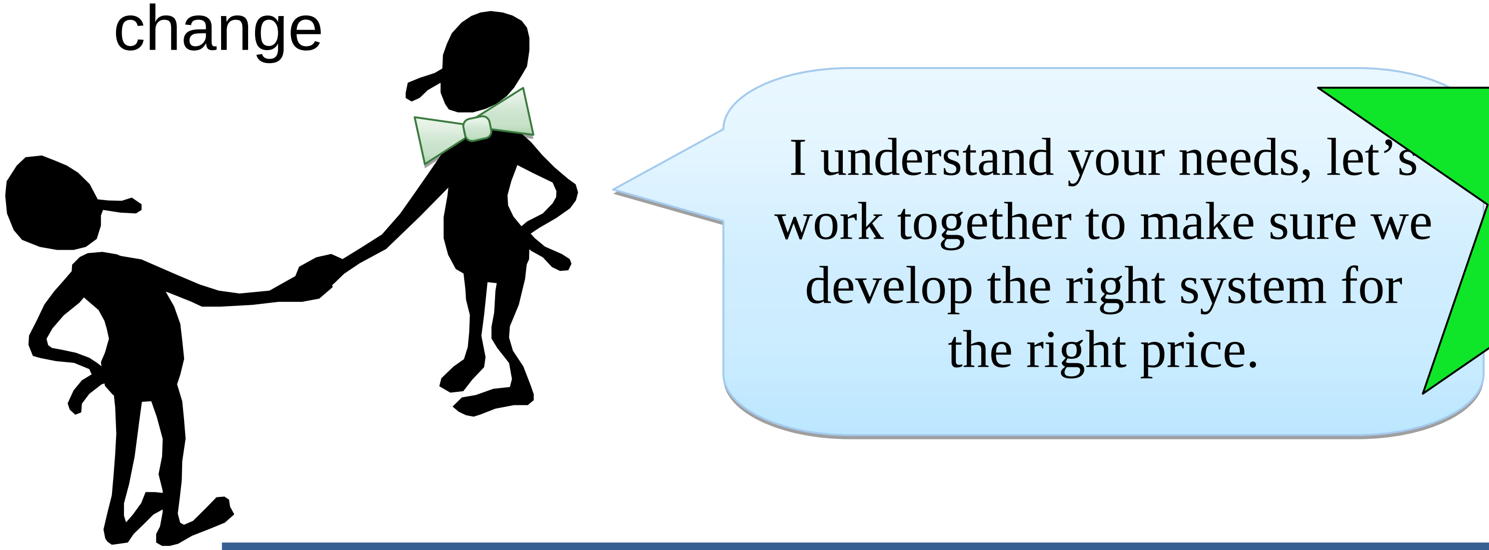
- Many managers (and customers) believe you can detail all the requirements upfront...
- ...and based on these can accurately predict the cost of the solution



A constant in software development is that requirements **always** change

Smart with Requirements

- Base early decisions on lightweight requirements and detail as and when it is needed
  - Use case outlines, feature lists or user stories
- Remember requirements are negotiable and priorities will change

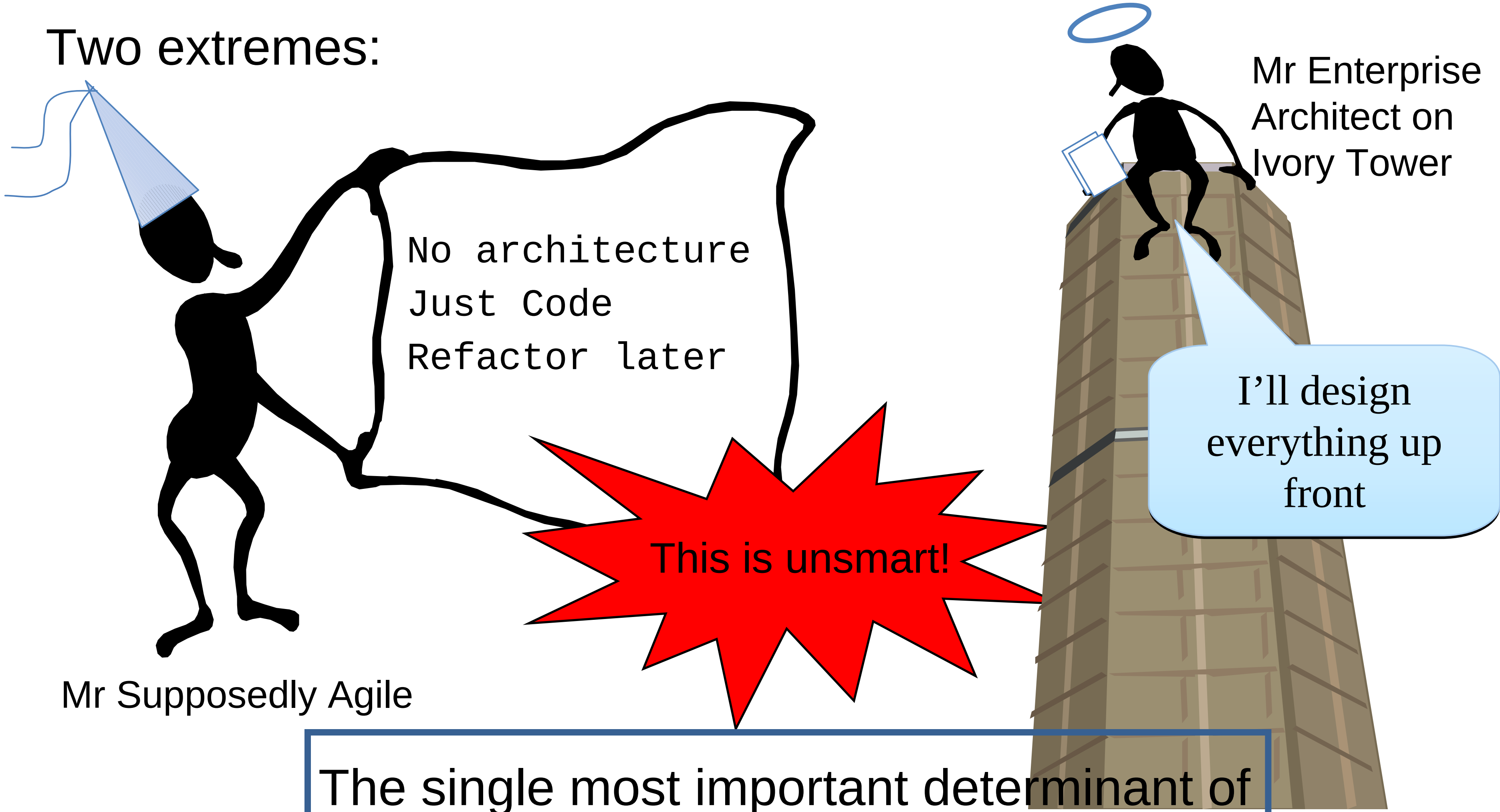

I understand your needs, let's work together to make sure we develop the right system for the right price.

This is smart!

Design your project for requirement changes

Not smart with Architecture

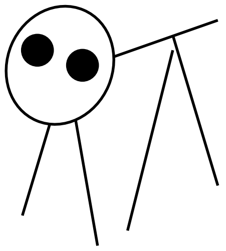
Two extremes:

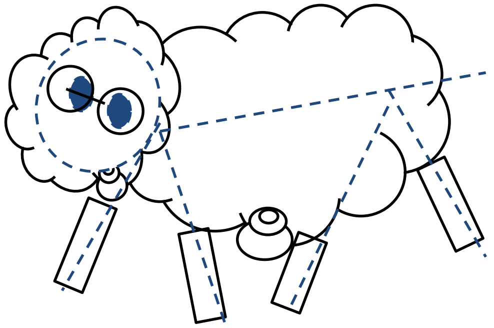


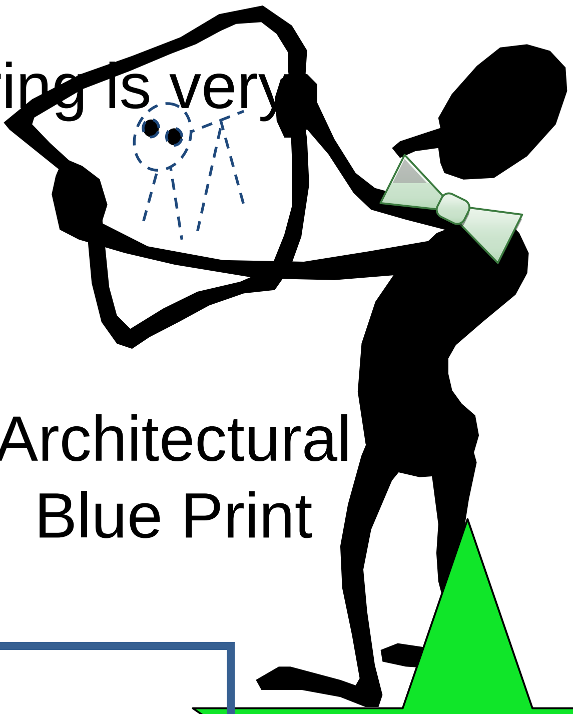
The single most important determinant of a software system's quality is the quality of its architecture

Smart with Architecture

- Focus on the skinny system
- But an architecture without executable code is a hallucination
- Refactor over releases, but large refactoring is very costly


Skinny System


Full Fledged System


Architectural Blue Print

Start to build a skinny system,
add muscles in later steps

This is smart!

IVAR JACOBSON INTERNATIONAL

Not smart with Service Oriented Architecture

Some people believe that

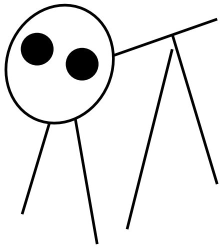
- SOA requires a new methodology
- SOA methods must major in paper-ware

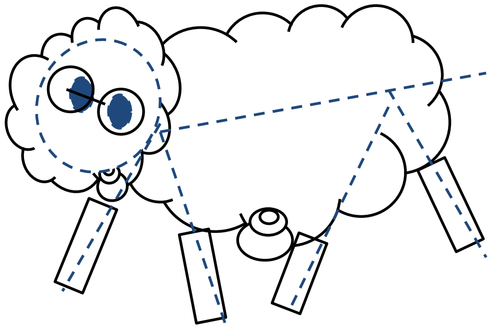
This is unsmart!

IVAR JACOBSON INTERNATIONAL

Smart with Service Oriented Architecture

- Use your foundation practices as a base
  - Use cases, components, or whatever you have
- Add system-of-systems practices
  - Create an architectural roadmap
  - Closing the Gap between business and IT,
  - Services
- Fill the roadmap project by project, starting with the skinny system.
- Do xSOA, executable SOA


Skinny System


Full Fledged System

This is smart!

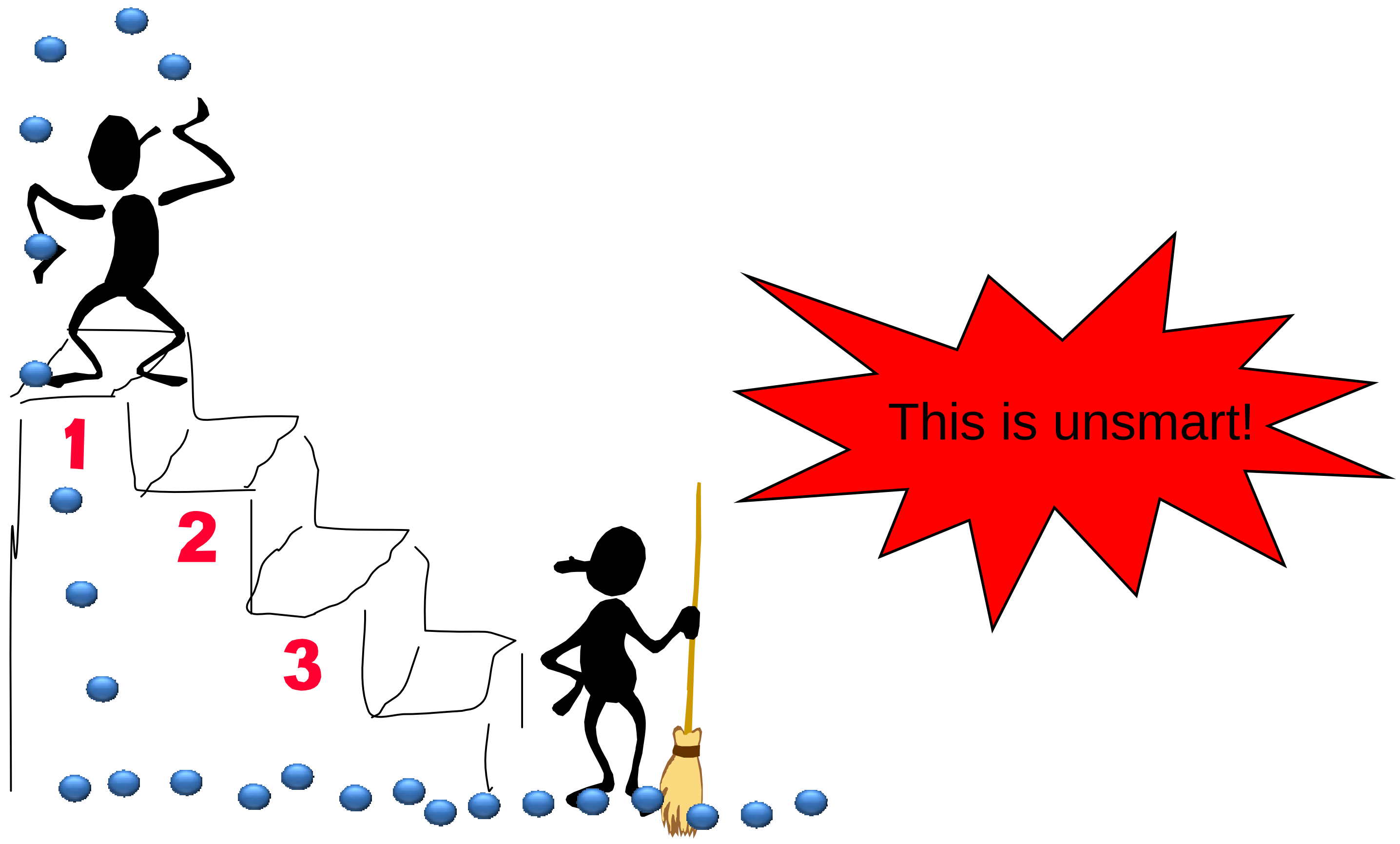
IVAR JACOBSON INTERNATIONAL

Not smart with Test

We have two classes of people: Developers and Testers

- *Developers are the creators…it is OK to create bugs as well*
- *Testers are the cleaners in the software world*

Testing is done as an after thought – too late and too expensive

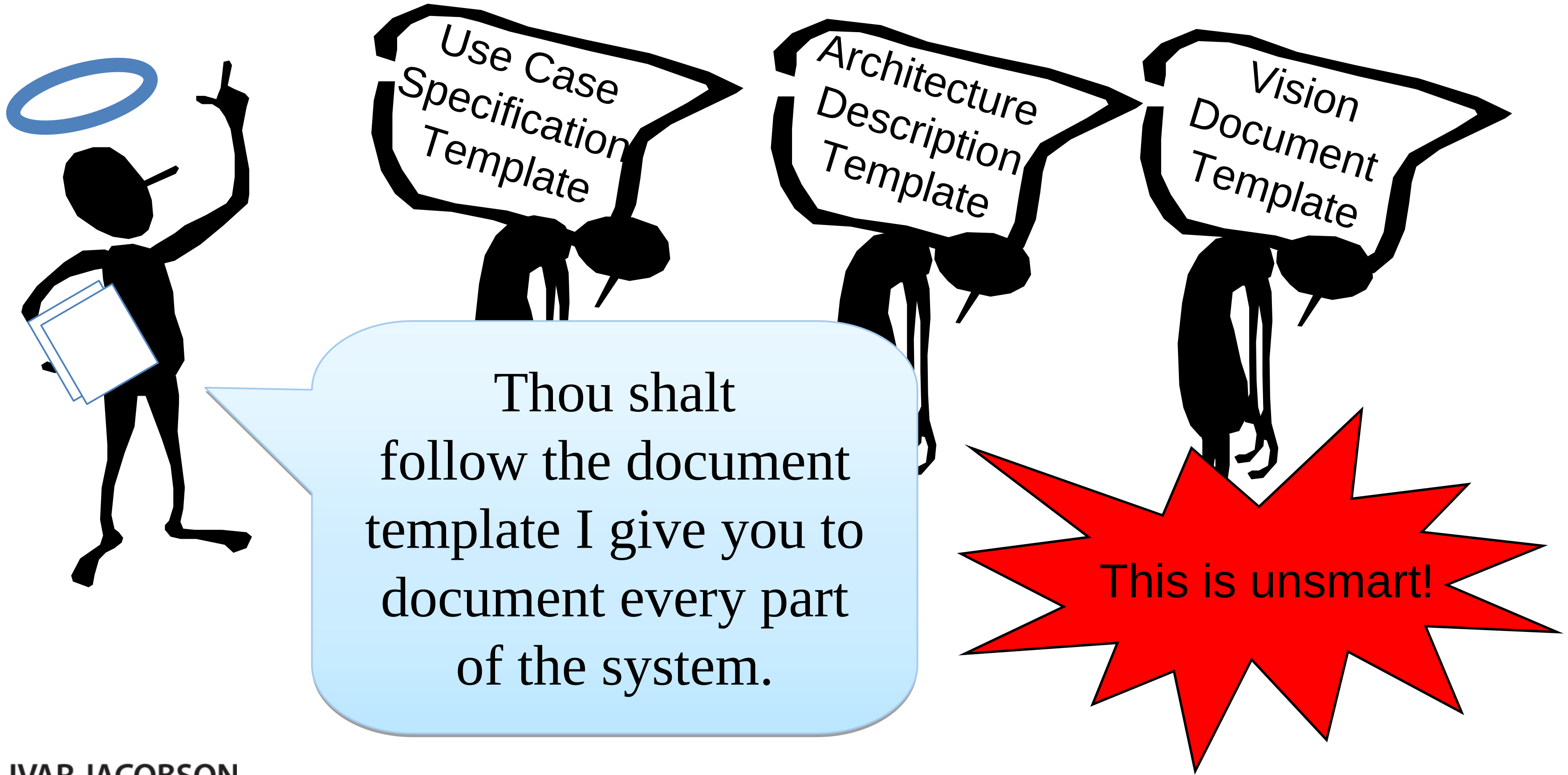


Smart with Test

Whatever you do **you** are not done
until **you** have verified
that **you** did what **you** wanted to do

Not smart with Documentation

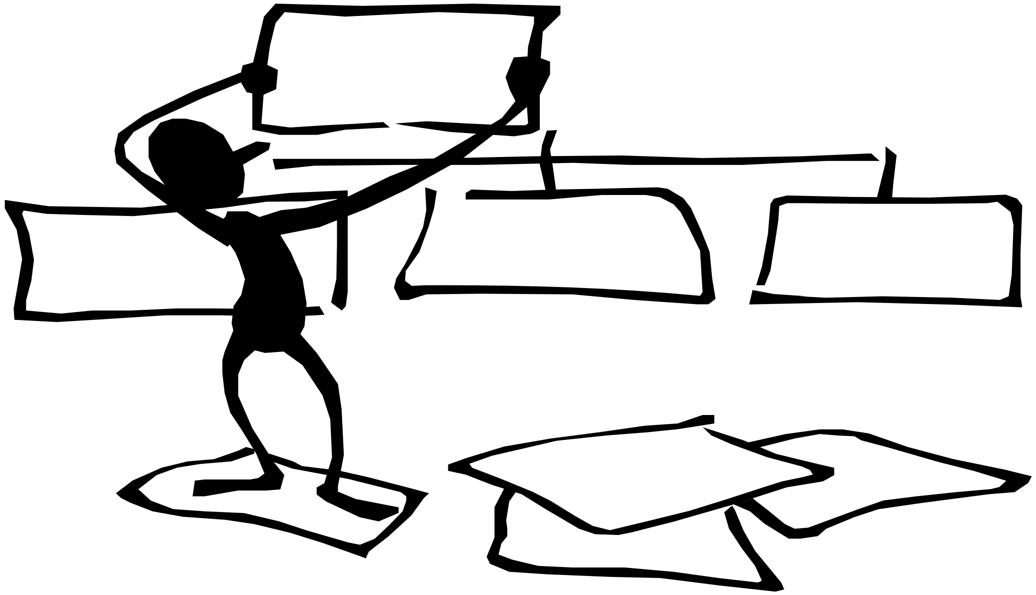
- There has been an over-emphasis on teams producing documentation


Use Case Specification Template
Architecture Description Template
Vision Document Template
Thou shalt follow the document template I give you to document every part of the system.
This is unsmart!

Smart with Documentation

Myth: The idea that you document software so people later can read what you did.

– Law of nature: People don't read documents



This is smart!

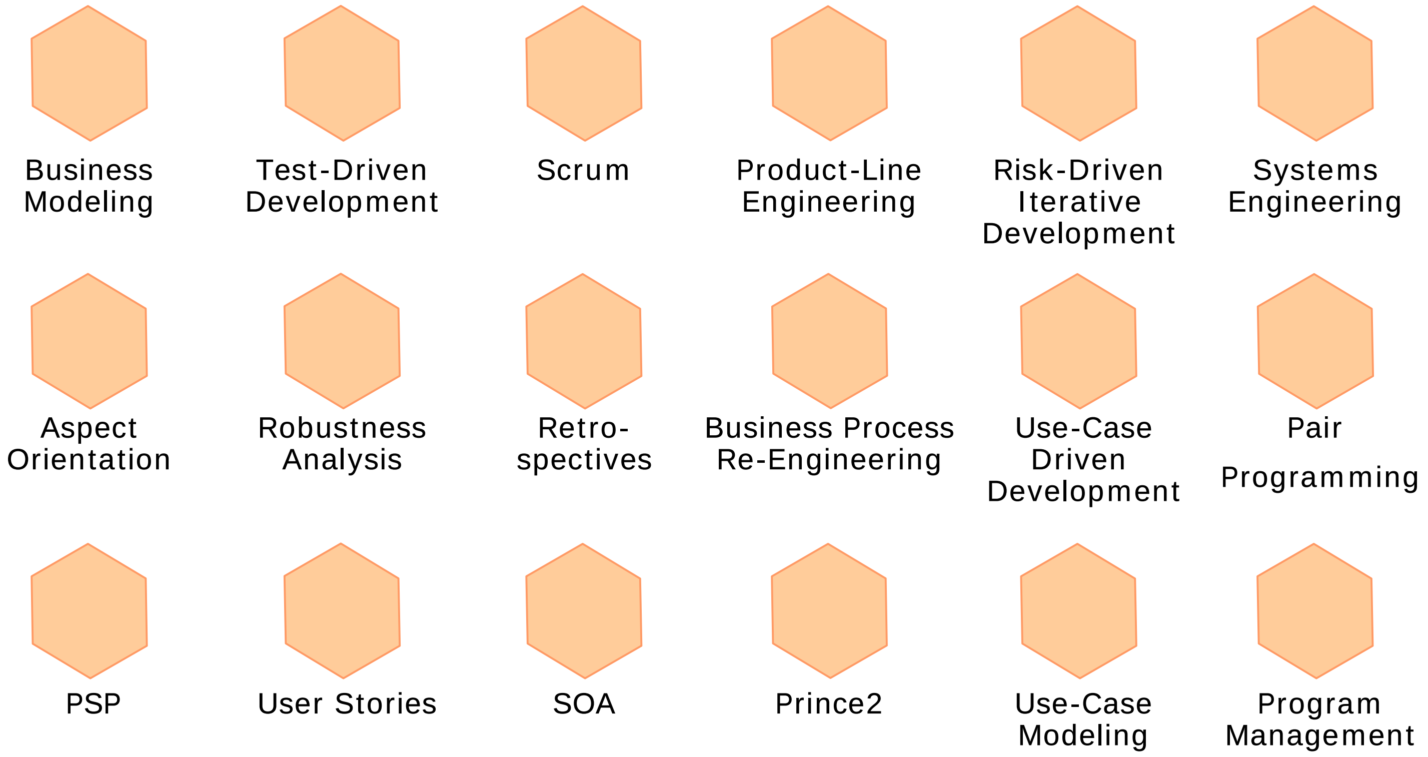
Focus on the essentials - the placeholders for conversations – people figure out the rest themselves

Agenda

1. What does Smart mean?
2. Smart Cases – Recognize it when you see it
3. How do you become Smart
4. What does Smart really mean?

How do you become Smart?

- You need knowledge in *good* (maybe best) practices
  - There are 100's of practices, some of them are good

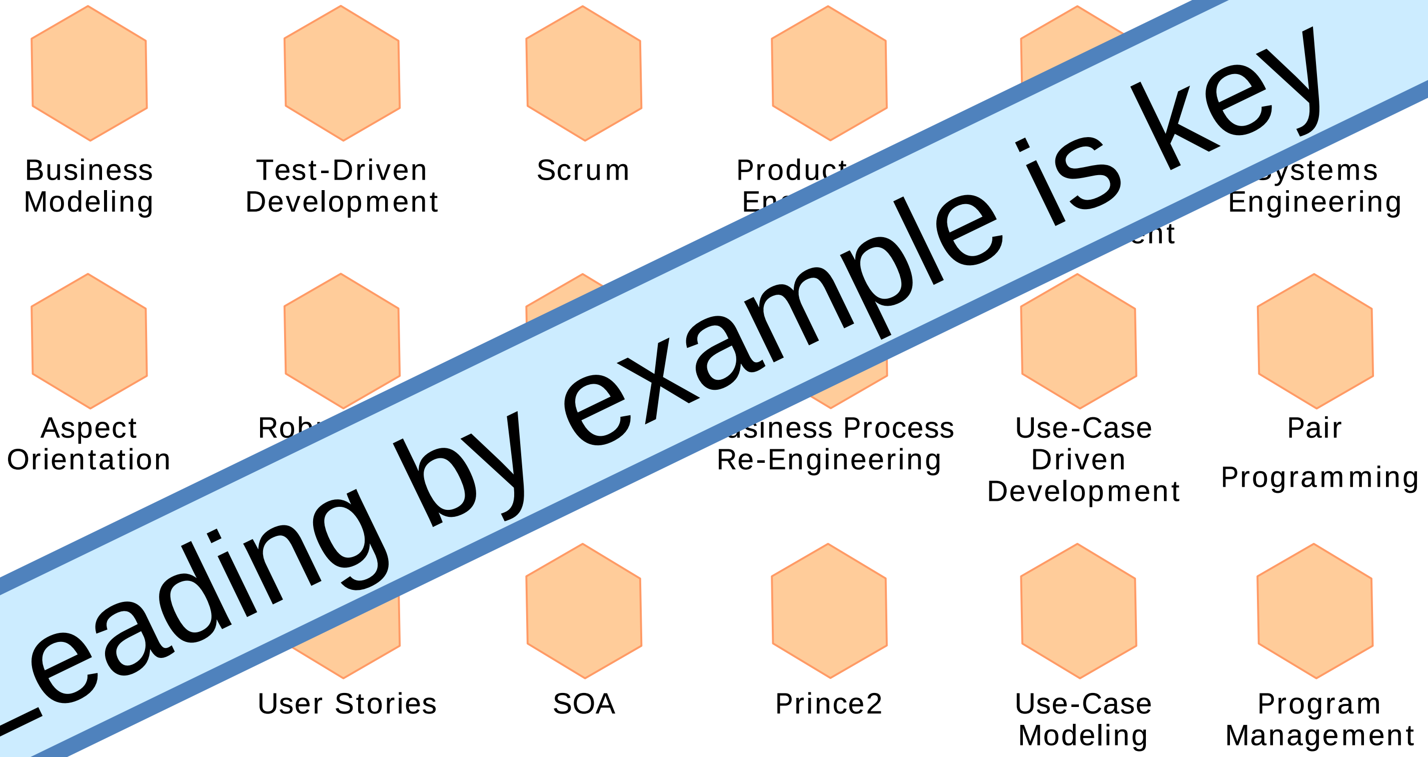


- And you need experience in using them

IVAR JACOBSON INTERNATIONAL

How do you become Smart?

- You need knowledge in *good* (maybe best) practices
  - There are 100’s of practices, some of them are good

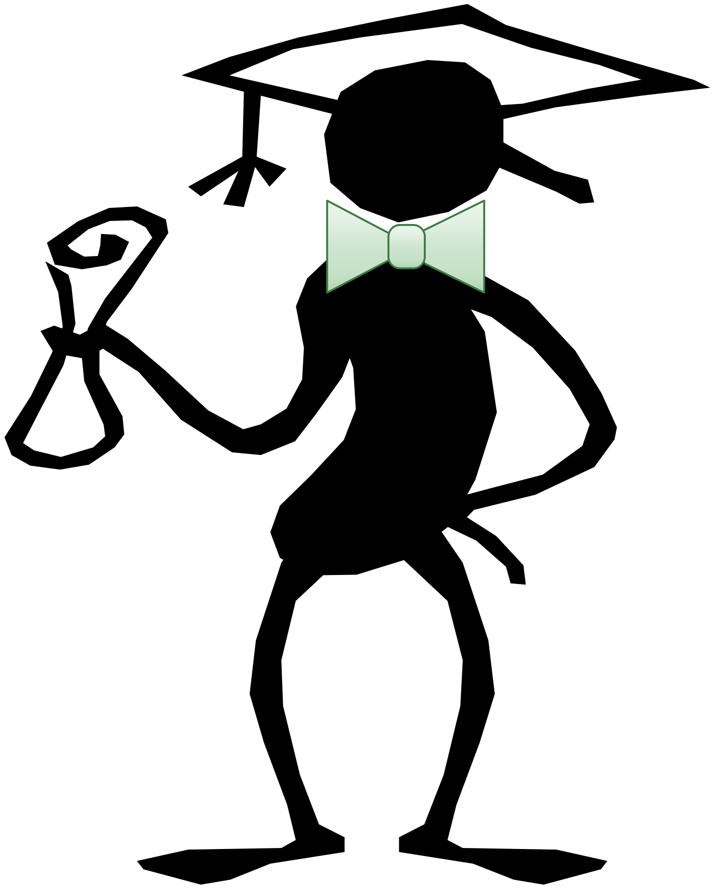


Leading by example is key

- And you need experience in using them

Of course, eventually it comes back to you, but

We can all become smarter.



IVAR JACOBSON INTERNATIONAL

Thank You

Contact me at ivar@ivarjacobson.com

IVAR JACOBSON INTERNATIONAL

Thank You

ivar@ivarjacobson.com

IVAR JACOBSON INTERNATIONAL