# Software Engineering and Scientific Computing

**Barbara Paech, Hanna Remmel**
Institute of Computer Science
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
paech@informatik.uni-heidelberg.de

RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

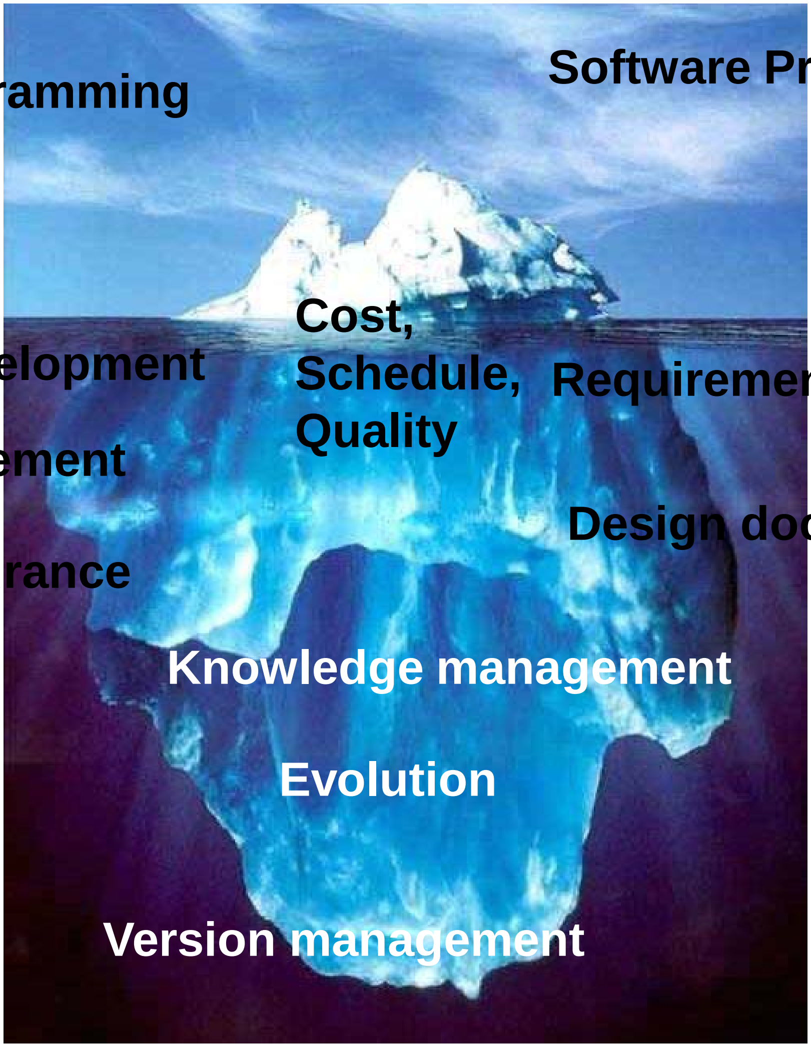
# AG Software Engineering, Uni HD

- **Profile Quality Engineering**
  - Requirements Engineering
  - Rationale Management
  - Quality Assurance

- Prof. Dr. Barbara Paech
  - Since 8 years in HD
  - before Fh IESE, Kaiserlautern
  - 6 PhD students
  - 2 Children

- Hanna Remmel
  - Master in Computer Science
  - Worked 7 years as a software developer
  - PhD student since April 2010

## Software development is complex



## Who are you?

- Your name
- What are you doing at Uni HD?
- What do you know about and what do you practise in software engineering?
- What are YOUR biggest problems with software engineering?
- What do you want to learn about software engineering and scientific computing?

## Goals of this course

- Understand the **complexity** of software development
- Know the **interests and responsibilities** of the project team
- Know  the **basic practices** of software development

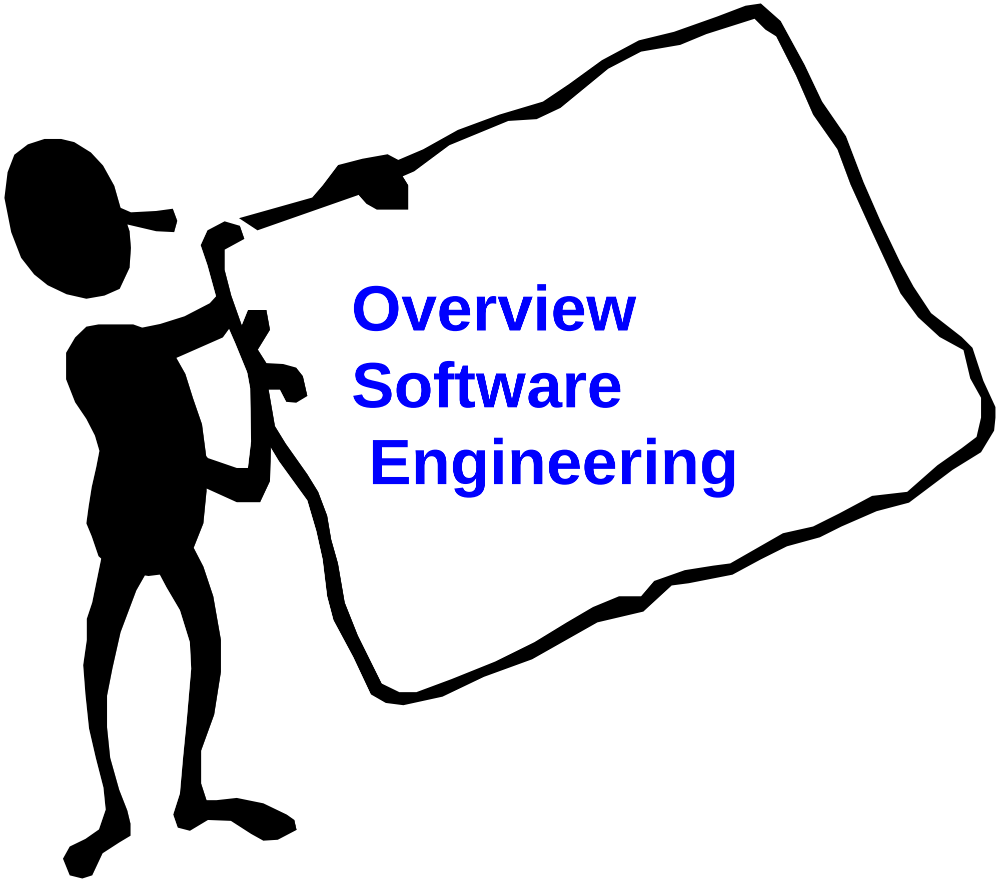
- Main contents
  - Version management
  - Build management
  - Issue tracking
  - Project management
  - Testing
  - Documentation and Modeling
  - Knowledge Management

## Course Overview

| Time | First Day | Second Day | Third Day | Fourth Day |
|---|---|---|---|---|
| 09:00-10:30 | Lecture | Lecture | Lecture | Free exercises |
| | | Break | | |
| 10:30-11:00 | Break | eXtreme Hour | Break | |
| 11:00-12:00 | Lecture | | Lecture | |
| 12:00-13:00 | Break | Break | Break | |
| 13:00-16:00 | Exercises | Exercises | Exercises | |

## Schedule First Day (Tuesday)

| | |
|---|---|
| 9:00 | Introduction to each other,<br>Introduction to Software Engineering |
| 10:30 | Break |
| 11:00 | Software Engineering in Computational Science Projects<br>Version management concepts |
| 12:00 | Lunch |
| 13:00<br>Incl. a short break | Tools, Exercises<br>Version Management<br>Issue Tracking<br>Build Management |
| 16.00 | End |



## Overview Software Engineering

# Overview

1. Terminology
2. Motivation
3. General Structure
    1. What to do? (Activities)
    2. What to produce?  (Results, Products)
    3. Who? (Actors)
    4. How? (Methods, Tools, Best Practices)

## What is Software Engineering?

- Software
  - Software is a collection of computer programs, procedures, rules, **corresponding documentation and data** (ISO/IEC 12207:2008)
- Engineering
  - Systematic process and prodcuct
  - Adherence to standards, consideration of quality and cost
  - Usage of models

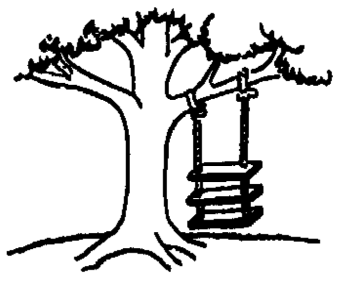
[Standish group]

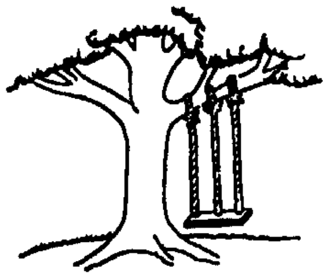
## SE is difficult!

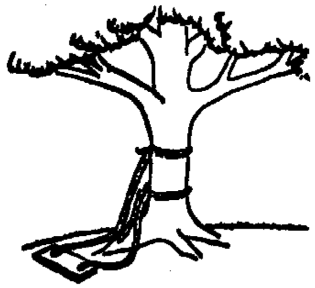
- Just 32 % of the projects successful, 25 % without result, 44 % not within schedule
- Time overrun up to 63%, cost overrun up to 45 %
- What is important for success?

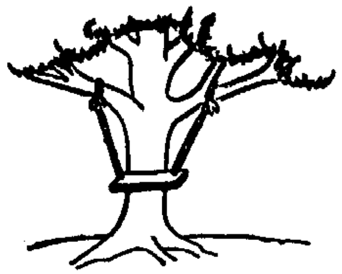
| | |
|---|---|
| Management support | 18 |
| User involvement | 16 |
| Experienced project managers | 14 |
| Clear business goal | 12 |
| Reduced Scope | 10 |
| Standard SW Infrastructure | 8 |
| Fixed Requirements | 6 |
| Formal Methods | 6 |
| Reliable estimation | 5 |

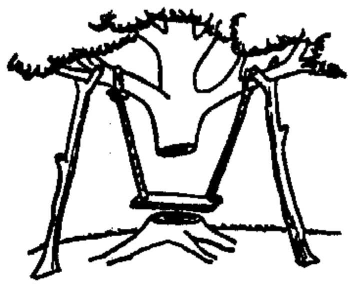
# Joint understanding of all stakeholders

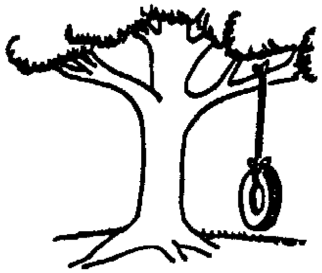

As proposed by the project sponsor


As specified in the project request


As designed by the senior analyst


As produced by the programmers


As installed at the user's site


What the user wanted

## Software is big

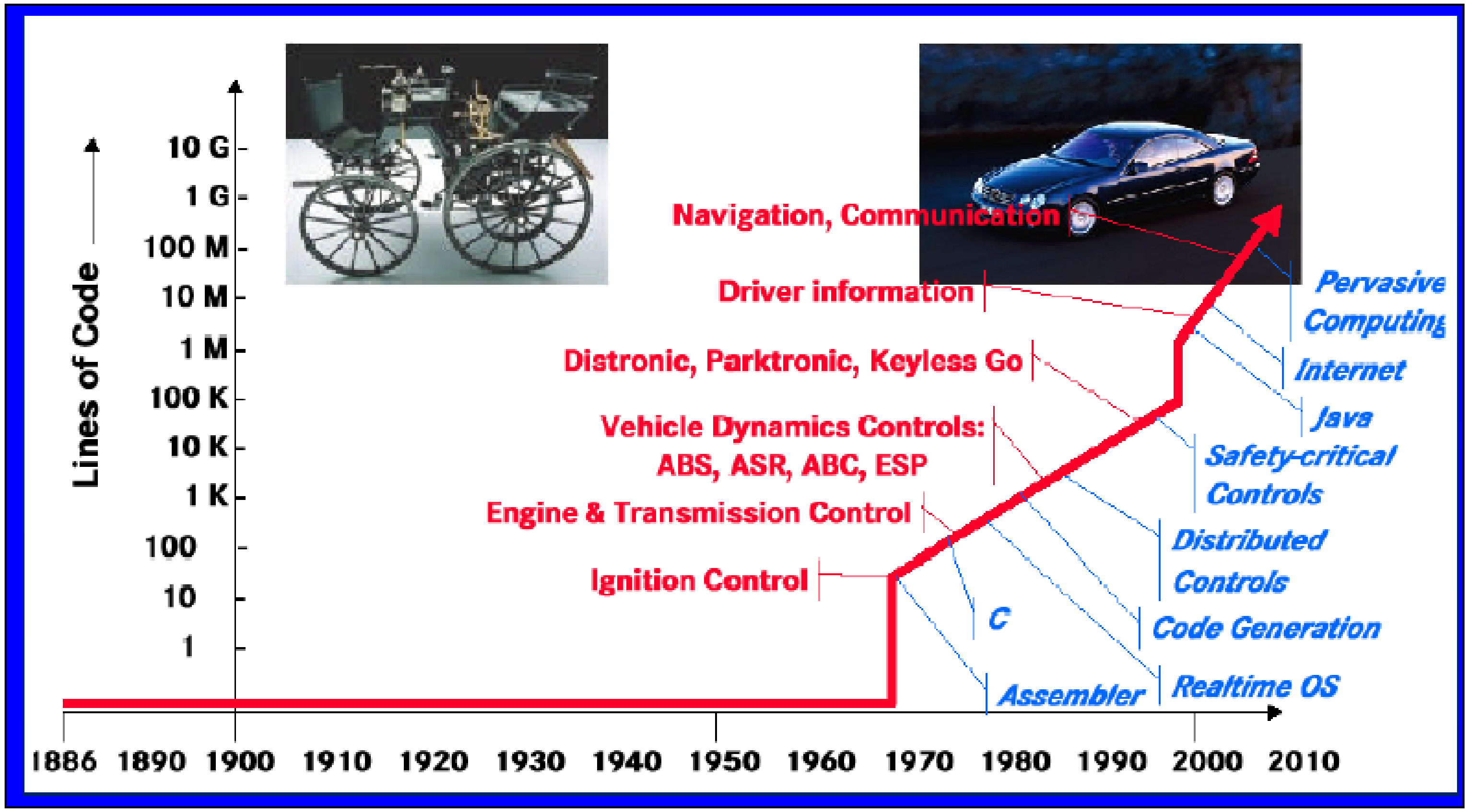
- Enterprise-Ressource-Planning Software R/3 von SAP

| Year | Lines of Code | Number components |
|---|---|---|
| 1994 | 7 Million | 14.000 |
| 1997 (Rel. 3.1.) | 30 Million | 200.000 |
| 1999 (Rel. 4.5.) | 50 Million | 400.000 |

- => Team work
  - Communication
  - Knowledge management
  - Project management

# Rapid Technology Change

**DaimlerChrysler 2003**



## Quality is difficult

- Error rates (M. Cusumano, MIT 1990)
  - 1977: 7-20 errors in 1000 LOC
  - 1994: 0,05-0,2 errors in 1000 LOC
  - Increase factor 100 in 17 years
  - **But: complexity increase factor 10 in 5 years**
- 0,1 errors means:
  - 18 plan crashes per day
  - 22.000 money mistransfers per hour

## Quality in small programs

- Software characteristics depend on goals [Weinberg,Schulmann, 1974]

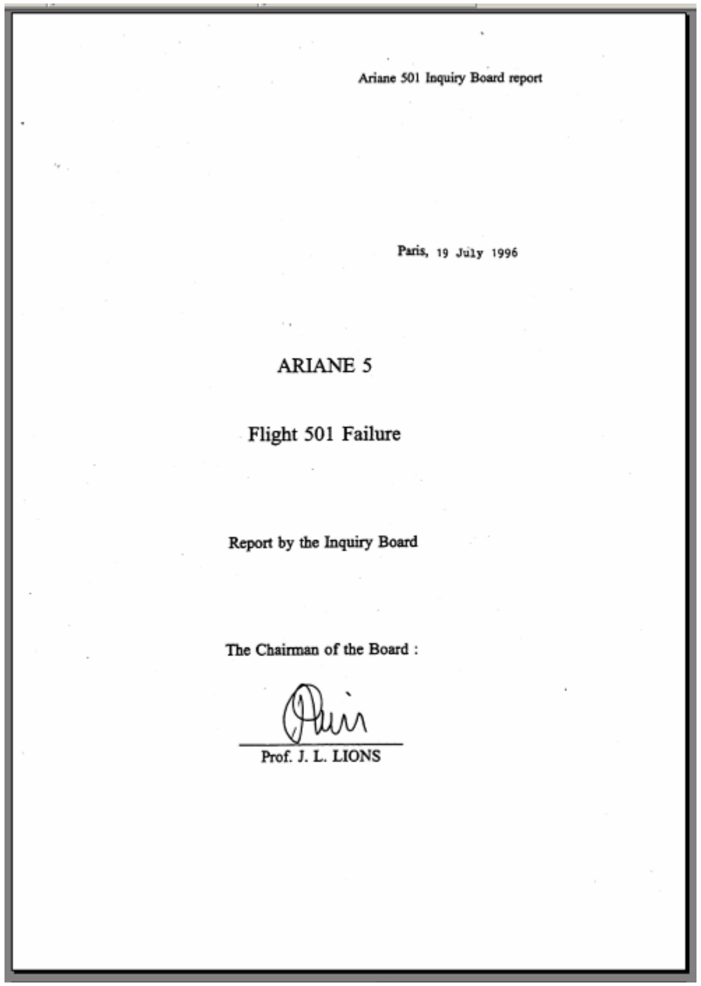
| **Goal** | **Effort** | **LOC** | **Memory** | **Understand-ability of the code** | **Understand-ability of the output** |
|---|---|---|---|---|---|
| **Effort** | 1 | 4 | 4 | 5 | 3 |
| **LOC** | 2-3 | 1 | 2 | 3 | 5 |
| **Memory** | 5 | 2 | 1 | 4 | 4 |
| **Understandability of the code** | 4 | 3 | 3 | 2 | 2 |
| **Understandability of the output** | 2-3 | 5 | 5 | 1 | 1 |

# Quality in big programs

- Well-known example: Ariane 501
- Ariane5 successor of Ariane4-family with over 100 successful starts
- 6-12t carriage (vs. 2-5t A4)
- First start  4.6.1996

## „Result“

- Rocket destroys itself after a few minutes
- High damage
  - Carriage lost, cost > 500 M€
  - 3 year delay of further missions
- Investigation committee
  - Report from 19.6.1996 (only 14 days after !!!)
  - see http://ravel.esrin.esa.it/docs/esa-x-1819eng.pdf



Ariane 501 Inquiry Board report

Paris, 19 July 1996

ARIANE 5

Flight 501 Failure

Report by the Inquiry Board

The Chairman of the Board :

[illegible]

Prof. J. L. LIONS

## Causes

- Root problem: conversion error, missing exception handling (programming error).
- Missing exception handling to save execution time (cost).
- Un-documented assumptions about value ranges (distributed development).
- Planned travel route not included in the requirements specification (management).
- Shut-down in case of errors typical for hardware problems (culture).
- Unnecessary code copied from A4 (re-use).
- Copied code not tested (testing, re-use).
- Missing Review (quality assurance)

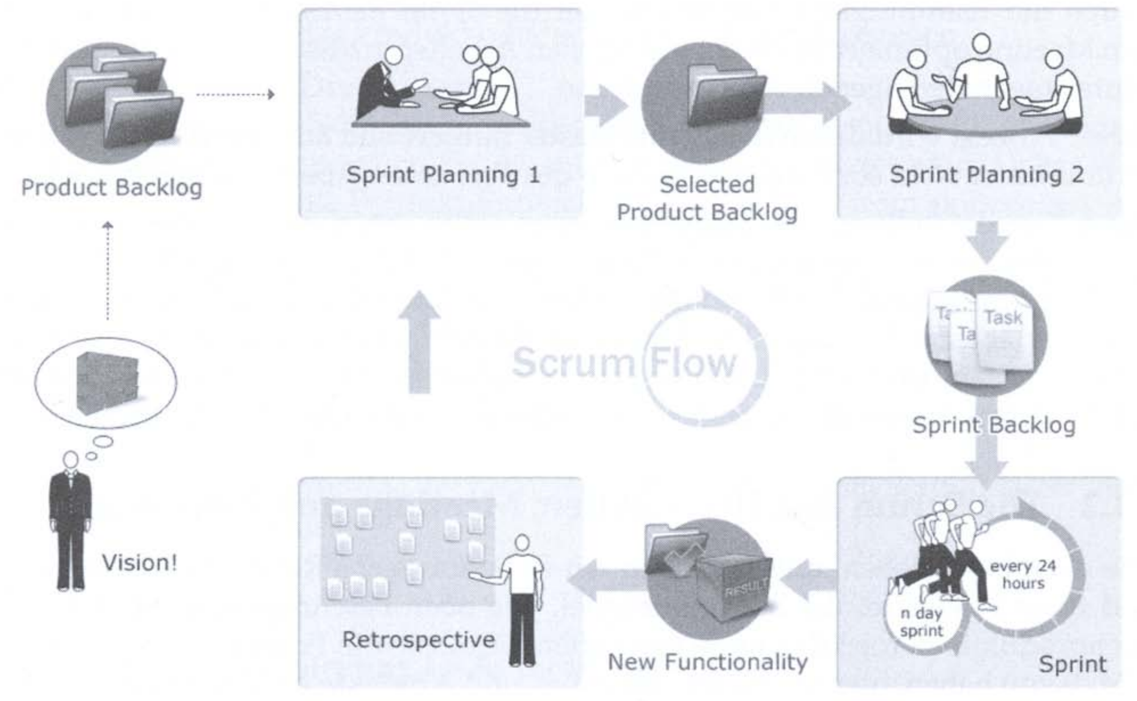
# Software Engineering (SE)

- Development
  - Of big programs
  - With high quality
  - By many team members
  - With cost and time limits

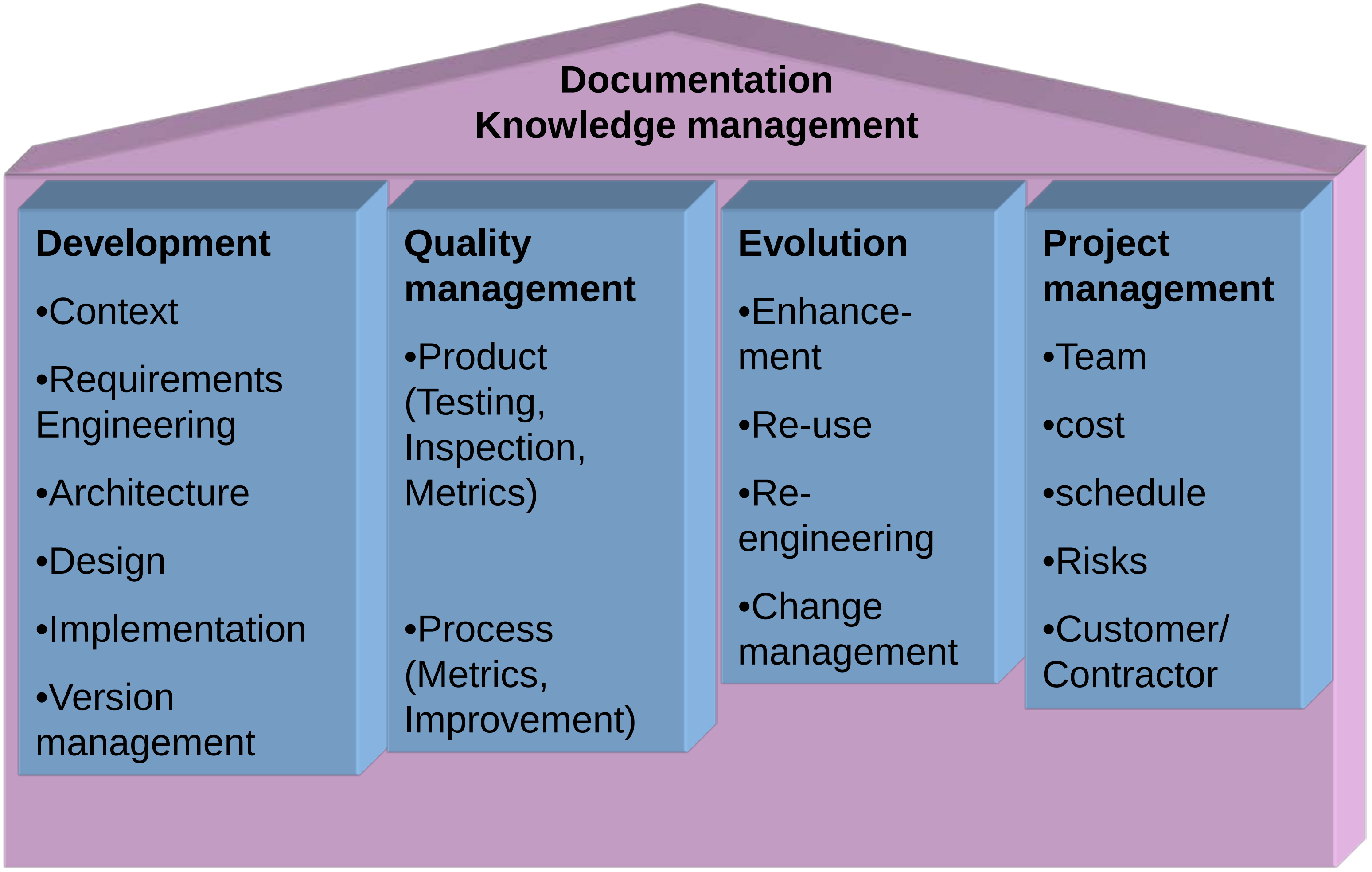
  - using Engineering principles
    - Planning
    - Work distribution
    - Methods and Tools (Standardisation, Quality)
    - Models to support knowledge management

# Software Engineering is a Process

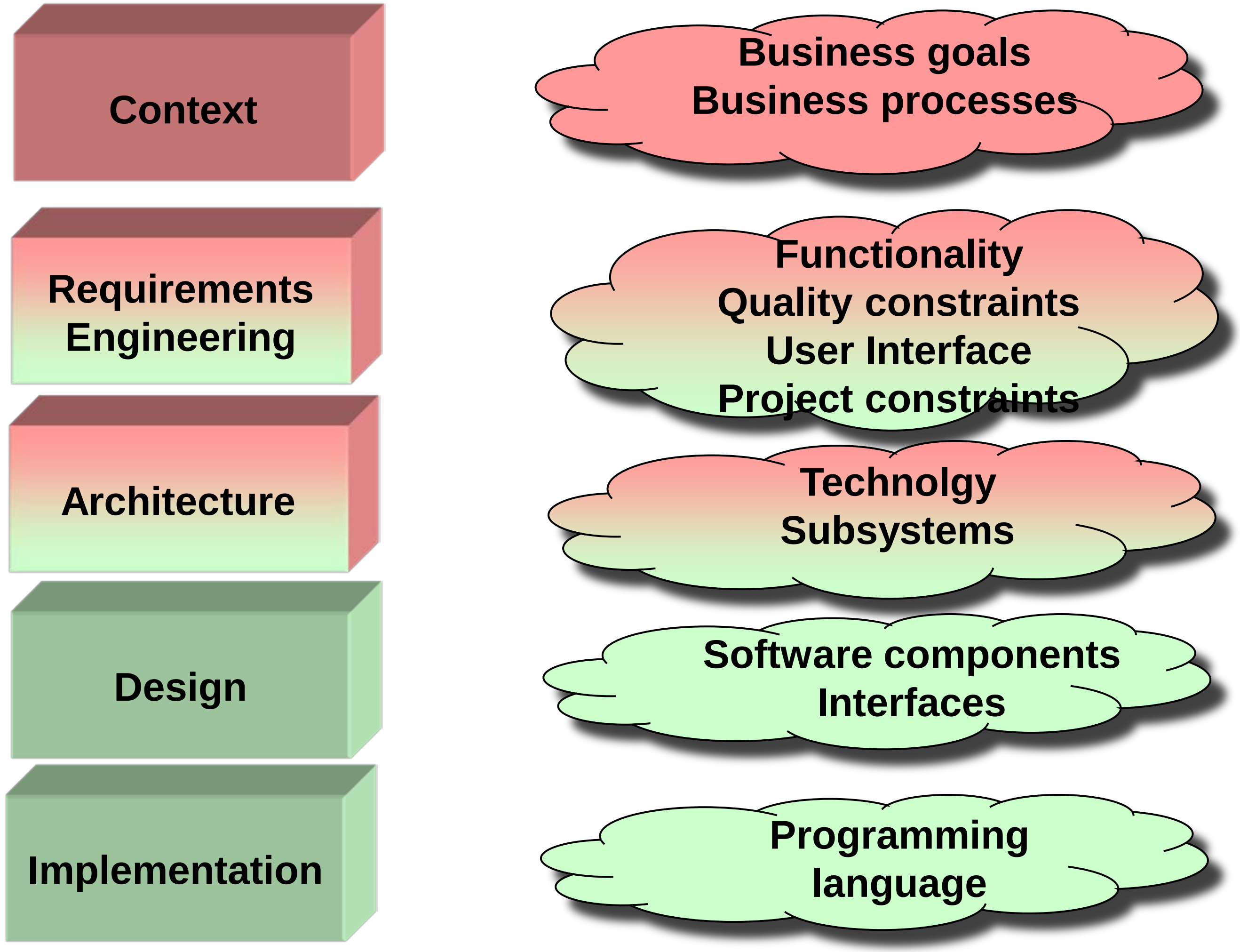
- actors (WHO)
- activities (WHAT)
- results (WHAT)
- guidelines (HOW)
- context (HOW)



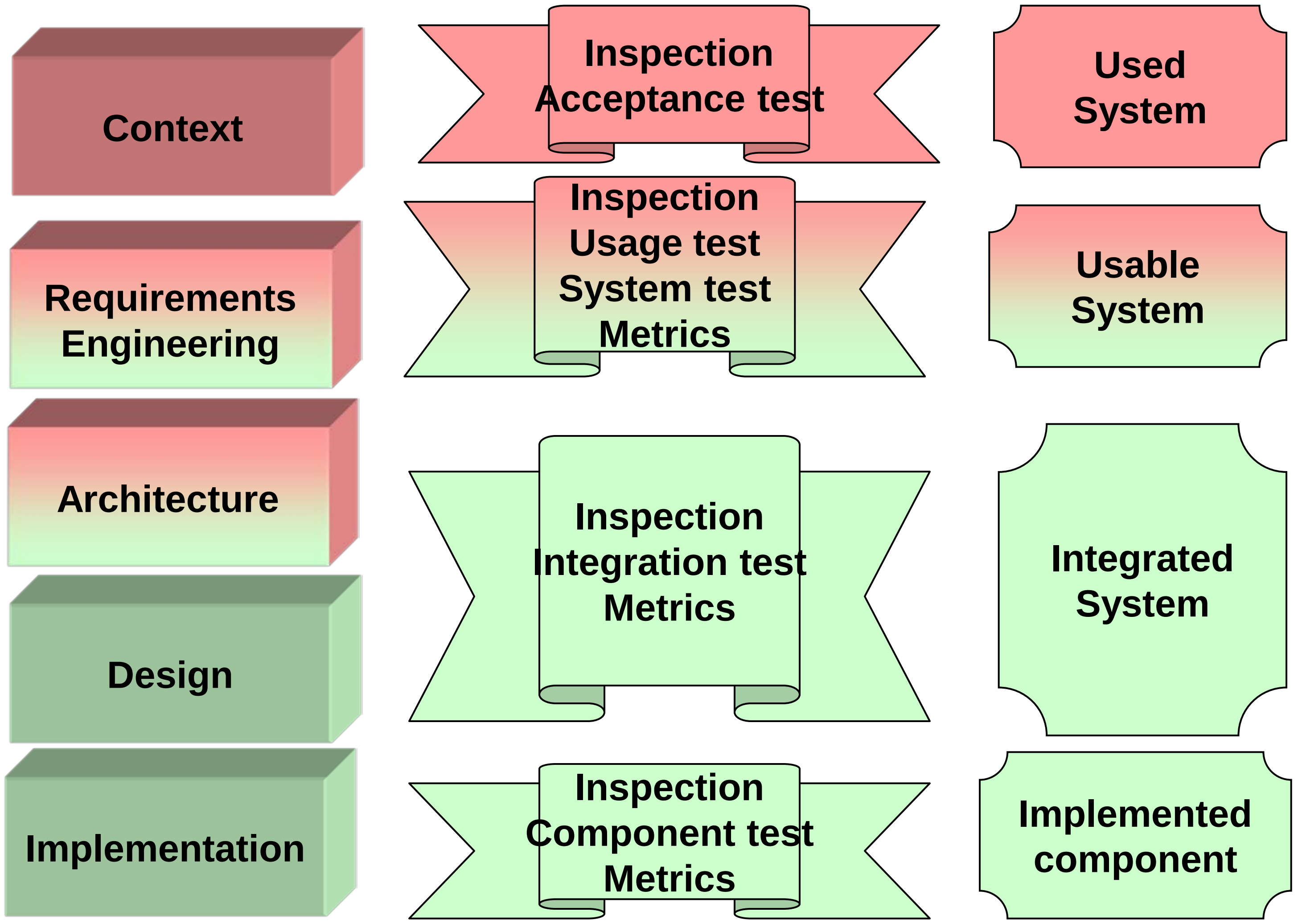
## Responsibilities



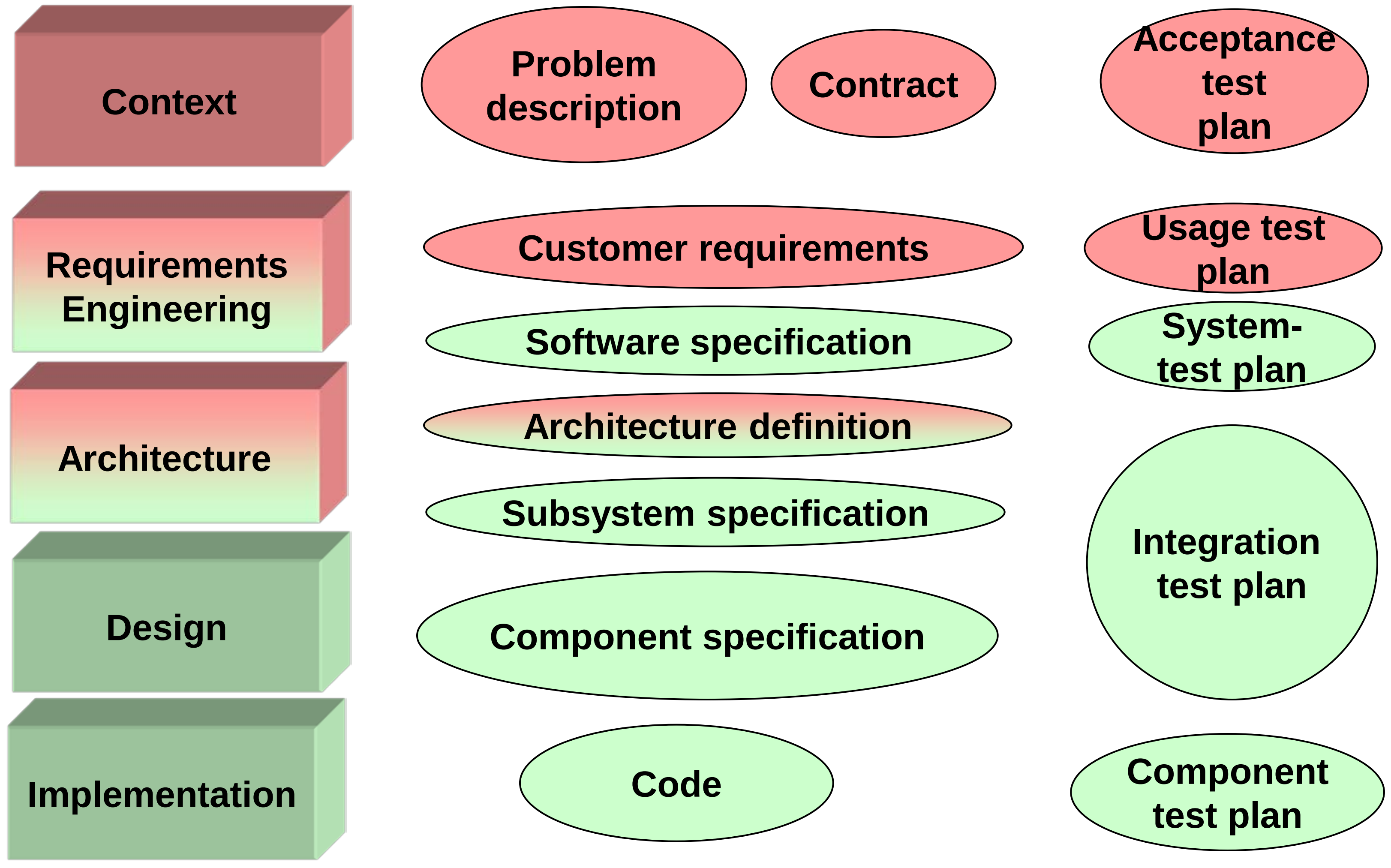
# Development Decisions



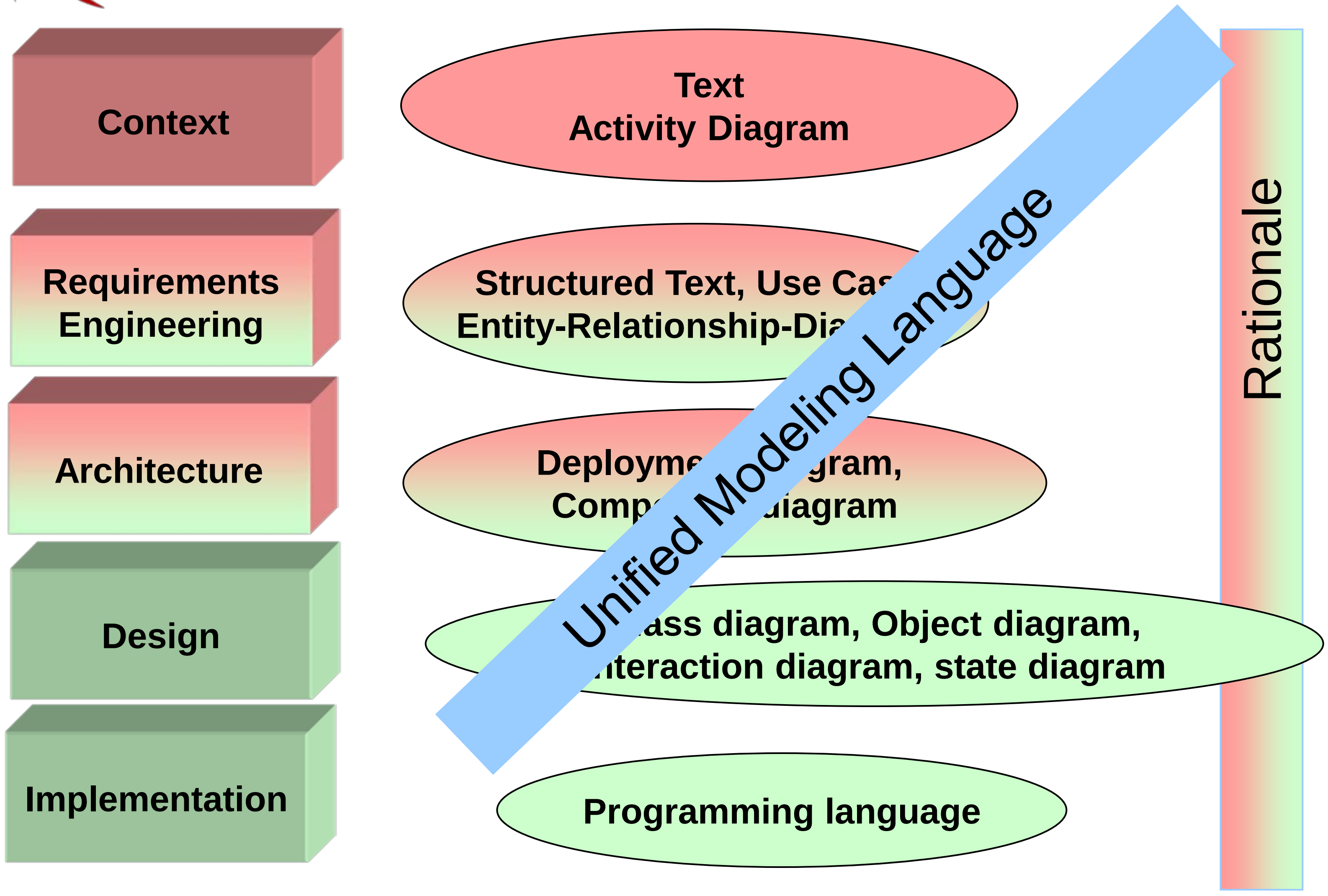
# Quality assurance



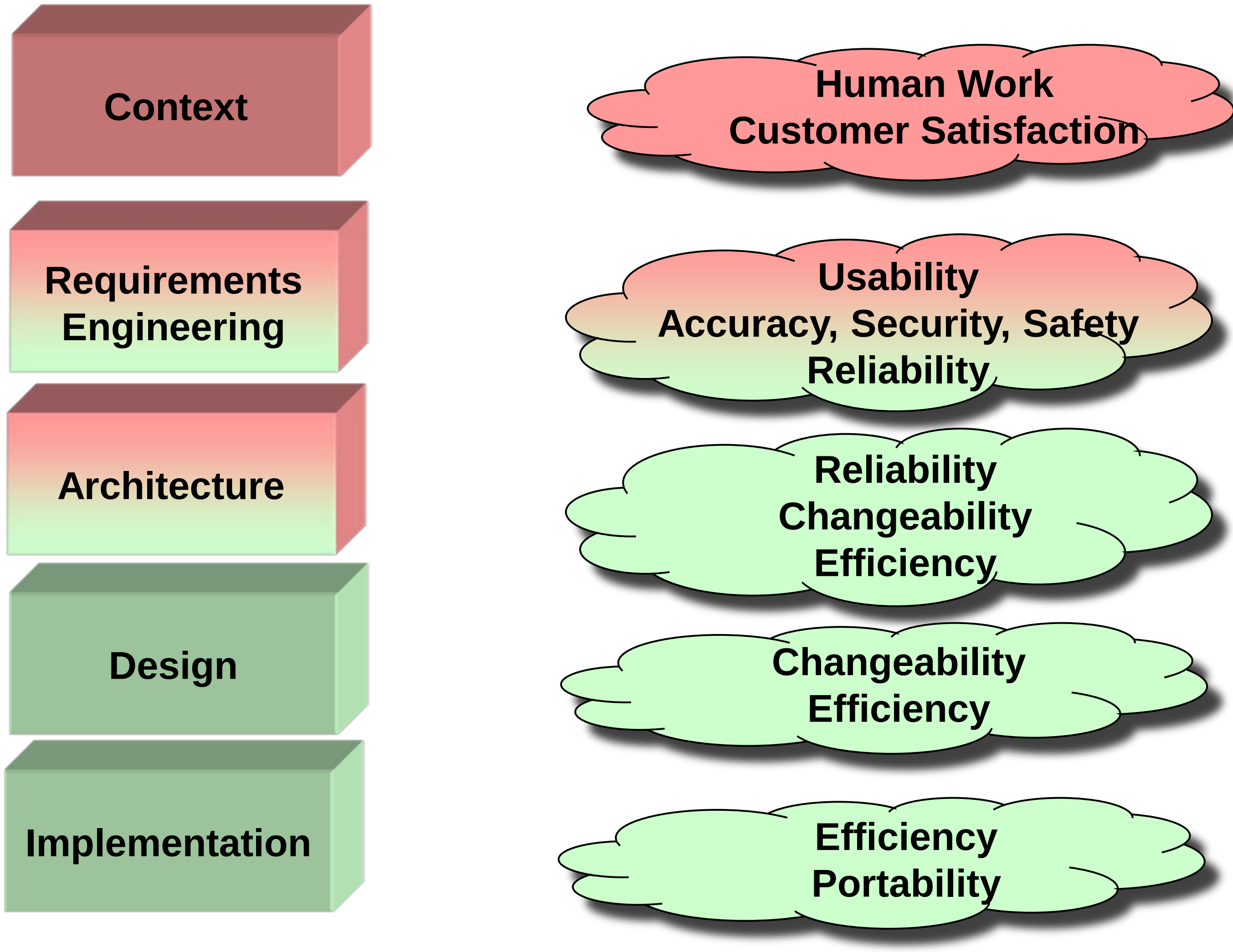
# Documentation



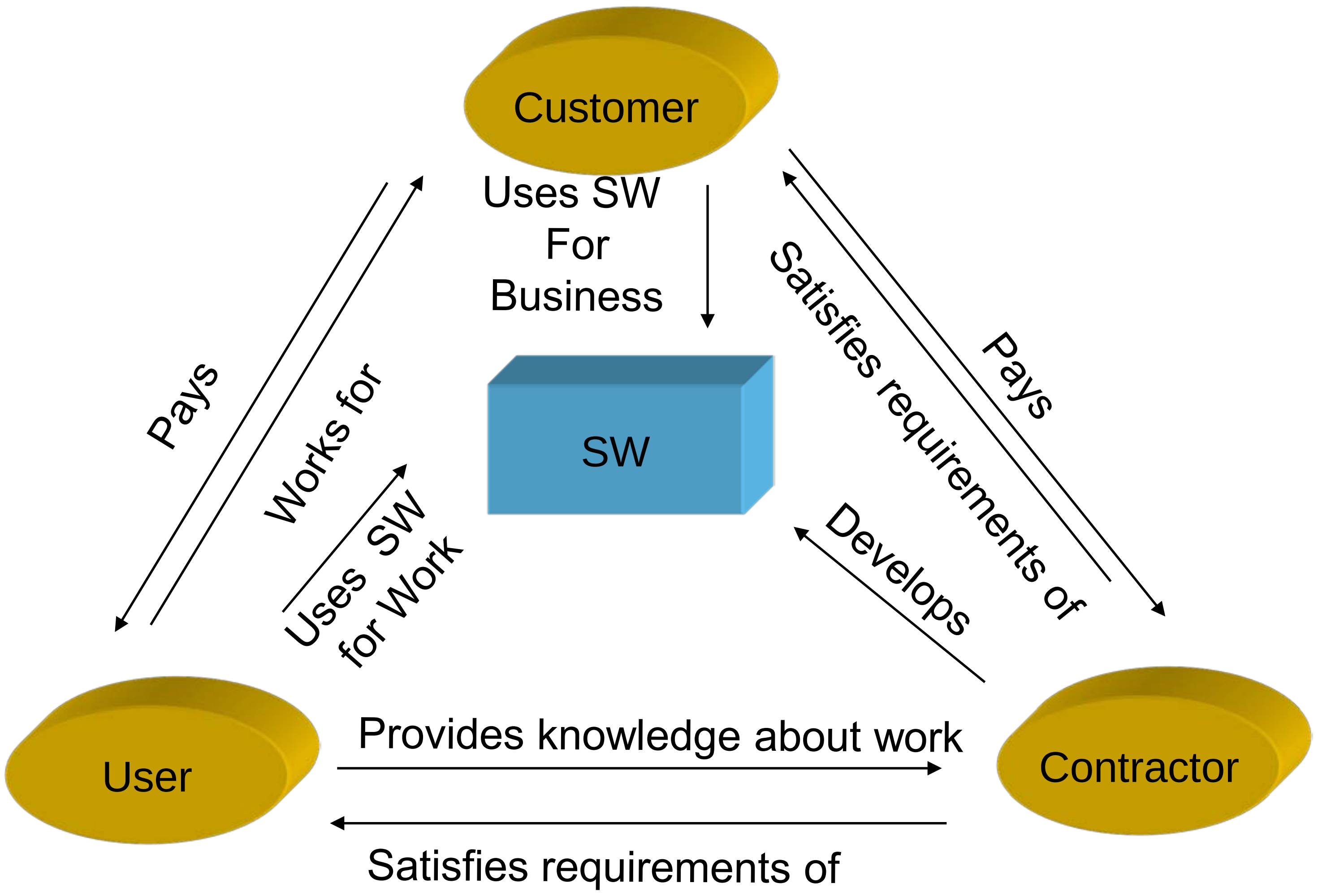
# Modeling


Context
Requirements Engineering
Architecture
Design
Implementation
Text
Activity Diagram
Structured Text, Use Cas
Entity-Relationship-Dia
Deployme gram,
Comp diagram
ass diagram, Object diagram,
nteraction diagram, state diagram
Programming language
Unified Modeling Language
Rationale

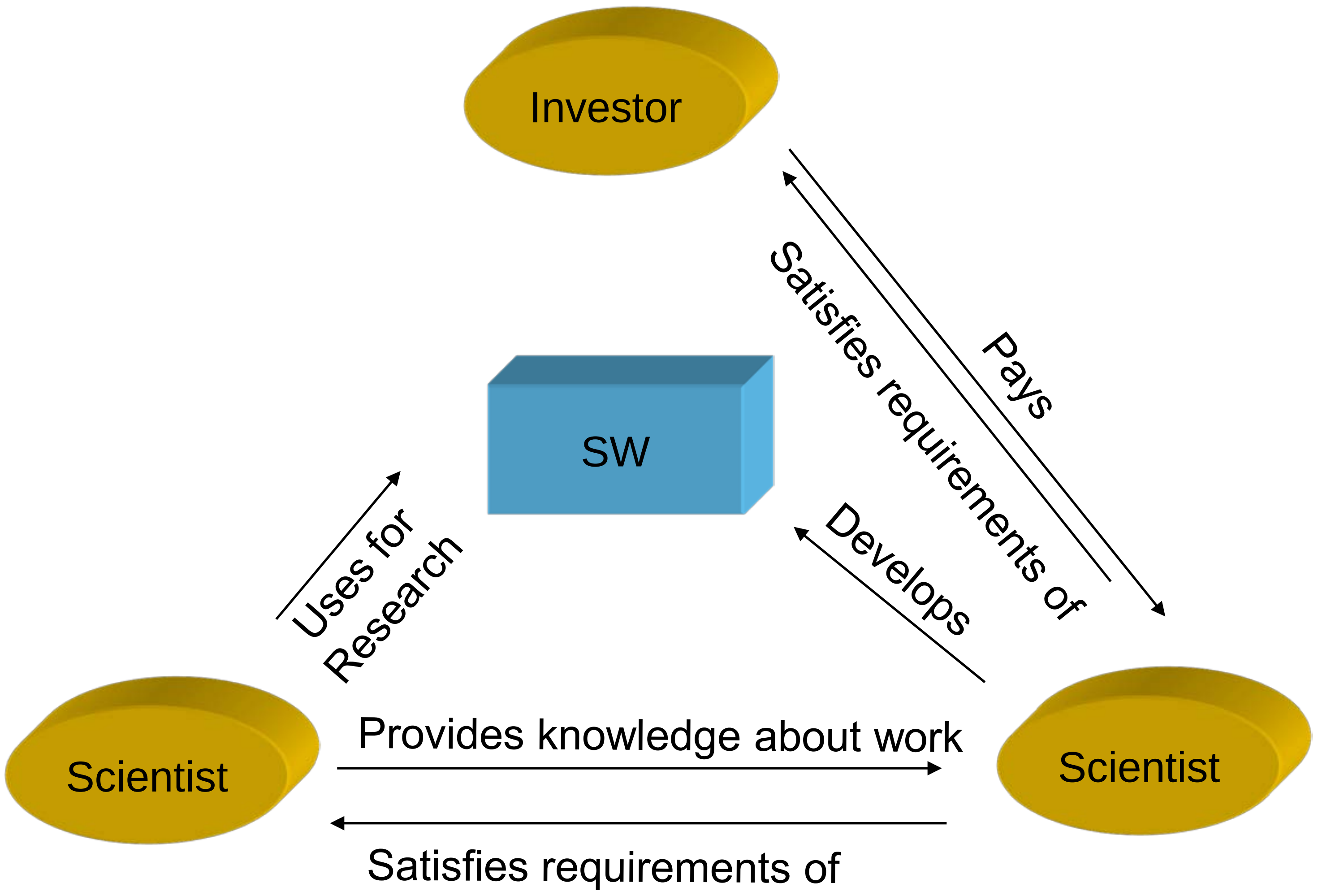
# Software Quality ISO 9126/DIN66272



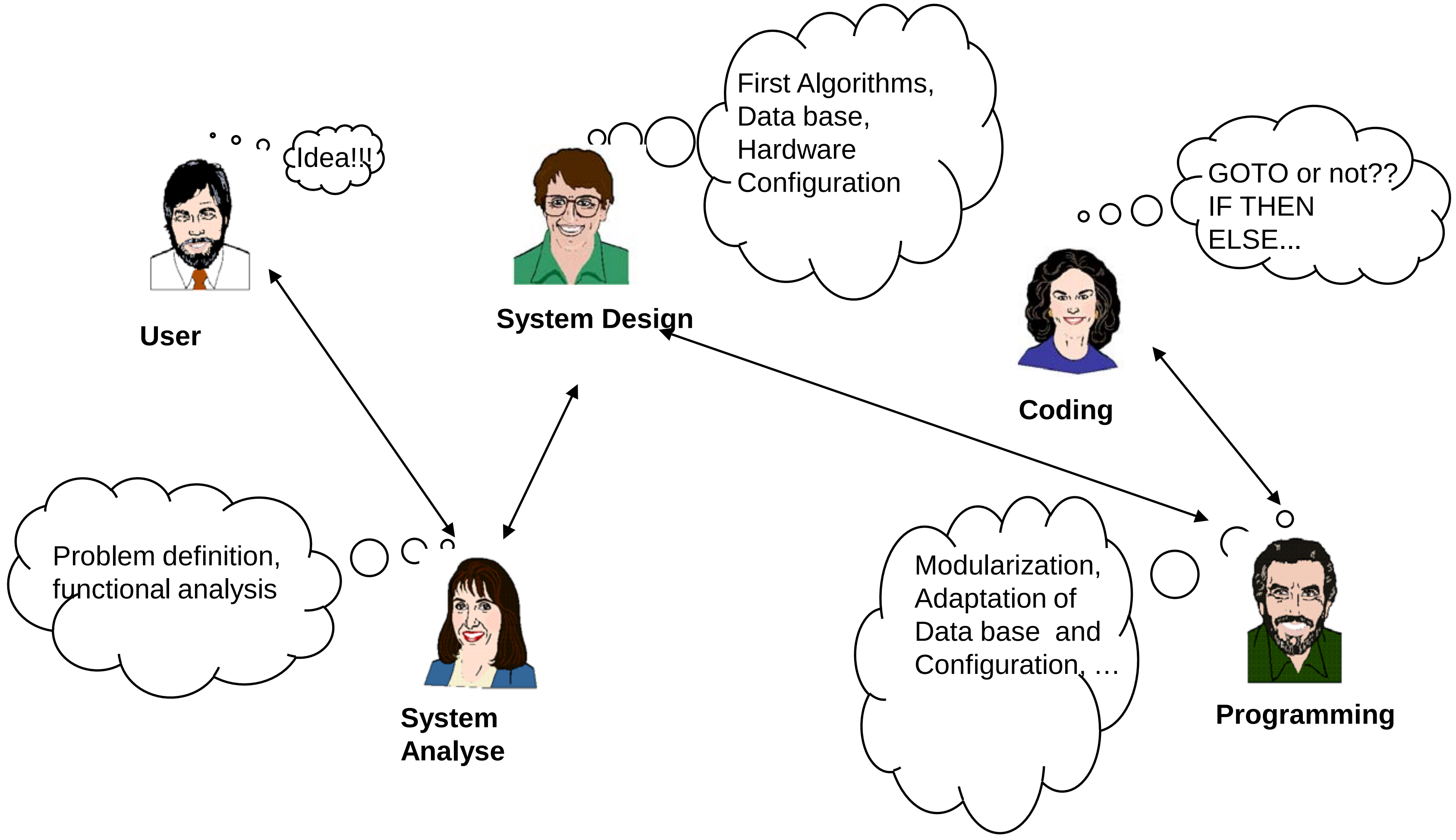
# Project Participants traditional SE



# Project Participants Scientific SE


Investor
SW
Pays
Satisfies requirements of
Develops
Uses for Research
Scientist
Provides knowledge about work
Satisfies requirements of
Scientist

# Software Engineering Methods in the Past

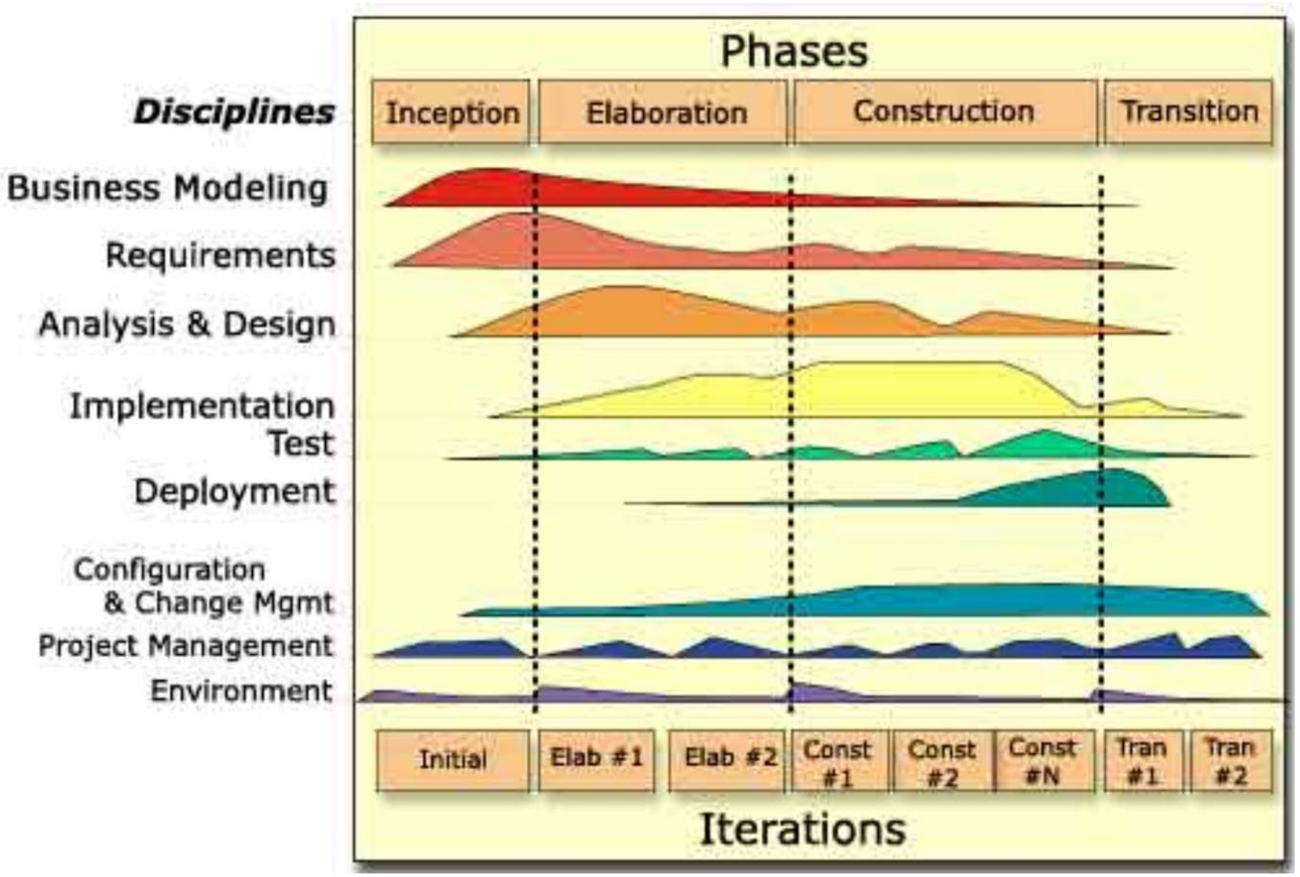


Idea!!!
User
First Algorithms, Data base, Hardware Configuration
System Design
GOTO or not?? IF THEN ELSE...
Coding
Problem definition, functional analysis
System Analyse
Modularization, Adaptation of Data base and Configuration, …
Programming

[Yourdon/Constantine 1979]

http://www-306.ibm.com/software/de/rational/

## Software Engineering Methods Today



## Summary

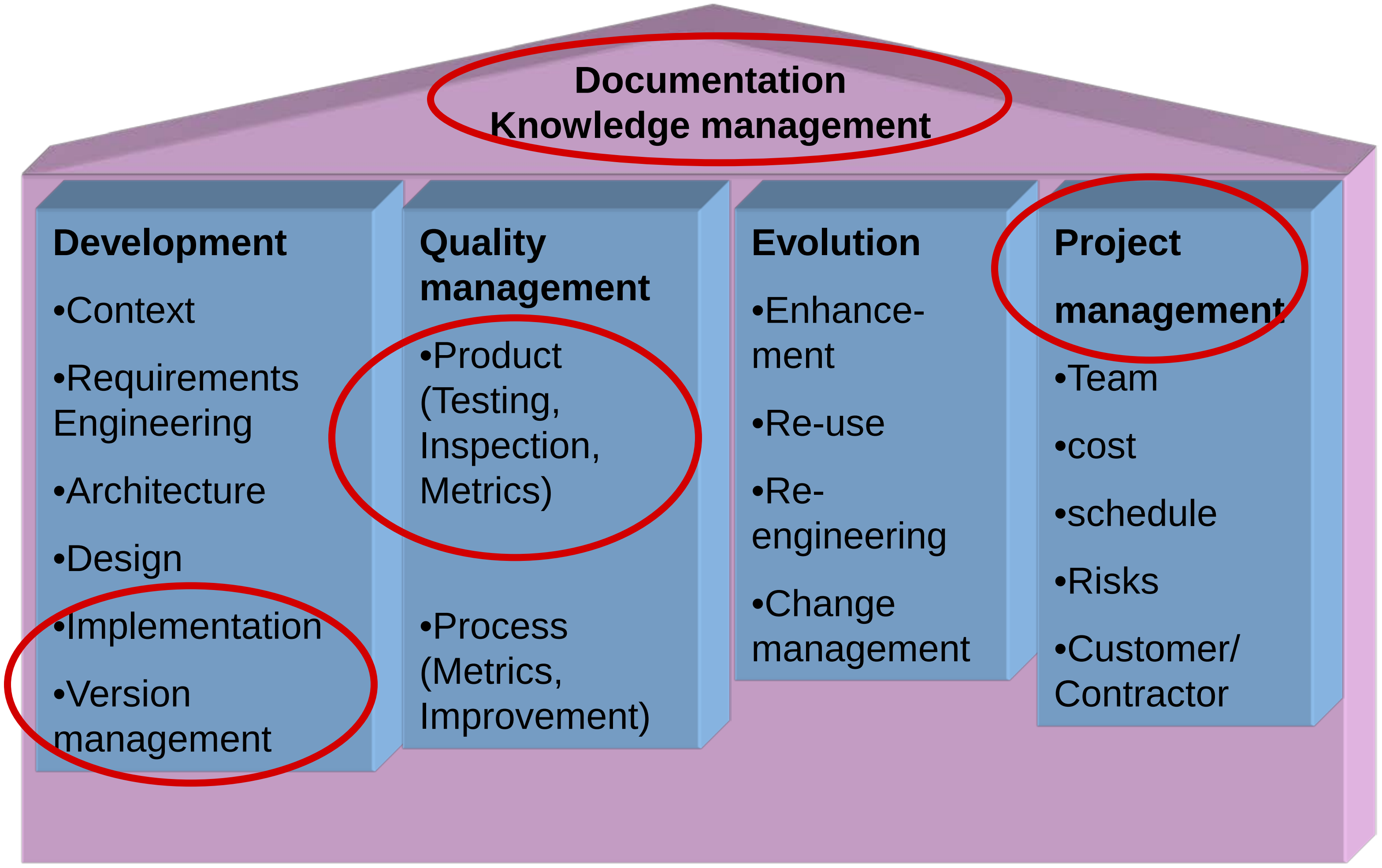
- SE creates socio-technical Systems
- SE focuses on quality, cost und effort
- SE shapes product / system and process

## Literature

- J. Ludewig, H. Lichter, *Software Engineering,* dpunkt 2007
- I. Sommerville, *Software Engineering*, Addison Wesley, 2008

- G. Weinberg, E. Schulmann, *Goals and Performance in Computer Programming*, Human Factors 16, p.70-77,1974
- E. Yourdon, L.L. Constantine, *Structured Design – Fundamentals of a Discipline of Computer Program and System Design*, Prentice Hall, 1979

# In this course: Programming in a small team



# Programming in a small team

What is
Ron doing?

Project management
Issue Tracking

I want to explain
my ideas to Hermione

Modeling
Knowledge Management

I want to change
Ginnys code

Version management,
Build management

I want to check
Harrys changes

Quality assurance
Testing