FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Information Processing Lecture

Prof. Dr. Andreas Pech

Fachbereich 2 Informatik und Ingenieurwissenschaften

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Information Processing

Three parts:

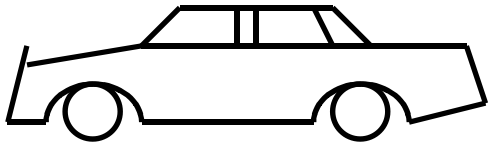
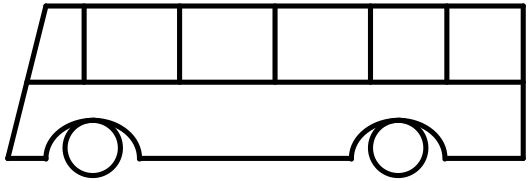
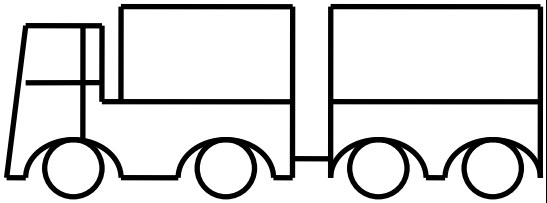
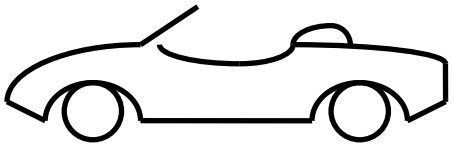
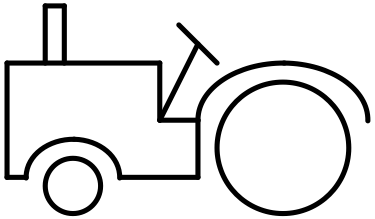
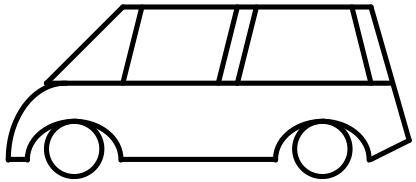
1 Software Engineering

2 Information Processing Lab

3 Project

Sensitization

Sensibility
The problem: Do we understand?

What the user Wants 	**How the user explains to the programmer** 	**What does the programmer understand** 
What the Programmer wants to build 	**What the programmer actually builds** 	**What the user actually needs** 

Quell: sd&m

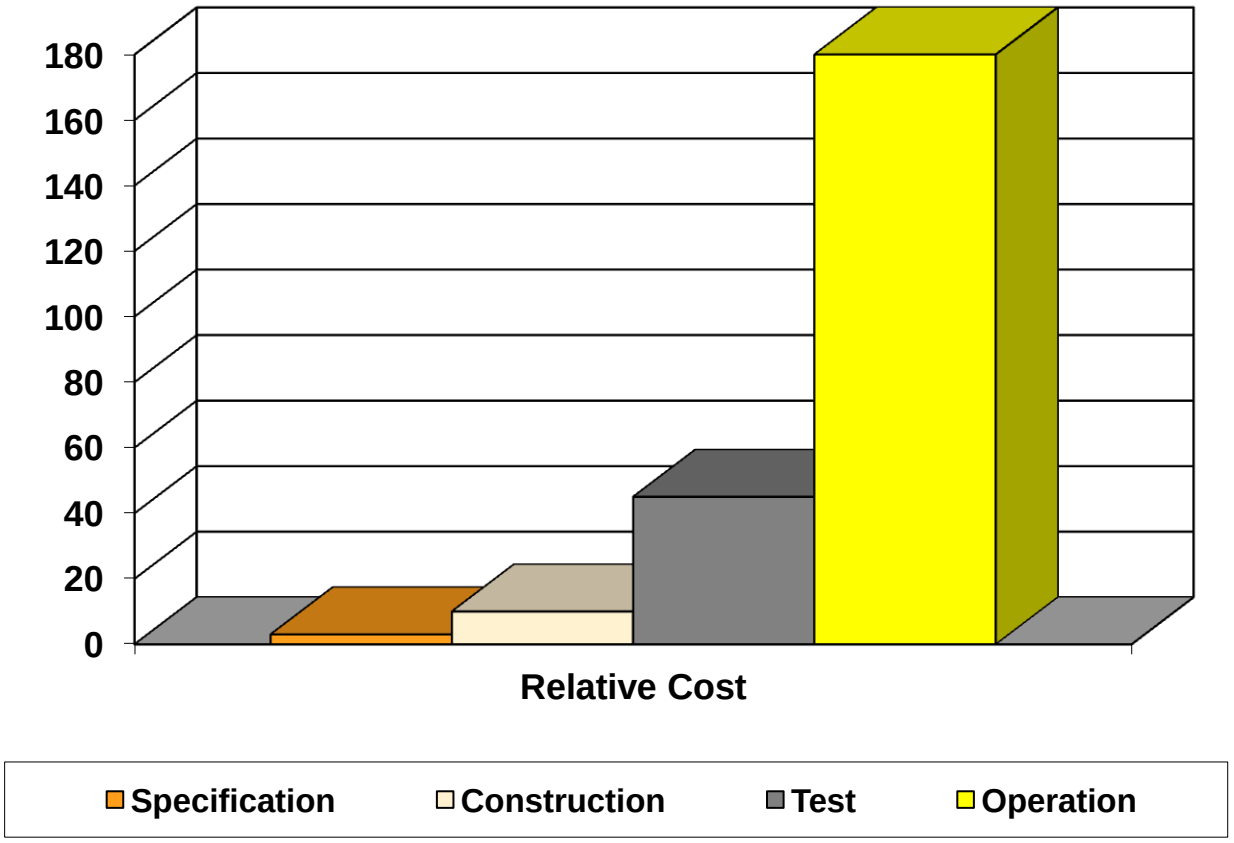
Sensibility
Project Planning

The most common mistakes are made where the remedy is very cheap, namely in the definition and design

These errors remainusually undetected in the early stages

Only in later phases they are discovered at considerable expense and eliminated

Sensibility
Costs for Troubleshooting



(Quelle: Balzert Softwaretechnik II, S. 287)

Sensibility
A Case-Study

Reasons for any delay / failure of Software projects (% of all projects):

- Project specifications are not fully described (51%)
- Poor planning and assessment of the project sequence (48%)
- inadequate / no project management (42%)

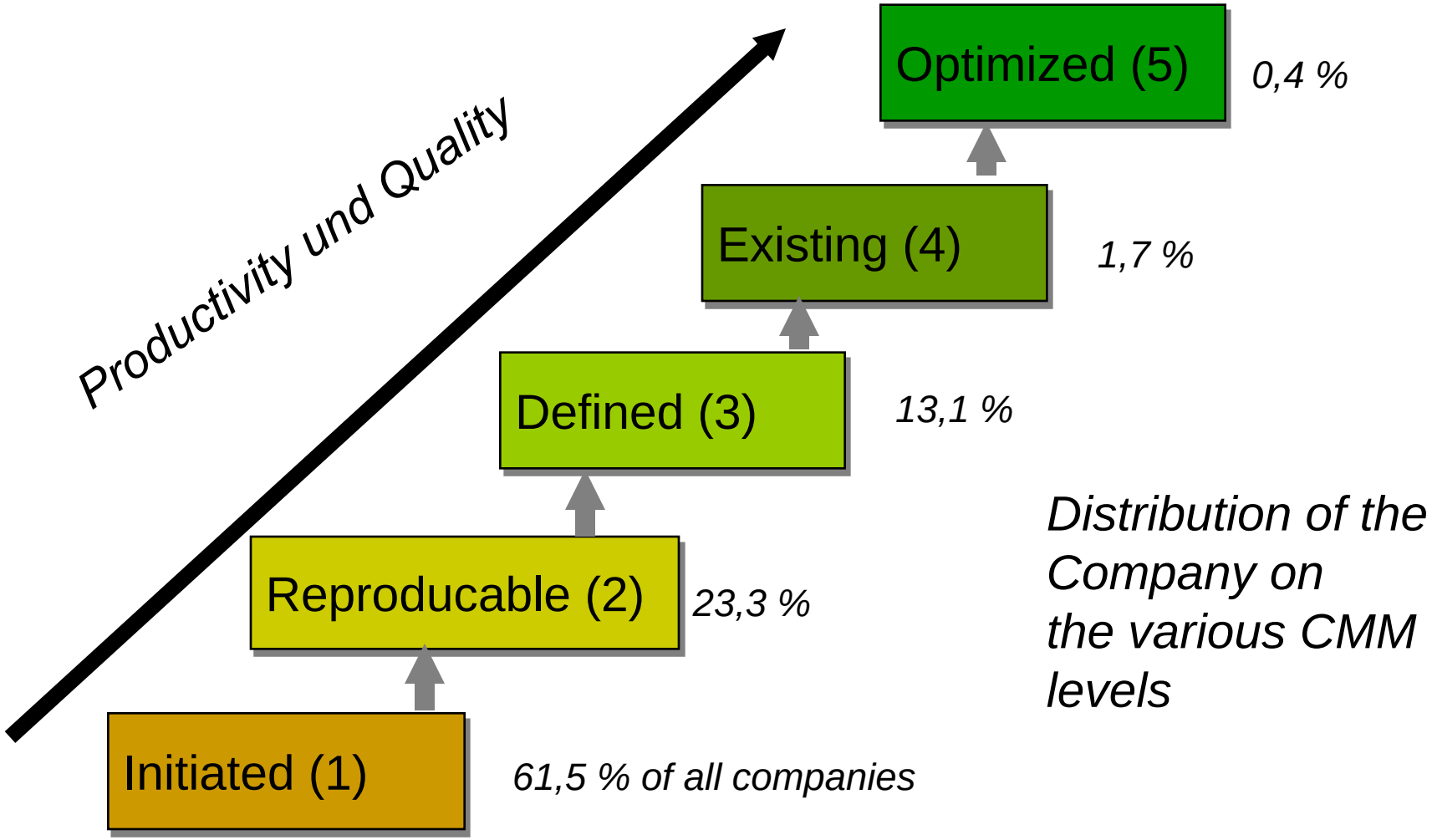
The reasons for this are:

- only 30% of all respondents describe all essential characteristics of the system
- Not sufficiently defined approach of the model hamper the ability to plan and forecast
- The interdisciplinary project structures are insufficiently taken into account

Problem areas:

- Project management of cross-company and multidisciplinary development processes
- Full process and disciplinary comprehensive product description

Sensibility
Capability Maturity Model (CMM)



Distribution of the Company on the various CMM levels

Sensibility
Capability Maturity Model (CMM)

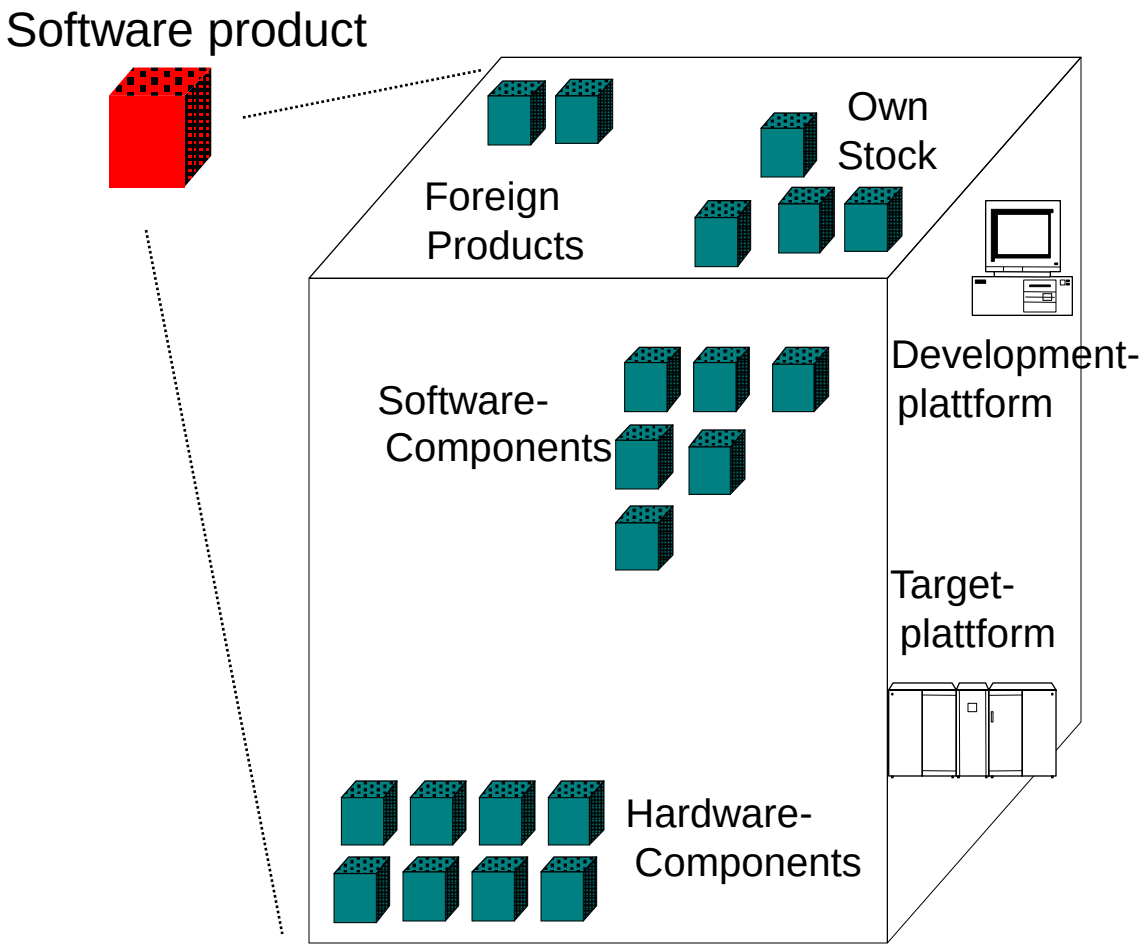
Level 1:

- No list software parts
- No quality assurance
- No functioning project management
- No functional supplier management
- Requirements unclear and incomplete

Level 2:

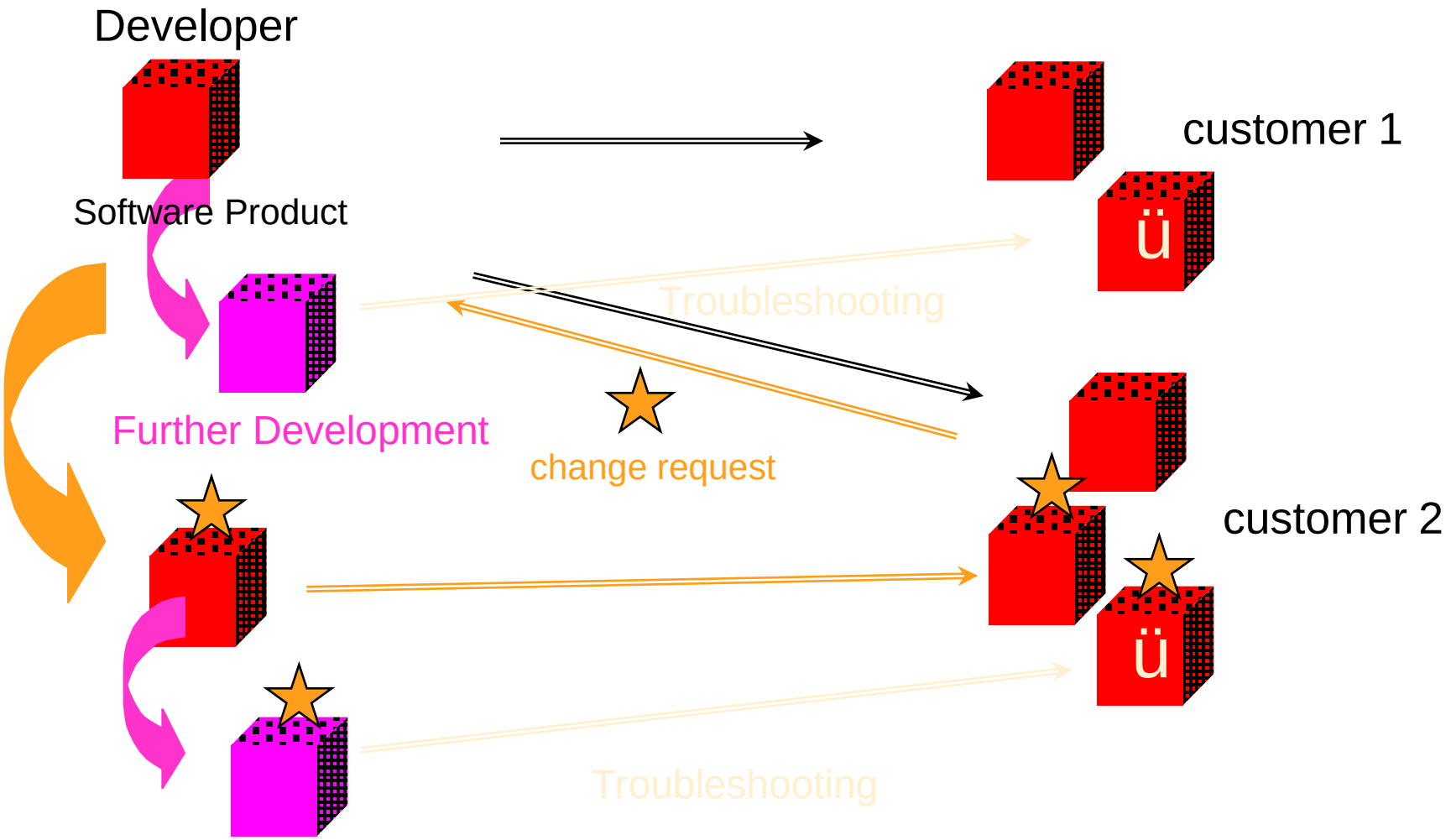
- A similar project can be properly planned and implemented on time

Sensibility
COMPONENTS OF A SOFTWARE PRODUCT



Quelle: sd&m

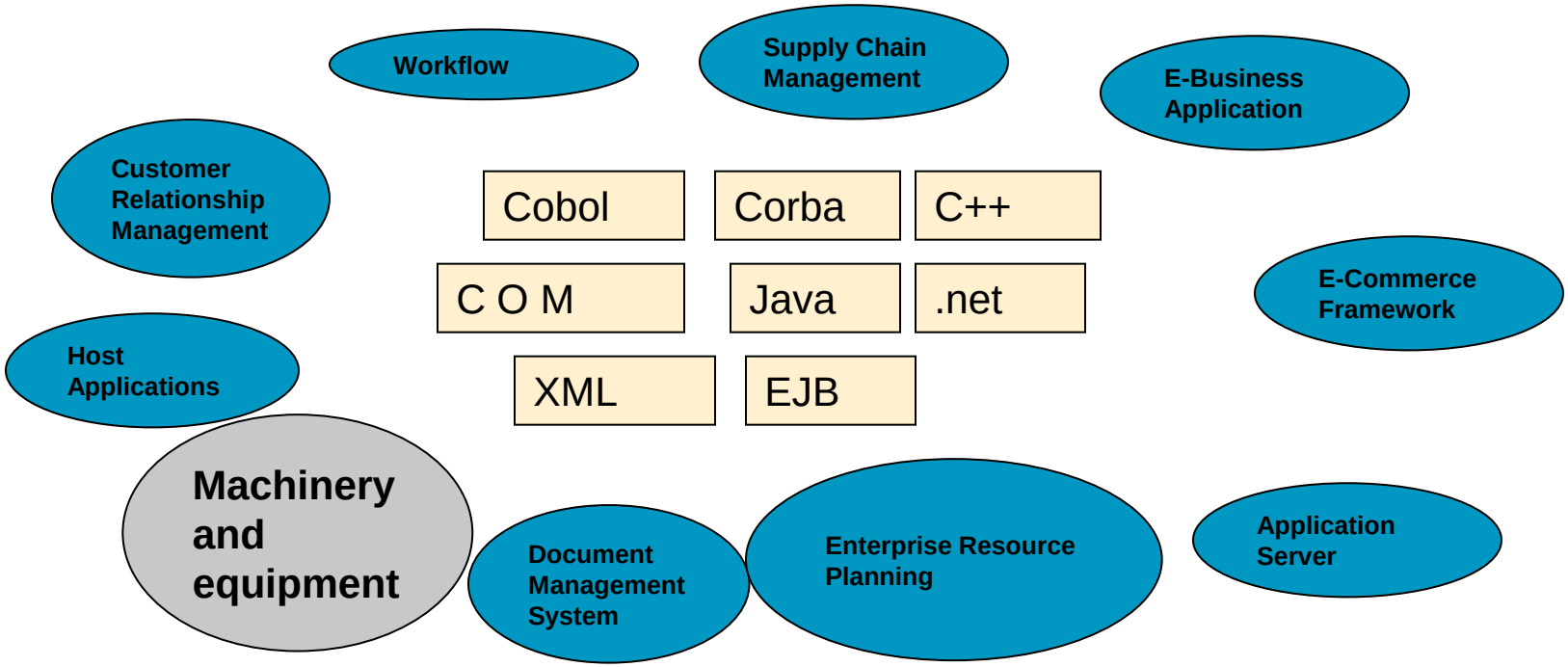
Sensibility
Even small projects are complex



Quelle: sd&m

Sensibility
The complexity problem

New technologies and new requirements increase the heterogeneity of application and system architectures and thus the complexity of the software



Examples of software errors

Ariane 5

Explosion of the Ariane 5

On 4 June 1996, the ESA launched an unmanned rocket with four satellites on board French Guyana. Ariane 5 exploded after 40 seconds of the start. The loss for rocket and satellite was about 500 million dollars. Development costs about 7 billion dollars.

Cause of the crash:

The on-board computer crashed 36.7 seconds after launch when he tried to convert the value of the horizontal velocity of 64 bit floating number number into 16-bit signed integer: - + b1 b2 ... b15. The corresponding figure was greater than 2 ^ {15} = 32768 and produced an overflow. The steering system broke down and passed control to a second identical unit. Self-destruction was triggered because the engines were threatening to burst.

Ariane 5

Comments

- The software came from the Ariane 4, but the Ariane 5 flew faster!
- The software was unnecessary for the actual flight and only served to launch preparations. In order to allow a possible restart in the event of a short interruption of the countdown, the program remained active for 40 seconds during the flight.
- The backup computer used exactly the same program.
- The conversion was not secured because they believed that the number could never be so great.

Loss of the Mars Climate Orbiter

Start of the Mars Climate Orbiter NASA on 11 Dec 98

Objectives:

Achieving Mars - orbit support of the Mars Polar Lander (Start 01 Mar 99) mapping of the entire surface. Then three years active for any other Mars landings

Loss of the Mars Climate Orbiter

The loss and its causes:

Actuated orbit on 23rd Sept 99: 170 km lower than planned. Crash or rebound!

Reason: Two groups of NASA are involved in the project. One calculated in meters, the other in inches, feet ... Wrong control statement - Incorrect path - loss Mars Polar Lander was supposed to land on 3rd Dec 99

Preprogrammed crash

Banana Software: Let the software come to maturity at the customers site

'It's not a bug, it's a feature'

The Loss !!!!!

The annual loss due to software problems in Germany is estimated to be about 84.4 billion euros.

For the correction of the errors, an estimate of 14.4 billion euros will be spent in the IT departments of the companies concerned.

- The actual damage, so the loss of productivity in the departments that work with the software, amounts to approximately 70 billion euros.

A Huge task!! because..

The complexity of software is doubling every 18 months.

Software engineering

What is SE ?

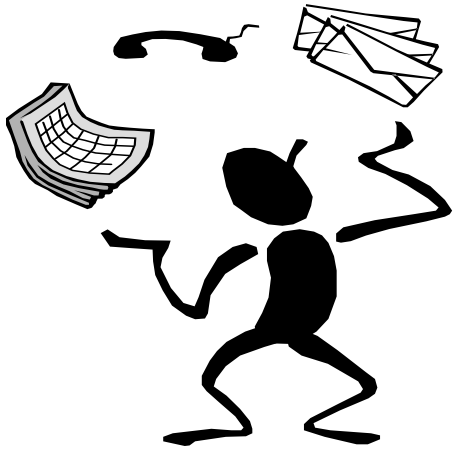
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Description

Software Engineering includes the following aspects :

- It is an engineering activity that deals with the technical problems of software and its application. For this purpose, there are methods and tools for the conception, design, modeling, quality control and maintenance.
- The software developer with the help of scientific knowledge and the tools of information processing sets up an idea to a real product for which there is a need in the market.
- The software developer is a specialist in the design science, production engineering, quality assurance and project organization for software.
- The software is provided according to rules of existing tools, finance, computer hardware and qualified personnel and in a time frame covered a reliable product.

Early stages of the project



Building a house without precise planning?

In this idea, no one would probably come!

- Precise planning for development of software?
- In many industries still very exotic!

Software engineering

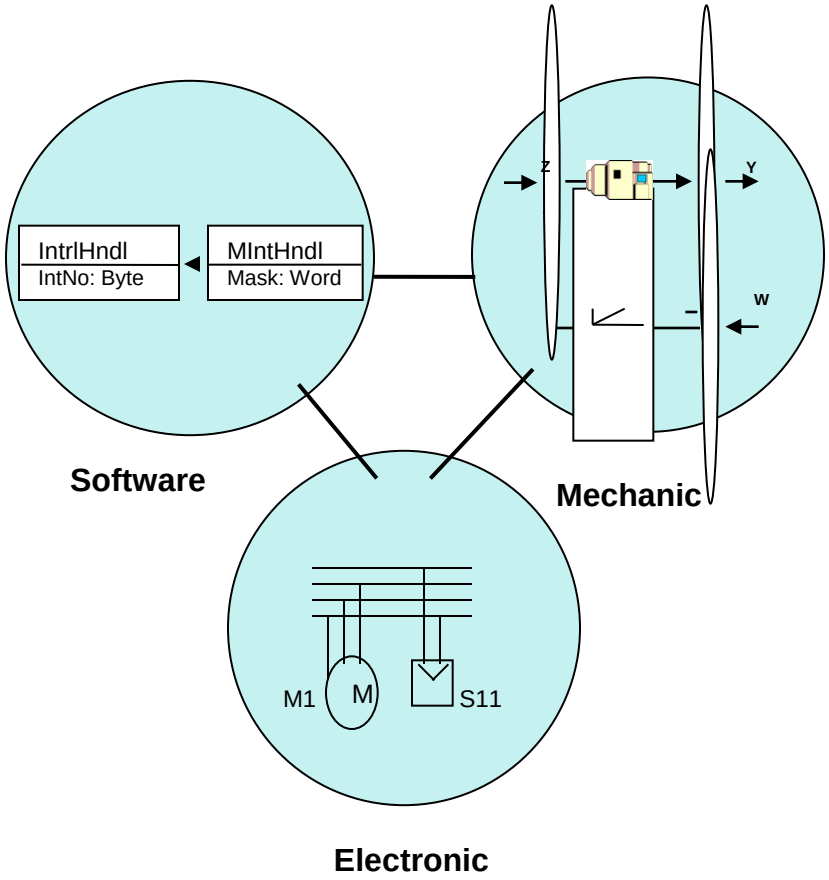
Engineering (ingenium) ~ ingenious
Optimal combination of old with new

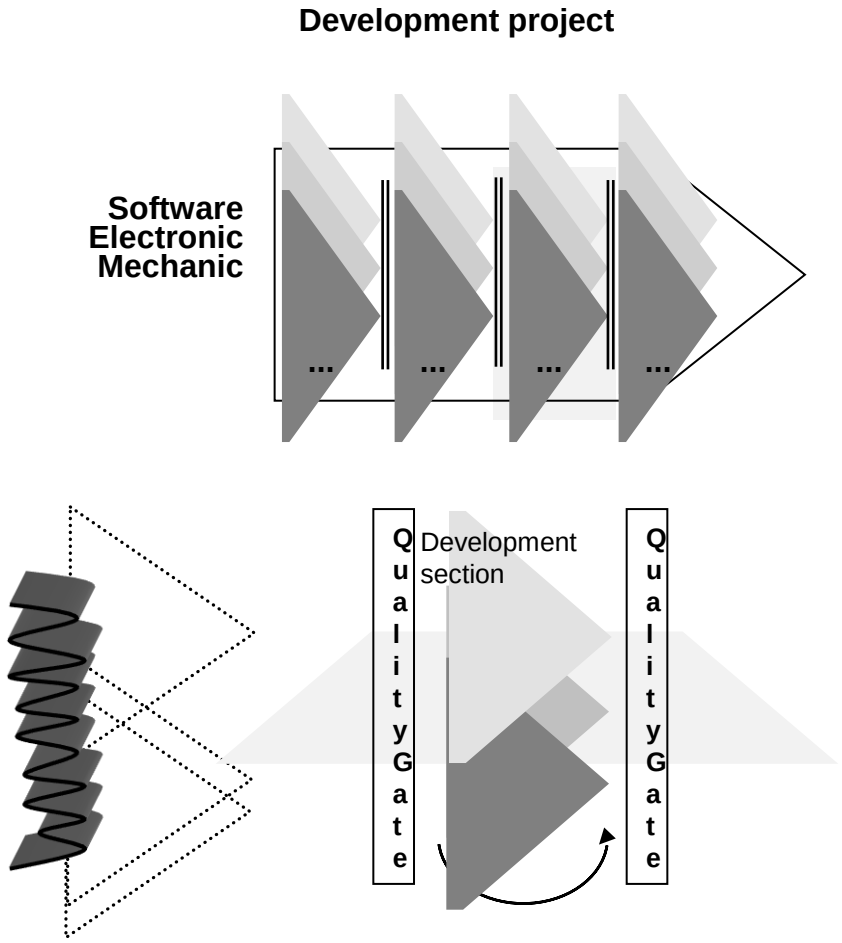
American Definition:
the planing designing, construction or management of machinery, roads, bridges, buildings etc.

Successful projects will be achieved by

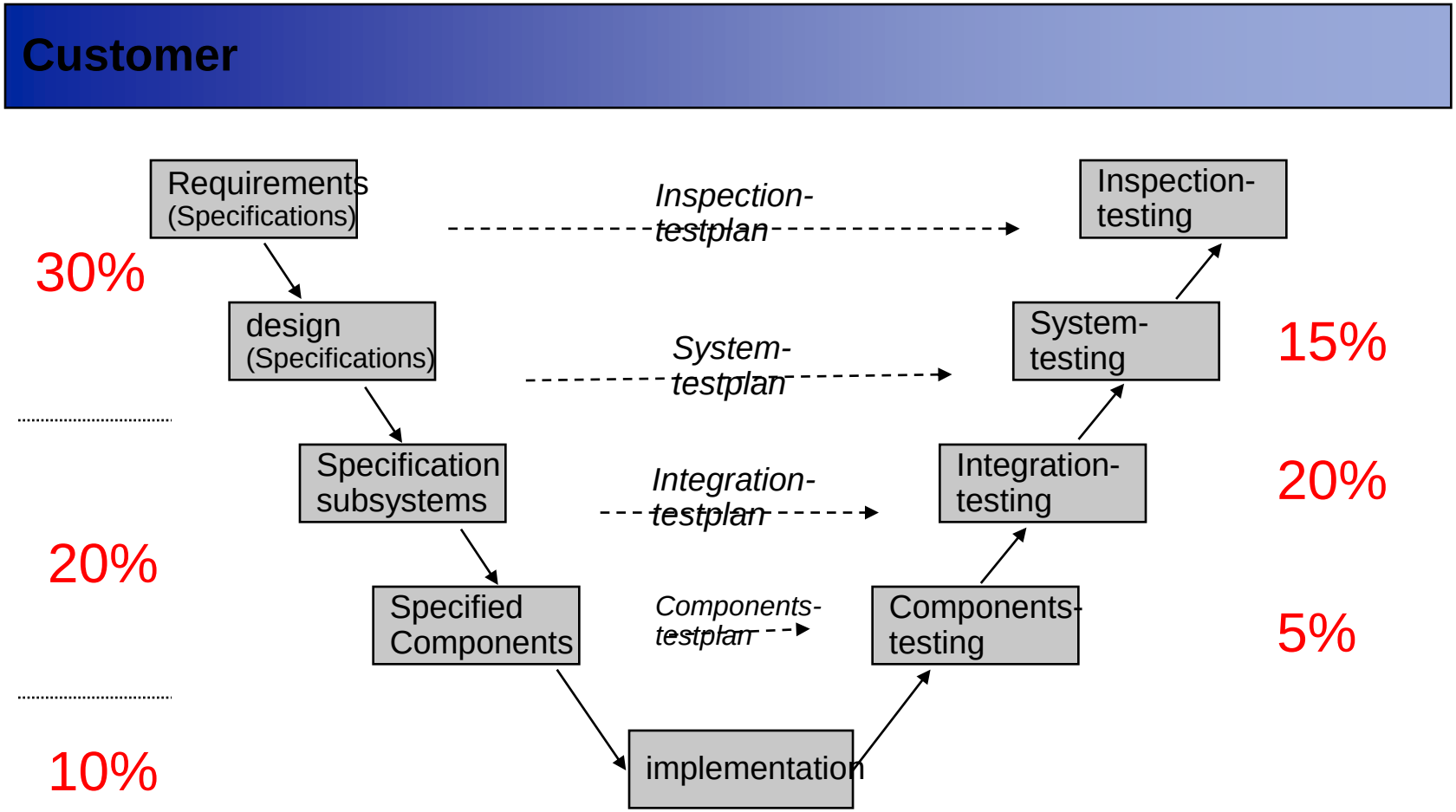
- Skilled guidance and organization
- Inclusion of "good" techniques

Find a uniform view



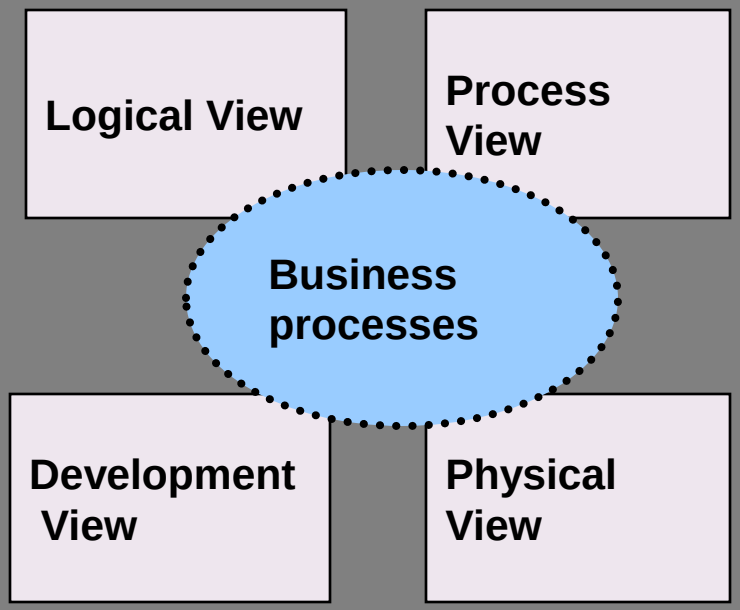


Time spent (large project): approx. 500000 (lines of code)

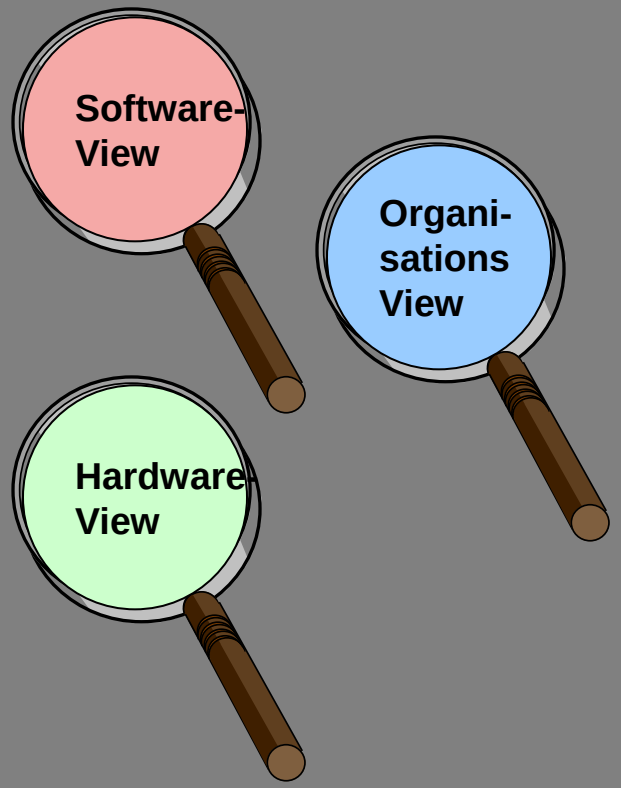


Quell: itq

Architecture - Element of engineering - view for system architectures -



or



Philippe Kruchten´s 4 + 1 Architecture

One of the possible definitions

To summarize software engineering is defined as the science of systematic development of software, starting with defining the requirements to acceptance of the finished product and the subsequent maintenance phase.

It proposes established approaches for specific tasks, often combined with new technologies that are tested prior to implementation on the applicability.

The central means for documenting software engineering results are UML diagrams.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Note

„For the development of large software products with millions of lines of code you need other procedures as for solving smaller exercises. "

"Software engineering is related to theoretical computer science, such as electrical engineering to physics."

Requirements Analysis

Requirements Analysis

The requirements analysis is a very important phase in the course of a project.
Actually, even the most important!
Here the basic foundations for the later stages are provided.

- Problem areas
- Different views
- Inaccurate requirements
- Build systems today to meet requirements of tomorrow
- Language barriers
- Dynamic development of requirements moment during the project
- Incorrect interlocutor
- Lack of target definition…

Results

- Faulty software
- Developed on the past requirements
- Time and costs not met
- Unsuitable for the future

Requirements analysis
Request channels

A requirement is a statement about a settled property or service to be provided a system and to be included processes, and people.

Requirements may be introduced through different channels. The most important are:

- Buyer / customer
- Management
- System users
- Service
- Marketing / Sales
- Developer
- Accountants and auditors

To perform a comprehensive analysis of requirements, Many of these channels should be used.

Functional requirements

Functional requirements describe in detail the functionality a system must have. This functionality can be prescribed by a solution flow, workflow or technological process.

Technical aids or tools are used to meet these requirements, such as, eg operating system, database, or hardware platform, are however no functional requirements .

Data
- Structure
- Use
- Transmission
- …

Behaviour
- Visible dynamic system behavior
- Functional interaction
- …

Functions
- Input / output
- Processing
- …

Error
- Behavior
- Expenditure
- …

Non-functional requirements

Non-functional requirements describe the boundary conditions which must be followed to the system itself and the creation and operation of the system. These include

- Standards
- Languages
- Documentation
- Operating Systems
- Technologies,
- Performance requirements (performance, throughput, response time)
- Reliability
- Modifiability
- Usability / ergonomics
- Reusability
- Expandability

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirements analysis
Classification according to priority

The requirements should be prioritized to distinguish between "important" and necessarily implemented requirements of and "less important" requirements.

In determining the priority of different criteria play a role. The focus is always the question:

- How critical is the non-implementation of the requirement?

Examples of such criteria are :

- What is the legal liability, the requirement (must-, should-, request)?
- Which groups of users are affected?
- Are people in danger?
- Threatens financial or reputational damage?
- Is the implementation basis or prerequisite for the implementation of other requirements?
- Can the implementation within this project or another be reused?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirements analysis
Classification according to criticality

Start the system (in a safe state)

Initial values for variables in safety-critical systems

Treatment manual intervention at all stages

Behavior in an offline state

Time-out requirements

Behavior due to unexpected inputs

Processing rules for sensor information, particularly when sensors are partially failed.

Rules for consistency, range and plausibility check of the input data (manual, sensor, memory).

Behavior and expenditure under heavy system load

Alarm management for "simultaneous" occurrence of critical conditions

Treatment of stale data

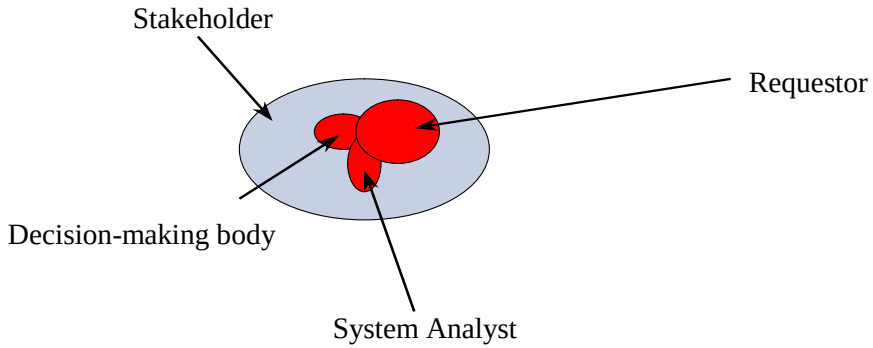
Reversal of commands for actuators

Fail-safe system state

Error detection and replacement of the system from a fault condition out

Requirements analysis
The roles in the process of requirements specification

Requester: Person (s) who cause a change or provide ideas / contract for the new development.
Systems Analyst: Performers person (s), who takes as an order receiver implementation of development activities (accept).
Decision-making body: Person (s) with budget responsibility, who are on the (usually economic) feasibility of a project. This committee decides on the basis of documents provided on the continuation and corrections of a development project.
Stakeholders: Are all in some way involved in the project. Thus, the stakeholders form a superset of all other previously mentioned include roles. They include the customer, the end user, marketing, systems analyst, developer.



Requirements analysis
basic concept

Now it comes to the clarification of the task or to the elaboration of the common understanding of all stakeholders. In particular, the conditions should be created, then the technical solution of the system are to be implemented to model and create the system specification.

Continue to be accurate in technical feasibility considerations, and the technical solution. This will also involve the development of coarse costs and possible implementation dates. All of these results are presented in the general concept that is making the initial document in the project.

There are to answer the following questions:

Those who read the rough draft?

What is the exact problem?

How far should the statements go to technical feasibility or technical realization?

How is a cost / benefit assessment look like?

What are possible implementation dates and costs from?

Who wants the project, who does not?

Who is funding the project?

Requirements analysis basic concept (specifications)

As a result of this phase, the basic concept arises in the form of a document (text, graphics, charts, etc.) whose extent depends very strongly on the problem to be solved and the project.

The basic concept serves as a basis for the further steps of the project and is the basis of the system specification.

Content of basic concept

- Listing of the principal requirements of the developed system (for the most part very un detailed) - this can be professional, technical, organizational, qualitative and non-functional nature
- A description of the problem to be solved at the highest level of abstraction
- General workflows, processes and boundary conditions
- Statements on the technical feasibility and implementation
- Results from preliminary studies and laboratory tests
- Coarse statements of project cost and implementation dates
- Might also looking statements, which should not be achieved with the system (exclusion criteria)
- The basic concept of which lies in the language of the client, with graphics should complement the text makes sense.

Requirements analysis
Requirements Management (Feature Request)

"Through requirements management, a defined procedure is established, which often allows to master complex projects demand changes for the vast number of requirements. This can only be handled systematically with the introduction of requirements management techniques.

The focus is on the management of changing requirements during the project, the traceability and the hallmark of dependencies between the individual requirements. In this case, it is necessary to develop a process that collects and checks changes and processes it. Here are relationships noted that exist between requirements, because that would lead to consistency problems.

In traditional process models, such as the V-Model are applicable requirements at the outset and then do not normally change.

It makes sense for a model that all main activities goes through several iterations and thus the whole system is gradually evolved (incremental process). This iterative procedure is readily with classical models combined.

Requirements analysis
Quality criteria for requirements

Complete
- Cover the entire functionality

Correct
- correct reproduction of the functionality required by Stakeholder32

Contradiction-free
- consistently over other requirements and internally consistent

Clear
- Exclusion of ambiguous interpretations of a request

Testable
- Verification of the implemented requirements by test cases

Understandable
- Intelligence for all stakeholders

Feasible, Viable
- Implementation under technological and system-specific aspects

Necessary
- Usability of the requirement resulting functionality

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Requirements analysis
Quality criteria for requirements

Trackable

- uniquely identifying and referencing the request

Evaluated

- Option of prioritizing the requirements

Reusable

- Generality, can also be used for other projects

Economical

- "As much as necessary, as little as possible"