

**ruynk**

# **Zertifizierungswissen**

## **Requirement Engineering**

### **REQB - Syllabus überarbeitet**

**Englisch**

**...mit Kommentaren in Deutscher Sprache**

**1. Version: Okt. 2012**

**Aktuelle Version: Juni 2017**

**Bearbeitet: Ruy Kuhlmann**

# **Syllabus**

## **Requirements Engineering**

### **Foundation Level**

**Auseinander genommenen Text und wieder zusammengeschnürt zu einer, nach meinem besten Wissen und Können, besseren und logischeren Zusammenfassung.**

Dadurch hängen die Themen besser zusammen.

# Inhalt

1. Definitionen (DE: Begriffsdefinitionen).....	6
2. Standards.....	7
3. Requirement.....	8
Purpose.....	8
Classification.....	8
Basic types.....	8
Life Cycle of REQ.....	8
Problems of REQ.....	9
Quality of REQ.....	9
REQ Spec must be.....	9
REQ Management.....	10
REQ Development.....	10
Verification of REQs.....	10
Techniques.....	10
4. On Projects.....	11
Project Vision:.....	11
Influence Factors on Vision:.....	11
Identifying Stakeholders:.....	12
About the Customer:.....	12
Contract:.....	12
Agreement:.....	12
Contract should include:.....	13
Errors on REQ ENG:.....	13
5. On Risks.....	13
Two Types of Risks:.....	13
Pot Risks Treatment:.....	14
Risk Management Activities:.....	14
Risk Management Plan:.....	14
5.1. Failure Mode and Effects Analysis (FMEA):.....	15
6. REQ.ENG.....	15
REQ ENG Process - Activities:.....	15
Negative Influences on REQ ENG:.....	15
REQ ENG contributes to:.....	16
Roles within REQ ENG:.....	16
Stakeholders:.....	17
Typical Stakeholder:.....	17
Methods of REQ ENG:.....	17
Tasks of REQ ENG:.....	17
Soft Skills of a Prof for REQ ENG:.....	18
6.0. Techniques for identifyng REQs:.....	18
Purpose:.....	18
Techniques:.....	18
REQ Identification from existing Docs:.....	19
Workshop goals:.....	19
Functional REQs:.....	19
Non Functional REQs:.....	20
Standard Content of a REQ DOC (for a REQ DOC see IEEE 830):.....	20
6.A. Description of REQs (Clear and Accurate!).....	20
Steps for the construction of REQs.....	21
6.1. SPEC of REQ (REQ Specification = Explicit Set Of REQs (to be satisfied by material, product or service) ).....	21
6.2. Solution SPEC.....	21
Purpose:.....	22

On Fomalization:.....	22
Procedure of Solution Specification:.....	22
Activities:.....	22
--- REQ ENG Process ---.....	23
6.3. REQ Analysis / OO Analysis (DE: Der beruehmt-beruechtigt Kap 7 hierunter...)	23
Goal of REQ ANALYSIS:.....	23
Different Perspectives of the System:.....	23
Different Models (DE: Angelehnt an die UML...):.....	24
Basic types of Models.....	24
OO Analysis:.....	24
Behavioral Diagrams:.....	25
Structural Diagrams:.....	25
SysML: System Modeling Language.....	25
Procedure of REQ Analysis:.....	25
6.4. Estimates.....	26
Influences on Development-Costs:.....	26
Cost Estimation:.....	26
1. By Analogy.....	26
2. Algorithmic Procedure.....	27
6.5. Prioritization = Establish relative Importance of REQ.....	27
Procedure:.....	28
Prioritization Scale:.....	28
Evolution of REQs:.....	28
Reasons for REQ Development + Changes:.....	28
6.6. SignOff = Agreeing in REQ is called SIGNOFF.....	28
REQ Signoff signed by:.....	28
Purpose:.....	29
Advantages of REQ Agreement:.....	29
6.7. Traceability (see Change Management too).....	29
Goals of Traceability:.....	29
Types of Traceability:.....	29
6.8. What Influence REQ ENG.....	29
7. Models for Development Processes.....	30
7.1. SE Process-Model.....	30
7.2 Product Life Cycle.....	30
7.3. Planing.....	30
7.4. Development.....	30
7.5. V-MODEL: .....	31
7.6. Rational Unified Process (RUP) (From IBM):.....	31
7.7. Agile Approaches.....	31
REQ DEV:.....	31
REQ DOC are:.....	31
7.7.1. Extreme Programming.....	32
7.7.2. SCRUM (DE: Gedraenge bedeutet das auf deutsch...)	32
7.7.3. Maturity Model.....	32
8. Change Management.....	33
Source of Change:.....	33
Change Management is the Process of:.....	34
Purpose:.....	34
Activities:.....	34
Describes:.....	35
Basis for Decision:.....	35
Stakeholder on the CCB:.....	35
Impact of Changes on the Project.....	35
9. On Quality Assurance.....	36
9.1. On the quality of REQs.....	36
Quality of REQs:.....	36

Tools and techniques for Q.-Improvement + QA on REQs.....	36
Elements to improve Quality of REQ SPEC:.....	37
9.2. About using REQ for QA.....	37
REQ ENG and Testing:.....	37
Acceptance Criteria = Success Criteria:.....	37
REQ ENG provides "Test Basis":.....	37
REQ an the Test-Process:.....	38
10. Metrics (DE: Innerhalb Projekte).....	38
Metrics:.....	38
11. Tools (EN: last and almost least).....	39
Tools may support:.....	39
Advantages of Using Tools:.....	39
Categories of Tools:.....	39
Analysis before Selecting a Tool:.....	40
Costs to be considered:.....	40

## Copyright

Dieses Dokument ist Eigentum von R. Kuhlmann. Es darf auch ohne meine Genehmigung vervielfältigt, verwertet, und auch dritten Personen zugänglich gemacht werden.

Die hier enthaltene Daten wurden vom REQB Syllabus (Version 1.3 vom Jahre 2011) destilliert und neu organisiert.

Beim Erwähnen bzw Weiterverwenden bitte die Quelle (dieses Dokument hier) erwähnen.

# 1. Definitionen (DE: Begriffsdefinitionen)

**Requirement (REQ):** Condition or Capability to solve a problem OR achieve a objective. MUST be meet to satisfy a Contract, Specification (Spec), etc.

**REQ Engineering (Eng):** Is a subset of SW Engineering (SE)

**REQ ENG Process:** Identification, Structuring and Managing of REQs

**Basic roles:** Client or Customer. Contractor, Supplier or Vendor

**REQ Management:** Is a working Frameeework for REQ.ENG.

**Risk (1) ISO 31'000:** Effect of uncertainty on objectives (pos and neg)

**Risk (2) ISTQB:** Chance of an event, hazard, threat resulting in undesirable consequences or potential (pot) problems.

**Solution (Sol):** Implementation of REQ

**Fault:** Flaw in Component OR System (Sys) (!= Defect)

**Defect:** During execution, may cause a failure of the Component or System. (!= Fault)

**Functional:** Function - Behaviour = WHAT the System does

**Not Functional:** also called Quality attributes = HOW the system does it (Example: testability, serviceability, (DE: Im Skript steht auch "tools". Das ist natuerlich Quatsch: ein Tool ist ein Werkzeug, nicht eine nicht funktionale Eigenschaft)

**Priority:** Evaluation of the importance/urgency of a REQ

**Commitment:** Degree of obligation of meeting a REQ (DE: Achtung, bei den Agilen Methoden wird erwartet, dass die agilen Entwickler sich "verpflichten" vor jedem agilen "Sprint" oder so: Es ist nicht undenkbar, dass die Verfasser des Syllabus irgendwie darauf Bezug nehmen wollten und dannach es vergessen haben, dies zu verdeutlichen)

**Validation (Val):** Process of confirmation that something fulfills customers REQ (What the customer needs) (DE: Passt zur Kunde)

**Verification (Verif):** Provides Checkpoints, Focus on incremental Confirmation of the implementation of REQs (DE: Dass es funktioniert)

**Techniques for Val and Verif:**

- Reviews

- Audits
- Checklists
- Testing

**Process Model:** A Process Model is a description of a development process. Define standard procedures for development.

**MoSCoW:** Must - Should - Could - Would

**Metric:** Metric is a measurement scale and the method used for measurement (ISO 14'598)

## 2. Standards

**ISO 9'000:** REQs of a QMS (Quality Management System)

**ISO 9'126 (Now ISO/IEC 25'000):** Quality Model w/ 6 Categories

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

**ISO 12'207** = SW LCP (Life Cycle Process)

**ISO 15'288** = Syst. LCP

**ISO/IEC 15'504** = Software Process Improvement and Capability Determination (SPICE + CMMI)

**SWEBOK (SW ENG. BOK -Body-of-Knowledge-)** = ISO techn. Report 19'759

**IEEE 610** = SE Glossary

**IEEE 830** = SW REQ Spec

**IEEE 1'233** = Syst. REQ Spec - Syst. Func.  
**IEEE 1'362** = Syst. Def. - Syst. Perf.

## 3. Requirement

**NOTE:** REQ may evolve throughout the project life cycle.

### Purpose

REQs are Foundation for Planning, Executing, Monitoring. Are component of Agreement, Order, Project Plan. Setting of System boundaries, Scope of delivery.

### Classification

1. Process REQ (related to Development and Delivery process) } User or Customer (EXTERNAL)
2. Product REQ (Functional and not functional Prod REQ) } Development Team (INTERNAL) => USER != CUSTOMER

### Basic types

- Customer
  - Desires, needs, expectations
  - Limitation on business (DE: Wieso ist das ein "Grundtyp" von REQ?)
- Solution/System
  - Spec of customer needs
- Product/Component
  - Functions, Characteristics of the solution
  - Basis for detailed Analysis and Design (A+D)

### Life Cycle of REQ

Status assigned to a REQ.

- New (proposed)
- For Review
- Approved
- Conflicted
- Implemented
- Modified
- Deleted
- Tested
- Deployed

### **Problems of REQ**

- Objectives unclear
- Communication - language => Knowledge
- Formulation vague or too formal
- Volatility (Changes!)
- Bad quality (DE: Zu allgemein, im Grunde ist alles, was ein Problem ist, "bad quality")
- Gold plating
- Insufficient user involvement
- Missing stakeholders
- Inaccurate planing
- Minimal secification

### **Quality of REQ**

REQ must be:

- Correct (function)
- Feasible (possible)
- Necessary (really needed)
- Prioritized
- Unambiguous
- Verifiable
- Singular (Granularity)
- Design-Independent ("what", not "how")

### **REQ Spec must be**

- Complete
- Consistent (no conflicts)
- Modifiable (allows changes)
- Traceable (link to source)

## REQ Management

Processes for the overall Management of REQ (DE: ja, logisch!)

### The continuous process of:

- Documenting
- Analyzing
- Tracing
- Prioritizing
- Communicating
- Agreeing on REQs
- Management of REQ Changes

## REQ Development

Collection of activities, Tasks, techniques and Tools to:

- Identify
- Analyze and validate
- Process of transforming needs into REQs

The Purpose of REQ Development ist to:

- Elicit        }
- Analyze     } 1. Customer,
- Establish   } 2. Product,
- Validate    } 3. Prod. Component REQ

(DE: Doppelt gemoppelt, ob "customer" REQ oder "prod" REQ, es sind doch alle REQs.

Einfach: Der Zweck des REQ DEV ist doch DEV of REQs, und zwar alle)

## Verification of REQs

- Verif has to be planed at beginning of a project
- Verif has to be done continuously

## Techniques

- Review
- Simulations
- Presentation

- Demonstrations

Reasons of neglecting REQs:	->	Consequences of neglecting REQ
* High time pressure	}	* REQ imprecise
* Orientation to fast results	}	* REQ ambiguous and Diff interpretations
* Fixation on Costs	} =>	* REQ contradictory and changing
* Considering only Func. REQ	}	* Missing REQs
* Misinterpretations (assuming things as known)	}	(DE: Dieses letzte wohl das Wichtigste)

## 4. On Projects

### Project Vision:

- Defining customers + Markets
- Defining objectives
- Common understanding of all stakeh
- What will change?
- Why is necessary?
- What happens after termination?
- Who profits?
- What are the costs?
- What are the risks?

### Influence Factors on Vision:

- Customers
- Strategy
- Competition
- Products
- Technologies

- Available Ressources

### **Identifying Stakeholders:**

All stakeh must be identified.

\* **Each Stakeh provide new REQs**

\* **If not all stakeh are identified -> RISK !**

- Analysis of Business process
- Determining Process + Product Owner
- Analyzing Organizational Structures
- Analyzing Market
- Dividing stakeh into groups
- Determining relationships
- Identification of potential conflicts
- Analysis of conflicts + Sources + win-win Opportunities
- Identification of risk minimizing stakeh
- Identification of stakeh perspectives

### **About the Customer:**

- Customer Needs must be satisfied
- Must be always involved
- Goal: Understand the customer
- One REQ has different Priority for different stakeh

**Project Management:** Projects fail because of Bad REQs

### **Contract:**

Describes what the customer wants

Only interests of the customer (not of other stakeh!)

### **Agreement:**

- Compliant with available ressources
- Based on estimation
- REQ provides input information for estimates

### **Contract should include:**

- Short description
- List of prioritized high level REQs
- Acceptance criteria for REQs
- List of Products (Docs, Code, Working SW, etc)
- Deadlines for Development + Delivery
- Other needs + Expectations (preferred technology, ... (DE: Ist das doch nicht auch ein REQ?))

### **Errors on REQ ENG:**

- Unclear REQ and Responsibilities
- Changes
- Unstable Prod and Design Basis
- Gap between expectations and Content
- Insufficient customer involvement
- Milestones not reachable
- Imprecise Expenses
- Imprecise estimate of impact of REQ Changes
- Lack of Traceability

(\*\*\* Das ist sicherlich irgendwo wiederholt)

## **5. On Risks**

### **Process of**

- Identification
- Assessment
- Prioritization
- Planing Reaction
- Resolving
- Monitoring

### **Risks**

Allows Identifying Potential factors

### **Two Types of Risks:**

#### **1. Project:**

- Organisational (Staff, political, stakeh)
- Technical
  - Def of right REQ
  - Existing Constraints
  - Not ready on time
  - Low quality of Design, configuration data, test data and testing
- Supplier (Contracts, fault from 3. party)

## **2. Product:**

- Low quality of Doc
- Potential of harm of SW/HW
- Poor characteristics
- Poor data integrity
- Violation of data standards
- SW not perform functionality
- SW does not satisfy stakeh needs
- Bad quality (DE: Was genau? Das ist super allgemein...)

## **Pot Risks Treatment:**

- Avoidance
- Reduction
- Sharing
- Retention

## **Risk Management Activities:**

- Identification
- Analysis
- Mitigation

## **Risk Management Plan:**

Created before and after the project plan.  
Should provide effective security controls

### **Includes:**

- List of Risks
- Probability of occurrence
- Severity of impact (Costs!)
- Mitigation strategies
- Risk assessment Matrix

### **5.1. Failure Mode and Effects Analysis (FMEA):**

- To prioritize potential failures according to severity
- According to frequency of occurrence, and
- According to how easily can be detected

FMEA Documents current Knowledge

Oft is FMEA used during the decision stage to avoid future failures

In later phases then, it can be used for process control

RESULTS are actions preventing or reducing severity or likerlihood of failures

#### **Implementation steps:**

- Identification of severity
- Identification of occurrence
- Techniques for detection

## **6. REQ.ENG.**

### **REQ ENG Process - Activities:**

- Identification
- Analysis
- Specification
- Agreement on REQs
- Changes on REQs
- Validation and QA

### **Negative Influences on REQ ENG:**

#### **INTERN:**

- No Knowledge on users
- Ineffective REQ ENG Approach
- Insufficient personell experience

#### **EXTERN:**

- Lack of communication

- Unclear/changing Business-Objectives => Unstable REQs
- No Knowledge on SW-Development Process
- No involvement of users/stakeh.

### **REQ ENG contributes to:**

1. Project conception (Identification of customers needs)
2. Contract negotiations (
  - Determining of initial scope
  - Determining costs of development
  - Agreeing on priorities)
3. Project definition (
  - Roles, Tasks, Activities
  - Business decision
  - Architecture Design
  - Test process deliverables)
4. Process Execution (
  - Basis for REQ DEV
  - REQs Verif and Val
  - Review Plans)

### **Roles within REQ ENG:**

#### **1. REQ Manager:**

- Documenting
- Analyzing
- Tracing
- Prioritizing
- Agreeing (...on REQs)
- Controlling changes
- Communicating to relevant stakeh

#### **2. REQ Developer (technical oriented person)**

- Elicitation
- Analysis
- Prioritizing

...of REQs

**Stakeholders:**

- Actively involved in the project
- Have conflicts of interest with each other (< Contradictory REQs)

**Categories:**

- Customers
- End users
- Managers
- Organisers
- Development engineers
- Regulators
- Domain Experts

**Typical Stakeholder:**

- Customer
- End User
- Product Manager
- System Analyst
- Business Analyst
- Business Representatives
- Marketing
- Sales
- SW-Developer
- QA
- Technical Experts
- Change Manager

**Methods of REQ ENG:**

(The customer at the center!)

- Customer oriented A+D
- Prototyping Approach
- Using Demonstrations

**Tasks of REQ ENG:**

- Analysis of Business Processes

- Identif + Analysis of REQs
- Structuring + Modeling REQs
- QA on REQs + SPECS
- Creation of the REQ Spec
- Risk Analysis (in the context of REQs)
- Change Management of REQs
- Agreement on REQs w/ Stakeh

(DE: Im Grunde genommen ist es doch ein PM on REQs)

### **Soft Skills of a Prof for REQ ENG:**

- Moderation
- Ability to convince
- Ability to communicate
- Analytical + clear thinking
- Methodological competence
- Self confident maner
- Language
- Precision
- Act on a structured way
- Stress resistance

## **6.0. Techniques for identifyng REQs:**

### **Purpose:**

- Identifying all Functions, Limitations, Expectations
- Orientating REQs to Project Vision
- Detailing high level REQs
- Describing Functions + Services (clearly (DE: logisch!))
- Excluding Functions + Features

### **Techniques:**

- Quastionnaires
- Interviews
- Self-Recording
- Customer Representatives
- Study of existing documents
- Reuse (only applicable to certain projects)
- Brainstorming

- Field observation
- Apprenticing
- Workshops

### **Observations to the techniques:**

**Self recording:** The user describes also changes and desires and needs

**Customer Representatives:** Main rule of Agile Methods

**Reuse (1):** Allows starting implementation earlier

**Reuse (2):** Can demand costly Change Management

**Workshop:** Meeting focused on specific topics

### **REQ Identification from existing Docs:**

- Process Models and Maps
- Process Descriptions
- Organizational Charts
- Product Specs
- Procedures
- Standard and Instructions

### **Workshop goals:**

- Identifying REQs
- Discovering hidden REQs
- Developing better REQs
- Prioritizing REQs
- Reaching consensus on REQs
- Reviewing results
- Resolving issues

**Problem:** Consensus is difficult to reach!

### **Functional REQs:**

What the system does

Specify the functions

(ISO/IEC 25'000)

- Suitability
- Accuracy
- Interoperability

- Functionality
- Compliance
- Security

### **Non Functional REQs:**

How the system does

Describe the quality

Are difficult to describe (oft not documented)

Must be clear + measurable

- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

### **Standard Content of a REQ DOC (for a REQ DOC see IEEE 830):**

- Introduction
- Secrecy clause
- Regulations
- Standards
- Stakeholders
- Purpose of the Product
- Description of the System
- Functional REQs
- Non Functional REQs
- Assumptions
- Dependencies
- Risks
- Safety REQ
- SW Quality Attributes
- Acceptance

## **6.A. Description of REQs (Clear and Accurate!)**

### **Must be**

- measurable
- unambiguous
- clear
- accurate

## **Steps for the construction of REQs**

- Determination of affected processes (Functionality, inputs and outputs)
- Classification of System Activity (Interaction w/ User, Interfaces)
- Determination of legal commitment (Clarification of legal terms)
- Refining Description of Process (of objects and Integration)
- Logical and time constraints (Boundary conditions)

## **6.1. SPEC of REQ (REQ Specification = Explicit Set Of REQs (to be satisfied by material, product or service) )**

- Serves to track and manage REQs
- Specified in a structured way
- REQ SPEC also called SPEC of REQ = REQs + SOLUTION

### **Contains:**

- Customer needs
- Limitation
- Acceptance Criteria

### **User Stories:**

Agile SW development Methodology

### **Intention of User-Stories:**

Respond faster to rapidly changing REQs

### **Composed of:**

- Written description
- Details of the story
- Tests: Completion criteria

## **6.2. Solution SPEC**

also called **FUNCTIONAL SPECIFICATIONS**

or        **SYSTEM REQ SPECIFICATION**  
or        **SW REQ SPECIFICATION**

Describe the Solution Area  
Focuses on Functionality

**Purpose:**

- Basis for common understanding of scope + functionality
- Ensuring team consensus (before writing source code, manuals, preparing data, testing)
- Detailed description of functionality (Interaction) - (for development team)
- Basis for Tests (Testteam)

**On Fomalization:**

REQ SPEC may be created on diff degrees of Formalization:

- Non Formal (common language)
- Semi-Formal (structured, includes some formal notation)
- Formal (mathematical description of the SW)

**Procedure of Solution Specification:**

Activity for formalizing Results of REQ.

- Identification,    }
- Analysis and       } -> Leads to: -> REQ AGREEMENT
- Specification     }

**Activities:**

- Identification of stakeh
- Def of Vision and objectives
- REQ determination
- Structured REQ SPEC
- Description of System Environment
- Determination of the solution

- REQ Analysis
- Modeling of the Problem
- Modeling of the Sol

### **OUTPUT:**

Solution Specification as starting point for SW + HW + DB Design.  
Describes

- ... (DE: Das wohl, was man schon kennt, dazu: )
- Operational constraints
- User-Interface constraints

### **--- REQ ENG Process ---**

## **6.3. REQ Analysis / OO Analysis (DE: Der beruehmt-beruechtigt Kap 7 hierunter...)**

**ANMERKUNG:** In diesem Kapitel wird versucht Abbildungen zwischen "Model" und "Method" zu machen ohne viel Erfolg. Ausserdem wird versucht, etwas wie ein "Model" zu benutzen ohne weitere Definition bzw. Begriffsbestimmung. Vielleicht wurde versucht, einen Ansatz zu der UML zu finden, ohne es explizit erwaeht au haben?

### **Goal of REQ ANALYSIS:**

To create a solution for the implementation of REQs. (DE: Das ist der Goal des REQ SPECS doch)

Takes into account diff stakeh, conflicting REQs, relationships + dependencies among REQs.

### **Different Perspectives of the System:**

- Logical view (functional REQs)
- Process View (Communication, Interaction, not functional REQs)
- Implementation View (Components or Modules)

- Installation View (Integration, system Architecture)

### **Different Models (DE: Angelehnt an die UML...):**

1. Context Model
  - Static description of System
  - Basic Architecture
2. Functional Decomposition (Static)
  - Decomposition of System
3. Data Flow Model (Dynamic)
  - Graphics of flow of data
4. State Transition Model (Dynamic)
  - Behaviour of system in a serie of events
5. Entity Relationship Model (cause and effect)
  - Abstract, conceptual representation of data

(DE: Ich wundere mich woher hat man diese Brocken entfuehrt)

### **Basic types of Models**

(DE: Es werden zwar 3 angegeben, doch das letzte ist sinnlos: "Conceptual Model" soll fuer "technological Specs" erhalten. Techn. Specs haben jedoch keine gemeinsame Verbindungen mit etwas abstract wie "Concepts".)

#### **1. REQ Model:**

- Describe problem area
- Serves for REQ Analysis
- Serves for cost estimation
- Basis for Solution Model (!!)

(DE: Scheint mir eher eine REQ SPEC zu sein)

#### **2. Solution Model:**

- Describes solution area
- Simultan to REQ Model (!!)
- Shape of implementation of functional + not functional REQs
- Basis for System Design

(DE: Scheint mir eher eine SOL SPEC zu sein)

### **OO Analysis:**

UML (Unified Modeling Language):  
Unified Notation for Analysis + Design of Systems.

## **Behavioral Diagrams:**

- Activity Diagrams
- Use Case Diagrams
- State Machine Diagrams
- Timing Diagrams
- Sequence Diagrams (work flow)
- Communication Diagrams
- Interaction Overview Diagrams

## **Structural Diagrams:**

- Class Diagrams (logical structure)
- Composite Structure Diagrams
- Component Diagrams
- Deployment Diagrams
- Object Diagrams
- Package Diagrams

## **SysML: System Modeling Language**

- More flexible and expressive semantic (than UML)
- Easy to learn + Apply (smaller than UML)
- Supporting models + Views
- Adds 2 new Diagrams: Requirement + Parametric
- REQ Diagram: For function, performance + interface REQ
- Parametric Diagram: Define performance + quantitative constraints

Requirement Model as Business Model presenting the Business Problem (to be solved) by the Solution Model

=>

Solution Model is more detailed and includes techn. SPEC. It is the base for Development + Test.

## **Procedure of REQ Analysis:**

- Analysis
- Description of Sol

- Cost estimate
- Prioritization

(DE: Das ist schon fast das REQ SPEC.)

## 6.4. Estimates

### Types of estimates:

- Costs
- Time
- REQ (DE: Soll man etwa REQ abschätzen? Was ist das fuer einen Quatsch?)

### Influences on Development-Costs:

- Project type
- Project maturity
- Design + Testing Methods + Tools
- Technology
- Complexity of planned solution
- Quality objectives
- Team qualifications + experiences
- Team distribution

### Cost Estimation:

#### 1. By Analogy

Based on experience.

#### Comparison may include:

- Nr of REQs
- Scope of Sol
- Technology used
- Staff Characteristics (skills, experience,...)

#### Delphi Method:

- Involves a panel of experts
- Structured communication technique
- Conduct interactive forecasting

### **Agile estimations:**

- "Planning Game" or "Planning Poker" method
- "Burn-Down-Rate": something to measure team ability
- "Retrospective sessions" after each sprint to compare figures

## **2. Algorithmic Procedure**

### **Using:**

**EKLOC** = Estimates Thousand of Lines of Code

**ESLOC** = Effective Source of Lines of Code

### **a. Putnam Equation:**

Effort = (size / productivity \* time <sup>4/3</sup>) <sup>3</sup> \* B

(DE: Das Ganze bringt eher wenig weil alle Beträge sowieso auch aus Abschätzungen stammen)

### **b. Function Points:**

A function point measures the amount of business functionality that an information system provides to a user. Basis:

- Internal logical files
- External interface files
- External inputs + Inquiries
- External outputs

### **c. CoCoMo (Constructive Cost Model)**

Function of program size,

...now coming complex formulae, depending on project type and a table of coefficients.

(DE: Alles reine Abschätzung...!)

## **6.5. Prioritization = Establish relative Importance of REQ**

**Procedure:**

- REQ grouping (influencing REQs depending on each other)
- REQ Analysis (stakeholders agreeing on level of importance + impact analysis)
- Creation of REQ Project Plan (Plan w/ REQ w/ high priority first, assigning responsibilities)
- Planning System Increments Testing (Design of Test for each increment)

**Prioritization Scale:**

- Critical (not acceptable without)
- Could (can wait until a later release)
- Would (be nice to have but product without does not matter that much)

**NOTE:** Moscow is better.

**Evolution of REQs:**

REQ are not stable -> Develop during the project life cycle

**Reasons for REQ Development + Changes:**

- New Insights
- New Customer Needs
- New Regulations
- Changes in the Business
- New Products + Systems

**6.6. SignOff = Agreeing in REQ is called SIGNOFF**

Formal Agreement that content and scope are accurate and complete

The REQ SignOFF is the final task of REQ Analysis + Design.

**REQ Signoff signed by:**

- Project managers
- Customer Business representatives
- Business and System Analysts
- REQ Engineer
- QA, Tester + Development Teams

**Purpose:**

Ensuring REQ are stable and changes are managed formal via Change Management.

REQ SignOFF is a Project Milestone.

**Advantages of REQ Agreement:**

- Ensures Development of right Prod
- Minimize Risk of misunderstandings
- Basis for further Design work

**6.7. Traceability (see Change Management too)**

Is a solution to manage developing REQs

Provides a check for all important steps of the Development Process

Important for Testing, Verif + Val

**Goals of Traceability:**

- Impact Analysis
- Coverage Analysis
- Proof of implementation

**Types of Traceability:**

- Horizontal (Relation between diff REQs)
- Vertical (Dependencies between artifacts)

**6.8. What Influence REQ ENG**

- Product to develop
- Environment of production
- Business Domain
- Legal factors
- Safety factors
- Environmental facts

- Time and Costs Constraints
- Cultural factors
- Technology constraints
- Design Constraints (DE: Was ist das fuer etwas? Wie ist es hierher rein gerutscht?)

## **7. Models for Development Processes**

### **7.1. SE Process-Model**

#### **Standard format for:**

- Planning
- Organizing
- Running

### **7.2 Product Life Cycle**

#### **Product development:**

- Planning
- Development
- Maintenance
- End of Life

### **7.3. Planing**

- Vision
- Strategy
- Business plan
- Cost-Benefit Analysis

### **7.4. Development**

- Specification
- Draft
- Implementation

OR  
(BETTER!)

- Analysis
- Design
- Implementation
- Testing

## **7.5. V-MODEL:**

**Defining -> Syst. Analysis -> Syst. Design -> Techn. Design -> Component (Module) Design -> Implementation -> Unit Testing -> Integration Testing -> System Testing -> User acceptance Testing**

## **7.6. Rational Unified Process (RUP) (From IBM):**

Iterative model

Inception -> Elaboration -> Construction -> Transition

## **7.7. Agile Approaches**

To communicate and track REQS, agile approaches use

- Produkt Backlog
- Story Cards
- Screen Mock up

AND Work Assignments are regularly adjusted...

### **REQ DEV:**

Customers Needs and Ideas are iteratively

- Elicited
- Elaborated
- Analyzed
- Validated

### **REQ DOC are:**

- User Stories
- Scenarios
- Use Cases

- Product Backlogs
- Result of Iterations (DE: Ist das Produkt die Dokumentation von sich selbst? Oder ist die Dokumentation selbst ein "result" der Iteration? Aber in welcher Form in diesem Fall?)

### **7.7.1. Extreme Programming**

#### **For changing customer REQs:**

- Short Development Cycles => Improve Productivity (DE: Die Effizienz steigt, die Effektivität sinkt...)  
=> Checkpoints to adapt new REQ (DE: Man muss also bis zum Checkpoint warten, um neue REQ aufzunehmen?)
- Pair Programming
- Extensive Code Review
- Unit Testing of Code
- Flat management structures
- Expecting changes in customer's REQ
- Frequently communication w/ customer
- No separate REQ-Phase

### **7.7.2. SCRUM (DE: Gedränge bedeutet das auf deutsch...)**

Is an agile framework

Roles:

- Scrum master
- Product Owner
- Team

Dividing Development into "Sprints"

2 Backlogs:

- Product Backlog: List of potential features, prioritized by Business Value (Product Owner is the owner)
- Sprint Backlog: List of Work, Features, broken into tasks (Team is the owner)

Needs Customer Feedback

Expect REQs evolve

REQs not fully specified

Features can be reprioritized

Use "User Stories" ("I as cat need a tool to catch mouses")

### **7.7.3. Maturity Model**

(DE: Ich frage mich ernst, wie man "Determination" mit "E" uebersetzt im "SPICE")

Has 5 Categories:

- 1. Customer Supply**
- 2. Engineering**
- 3. Supporting**
- 4. Management**
- 5. Organization**

Capability Process:

- 0. Incomplete**
- 1. Performed**
- 2. Managed**
- 3. Established**
- 4. Predictable**
- 5. Optimizing**

Capability Maturity Model Integrated (CMMI)

Define 5 maturity levels for

- Development
  - Services
  - Aquisition
- 1. Initial (DE: Chaos)**
  - 2. Managed (DE: Prinzipien)**
  - 3. Defined (DE: Definition)**
  - 4. Quantitatively managed (DE: Messbar)**
  - 5. Optimizing (DE: Verbessern)**

## **8. Change Management**

Changes will allways happen.

Stability of REQs allows to meet deadlines (of project).

Depending of complexity and impact, a change should be analysed to establish risks and evaluate the value of the modification against risks.

### **Source of Change:**

- New or Extended functionality, Modifications of Functionality
- Defects found in SW/Doc

- Problems such as bad performance
- External factors: Regulations, ...

REQ, Change Requests and Defect Lifecycle is very similar and managed by the same tool

Change != Defect

**Defect:** Flaw in Component or System, can cause malfunction.

**Change:** Modification of Features, REQs, Functions.

### **Change Management is the Process of:**

- Requesting }
  - Determining attainability }
    - Planning }
      - Implementing }
        - Evaluating } - Changes

### **Purpose:**

Ensure Traceability of changes, allow and support the process of Changing.

### **Activities:**

- Identification of pot. Changes
- Request new Functionality
- Analysis of Change Request
- Evaluating }
- Planning }
- Implementing }
- Reviewing }
- Closure }

- Roll off } - the Change

**Change REQ:** RFC = Request For Change  
Submitted as a formal Change Request Document.

**Describes:**

- Reason for Change
- Priority
- Requested Solution
- Name of the submitter
- Date
- Date of implementation
- Costs (estimated)

**CCB** = Change Control Board:  
Changes will be checked and decided by the CCB.

**Basis for Decision:**

- Risk
- Impact
- Required effort

**Stakeholder on the CCB:**

- Project Management
- Development Management
- REQ Engineers
- QA + Test Management
- Business Management
- Customer

**Impact of Changes on the Project**

**1. Changes of:**

- Schedule
- Budget
- Resources

**2. More Work:**

- Update of Design + Analysis
- Update of Spec
- Update of Doc
- Update of Test Plan
- Update of Training Need
- Change of Programming
- Change in Tests + Test Scope

## **9. On Quality Assurance**

### **9.1. On the quality of REQs**

#### **Quality of REQs:**

REQ are the basis for System Development

REQ have to be documented and tested against quality criteria

- Correctness
- Understandable
- Feasible
- Traceable
- Identifiable
- Testable

#### **Tools and techniques for Q.-Improvement + QA on REQs**

- Standard + Templates
- Reviews + Inspections
- Traceability
- Prototyping
- Observance of Q.-Criteria:
  - Completeness
  - Correctness
  - Compliance w/ Standards

## Elements to improve Quality of REQ SPEC:

- Outlining Purpose of Document, scope, specifications, glossary
- Outlining objectives at diff levels
- Defining Design + Implementation constraints
- Grading/Prioritizing of REQs
- Clear Statements (what - not how)
- Documenting Legislation + Assumptions + Business Rules + Dependencies
- Avoiding supplementary descriptions of diagrams that are clear (DE: Das ist auch "clear". Schein Ballast zusein.)
- Replace abstract text w/ diagrams
- Clearly specified user rights + privileges
- Use of structured presentation
- Use of simple, precise language

## 9.2. About using REQ for QA

### REQ ENG and Testing:

REQ ENG is closely connected to testing.  
GOOD REQ => GOOD TEST CASES

### Acceptance Criteria = Success Criteria:

- Standards to satisfy customers expectations
- Should be agreed upon by both sides
- Basis for Project Quality Plan
- Basis for Acceptance Testing
- Criteria must be measurable
- Criteria must be realistic
- Criteria must be agreed on

### REQ ENG provides "Test Basis":

- Enabling functional Coverage
- Basis for Black-Box }
- Basis for Spec-Based }
- Enabling equivalence partitioning }

- Enabling boundary value analysis        }
- Enabling Decision Table                    }
- Enabling State Transicion                 }
- Enabling Use Case                           } - Testing

(DE: Viel mehr als "enabling" soll es moeglich sein, die REQs zu benutzen wenn bestimmte Testmethoden geplant sind, und dies in den REQs vorgesehen wurde).

### **REQ an the Test-Process:**

Well defined REQ

- Reduce the risk of project failure
- Allow careful testing
- REQ validated through static tests
- REQ accepted by Test Managers
- Testers participate in Reviewing of REQs to ensure their testability, pointing to weak points and possible defects

## **10. Metrics (DE: Innerhalb Projekte)**

- Metrics makes possible to quantify the status of a project
- Metrics makes possible to quantify the quality of a project
- Measurements must be compared to reference data

### **Metrics:**

- Project costs
- Project tracking
- Project stability (how many changes)
- Process improvement (reasons for changes)
- Quality of Specs
- Number of defects (faults in REQ)

## **The change rate can be a measure for the quality of REQs:**

The higher the Change-Rate of REQs, the higher the Risk of the project.

# **11. Tools (EN: last and almost last)**

## **Using Tools in REQ ENG:**

Help to keep Docs consistent + up to date

### **Tools may support:**

- REQ Identification
- REQ Storage
- REQ Modeling
- REQ Documenting (DE: Doppelt gemoppelt?)
- Define REQ Traceability
- Maintain REQ Traceability

### **Advantages of Using Tools:**

- REQ stored in one place
- REQ Accessible for all stakeh
- Support REQ traceability
- Allow verify REQ Coverage
- Manage REQ Changes
- Force usage of defined templates
- Force using defined modeling notation
- Automate some activities

### **Categories of Tools:**

- Elicitacion (Mind Mapping)
- Modeling (UML, SysML)
- Prototyping }
- REQ Management }
- Defect Management }

- Change Management }
- Project Management } - Tools

## **Analysis before Selecting a Tool:**

### **To be considered:**

- Modeling notation
- Modeling notation in Future
- Desired functionality
- Costs (for how many projects)
- Integration w/other tools
- Information exchange w/ other tools
- Easy of use + learning
- Online Help + other help

### **Costs to be considered:**

- Purchasing cost
- Cost of training
- Costs of Extending functionality
- Integration costs w/ other tools

=====

## **Mit den besten Wünschen:**

### **Unternehmen**

- **AMTRS** Systemanalyse – Ruy Kuhlmann
- **"ruynk" Projektkoordination** – R. C. N.-Kuhlmann Projektmanager
- **IT-Wissen "Ryusui"** - Kurse und Seminare
- **TST Tsubame Software Tools** - Business Support Software Solutions

### **URLs:**

- [www.amtrs.de](http://www.amtrs.de) ---> AMTRS: Analysieren, Modellieren, Trainieren, Reorganisieren, Systematisieren
- [www.ruynk.com](http://www.ruynk.com) ---> Projektmanagement, Projektkoordination, FlePA
- [www.ruynk.de](http://www.ruynk.de) ---> Private Homepage
- [www.ryusui.de](http://www.ryusui.de) ---> IT-Wissen – Kurse und Seminare
- [www.tsubame.de](http://www.tsubame.de) ---> Business Support Software Solutions

**Blogs:**

- [RUYNK: Über Projektmanagement und FlePA](#)
- [Tsubame SW: Über SW-Entwicklung](#)