Syllabus

REQB®
Certified Professional for Requirements Engineering

Foundation Level

Version 1.2

July 1st, 2008

Overview of Changes

Version	Date	Comment
0.1	Apr. 17, 2006	First version of the syllabus; creation of a basic structure for the syllabus
0.2	Jul. 20, 2006	Expansion to Version 0.1
0.3	Sep. 4, 2006	Further expansion and revision of Version 0.1
0.4	Oct. 10, 2006	Revised Version 0.3
0.5	Dec. 15, 2006	Revised Version 0.4
0.6	Feb. 7, 2007	Completely revised Version 0.5
0.7	Apr. 10, 2007	Revised version for review
0.8	Jun. 15, 2007	Alpha version
0.9	Sep. 1, 2007	Beta version
1.0	Jan. 15, 2008	Released version 1.0
1.1	Mai 29, 2008	Updated version 1.1
1.2	Jul. 1., 2008	Updated version 1.2

Main Idea

The central theme for this syllabus was that the complexity of software and our dependency on software continues to increase. The result is a high level of dependency on the freedom from error in the software. The Requirements Engineering Qualifications Board (REQB) has therefore decided to create uniform international standards in the area of Requirements Engineering. For standards are like languages - it is only if you understand them that you can work effectively. In order to now create such a uniform language in this important area of requirements engineering, international experts got together in REQB and developed this syllabus.

Content

Introduction

Purpose of the Syllabus

This syllabus defines the basic level (Foundation Level) of the training program to become a REQB Certified Professional for Requirements Engineering (short form CPRE). REQB developed this syllabus in cooperation with the Global Association for Software Quality.

The syllabus is to serve as a foundation for training providers who are seeking accreditation as teachers. All areas of this syllabus must correspondingly be incorporated in the training documents. The syllabus should, however, also serve the learner as preparation for certification. All areas listed here are thus relevant for the examination, which can be taken either after accredited courses or in open examination.

Examination

The examination to become a Certified Professional for Requirements Engineering is based on this syllabus. All sections of this syllabus can thereby be tested. The examination questions are not necessarily divided into the individual sections. A question may refer to several sections.

The format of the examination is Multiple Choice.

Examinations can be taken after having attended accredited courses or in open examination (without a previous course). You will find detailed information regarding examination times on the website of gasq (www.gasq.org) or on REQB's website (www.reqb.org).

Accreditation

Providers of a REQB Certified Professional for Requirements Engineering Course must be accredited by the Global Association for Software Quality. Their experts review the training provider's documentation for accuracy. An accredited course is regarded as conforming to the syllabus. At the end of such a course, an officially

Certified Professional for Requirements Engineering examination (CPRE exam) may be carried out by an independent certification institute (according to ISO 17024 rules).
Accredited Training Providers can be identified by the official REQB Accredited Training Provider logo:

Internationality

This syllabus was developed in cooperation between several international experts. The content of this syllabus can therefore be seen as the international standard. The syllabus thereby makes it possible to train and test internationally at the same level.

K Levels

The syllabus has been divided into different K levels. This makes it possible for the candidate to recognize the "knowledge level" of each point.
There are 3 K-levels in the current syllabus:

- K1 - remember, recognize, recall
- K2 - understand, explain, give reasons, compare, classify, summarize
- K3 - apply in a specific context

1. Basics

Learning target

What is a requirement?

What is the meaning and purpose of requirements?

How can requirements be classified?

What types of requirements are there?

What problems are there concerning requirements?

What concepts are important in connection with requirements?

What is the difference between RM (Requirements Management) and RE Requirements Engineering)?

What important norms and standards exist?

Why is Requirements Engineering important?

1.1 Requirement

Definition of what is meant by the term “A Requirement” (K 1)

Glossary:

IEEE 610.12: *A requirement is a condition or a skill that a user needs in order to solve a problem or arrive at a goal.*

What is the meaning and purpose of requirements? (K 2)

- Foundation for assessment, planning, execution and monitoring of the project activity
- Customer expectations
- Component of agreements, orders, project plans…
- Setting of system boundaries, scope of delivery, contractual services

Classification of requirements (according to Ebert05) (K 2)
Requirements consist of process requirements and product requirements

Process requirements: costs, marketing, processing time, sales and distribution, organization, documentation
Describe needs and limitations of the business processes.

Product requirements consist of functional and non-functional product requirements. Both can be regarded from the point of view of the user (external) or customer and from the point of view of the developer (internal).

Functional product requirements from the user's point of view: user interface, applications, services
Functional product requirements from the customer's point of view: user interface, applications, services

Note: User and customer can be different!

Functional product requirements from the developer's point of view: architecture, power supply, load distribution
Functional requirements describe the function of the system

Non-functional product requirements from the point of view of the user: reliability, performance, usability
Non-functional product requirements from the point of view of the customer: reliability, performance, usability
Non-functional product requirements from the point of view of the developer: testability, serviceability, tools

Non-functional requirements describe the quality attributes of the system.

Types of requirements: customer requirements, solution/system requirements, product/component requirements (K 1)

Problems with requirements (K 2)

- unclear objectives
- communication problems
- language barriers
- knowledge barriers
- vague formulation
- too formal formulations
- ambiguous, overly specified, unclear, impossible, contradictory requirements
- instability of the requirements
- bad quality of the requirements
- gold plating
- insufficient user involvement
- overlooked user classes
- inaccurate planning
- minimal specification

Quality criteria for requirements (according to Wiegers05): (K 2)

1. Each requirement must be complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable
2. The requirements specification must be complete, consistent, modifiable and traceable

For training companies: explanation of the individual quality criteria

Solution (K1)

A solution is the implementation of the requirement

Commitment (K1)

Commitment is the degree of obligation

Defining the commitment through key words

For training companies: explanation of key words

Observing legal responsibilities, especially in case of faults

Fault (K 1)

Deviation of the current state from the target state

Priority of requirements (K 1)

Evaluation of the importance/urgency

Criticality of requirements (K 1)

Evaluation of the risk of a requirement by evaluating the damage in case of non-fulfillment of a requirement

Validation (K1)

Process of confirmation that the specification of a phase or the entire system fulfills the customer's requirements

Verification (K 1)

Comparison of an intermediate product with its specifications. It is thereby determined if the software was developed correctly and if the specifications that were determined during the previous phase were fulfilled

Delineation between requirements management and engineering (K 2)

Requirements Management (RM) includes processes for the identification and management of requirements

Requirements engineering includes the basic engineering skills

For training companies: deepening the individual areas

1.2 Standards and Norms

ISO 9000:

Requirements of a quality management system:

- defined concepts and basics of a QMS
- domain or industry neutral

ISO 9126:

Defines a quality model with six categories:

Functionality, reliability, usability, efficiency, maintainability, portability

IEEE 610:

Standard Glossary of Software Engineering Terminology

IEEE 830:

Recommended Practice for Software Requirements Specifications

IEEE 1233:

Guide for Developing of System Requirements Specifications

IEEE 1362:

Guide for Information Technology – System Definition

Process norms:

ISO 12207:

Standard for Software Life Cycle Process

ISO 15288:

System Life Cycle Process

ISO 15504:
Software Process Improvement and Capability Determination (SPICE)

Capability Maturity Model Integrated (CMMI)

In order to pass the examination, it is not necessary to know the contents of all the norms. It is, however, important (K 1) to know which norms are of importance for Requirements Engineering.

Requirements Engineering is of vital importance. And yet it is neglected time and again.

For training companies: highlighting the importance of Requirements Engineering and the reasons why it is often neglected (K 2).

- *Neglect due to high time pressure*
- *Neglect due to an exclusive orientation toward fast results*
- *Neglect due to an exclusive fixation on costs*
- *Neglect due to misinterpretations (many things are seen as given)*

Possible consequences of neglecting Requirements Engineering (K 2):

- Requirements become imprecise
- Requirements are ambiguous
- Requirements are contradictory
- Requirements that change
- Requirements that do not fulfill the criteria
- Requirements that can be interpreted differently
- Missing requirements

2. Procedure and Processes

Learning Goals:

What different process models are there?

How do the different process models differ?

What characterizes the Requirements Engineering process?

What are the phases of this process?

2.1 Process Models

Procedural models (K 2)

Method-independent process description of development processes

Roles, activities, phases and documents are thereby taken into account

Product life cycle (PLC) (K 2)

Basic phases: planning, development, maintenance, end of life

The planning phase includes: vision, strategy, business plan, and cost benefit analysis

The development phase includes: specification, draft, and implementation

Defines various phases of the product development

General V Model (K 2)

Steps of development:

- Definition of requirement, determination of requirement (performance specifications)
- Functional system draft, systems analyses (functional specifications)
- Technical system draft, architecture draft (software design)

- Component specification
- Implementation

For training companies: graphic portrayal of the General V model; in depth description of the General V Model

Rational Unified Process (RUP©) (K 2)

Procedural model by IBM Rational ©

Thereafter there is an iterative development process with

- 4 phases (inception, elaboration, construction, transition)
- 9 disciplines (among others requirements discipline)

For training companies: deepening of the RUP© with graphic presentation; deepened study of the requirements discipline

Extreme Programming (K 2)

Procedural model by Kent Beck et al

Requirements management as a main component

Completely without any requirement investigation

For training companies: also explain at least three further agile models including Scrum and Crystal.

Degree of Maturity Model (K 2)

For the identification and improvement of the process maturity (process assessment and process improvement)

Definition of degrees of maturity and corresponding specifications of the processes

For training companies: deepening, using the example ISO 15504/SPICE; with a description of the typical requirements for Requirements Engineering

2.2 Requirements Engineering Process (K 2)

Non-core process, which concerns all phases of the systems development

- Identification of requirements
- Analysis of requirements
- Specification of requirements
- Changes of requirements
- Verification
- Quality assurance

Description of the negative influencing factors on the processes

Description of the different points of view (customer, supplier point of view)

Method of the Requirements Engineering process with the customer at the center

3. Project and Risk Management

Learning Goals:

Why is Requirements Engineering important in projects?

What errors can arise in Requirements Engineering?

What risks are there with requirements?

3.1 Project Management

Description of the necessity of RE in projects (K 2)

RE should contribute to the following areas: (K 1)

- Project conception
- Contract negotiations
- Project definition
- Project execution

For training companies: more detailed description of RE in these areas.

What errors can occur in Requirements Engineering? (K 2)

- Unclear requirements
- Changing requirements
- Unstable product and design basis for sub-orders
- Unclear responsibilities
- Gap between customer expectations and project contents
- Insufficient customer management
- Project definition with milestones that cannot be achieved
- Imprecise expense estimate
- Imprecise estimate of impact

3.2 Risk Management

Explain the necessity of risk management (K 3)

Efficient risk management as a key to lowering project risks

For training companies: detailed description of the development of countermeasures and techniques of risk management (such as Failure Mode and Effect Analysis).

4. Responsibilities and Roles

Learning Goals:

What basic roles are there in Requirements Engineering?

What is a stakeholder?

What are the tasks of Requirements Engineering?

What is the task of a Professional for Requirements Engineering?

What characterizes a Professional for Requirements Engineering?

4.1 Basic Roles

Basic roles: (K 2)

Client (= customer)

Contractor (= supplier)

The client formulates his needs

The contractor delivers solutions

Stakeholder

A stakeholder is a person or a role that has an interest

Stakeholders can be either natural persons, legal entities or abstract persons

Stakeholders often have conflicts of interest among each other

For training companies: description of typical stakeholder (e.g. Managing Director, Project Manager, client)

Important: Identification of all stakeholders in order to take adequate consideration of all perspectives

4.2 Tasks of Requirements Engineering

Tasks: (K 2)

- Analysis of business processes
- Identification and analysis of requirements
- Quality assurance of requirements and specifications
- Creation of the requirements specification
- Risk analysis

The Professional for Requirements Engineering identifies wishes and aims.

Knowledge of a Professional for Requirements Engineering: (K 1)

- Skill of moderation
- Self-confident manner
- Ability to convince
- Language skills
- Ability to communicate
- Precision
- Analytical, clear thinking
- Ability to act in a structured way
- Methodological competence
- Stress resistance

5. Identification of Requirements

Learning goals

What should a contract contain?

What should be considered when evaluating requirements?

What characterizes a typical project vision?

How can stakeholders be identified?

What are the goals of the identification of requirements?

What techniques are there for identifying requirements?

What characterizes functional and non-functional requirements?

How do they differ?

What contents should be covered by a requirement document?

What characterizes good requirements?

What are the standard contents of a requirements document?

How does one construct a requirement?

5.1 Customer

The customer must always be involved. The goal is to understand the customer and to develop a mutual understanding of one another. The contractor should thereby always put himself in the customer's position.

Contract: (K 2)

- In the agreement it should be formally specified and described what the customer wants
- It must be ensured that the interest of the customer takes center stage.
- It is important to set realistic deadlines and prices and to make realistic project plans

When evaluating the requirements, different points of view must be taken into consideration.

5.2 Project Visions and Goals

In the beginning we have the development of project visions. This is the first step of Requirements Engineering

It is crucial to have a clear definition of project visions.

For training companies: presentation of typical project visions

Important questions regarding project visions: (K 2)

- What will the project change?
- Why is the project necessary?
- What happens once the project has been terminated?
- Who will profit from the project?
- Which costs are we willing to bear?
- What risks are we willing to assume?

For training companies: presentation of typical influences on project visions (customers, strategy, etc.)

For each project, the vision must be set anew

5.3 Identifying Stakeholders

All stakeholders on the customer and supplier side are to be identified

Interest groups are to be brought together.

For training companies: explanation of the identification and evaluation of stakeholders

5.4 Techniques for Identifying Requirements

Aim of the identification of requirements (K 2)

Identify all desired functions, characteristics, limitations and expectations

Orient the requirements toward the project vision

Functions must be described clearly

Functions that the customer does not want are to be excluded

Techniques (K 1)

- Questionnaires
- Interviews
- Self-recording
- Representatives of the customer on site
- Identification on the basis of existing documents
- Reuse (Reusing the specification of a certain project)
- Brainstorming
- Field observation
- Apprenticing
- Workshops after each specified process

For training companies: techniques, including describing advantages and disadvantages

For training companies: description of questioning methods for interviews

5.5 Functional and Non-functional Requirements

Functional Requirements (K 2)

Description of the functions of the system

Trigger a process

Non-functional Requirements (K 2)

Describe attributes of the functions or their quality characteristics

Difficult to describe, therefore often only formulated vaguely

Often hard to track and test

Precise and clear description necessary for validation

For training companies: examples of functional and non-functional requirements

Quality characteristics according to ISO 9126: (K 2)
Functionality, reliability, usability, efficiency, maintainability, portability

For training companies: detailing the individual quality characteristics
For training companies: description of limitations (for example technical specifications)

5.6 Descriptions of Requirements

Content of the requirement text:
Who? What? How? When? With whom? By what affected?

Important guidelines for the creation of the requirements document: (K 2)

- The requirements must be unambiguous, precise and understandable
- Superfluous information should be avoided
- Templates as an aid to limit language

Standard contents of a requirements document: (K 1)

- Introduction
- Secrecy clause
- Regulations
- Standards
- Stakeholder
- Purpose of the product
- Description of the system
- Functional requirements
- Non-functional requirements
- Assumptions
- Dependencies
- Risks
- Safety requirements
- Software Quality Attributes
- Acceptance

Construction of a requirement (K 2)

1. Determination of the process
2. Classification of the system activity
3. Determination of the legal commitment
4. Refining the process
5. Logical and time constraints

For training companies: description of the individual steps in the construction of a requirement

6 Specification of requirements

Learning goals:

What is a requirement specification?

What characterizes a requirement specification?

What is a solution specification?

What characterizes a solution specification?

What standards are important for requirement specifications and solution specifications?

What is the typical procedure when it comes to the specification of requirements?

What different degrees of formalization exist for the specification of requirements?

What can be the consequences of errors in the requirements?

What possibilities are there for avoiding requirement errors?

6.1 Specification

In the specification, requirements are specified in a structured way and are modeled separately.

The specification serves to track and manage requirements.

Requirements specifications (K 2)

Are also called performance specifications

The creation should be the customer's task

Solution specifications (K 2)

Are also called functional specifications

The basis for the further system development

Important standards: (K 1)

IEEE 1362 (system performance specifications), IEEE 830 (software requirements specification) - IEEE 1233 (system functional specifications)

For training companies: more detailed description of the performance specifications and the functional specifications

6.2 Procedure

Specification as an activity for formalizing the results of the requirements analysis (K 2)

For training companies: description of steps for the specification of problems and solutions (determining the solution space, describing the customer contacts, etc.)

The identification phase is terminated when all agreements that are necessary for the project have been signed

6.3 Formalization

Different degrees of formalization

- Non-formal
- Semi-formal
- Formal

For training companies: description and differentiation of the degrees of formalization (K 2)

6.4 Quality of Requirements

Requirement errors as a cause of high costs (K 2)

The later errors are detected, the higher are the costs

Therefore the use of manual verification (are we producing the product correctly?) and validation (are we producing the right product?)

For training companies: description of measures for quality improvement and assurance

7. Requirements Analysis

Learning Goals:

What is the goal of requirements analysis?

What is the procedure during requirements analysis?

What are the different models of requirements analysis?

What characterizes UML?

What characterizes SysML?

What is the reason for the cost estimate?

What are the important factors for cost estimates?

What is the procedure when prioritizing?

What should be considered when agreeing on requirements?

7.1 Requirements and Solutions

Goal of the requirements analysis: solution for the implementation of the requirements (K 2)

Procedure:

1. Analysis of the needs
2. Description of the solution
3. Cost estimate and prioritization

7.2 Methods and Techniques

Different aspects of a system are represented through different views

Models are developed through suitable methods of analysis

Differentiation between types of models (K 2)

Requirements models

Solution models

For training companies: detailed description of the two types of models and their views

Different models (K 1)

- Context model
- Functional decomposition
- Data flow model
- State transition model
- Petri Net
- Entity Relationship Model

For training companies: detailed description of the various models

7.3 Object-oriented Analysis

UML (Unified Modeling Language)

UML provides diagrams for different views of the system

Use case diagrams, class diagrams, activity diagrams, state diagrams, object diagrams, component diagrams, package diagrams, etc.

For training companies: examples of diagrams (at least for use case diagrams, class diagrams, activity diagrams and state diagrams)

SysML (Systems Modeling Language) as an expansion of the UML

7.4 Cost Estimates

Cost estimates connect Requirements Engineering with the project management

Types of estimate (K 2)

- Costs
- Time
- Requirements
- Quality

Cost estimates help to recognize the cost for change

For training companies: description of the determining factors for the development costs

Estimation procedure (K 1)

- Conclusions by analogy
- Algorithmic procedure
- Function point procedure
- Constructive cost model
- Delphi method

For training companies: explanation of these estimation procedures

Estimation procedures are always based on historical data and framework conditions

7.5 Prioritization

Procedure

1. Grouping of the requirements
2. Analysis of the requirements
3. Creation of the project plan
4. Testing the increments

7.6 Agreeing on Requirements

Agreements (K 2)

- Formal agreement as project basis
- The list of requirements must be binding
- A constant review of the requirements list is necessary

For training companies: description of the advantages of binding allocations

8. Tracking of Requirements

Learning Goals:

What is traceability?

Why do requirements continue to develop?

What is the purpose of traceability?

What kinds of traceability are there?

What characterizes change management?

How is the Change Control Board constituted?

What are metrics?

What makes metrics possible?

8.1 Tracing within the Project

Traceability (K 2)

Requirements are not stable, but continue to develop

Reasons for continued development

- New insights
- New customer needs
- Continued work
- New connections within the project

Traceability as a solution:

- Provides a check that all important steps of the development process have been carried out

Goals: Impact analysis, coverage analysis, use-of-potential analysis, proof of implementation, use of the requirement, etc.

In order to ensure good traceability, it is important to label the requirements precisely.

Types of traceability

Horizontal and vertical tracing

For training companies: describe what horizontal and vertical tracing is

8.2 Change Management

Changes of the requirements (Change Management) (K 2)

Changes are checked and decided on by a Change Control Board

Makes decisions regarding change requests

<u>The Change Control Board consists of (K 1)</u>

- Project management
- Development
- Quality assurance
- Business management, if applicable
- Customer, if applicable
- etc.

For training companies: Tasks of the Change Control Board

A structured process is necessary for changes of requirements

The analysis of the meaning of each change is important. Hasty solutions are problematic.

Large changes of the requirements can be so serious that they represent a contractual change.

For training companies: description of the life cycle of a requirement

For training companies: explain the impact of changes of requirements

For training companies: distinction between error management and change management

8.3 Metrics

Metrics make it possible to make a quantifiable statement regarding the project status and quality

Classification figures must always be compared to reference data

Metrics for requirements: (K 1)

- Project costs
- Project tracking
- Project stability
- Process improvement
- Quality of the specification
- Number of errors
- Type of error

Measurement of the requirements quality: (K 2)

Are the requirements correct?

Are the requirements understandable?

The higher the change rate, the more a project is at risk

9. Quality Assurance

Learning Goals:

What factors influence Requirements Engineering?

How can quality assurance be improved?

What are acceptance criteria?

What methods exist for quality assurance?

9.1 Influencing Factors

Some factors that influence the success of Requirements Engineering: (K 2)

- The product that is being produced
- The environment in which the product is produced
- Industry
- Time pressure
- Cost pressure
- Social factors

Such factors must be taken into consideration when it comes to quality assurance

9.2 Quality Assurance through Testability

Requirements engineering extends across the entire life cycle

Requirements engineering is closely connected to testing. Good test cases require good requirements that can be tested (The involvement of testers for the specification is therefore very important) (K 2)

Acceptance criteria

Every requirement has at least one acceptance criteria

This is the basis for the acceptance test

Methods:

Functional coverage

Equivalence partitioning

For training companies: description of the methods

10. Tools

Learning Goals:

How do tools help with regards to Requirements Engineering?

What activities can be taken over by tools with regards to Requirements Engineering?

What requirements are there on tools in the area of Requirements Engineering?

What must be taken into consideration in terms of cost regarding tools?

10.1 Advantages of Tools

Tools for storage and administration of requirements facilitate Requirements Engineering. They can take on mechanical activities or ensure overview. It is thus possible to keep difficult static documents consistent and current. The selection of a tool must occur before the product is developed. This can otherwise cause substantial problems. (K 2)

For training companies: description of the requirements placed on tools in the area of Requirements Engineering

For training companies: description of the activities of Requirements Engineering that can be supported by the use of tools

10.2 Categories of Tools

- Text processing, table calculations
  MS Excel, MS Word etc.
- Modeling tools
- Tools for requirements management
- Tools for error management

- Open source tools
- …

The cost for tools thereby varies greatly. The choice of a tool must therefore be made very carefully. A hasty choice can result in high costs. (K 2)

11. Literature

Beck, Kent: Extreme Programming. Munich 2003

Beck, Kent: Extreme Programming Explained: Embrace Change. Boston 2000

Beck, Kent: Test Driven Development. By Example. Amsterdam 2002

Beck, Kent: Refactoring: Improving the Design of Existing Code. Addison-Wesley Longman 1999

Boehm, Barry: Software Engineering Economics. Englewoods Cliffs, NJ 1981

Bundschuh, Manfred; Fabry, Axel: Aufwandschätzung von IT-Projekten. Bonn 2004

Cockburn, Alistair: Agile Software Development. Addison Wesley 2002

Cockburn, Alistair: Writing Effective Use Cases. Amsterdam 2000

DeMarco, Tom et al.: Adrenalin-Junkies und Formular-Zombies – Typisches Verhalten in Projekten. Munich 2007

DeMarco, Tom: Controlling Software Projects: Management, Measurement and Estimates. Prentice Hall 1986

DeMarco, Tom: The Deadline: A Novel About Project Management. New York 1997

Ebert, Christof: Systematisches Requirements Management. Anforderungen ermitteln, spezifizieren, analysieren und verfolgen. Heidelberg 2005

Evans, Eric J: Domain-Driven Design: Tackling Complexity in the Heart of Software. Amsterdam 2003

Graham, Dorothy et al: Foundations of Software Testing. London 2007

Gilb, Tom; Graham, Dorothy: Software Inspection. Reading, MA 1993

Hull, Elizabeth et. All: Requirements Engineering. Oxford 2005

IEEE Standard 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology

IEEE Standard 1233-1998 IEEE Guide for Developing System Requirements Specifications

IEEE Standard 829-1998 IEEE Standard for Siftware Test Documentation

IEEE Standard 830-1998 IEEE Recommended Practice for Software Requirements Specifications
IEEE Standard 1362-1998 IEEE Guide for Information Technology-System Definition – Concept of Operations (ConOps) Document

IEEE Standard 1220-1998: IEEE Standard for Application and Management of Systems Engineering Process

ISO 9000

ISO 9126

ISO 12207

ISO 15288

ISO 15504

Jacobsen, Ivar et al.: The Unified Software Development Process. Reading 1999

Lauesen, Soren: Software Requirements: Styles and Techniques. London 2002

Mangold, Pascal: IT-Projektmanagement kompakt. Munich 2004

McConnell, Steve: Aufwandschätzung für Softwareprojekte. Unterschleißheim 2006

Paulk, Mark, et al: The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA 1995

Pfleeger, Shari Lawrence: Software Engineering: Theory and Practice, 2nd edition. Englewood Cliffs, NJ 2001

Pohl, Klaus: Requirements Engineering. Grundlagen, Prinzipien, Techniken. Heidelberg 2007

Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK ® Guide). PMI 2004

Robertson. Suzanne; Robertson, James: Mastering the Requirements Process, Harlow 1999

Rupp, Chris: Requirements-Engineering und Management. Professionelle, Iterative Anforderungsanalyse in der Praxis. Munich 2007

Sommerville, Ian: Requirements Engineering. West Sussex 2004

Sommerville, Ian: Software Engineering 8. Harlow 2007

Sommerville, Ian; Sawyer, Pete: Requirements Engineering: A Good Practice Guide. Chichester 1997

Sommerville, Ian; Kotonya, Gerald: Requirements Engineering: Processes and Techniques. Chichester 1998
Spillner, Andreas et all: Software Testing Foundations. Santa Barbara, CA 2007

Thayer, Richard H.; Dorfman, Merlin: Software Requirements Engineering, 2nd edition. Los Alamitos, CA 1997

V-Modell® XT: http://www.vmodellxt.de/

Wiegers, Karl E.: Software Requirements. Redmond 2005

Wiegers, Karl E.: More About Software Requirements: Thorny Issues and Practical Advice. Redmond, Washington 2006