International
Requirements
Engineering
Board

IREB

IREB Certified Professional for Requirements Engineering
– Foundation Level –

Syllabus

Version 2.1, Valid from 1st September 2010
(based on the German Version 2.1, 1st September 2010)

Acknowledgement

This syllabus has been written by the following members of the board: Karol Frühauf, Emmerich Fuchs, Martin Glinz, Rainer Grau, Colin Hood, Frank Houdek, Peter Hruschka, Barbara Paech, Klaus Pohl and Chris Rupp. They have been supported by the IREB members Ian Alexander, Joseph Bruder, Samuel Fricker, Günter Halmans, Peter Jaeschke, Sven Krause, Steffen Lentz, Urte Pautz, Suzanne Robertson, Dirk Schüpferling, Johannes Staub, Thorsten Weyer and Joy Beatty. We thank everybody for their honorary contributions.
Copyright © 2010 for this syllabus is with the authors mentioned above. The rights have been transferred to the IREB International Requirements Engineering Board e.V.

Preamble

Purpose of the Document

This syllabus defines the foundation level to gain the certificate "Certified Professional for Requirements Engineering" established by the International Requirements Engineering Boards (IREB).
IREB offers this syllabus and associated exams in different languages. For training providers the syllabus is the basis for structuring their offerings and creating course materials. Students can use this document (and background literature) to prepare themselves for the exam.

Contents of the Syllabus

This foundation level syllabus addresses the needs of all people involved in the requirements process. Beyond the key people (i.e. business analysts and requirements engineers) this includes project and IT managers, domain experts, software developers and testers.

Key areas addressed and excluded:

The foundation level covers generally valid requirements know-how knowledge for all kinds of systems (like classical information systems, embedded systems, safety critical systems, …). A training course may cover more domain specific details – if needed by the audience. However, this syllabus does not claim to deal with domain specific issues.
The syllabus does not prescribe a specific process model (e.g. waterfall, iterative or incremental).
It mainly defines the knowledge that a business analyst or requirements engineer should have, without prescribing exact interfaces to other roles in the development process. It is not intended to favor any published process for requirements engineering or overall software development.

Degree of Detail

This syllabus should be the basis for internationally consistent training and learning and examinations. To reach this goal the syllabus provides strictly phrased educational objectives for each chapter (cf. next section) and informal explanations to detail the educational objectives, augmented by references to literature where necessary.

Educational Objectives / Cognitive Levels of Knowledge

One of two cognitive levels is assigned to each module of the syllabus. The higher level includes the lower level. The statements for educational objectives are phrased using the verbs "knowing" for level 1 and "mastering and using" for level 2. These two verbs are placeholders for the following verbs:

- **L1 (knowing):** enumerate, characterize, recognize, name, reflect
- **L2 (mastering and using):** analyze, execute, justify, describe, judge, display, design, develop, complete, explain, elucidate, elicit, formulate, identify, interpret, reason, translate, distinguish, compare, understand, suggest, summarize

⚠ All terms defined in the glossary have to be known (L1), even if they are not explicitly mentioned in the educational objectives.

This syllabus uses the abbreviation "RE" for Requirements Engineering.

Structure of the Contents

This syllabus consists of 9 chapters. Each chapter covers one major educational unit (EU). Following the main title of each chapter, the highest cognitive level of this unit is mentioned. Each chapter also includes the duration suggested to teach it; below this header you will find a list of terms defined in the glossary (cf. appendix A).

Example: 1 Introduction and Foundations (L1)
Duration: 1 ¼ hours
Terms: Requirement, Stakeholder, Requirements Engineering, Functional Requirement, Quality Requirement, Constraint

This example shows that chapter 1 only contains education goals on level 1. It states that 75 minutes should be sufficient to teach the contents of this chapter. Each chapter can be further structured into units. Next to the headers of the units you will again see their corresponding cognitive levels.
Following the chapter headers the educational objectives are enumerated (EO1.1, EO1.2,...). The numbering scheme for these objectives is aligned with the chapter and section numbering.

E.g.:   E0 3.1.2
This is the second objective in chapter 3, section 1.

The Exam

This syllabus is the basis for the exam for the foundation level certificate.

Questions in the exam can cover the content of more than one chapter. All chapters 1 – 9 are relevant for the exam. The exam is a multiple-choice test.

Exams can be held immediately after training courses, but also independently from courses (e.g. publically announced exams of the certification authorities). A list of recognized certification authorities can be found at http://www.certified-re.com

Contents

EU 1 Introduction and Foundations (L1)

Duration: 1 ¼ hours
Terms: Requirement, Stakeholder, Requirements Engineering, Functional
              Requirement, Quality Requirement, Constraint

Educational Objectives:
EO 1.1 Knowing symptoms of inadequate RE and reasons for it
EO 1.2 Knowing the four major activities of requirements engineers
EO 1.3 Knowing the role of communication in RE
EO 1.4 Knowing skills of a requirements engineer
EO 1.5 Knowing three kinds of requirements
EO 1.6 Understanding the role of quality requirements

Good RE is important since many problems in software and system development
have their origin in this discipline. Correcting them later results in high costs.
Typical symptoms of inadequate RE are unclear and missing requirements. Some of
the typical reasons for inadequate RE are

- the wrong assumption by stakeholders that many things are self-explaining
  and need no explicit treatment
- communication problems based on different know-how and experience
- project pressure exerted by contractors asking for early delivery of
  productive systems

The four major activities within RE are eliciting requirements, documenting
requirements, checking and reconciling requirements, and managing requirements.

Natural language and models are the most important means to communicate
requirements. We need to establish a common terminology among participating
stakeholders. Also the medium for communication (writing or talking) plays an
important role. When communicating, all participating stakeholders have to deal
consciously with focusing and simplification.

This is especially true for the most important role in RE: the business analyst or
the requirements engineer. Besides communication skills and method skills he or
she should have the following skills: analytical thinking and reasoning, empathy,
conflict resolution skills, moderation skills, self-confidence and the power of
persuasion.

We distinguish three kinds of requirements: functional requirements, quality
requirements and constraints. The latter two are often called non-functional
requirements.

Quality requirements have to be documented explicitly with respect to the
following considerations:

- Itemization of functionality (e.g. with respect to security aspects or precision of calculations)
- Reliability
- Usability
- Efficiency
- Changeability
- Transferability

Even though quality requirements are documented using Natural Language, their relation to other statements have to be traceable and the verification of quality requirements has to be ensured by quantity indicators or via transformation into additional operational functionality.

EU 2 System and System Context (L2)

Duration: 1 ¼ hours
Terms: System Context, System Boundary, Context Boundary
Educational Objectives:
EO 2.1 Knowing system context, system boundary and context boundary
EO 2.2 Mastering and using system boundary and context boundary

EU 2.1 System, System Context, and Boundaries (L1)

For requirements, the origins (and often their justification) are found in the context of the planned system. The origin is the set of all context aspects that initiated or influenced the creation of requirements. Among the potential sources or origins of requirements in the system context are:

- People (stakeholder or groups of stakeholders)
- Working systems (technical systems, software and hardware)
- Processes (business processes, technical or physical processes)
- Events (technical or physical)
- Documents (e.g. standards, laws, system documentation)

The task of defining the system boundary is to determine which aspects should be covered by the planned system and which aspects will not be covered by the planned system. When determining the system boundary you have to identify the part of the environment that will interact with the planned system.

EU 2.2 Determining System and Context Boundaries (L2)

Sometimes the system boundary is only clear towards the end of the requirements process. Before that, the required functionality and quality aspects of the planned system may be incomplete or simply not known yet. There will be grey zones in which the system boundary will reside. Besides shifting the boundary within the grey zone it is also possible that the grey zone moves during the RE process. For

example, when context aspects suddenly become relevant for the planned system after shifting the system boundary.

Also the context boundary can move over time. For example, when a new law that was considered important turns out to be irrelevant for the new system, the system context has to be reduced in that area.

There may also be a grey zone for the context boundary. It consists of those aspects in the system environment, where it is still unclear whether they will influence the planned system or not.

To document the system context (especially the two boundaries of the system and the context) context diagrams or use case diagrams are often used. In context diagrams, sources and destinations in the environment are modeled, i.e. origin or destination of information flows between the systems and the environment. In use case diagrams, actors (e.g. people or other systems) define the environment. Their relationships with use cases of the planned system are modeled.

EU 3 Requirements Elicitation (L2)

Duration: 1 ½ hours
Terms: none

Educational Objectives:
EO 3.1.1   Knowing different kinds of requirements and sources for requirements
EO 3.1.2   Knowing the impact of requirements sources and the consequences of ignoring requirements sources
EO 3.1.3   Knowing what to capture about stakeholders
EO 3.1.4   Knowing important principles in managing stakeholders (stakeholder rights and duties)
EO 3.2.1   Mastering and using the Kano model
EO 3.3.1   Knowing factors influencing the choice of elicitation techniques
EO 3.3.2   Knowing advantages and disadvantages of selected elicitation techniques;
EO 3.3.3   Mastering and using some elicitation techniques like questioning techniques, creativity techniques, document centered techniques, observation techniques and supporting techniques

EU 3.1 Sources of Requirements (L2)

Eliciting requirements for the planned system is an important RE activity. On the one hand this activity is driven by the system context, on the other hand by the various sources for requirements. Sources are first of all the stakeholders of the project, but also existing documents or existing systems.

It is the job of a requirements engineer or business analyst to collect goals and detailed requirements from all the sources mentioned above. If sources are neglected, this can have very negative consequences for the overall project.
One of the important steps in RE is finding and documenting your stakeholders as important sources for requirements. In this stakeholder list you should include:

- Stakeholder name
- Stakeholder role
- Additional contact information
- Availability (when and where during the project)
- Relevance for the project
- Knowledge areas and depth of knowledge
- Personal goals and interests in this project

Depending on the culture of the company it may be advisable to have an agreement with the stakeholders (orally or in writing) about their jobs, responsibilities, authority, etc.
Such agreements lead to rights and responsibilities for stakeholders. Effective stakeholder management increases their motivation and guards against conflicts. Stakeholders should be actively involved in the project, and not only informed about decisions and results.

EU 3.2 Categorization of Requirements according to the Kano Model (L2)

During elicitation of requirements it is important to know which of the requirements are most important to achieve customer satisfaction. According to professor Noriaki Kano, customer satisfaction can be classified in 3 categories:

- Base factors
- Performance factors
- Excitement factors

EU 3.3 Elicitation Techniques (L2)

Different elicitation techniques are needed to find conscious, unconscious and subconscious requirements of stakeholders. The choice of technique is influenced by risk factors (human risks, organizational risks, risks in the domain) and by the granularity and the degree of detail needed for the requirements. For different systems and products, different techniques can be used, such as:
- Questioning techniques (e.g.. interviews, questionnaires)
- Creativity techniques (e.g. brainstorming, paradox brainstorming, change of perspective, analogies)
- Document based techniques (e.g. system archaeology, perspective based reading, reusability of requirements)
- Observation techniques (e.g. field observation, apprenticing)
- Supporting techniques (e.g. mind mapping, workshops, CRC cards, audio and video recordings, use case modeling, prototyping)

Knowing about the strengths and weaknesses of the different techniques and applying them effectively are key skills for requirements engineers. The best results can be achieved by combining different elicitation techniques.

EU 4 Requirements Documentation (L2)

Duration: 2 hours
Terms: Requirements Document, Requirements Specification

Educational Objectives:
EO 4.1.1   Knowing key reasons for requirements documentation
EO 4.2.1   Knowing three perspectives (views) of functional requirements
EO 4.2.2   Knowing advantages and disadvantages of natural language documentation
EO 4.2.3   Knowing the most important model based documentation techniques
EO 4.2.4   Knowing the advantages of mixing documentation techniques
EO 4.3.1   Knowing the advantages of standardized document structures
EO 4.3.2   Knowing the four most widespread document structures
EO 4.3.3   Knowing important chapters for a tailored standard structure
EO 4.4.1   Knowing activities relying and building on requirements documents
EO 4.5.1   Mastering and using quality criteria for requirements documents
EO 4.6.1   Mastering and using quality criteria for requirements
EO 4.6.2   Knowing the two most important style guidelines for requirements
EO 4.7.1   Mastering and using contents and meaning of a glossary
EO 4.7.2   Mastering and using rules for handling glossaries

EU 4.1 Structuring Documents (L1)

During RE it is necessary to document important information. Any more or less formal way of capturing requirements is called a documentation technique. This includes everything from writing various styles to using formal diagrams with defined semantics.
During the life cycle of a requirements document many people come in contact with the documentation. Documentation is a key supporting feature for goal-oriented communication. This support is necessary because requirements are long-lasting; they may be legally relevant and they should be accessible to many people. Requirements documents may become very complex.

EU 4.2 Documentation Style Alternatives (L1)

For the functional requirements — which are a key part of a requirements document — three perspectives of the planned system can be documented:

- ⯈ A structural perspective
- ⯈ A behavioral perspective
- ⯈ A functional perspective

All three perspectives can either be described using natural language, or modeled using conceptual models that are specific for the perspectives.
Therefore, your alternatives for creating requirements documents are:
- Using pure natural language documents
- Using conceptual requirements models like use case models, class models, activity models and state models (cf. chapter 6)
- Using mixed forms of requirements documents, combining natural language with models.

EU 4.3 Document Structure (L1)

The requirements about the proposed system are the essential parts of a requirements document. Depending on the intended purpose of requirements documents, they often contain information about the system context, approval conditions or characteristic properties of the implementation, in addition to the requirements information. In order to allow proper handling, the content of such documents needs to be structured accordingly.
Reference structures propose a more or less complete and flexible field-tested content structure for requirements documents. Common reference structures for requirements documents are among the following:

- IEEE 830–1998 (Reference structure for "Software Requirements Specification")
- IEEE 1233–1998 (Reference structure for "System Requirements Specification")
- Volere...

In practice it appears that there are a lot of positive outcomes with using reference structures for requirements documents. For instance, using reference structures simplifies the usage of the requirements documents in subsequent development tasks (e.g. definition of test cases). Generally the reference structures cannot be transferred to another requirements document as-is, as the content structure has to be adapted to domain-, company- or project-specific conditions.

EU 4.4 Using Requirements Documents (L1)

During software development projects requirements documents are the basis for a lot of different tasks, among them

- Planning
- Architecture design
- Implementation
- Test
- Change management
- Operation and maintenance
- Contract management

EU 4.5. Quality Criteria for Requirements Documents (L2)

In order to serve as a basis for follow-up activities in the development process a requirements document has to satisfy some quality criteria, for example:

- Unambiguousness and consistency
- Clear structure
- Modifiability and extensibility
- Completeness
- Traceability

EU 4.6 Quality Criteria for Requirements (L2)

In addition to the quality criteria for documents, each individual requirement should conform to requirements' quality criteria. Such qualities are:

- harmonized
- prioritized
- unambiguous
- valid and current
- correct
- consistent
- testable
- implementable
- traceable
- complete
- understandable

Besides the quality criteria there are two fundamental rules for natural language requirements that support readability:

- Short sentences and paragraphs
- Only one requirement per sentence

EU 4.7 Glossary (L2)

A frequent reason for conflicts during RE is a different interpretation of terms among the stakeholders. To avoid this problem it is necessary to capture all relevant terms in a glossary. A glossary is a collection of definitions of

- specific terms of the domain
- abbreviations and acronyms
- everyday words that have specific meaning in the context of the project
- synonyms
- homonyms

Some rules should be obeyed when working with a glossary:

- The glossary should be managed centrally
- The responsibilities for maintaining the glossary have to be clarified
- The glossary should be maintained during the whole project
- The glossary should by accessible to everyone in the project
- The use of the glossary is mandatory
- The origin of the terms should be mentioned in the glossary
- The glossary must be agreed upon among the stakeholders
- The glossary entries should have identical structure

It is advisable to start early with the glossary in order to reduce adaptation work later on.

EU 5 Documentation of Requirements using Natural Language (L2)

Duration: 1 hour
Terms: Requirements Template

Educational Objectives:
EO 5.1 Mastering and using the five transformation processes when perceiving and writing natural language requirements, as well as the consequences for phrasing requirements
EO 5.2 Mastering and using the five steps of writing requirements using a natural language template

EU 5.1 Language Effects (L1)

Natural language is often ambiguous and interpretable. When you use natural language for specifying requirements you have to be aware of that. In our perception and during documentation of the perceived information, transformation processes occur. Since these transformations are well known and follow a set of rules, requirements engineers can make use of these rules. By asking the right questions a requirements engineer can determine what the originator of the requirement really meant.

The five most relevant transformations for requirements engineers are:

- Nominalization (using nouns for complex processes)
- Nouns without precise qualification
- Universal quantifiers ("all" without precise reference)
- Incompletely specified conditions
- Incompletely specified verbs

EU 5.2 Constructing Requirements using a Template (L2)

There is a simple way to avoid the transformation problems mentioned
above when writing requirements. It is based on a template for complete and
unambiguous sentences. This template effectively supports the author to create
precisely phrased requirements.
Here are the five steps to write requirements according to the template:

- Determine legal binding force
- State the core of the requirement
- Characterize the activity of the system
- Include the objects to be manipulated
- Include logical and temporal conditions

The fixing of the binding force by using the verbs "shall", "should", "will" can be
done in the text of the requirement. If the binding force changes, the requirements
change too. Using attributes is another possibility for documenting the binding
force of requirements.
You will come up with the best results not by following the template slavishly,
but by training your requirements engineers using this method, and by using the
template as a tool.

EU 6 Model-based Documentation of Requirements (L2)

Duration: 5 hours
Terms: Model

Educational Objectives
EO 6.1.1   Knowing models and their capabilities
EO 6.1.2   Knowing defining elements of a conceptual modeling language
EO 6.1.3   Knowing the advantages of using requirements models
EO 6.2.1   Knowing the importance of goals in the RE process
EO 6.2.2   Knowing two kinds of goal decomposition techniques
EO 6.2.3   Mastering and using either-or-trees to model goals
EO 6.3.1   Mastering and using use case diagrams
EO 6.3.2   Mastering and using use case specifications
EO 6.4.1   Knowing the three modeling views for requirements
EO 6.5.1   Knowing the structural view for requirements modeling
EO 6.5.2   Mastering and using entity relationship diagrams and UML class
                diagrams
EO 6.6.1   Knowing the functional view for requirements modeling
EO 6.6.2   Mastering and using data flow diagrams and UML activity diagrams
EO 6.7.1   Knowing the focus of behavioral requirements models
EO 6.7.2   Mastering and using UML state charts

Note: In this chapter the cognitive level L2 ("mastering and using") does not comprise the verbs "create", "design", "develop", "formulate"; i.e. students should be able to read the models and compare them with natural language. Creating such models is part of the IREB advanced level module on "requirements modeling".

EU 6.1 Models (L1)

Using models makes it easier to grasp information about features of the system and their relationships. It also helps to understand them faster and document them unambiguously. A model is an abstraction of existing reality or a plan for reality to be created. Models have three important characteristics:

- Representation characteristic: models represent reality
- Abstraction characteristic: models abstract interesting facts from reality
- Pragmatic characteristic: models are developed for a specific purpose

RE uses a variety of conceptual models that normally represent reality using a set of graphical elements. Conceptual modeling languages are used, defined by a given syntax (modeling elements and their legal combinations) and semantics (the meaning of the modeling elements).

Requirements models are conceptual models that document requirements for the system to be developed. Among the advantages of using models to document requirements instead of natural language are:

- It is easier to understand and memorize pictures instead of text
- The models allow to concentrate on one perspective of the system
- By suggesting a modeling language for a defined, specific purpose the appropriate abstractions of reality can be stipulated

A well-defined mixture of natural language and requirements models combines the advantages of both ways of documenting requirements.

EU 6.2 Goal Models (L2)

A goal is an intentional description of one or more stakeholders about a wanted, characteristic property of the system to be developed (or the project that is responsible for developing the system.)
Thus goals are the highest level requirements, i.e. the most abstract formulation of requirements. All other requirements should be traceable back to these goals. Often they are defined using natural language. But they can also be documented in terms of models.
Goals (similar to any other more detailed requirement) can be decomposed into sub goals. We distinguish two kinds of decomposition:

- "AND decomposition": all sub goals must be achieved in order to meet the overall goal
- "OR decomposition": at least one sub goal has to be achieved in order to meet the overall goal

Such decomposition relationships are often modeled in form of and/or-trees.

EU 6.3 Use Case Models (L2)

Use cases help to examine and document the functionality of planned or existing systems from the perspective of users of the system. The use case approach uses two documentation techniques that complement each other:

- Use case diagrams and
- Use case specifications

Use case diagrams are easy-to-understand models, displaying the functionality of the system under study from an outside point of view. They can show some relationships among the functionality and the relationship with the context. Typical modeling elements for use case diagrams include:

- Actors (people or other systems) in the context of the system
- The system boundary
- Use cases

Different types of relationships between the above mentioned elements.
Use case specifications augment the use case diagrams by more precisely describing important characteristic properties of the use cases. Generally, for each use case you fill in a predefined template that includes the following paragraphs:

- A unique id for the use case
- The name of the use case
- A description of the use case
- The triggering event
- The actor
- The result
- Pre- and post-conditions
- Different kind of scenarios. These scenarios specify the required behavior, i.e. the sequence of steps to be performed by this use case. We distinguish a main scenario and potentially some alternative scenarios. You can also describe the behavior of the use case in case of exceptions (exception scenarios)

EU 6.4 Three Modeling Views (L1)

Requirements models can be classified into three categories based on the key abstractions they use. These three views are slightly overlapping to allow for controlled redundancy that can be used for checking the models.

- Structural View
- Functional View
- Behavioral View

Typical representatives for the structural perspective are entity relationship models (ERM) and UML class models. For the functional view, data flow diagram or UML activity diagram (including object flow) are popular. Examples for the behavioral view are state transition diagrams or UML state charts.

EU 6.5 Structural Requirements Models (L2)

In the structural view you document the structure of the systems' data as well as usage and dependency relationships inside the system context. Traditionally entity relationship diagrams have been widely used for modeling. They include three kinds of modeling elements:

- Entity types
- Relationship types
- Attributes

You can add the cardinalities to the relationships to document how many entities of this entity type are involved in the relationship (e.g. 1..M, M..N)
Today, ERM models can be substituted by UML class models to portray the structural view. A class diagram consists of a set of classes and associations between them. So, UML class diagrams consist of:

- Classes
- Associations (annotated with multiplicities and roles)
- Aggregation and composition relationships (whole-part structures)
- Generalization relationships (is-a relationship)

EU 6.6 Functional Requirements Models (L2)

Models for the functional view focus on processing input data from the environment of the system to create output data of the system. The coarsest (i.e. most abstract) form of functional models are the use case models mentioned in section 6.3.
Other functional models include data flow diagram — as defined in "Structured Analysis" according to Tom DeMarco. They use the following modeling elements:

- Processes (Nodes)
- Data flows
- Data stores
- Terminators (sources and sinks of information)

Since data flow diagrams normally exclude control flow between the processes and internal specification of the nodes they can be augmented by additional elements.

So-called mini specifications (mini-specs) are used to describe the internal behavior of individual nodes.

In UML 2.x data flows can be added to activity diagrams in the form of objects and object flows between activities. In that form they can be used as an alternative to data flow diagrams. Normally activity diagrams focus on the activities of a process and the control flow between them. In addition to decision nodes (if-then-else, case), activity diagrams allow the modeling of parallel (concurrent) execution of functions using fork and join constructs.

The most important modeling elements for UML activity diagrams are

- Actions
- Start and end nodes
- Control flow
- Object flow
- Decision nodes and re-uniting alternative control flows
- Fork and join
- Hierarchical decomposition

EU 6.7 Behavioral Requirements Models (L2)

The dynamic behavior of a system can be modeled using models for the behavioral view. In this view you concentrate on system states and on events that can change that state. UML state charts use the following modeling elements:

- State
- Specific start and end states
- Transition
- Concurrent state decomposition

Eu 7 Checking and Reconciling Requirements (L2)

Duration: 2 ½ hours
Terms: none

Educational Objectives
EO 7.1.1   Knowing what it means to check requirements
EO 7.2.1   Knowing about conflicts between requirements
EO 7.3.1   Knowing three quality aspects of requirements
EO 7.3.2   Mastering and using check lists for the three quality aspects "content", "documentation" and "level of agreement"
EO 7.4.1   Knowing the six principles for checking requirements
EO 7.4.2   Mastering and using the principles of checking requirements
EO 7.5.1   Knowing techniques for checking requirements
EO 7.5.2   Mastering and using the following checking techniques: commenting requirements documents, inspection, perspective based reading, checking via prototypes and using check lists

EO 7.6.1   Knowing activities for reconciling requirements
EO 7.6.2   Knowing types of requirements conflicts
EO 7.6.3   Knowing different conflict resolution techniques
EO 7.6.4   Knowing documentation techniques for conflict resolution

EU 7.1 Basics for Checking Requirements (L1)

The major goal of checking requirements is to find out whether they conform to quality criteria (e.g. correctness or completeness) that have been set beforehand. You want to detect and correct potential shortcomings as soon as possible. Since requirements documents are the basis for all further development activities, any kind of undetected error made during requirements analysis drastically increases the costs of development. This is because not only do requirements errors have to be corrected, but other artifacts like architectural design, source code and test cases also have to be reworked.

EU 7.2 Basics of Reconciling Conflicting Requirements (L1)

Unresolved conflicts in requirements can mean that some requirements of a group of stakeholders cannot be implemented, that the system is not used at all or is only marginally accepted by the users. The goal for reconciling conflicts within the requirements is to create a common and agreed understanding of the requirements among all relevant stakeholders.

EU 7.3 Checking Requirements Quality (L2)

We distinguish between three quality aspects of requirements: content, documentation and level of agreement. For each requirement each of these aspects can be reviewed using a set of checking criteria.
The eight checking criteria for the quality aspect "content" are:

- Completeness of the requirements document
- Completeness of each individual requirement
- Traceability
- Correctness and adequacy
- Consistency
- No premature design decisions
- Verifiability
- Necessity

The five checking criteria for the quality aspect "documentation" are:

- Conformity with the document format
- Conformity with document structure
- Understandability
- Clearness and unambiguousness
- Conformity with documentation rules

The three checking criteria for the quality aspect "level of agreement" are:

- Reconciliation
- Reconciliation after change
- Conflicts solved

EU 7.4 Principles for Checking Requirements (L2)

Checking requirements is based on a couple of principles. These principles ensure that during checking a maximum number of errors in the requirements can be identified. The six principles for checking requirements are:

Involve the right stakeholders
Separate error discovery and error correction
Check from different points of view
Switch between different styles of documentation
Construct development artifacts based on the requirements
Repeat checks

EU 7.5 Techniques for Checking Requirements (L2)

There are several techniques for systematic checks of requirements. They can be used in combination to gain a relatively complete result with respect to predefined quality criteria. Among the techniques are:

- Expert reports
- Inspections
- Walkthroughs

Additional techniques can be used, like:

- Perspective based reading
- Checking via prototypes
- Use of check lists

EU 7.6 Techniques for Reconciling Conflicting Requirements (L1)

Reconciling conflicting requirements aims at achieving a common understanding about the requirements among all relevant stakeholders. The tasks during reconciliation are:

- Conflict identification
- Conflict analysis
- Conflict resolution
- Documentation of conflict resolutions

During conflict analysis we identify different kinds of conflicts and then apply different strategies to resolve the conflicts. Kinds of conflicts are:

- Conflicts about the content
- Conflicting interests
- Value conflicts
- Relationship conflicts
- Structure conflicts

In projects, conflicts often have mixed reasons. When trying to resolve the conflict you should include all relevant stakeholders. For conflict resolution a number of techniques are available:

- Agreement
- Compromise
- Voting
- Forming Variants
- The boss is always right
- Consider all facts
- Plus-Minus Interesting
- Decision matrix

After resolving a conflict one way or another, the result should be documented. This documentation should include the origin of the conflict, the stakeholders involved, opinions of the different stakeholders, the means of resolving the conflict, potential alternatives, the decision and the reason for the decision.

EU 8 Requirements Management

Duration: 2 ½ hours
Terms: none

Educational Objectives
EO 8.1.1   Knowing purpose and definition of requirements attributes
EO 8.1.2   Knowing important types of requirements attributes
EO 8.2.1   Mastering and using views based on attributes
EO 8.3.1   Knowing techniques for prioritizing requirements
EO 8.3.2   Mastering and using techniques for prioritizing requirements
EO 8.4.1   Knowing advantages of requirements traceability
EO 8.4.2   Mastering and using different kinds of traceability
EO 8.4.3   Mastering and using forms of representation for traceability
               relationships
EO 8.5.1   Mastering and using versioning of requirements
EO 8.5.2   Mastering and using requirements configurations
EO 8.5.3   Mastering and using requirements baselines
EO 8.6.1   Knowing how to handle change requests
EO 8.6.2   Knowing the duties of a change control board
EO 8.6.3   Knowing all the elements of change requests

EO 8.6.6   Mastering and using different types of change requests
EO 8.6.7   Mastering and using a process to manage change requests

Eu 8.1 Adding Attributes to Requirements (L1)

To manage requirements over the whole system life cycle, it is necessary to collect information about the requirements in a structured way. The definition of the attribute structure for requirements can be done in tabular form or by creating an informational model of the requirements attributes.

Typical attributes are:

- Identifier (unique ID)
- Name
- Description
- Source
- Stability
- Criticality
- Priority

The "legal binding force" can be added to the requirement by using an attribute to provide additional information.
Attribute schemes should be tailored to the needs of the project, including project constraints. Among them are:

- Specific characteristics of the project
- Company specific regulations
- Domain specific regulations
- Constraints of the chosen development process

Eu 8.2 Creating Views of Requirements (L2)

In practice, we find that the number of requirements in a project grows constantly; as does the number of requirement relationships. To cope with the complexity of a large set of requirements, it is necessary to allow individuals to filter requirements based on their needs, which reduces the amount of information displayed. We distinguish two kind of filtering mechanisms:

- Selective Views: showing selected subsets of attributes for selected requirements (based on defined selection criteria)
- Compacted and compressed views: showing compressed information about selected requirements (again according to defined selection criteria)

EU 8.3 Prioritizing Requirements (L2)

Requirements can be prioritized at different points in time and in different activities according to different criteria. Preparing for prioritization of

requirements is based on a simple process:

- Define goals and constraints for prioritization
- Define prioritization criteria
- Select the relevant stakeholders
- Chose artifacts to be prioritized

Based on the results of the above activities, you can use one or more prioritization techniques. Among the techniques are:

- Ranking and top-ten selection
- One criterion classification
- Kano classification
- Carl Wiegers' priority matrix

EU 8.4 Tracing Requirements (L2)

During requirements management you should record, organize and maintain traceability information. The benefits of managing requirements traces are:

- Ease of traceability
- Identification of system's features that are not needed
- Identification of requirements not needed
- Support of impact analysis
- Support for reusability
- Support for accountability
- Support for system maintenance

There are three kinds of trace relationships that should be considered:

- Pre Requirements Specification Traceability
- Post Requirements Specification Traceability
- Traceability between requirements (e.g. levels of abstraction, derivations, …)

You should only trace what you really need. Traces can be established with a variety of techniques, such as:

- Textual references and hyperlinks
- Traceability matrixes
- Traceability graphs

EU 8.5 Requirements Versioning (L2)

Versioning and configuring requirements records the status of requirements and requirements documents available at certain points in the life cycle of systems and products, and reconstructs that status on demand. A version number for a requirement should include the following two parts:

- ▶ Version
- ▶ Increment

Requirements configuration combines a defined set of logical coherent requirements. Only one version of each requirement may be contained in a given requirements configuration. Requirements configurations are formed in two dimensions:

- ▶ Product dimension: the collection of all requirements in a given requirements configuration
- ▶ Version dimension: the different versions of one requirement

Requirements configuration should have these characteristics:

- ▶ The requirements of a configuration should be logically connected
- ▶ Requirements in a configuration should be consistent
- ▶ The configuration should have a unique ID
- ▶ Requirements attached to a configuration must no longer be changed
- ▶ Configurations allow one to return to earlier versions of it.

Requirements baselines are selected requirements configurations containing stable versions of requirements. They often define the releases of the system.

EU 8.6 Managing Change Requests (L2)

Requirements constantly change and evolve over the life cycle of a system. Systematically managing requirements change is part of a controlled requirements management process. In this process, the change control board is responsible for managing incoming change requests. The tasks of a change control board comprise the following:

- ▶ Classification of each incoming change request
- ▶ Determining the effort needed for the change
- ▶ Judging cost and benefit of the change request
- ▶ Defining new requirements based on the change request
- ▶ Deciding whether to accept or decline the change request
- ▶ Prioritize the accepted change requests
- ▶ Allocate changes to a baseline (and projects affected by this baseline)

Typical participants in a change control board are change managers, clients, architects, representatives of the users, quality staff and, of course, requirements engineers and business analysts.

For important changes to existing requirements, you can use the same mechanism: write change requests and send them to the change control board. The change request should contain the following information:

- A unique ID for the change request
- Title of the change request
- Description of the necessary change
- Rationale for the change
- Date of the change request
- Requester
- Priority as seen by the requester

You can distinguish three kinds of changes:

- Corrective changes
- Adaptive changes
- Exceptional changes

The process of managing changes includes the following steps:

- Impact analysis and assessment of the change
- Prioritization of the change request
- Allocation of the change to a change project
- Feedback to the requester about accepting or declining the change request

EU 9 Tool Support (L1)

Duration: 1 hour
Terms: none

Educational Objectives
EO 9.1 Knowing eight features that requirements management tools should
        provide
EO 9.2 Knowing five aspects to consider when introducing requirements tools
EO 9.3 Mastering and using seven aspects when evaluating tools

EU 9.1 Types of Tools (L1)

Tools specifically developed for RE are requirements management tools and requirements modeling tools. Beyond these, many other types of tools can be used to support the requirements process, including: test management tools, configuration management tools, WIKIs, office software, and general purpose graphics tools.
Specialized RE tools should at least

- maintain different kinds of information
- allow users to establish and maintain logical relationships between different kinds of information
- uniquely identify artifacts

- control access rights to the information in a flexible, but secure way
- support different views of the stored information
- organize the information (e.g. in hierarchies or based on attributes)
- make statistics available about the collected information generate various kinds of documents based on the collected information

Standard office tools support these features only to a limited extent. Specialized RE tools go beyond the capabilities of office tools, for example, by offering better support for traceability.

EU 9.2 Tool Introduction (L1)

An RE tool should only be selected when the RE process and techniques have been chosen. Successful tool introduction requires a clear understanding of responsibilities and processes in RE. The following aspects should be considered:

- Plan the resources needed for tool introduction
- Evaluate tools according to predefined criteria (see next section)
- Avoid risks by using the tool in pilot projects first
- Consider total costs for the tool (not only license fees)
- Training the users

EU 9.3 Tool Evaluation (L1)

There are many aspects to be considered when evaluating and choosing a RE tool. This process can be structured using these seven aspects:

- Project view (e.g. support for project planning)
- User view (especially usability)
- Product view (functionality)
- Process view (method support)
- Vendor view (e.g. services offered by vendor)
- Technical view (e.g. interoperability, scalability)
- Economic view (costs)

Evaluation criteria should be established for each aspect.

Appendix A: Glossary

This glossary contains the most important terms used in the IREB syllabus for the CPRE Foundation Level. A more complete glossary is planned for the next release of the syllabus.

Chapter 1: Introduction and Foundations

Requirement	According to IEEE a requirement is (1) a condition or capability needed by a user to solve a problem or achieve an objective (2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document (3) a documented representation of a condition or capability as in (1) or (2).
Stakeholder	A stakeholder of a system is a person or organization that (directly or indirectly) influences the requirements of the system to be developed.
Requirements Engineering	Requirements Engineering is a cooperative, iterative, incremental process, the goals of which are to make sure that (1) all relevant requirements are known and understood to a level of detail that is necessary (2) the involved stakeholders achieve a satisfactory level of agreement about the known requirements (3) all requirements are documented according to documentation guidelines or specified according to specification guidelines.
Functional Requirement	A functional requirement defines a function that has to be offered by the system to be created or one of its components.
Quality Requirement	A quality requirement defines a qualitative property that the system to be created or one of its functions has to offer.
Constraint	A constraint is an organizational or technical requirement that restricts degrees of freedom for designing and implementing the system to be created.

Chapter 2: System and System Context

System Boundary (a.k.a. Scope of Product)	The system boundary separates the planned system from its environment. It delimits the scope of the product that may be shaped and changed during the development project from aspects in the environment that are not allowed to be changed in this project.
Context Boundary (a.k.a. Scope of Work)	The context boundary separates the relevant part of the environment of the planned system from the irrelevant part, i.e. the part of the environment that has no impact on the planned system and thus no impact on the requirements for that system.
Context of the System	The context of the system is that part of the systems' environment that is relevant for understanding and defining requirements for the system.

Chapter 4: Requirements Documentation

Requirements Document/ Specification	A requirements specification is a document containing specified requirements, i.e. requirements in accordance with specification criteria.

Chapter 5: Documentation of Requirements using Natural Language

Requirements Template	A Requirements Template is a construction plan for the syntactic structure (= arrangement of words) of a single requirements sentence.

Chapter 6: Model-based Documentation of Requirements

Model	A model is an abstraction of existing reality or a plan for reality to be created.

Appendix B: List of Abbreviations

a.k.a.	also known as
EO	Educational Objective
ERM	Entity Relationship Model
EU	Educational Unit
IREB	International Requirements Engineering Board
L1	Cognitive Level 1
L2	Cognitive Level 2
RE	Requirements Engineering
UML	Unified Modeling Language

Publisher:

Contact: