REQB®

Approach to

Requirements Engineering

Version 1.0

2013

Overview of Changes

Version	Date	Comment
1.0	29/05/2013	First version

Acknowledgements

Editors: Folke Nilsson, Karolina Zmitrowicz.

Contributors: Alain Ribault, Beata Karpinska, Krzysztof Chytla, Eric Riou du Cosquer, Werner Henschelchen.

Table of Content

1 Introduction

1.1 REQB® Approach to Requirements Engineering

This document provides a short introduction to the Requirements Engineering processes, their context and their relationships with other development processes. The main aim of this document is to provide the reader with a general view of the structure of the Requirements Engineering process, its inputs, outputs and specific activities.

The document presents a REQB® definition of the Requirements Engineering processes. The REQB® approach is based on unified, standardized knowledge about Requirements Engineering. It follows international standards and utilizes best practices that have been adjusted to current market needs and real-life practices.

1.2 Context of Requirements Engineering

Requirements Engineering is a part of Software Engineering that focuses on identifying and analyzing requirements for a given business problem and proposing software and/or hardware solutions to meet the stated requirements in the best possible way.

Requirements Engineering is not performed in isolation. It is linked with other disciplines (Figure 1) and should be incorporated into the overall solution development process.

The starting point for Requirements Engineering is Business Analysis. In order to propose the best solution for a given business problem, it is important to define the problem correctly.

Business Analysis is a discipline that concentrates on identifying the business needs of an organization and determining solutions that will satisfy those needs and other business problems. These solutions may include development of software and/or hardware systems, business processes, organizational changes, etc. The business goals and needs identified during the Business Analysis are developed into requirements for the solution. Therefore, Requirements Engineering can be seen as a continuation or a part of the Business Analysis process. The output from Requirements Engineering is the solution design that is then implemented, tested and finally provided to the customer.

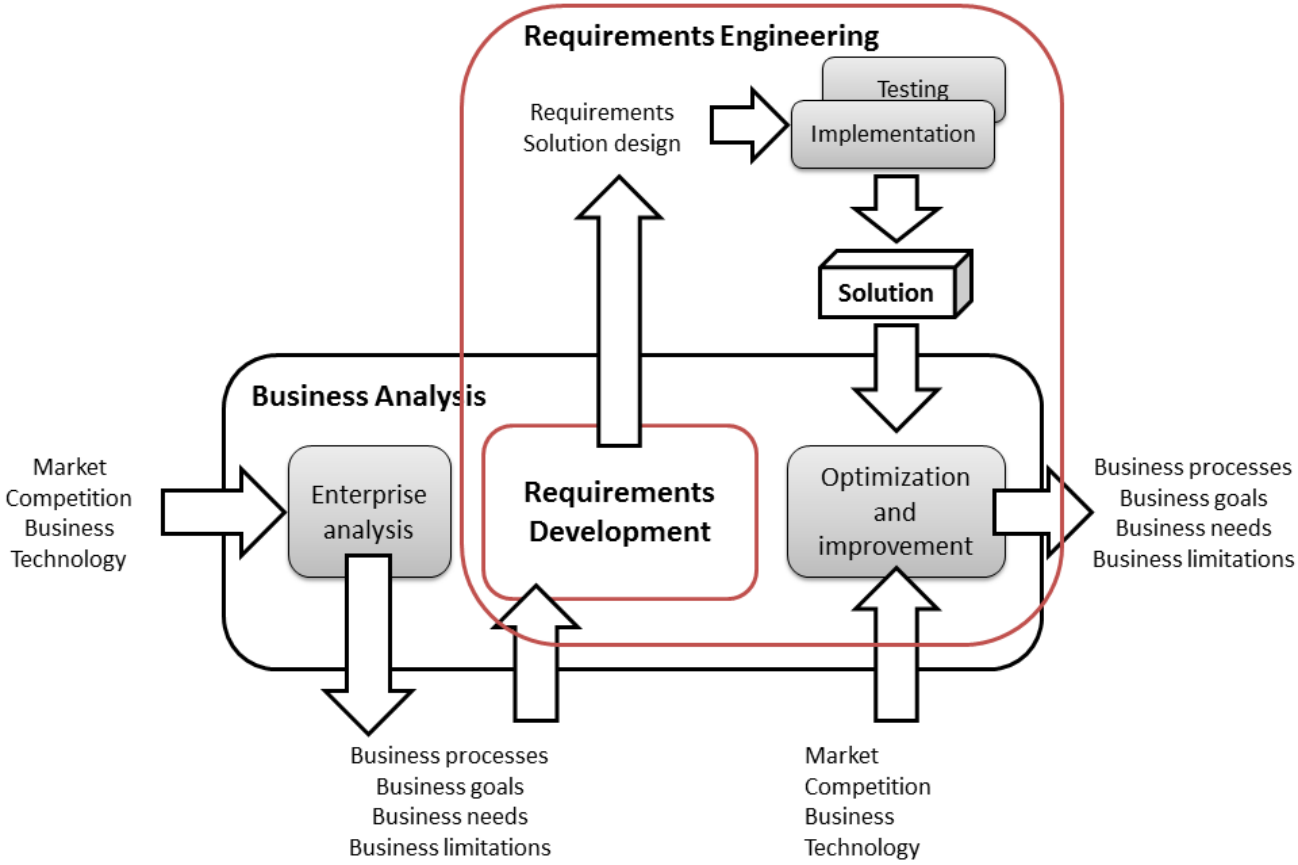

Figure 1 The context of Requirements Engineering

The implemented solution rarely stays unchanged for a long period of time. The market and business change, technology is evolving all the time and new connections within the business may appear. That is why Business Analysis–and Requirements Engineering–continues during the whole lifecycle of the solution. After delivering the solution Business Analysis then seeks for improvements that would allow increasing the market, improving the efficiency of operating or gaining competitive advantage. New business needs may appear. Those business needs are then developed by Requirements Engineering to propose new, improved solutions or to extend the existing solution with new, better features. For each identified business need the cycle begins anew.

1.3 Basic Terms and Definitions

To understand the discipline of Requirements Engineering and its main inputs and outputs, it is important to define the most important terms. This chapter introduces several definitions required to understand the Requirements Engineering process. The full list of terms can be found in the REQB© "Standard Glossary of Terms used in Requirements Engineering".

The main input to Requirements Engineering is a **business problem**. A business problem is the description of what a customer wishes to do in order to realize or improve its business processes.

The aim of Requirements Engineering processes is to define a **solution** for a given business problem. A solution is the answer to the needs of a customer. Those needs are usually expressed as business requirements and are inputs to Requirements Engineering. A solution may be a software system, service, new or improved business processes, etc. In general, a solution is the implementation of the requirements.

The output of the development process is called a **product**. A product is defined as a composition of software, hardware and other outputs of the production process such as documentation.

A **requirement** can be understood as a documented representation of a feature requested by a user or other stakeholder to solve a given business problem or achieve a specific objective. Usually requirements are documented together with supporting information – **requirement attributes**. The most important attributes for requirements are: **commitment**, expressing the degree of obligation for meeting the requirement and **priority,** expressing the business urgency of the requirement.

Defining and documenting requirements clearly and unambiguously is very important. They represent the customer's expectations and serve as the foundation for further development and testing. Requirements define the solution boundaries, scope of delivery, and may become the basis for contractual services.

Requirements may be related not only to a product, but to a business process or an organizational structure as well. For instance, the purpose of a project may be not only implementing components but also improving the organization's business process itself. Therefore, the common **classification of requirements** divides requirements into **product requirements** and **process requirements**.

Process requirements are either constraints on the development of the solution ("how to do the product") or requirements related to business processes while product requirements are related to the product itself and they describe "what is to be done". The specific types of product requirements are functional and non-functional requirements. **Functional product requirements** specify what the product should do; they describe functions of the product. **Non-functional product requirements** specify how the product should perform its functions (the quality attributes of the product).

In addition to the classification defining the different types of requirements, there is also a classification presenting the different abstraction levels of requirements (Figure 2). These abstraction levels represent the different levels of requirements analysis as well and they address different areas – either the business problem or the solution.

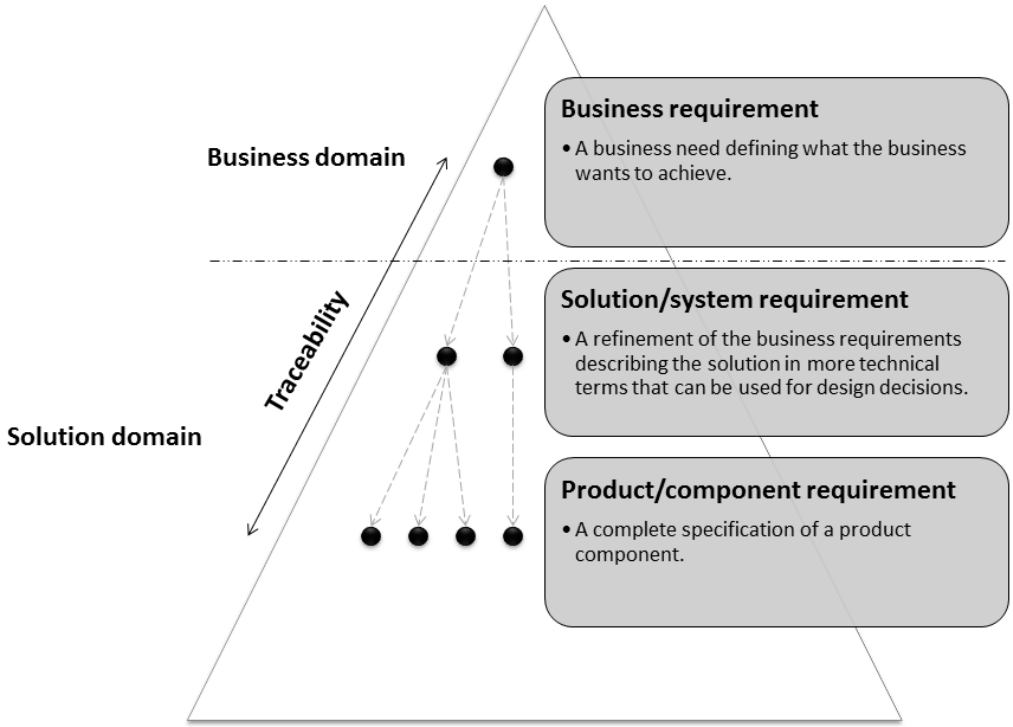

Figure 2 Abstraction levels of requirements

When working with requirements on the different levels, it is important to define and maintain **traceability**. Traceability is a connection that exists between artifacts on the different abstraction levels, for example, between business requirements and solution requirements. Traceability may also be useful between requirements and other artifacts such as test cases.

As one of the main purposes of requirements is to express stakeholders' expectations and needs regarding the planned solution, requirements must express real value in terms of perceived benefit for the stakeholders. To reduce the risk of having problems with requirements and to ensure the best possible quality of the future solution, **quality criteria for requirements** should be applied. Some of the most important quality criteria for requirements state that each requirement must be: correct, feasible to implement, unambiguous and verifiable (i.e., possible to verify if it has been implemented correctly).

A common solution for checking and monitoring the quality of requirements is **validation** and **verification**. Validation is usually conducted with the support of a customer and aims to confirm that the requirements or the requirements specification correctly represents the needs of the customer. Verification is a comparison of an intermediate product with its specifications to determine if the product was developed according to the specifications. The most common techniques for verification and validation are reviews which use common checklists covering the major quality criteria, and testing.

The whole process of developing and managing requirements to create a solution is called **Requirements Engineering**. Requirements Engineering (RE) is a discipline concerned with collecting and managing the requirements of the future product. Detailed characteristics and specific activities of Requirements Engineering will be described in the next chapter.

2 Requirements Engineering Process

2.1 Introduction

As mentioned in the previous chapter, Requirements Engineering includes processes needed for identifying, structuring and managing requirements. Specific activities covered by the general Requirements Engineering process include the following:

- Requirements Elicitation Requirements Analysis
- Requirements Specification
- Requirements Verification and Validation
- Traceability of Requirements
- Configuration and Change Management
- Quality Assurance

The Requirements Engineering process is a structured set of the activities listed above. The activities are categorized as part of the Requirements Management process or part of the Requirements Development process, depending on the purpose and the phase of the solution development. A complete process description should include a definition of the relationships with other related disciplines and areas (for example, Business Analysis or testing), input to and output from the activities, the stakeholder responsibilities for the specific activities and outputs, any required competencies and specifications for tools to support the activities.

The structure of the Requirements Engineering process depends on different factors, such as the organization culture and maturity, or the development process model used. The REQB® recommends the following categorization:

- Generic Requirements Engineering process
- Requirements Engineering process in development process models
- Requirements Engineering process in maturity process models

The **Generic Requirements Engineering** process should be a starting point for each organization involved in solution development work as it provides the most crucial processes for handling requirements. This Generic Requirements Engineering process can be adapted to the specific needs of an organization and considers the development process model to be applied. The result of this adaptation is the **Requirements Engineering process in development process models**. In addition to development process models, maturity process models also exist. Those maturity models are mostly used to evaluate the current maturity of a specific organization and introduce improvements needed to increase efficiency of the organization. The Requirements Engineering process is one of the main areas for improvement and most maturity process models provide special requirements and rules to be applied when improving the current Requirements Engineering process. Therefore, we can define another variant of the generic Requirements Engineering process – **Requirements Engineering process in maturity process models**.

The following section focuses primarily on the generic Requirements Engineering process which can be used as a basis for any organization.

2.1.1 Generic Requirements Engineering Process

The generic Requirements Engineering process covers the following activities (Figure 3).

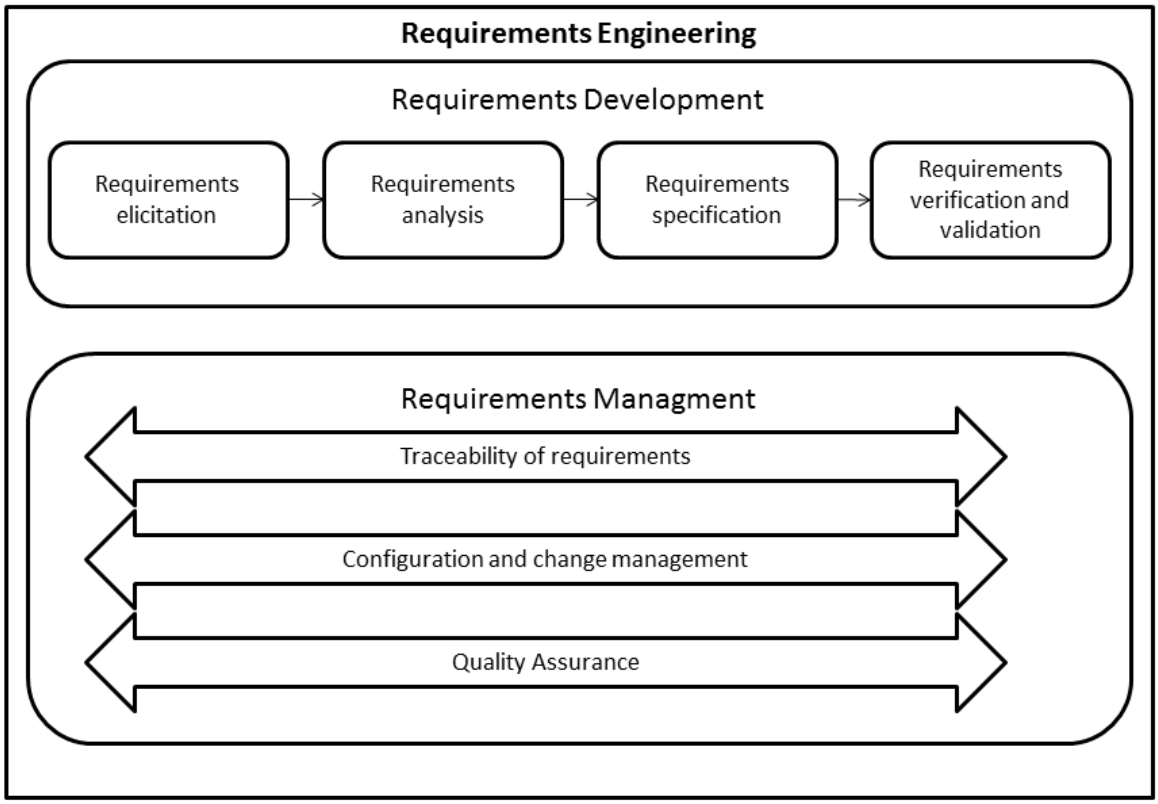

Figure 3 Generic Requirements Engineering process

Some of the specific activities can overlap and continue during the entire solution development effort.

The **Requirements Development** process and the **Requirements Management** process are interrelated. They can be seen as two sides of the same coin and cannot exist without each other. The Requirements Development process comprises all activities for the generation of the requirements and for the support of the development based on those requirements. The Requirements Management process provides the infrastructure and support to the Requirements Development process throughout the product lifecycle.

The Requirements Engineering process should be defined for a specific project or organization and tailored to the other supporting processes. It should be deployed with proper training and with appropriate tool support.

A common problem today is that very few organizations have a well-defined and standardized Requirements Engineering process. As there is no single process that is correct and applicable for all organizations, each organization should develop its own process based on the generic Requirements Engineering process defined by REQB®. The generic Requirements Engineering process should be adapted to the needs of a specific organization, taking into account the type of solutions being developed, business domain, business processes, organizational culture, and the level of competence and ability of the personnel responsible for the Requirements Engineering process.

Most Requirements Engineering activities can be supported by tools. A common application of tools is to facilitate the storage and administration of requirements. Most requirements management tools provide a common repository for requirements where requirements can be organized and traceability maintained. These tools can automate some mechanical activities and can provide different statistics and reports as well. Modeling tools allow the requirements engineer to model requirements in specific modeling notations such as UML or BPMN. Some modeling tools support generating requirements specifications from the models.

In addition to the tools that are dedicated to Requirements Engineering, other tools can be useful as well. Configuration Management tools can be used to define and maintain configuration integrity; Defect Tracking tools can be used to manage defects found in requirements. In case of complex projects, Project Management tools can be used to manage the Requirements Engineering project itself.

2.1.2 Requirements Development Process

As defined earlier, the Requirements Development process covers the following activities:

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Verification and Validation

These activities should not be considered as separate individual phases without any interconnections. Very often there is a high market pressure which forces shorter development cycles and increases the risk of frequent changes, upgrades or revisions to the solution under development. Therefore, it is rare to be able to implement the Requirements Development process as a linear process in which requirements are collected from the stakeholders, analyzed, specified and handed over to the development team which then creates the solution model and validates the requirements. In reality, requirements typically iterate toward a level of quality and detail that is sufficient to permit design and procurement decisions to be made. In most cases the requirements change as the product evolves which means that proper steps have to be taken to track and manage the changes.

The first step in the Requirements Development process is **Requirements Elicitation**. The main aim of this activity is to collect requirements from all interested stakeholders – not only users and sponsors, but the project team, the market and other external sources as well. To elicit requirements it is required to understand what problem is to be solved. This includes identifying the stakeholders and establishing (or understanding previously identified) high level business goals. Stakeholders provide the requirements and limitations; therefore it is very important to identify all the stakeholders in order to elicit the full set of requirements. Defined business goals set the boundaries and keep the focus for what is to be done – the goals control the solution scope. Requirements collected during Requirements Elicitation should support meeting the business goals.

Specific techniques that can be used for Requirements Elicitation include interviews, hosting representatives of the customer on site, and identifying requirements on the basis of existing business documents, observations, or requirements workshops. Many techniques exist and their application and usefulness depends on the objectives to be achieved. The best results are achieved when using a combination of different techniques.

Usually when eliciting requirements the main focus is put on functional requirements. This is a common mistake as non-functional requirements can be crucial for a specific solution (for example, for safety-critical solutions where reliability is a key characteristic). Non-functional requirements specify criteria that can be used to judge the operation of a product and they have a great impact on the customer's satisfaction in using the product.

The next step after Requirements Elicitation is **Requirements Analysis**. The main goal of the analysis is to create a solution for the implementation of the requirements. Creating the solution requires elaborating the business requirements into system/solution requirements and down to product component requirements, detecting and resolving conflicts between requirements and establishing the boundaries for the solution as well as the interactions of the solution with its environment.

The Requirements Analysis often includes verification and validation activities. To be able to propose the solution that meets the specified requirements, it is crucial to ensure that the requirements are correctly understood and approved by the stakeholders.

An important aspect of the Requirements Analysis is requirements acceptance and prioritization. Requirements acceptance is a formal agreement that the content and scope of the requirements are accurate and complete. The requirements should be accepted at different phases during the solution development as formal agreement is the base for the further activities. Requirements prioritization allows establishing each requirement's relative importance and may dictate the order of implementation so as to complete the most crucial requirements first so that their implementation can be verified as soon as possible.

Requirements Development is not only a process of discovering and analyzing requirements, it is also a process of facilitating effective communication of these requirements to the different stakeholders. The way in which requirements are documented plays an important role in ensuring that they can be read, analyzed, changed and validated. Documenting requirements is called **Requirements Specification**. The requirements specification is a document that formalizes the scope and content of the requirements to be

implemented in the planned solution and usually serves as a basis for a formal agreement on the requirements. Requirements Specification can be considered an activity or a final result of Requirements Analysis.

Solution and System Modeling is often a part of Requirements Analysis and Specification as it aims to develop models of a real-world problem to be solved by the specific solution. The main challenge for the Solution and System Modeling is to choose and develop appropriate diagrams which can form models describing the business solution from different points of view. These models should not only correctly represent the solution, but should be understood by both the customer and the supplier representatives.

The Solution and System Modeling can use several types of models, but in general three basic levels of models exist:

- Requirements model
- Solution model
- Conceptual model

Different levels of modeling and different views of the solution can be described by different diagrams. To have the full image of the solution, usually a combination of different views is used. This results in using different diagrams describing the solution model from specific perspectives.

To ensure the best quality of requirements, their specifications and models, validation and verification procedures should be used. **Requirements Verification and Validation** should be done continuously during the development of a solution to ensure that the product being developed meets quality criteria and will satisfy the stakeholders' needs. The best practice is to plan and perform verification and validation of requirements from the early phases of the solution. Verification and validation activities should also ensure that the requirements documents conform to company standards (templates) and are understandable, consistent, and complete. It also is important to validate the models developed during the analysis and the specification.

As requirements are the basis for solution development and testing, their quality is crucial for the success of the project. Clear, complete, consistent and testable requirements reduce the risk of project (or even product) failure, as they allow careful and accurate testing. It is recommended to involve testers in reviews of requirements as they can significantly help improve the quality of requirements by identifying weak points and possible defects.

Testability of requirements is supported by **Acceptance Criteria.** Acceptance Criteria describe criteria that must be met to approve the solution and should be agreed upon by both sides – the supplier and the customer – before starting the project. The best practice is to define Acceptance Criteria for specific requirements as a part of contract documentation.

2.1.3 Requirements Management Process

Requirements Management is a collection of managing and supporting activities which are needed to ensure that the Requirements Development process is executed properly during the product lifecycle. Change management and solution maintenance is also considered.

Generally Requirements Management deals with the following activities:

- Traceability of Requirements
- Configuration and change management
- Quality Assurance

The Requirements Management process operates in a larger context and has strong relations with other processes including Project Management, Analysis & Design, Configuration Management, Testing and Release Management.

Products are usually developed in the form of a project. Therefore, the project itself should be analyzed from a Requirements Engineering perspective.

According to well-recognized research (e.g., the Chaos Report), one of the main reasons why projects fail is due to issues with requirements. Neglecting Requirements Engineering can result in low quality requirements and – as a consequence – a low quality of the final product. Careful and structured Requirements Management is a necessary part of any project and should be considered as a part of **Project Management**.

An important challenge for product development is to handle the risks. This is usually done through the **Risk Management** process. The goal of Risk Management is to identify and then manage the risk according to the mitigation plan. During Requirements Elicitation and while developing the solution proposal, all associated risks should be defined so that the Risk Management process can properly address these risks.

Another important area of Requirements Management is **Traceability of Requirements.** This activity deals mainly with defining and maintaining traceability of requirements. Traceability is necessary as requirements are not stable – they continue to develop during the development lifecycle. Traceability provides a method for managing developing requirements and other artifacts related to those requirements.

Traceability also supports **Configuration and Change Management** as it allows analyzing the impact of changing a specific requirement or other configuration unit. In the case of more complex solutions, or businesses that change often, it is often not possible to manage changes in an effective way without traceability.

When defining the Requirements Management process, it is also necessary to define necessary **Quality Assurance** activities to be applied in order to ensure that the different Requirements Engineering processes and their products are of good quality.

As demonstrated above, Requirements Engineering deals with many tasks, from analysis of business processes, through identification, analysis and modeling of requirements, quality assurance, risk analysis and change management of requirements. Therefore some special roles for Requirements Engineering and specific competencies are needed.

The large variety of tasks has resulted in many different titles for people performing Requirements Engineering-oriented tasks. Different organizations use their own names, such as Requirements Engineer, Requirements Manager, Requirements Developer, Business Analyst, System Analyst, Solution Architect, System Architect, Designer, etc. Regional variants of these roles depending on culture, habits and traditions exist as well. A complication is that specific responsibilities and competencies needed for the roles mentioned above are rarely clearly defined or understood. To provide a common understanding of the responsibilities and roles, REQB® defined the roles of Requirements Manager and Requirements Developer. This classification results from the difference of the tasks performed in the Requirements Management and Requirements Development activities.

2.2 Requirements Engineering in Development and Maintenance Models

As mentioned in the Chapter 2.1 Introduction, the structure of the Requirements Engineering process depends on the development process model used. The purpose of this chapter is to provide examples of how different Requirements Engineering activities occur in some standard types of development models.

The general product life cycle (PLC) defines various phases of product development where the basic phases are planning, development, maintenance and the end of life. The development phase is often divided into some other phases, often performed iteratively. This generic PLC is a basis for different development models.

The most simple development process model – the **Waterfall model** – can be regarded as a linear representation of the generic Requirements Engineering process. This model is based on the assumption that it is possible to collect, define and analyze all the requirements in an early stage of development and that they will not a subject to change. The main weakness of this process model is poor flexibility which eliminated it from many modern projects where requirements are changing due to business, technology or market changes.

The **General V-model** has been a further development of the Waterfall model. The main improvement implemented in V-model is the distinction between development steps and testing steps, where each development step has a corresponding testing step. The model can be seen as a decomposition of requirements where each level has an associated testing activity.

This model is quite effective for Requirements Engineering in the development phases and gives a good base for Requirements Management with traceability and change management. However, Requirements Development in a V-model can be difficult as it is often impossible to ensure that all important requirements are found in early stages. Often the initial scope of the product increases due to

requirements coming from different stakeholders and surrounding systems or environments in later phases which can cause the need for extensive rework and change management.

In case of more complex or unstable solutions, it is recommended to use **iterative** or **incremental models.** The iterative/incremental approach is built on the fact that requirements are often difficult to specify, especially in commercial systems. Techniques for iterative and incremental development assume cooperation and partnership between developers and users. Requirements are often defined in terms of business objectives only – these business objectives are then elaborated in later iterations.

The incremental development model assumes that the list of requirements is divided into smaller parts and each part is developed in steps or increments. The idea is that each part can be implemented and, in best case, go live so that the time to market is short. It is also possible to get immediate feedback on the delivered increments, so the outstanding, not yet developed increments, can be improved. In this way the risk can be reduced (especially comparing to a “one go” strategy). Even if the project is terminated before completion, stakeholders should get some benefits from already completed increments.

The iterative approach is especially useful for large projects with a huge number of requirements. The main idea of the iterative approach is to develop a solution through repeated cycles (which can be regarded as a mini-PLC) and in smaller portions at a time. The purpose of such organization is to make the project more controllable and to systematically monitor and address risk.

All the process models described above belong to so-called traditional approaches. The most common challenge for traditional approaches is supporting frequent changes. Most of traditional models do not support changing requirements in an effective way. As a solution for this problem, so called Agile models appeared. Agile approaches, which are a form of iterative and incremental approaches, are based on the Agile Manifesto presenting a set of value prepositions and principles used in Agile environments. The implications of these principles on Requirements Engineering are quite significant. In Agile environments the Requirements Management process is not as formal as in the case of traditional models and many of the activities normally performed under Requirements Development are not explicitly described but are expected to be performed under the responsibility of, for example, a Product Owner.

2.3 Requirements Engineering in Maturity Models

The Requirements Engineering process can be improved just as any other process defined as a part of product/solution development or maintenance. Improvements can be applied to each of the Requirements Engineering activities. For example, Requirements Elicitation may be improved to collect business requirements in a more effective way. In order to do so, the organization may introduce some techniques ensuring the requirements are gathered faster, are complete and agreed upon by most stakeholders. Some of such techniques are interviews, brainstorming, initial prototyping using Personas, and scenarios.

Assessment of the current level of process maturity and further improvement of specific areas can be supported by **maturity models**, which often use defined maturity levels. Maturity levels serve for the identification, assessment and improvement of the process maturity (process assessment and process improvement).

Examples of maturity models that can be applied for Requirements Engineering processes are ISO/IEC 15504 (SPICE – Software Process Improvement and Capability Determination) or Capability Maturity Model Integrated (CMMI). Both models define five maturity levels for specific processes or areas and allow comparing the maturity of different organizations.

3 Conclusion

Requirements Engineering is one of the most important processes needed to ensure the quality of any product and the success of the system or software development projects. It corresponds to a set of activities, methods and skills briefly mentioned in this introduction document bur further explained in the REQB Foundation Level Syllabus.