# Rapid Requirements Engineering – Does a Specification Always Need to Come at the Start?

Michael Gerdom and Dr. Uwe Rastofer,
method park Software AG

Michael Gerdom studied computer science at the University of Erlangen-Nuremberg. He has gained several years' practical experience from customer projects in the area of implementing object-oriented technologies and UML in embedded systems environments. He is responsible at method park Software AG for heading the area of Training & Consulting. His current main areas of work as a consultant lie in system and software architecture, as well as in requirements engineering. He is a founding member of the iSQI German Software Architecture Board.

Dr. Uwe Rastofer studied computer science and gained his PhD in it at the University of Erlangen-Nuremberg. He has gained several years' experience as a consultant, trainer, developer and project manager in the area of distributed and embedded systems. His main areas of work at method park are in consulting and training within the fields of requirements engineering, system and software architecture, UML and object-oriented software technologies.

## Abstract

Requirements and the way they are dealt with are decisive to the success of a project. This statement is never really questioned in modern software engineering circles. Why is it, then, that a systematic requirements engineering (RE) system is so rarely established? Where do the problems lie when it comes to implementing such a system? This paper outlines the challenges and how these may be met using the example of the automotive industry.

## Approaches for Improving Requirements Engineering

If improvement programs are viewed according to CMMI or SPICE, all important aspects of requirements engineering are outlined there. CMMI, for example, identifies two process areas, namely requirements management (RM) and requirements development (RD), for dealing with this issue. On the one hand, it is about dealing with the requirements, and on the other about actually identifying what these requirements are. Thus far everything seems rather simple.

A typical procedure in requirements engineering encompasses the following steps:

- Identifying the sources containing requirements
- Identifying stakeholders
- Analyzing existing documents
- Documenting the requirements in a clear form
- Tracking the status of the requirements (implementation, testing)
- Managing change requests

The main goals of these steps lie in understanding the customer requirements prior to the start of development and in the ability to deal with changes. To achieve this, the requirements must be documented in a suitable way.

The theoretical basis for procedures and methods within the area of requirements engineering is thus very clear. Yet it is the practical feasibility which is decisive to success.

## Typical Points of Departure for Individual Projects

In order to assess the feasibility of a requirements engineering approach, it is necessary to look at the point of departure for such projects. In the automotive industry, development projects typically do not begin right from scratch. There is almost always a previous project which needs to be adapted and expanded upon. In order to create a requirements specification, the following information sources are therefore available:

- An often loose collection of changes and expansions (new customer requirements) which additionally need to be fulfilled by the new system.
- Distributed knowledge among the developers as to how the existing system has been implemented and the reasons for this.
- More or less comprehensive and current documentation on the existing system.

In only a few cases is there really a requirements specification for the existing system which may be built upon for the system needing to be newly developed. In most cases, the requirements are not explicitly documented, but are available only implicitly as knowledge stored in the minds of the developers. Furthermore, this knowledge often represents a mix of requirements and solution strategies.

It must also be borne in mind that projects are always carried out under enormous time and cost pressures.

## Typical Procedures in Projects

The actual situation for executing follow-on projects is represented in the following diagram. In the case of a further development, decisions or assumptions are made based on customer requirements, the knowledge of the developers and the implementation to date. These decisions are then directly implemented, and perhaps also documented, in the design data. The colors in the diagram highlight how both the knowledge of the developers and the implementation phase are of a sufficiently high level of quality. The quality of the documents diminishes rapidly, however, from the design and architecture phases right through to the specification stage. In the procedure used to date, this applies both to the existing system and to the new system.

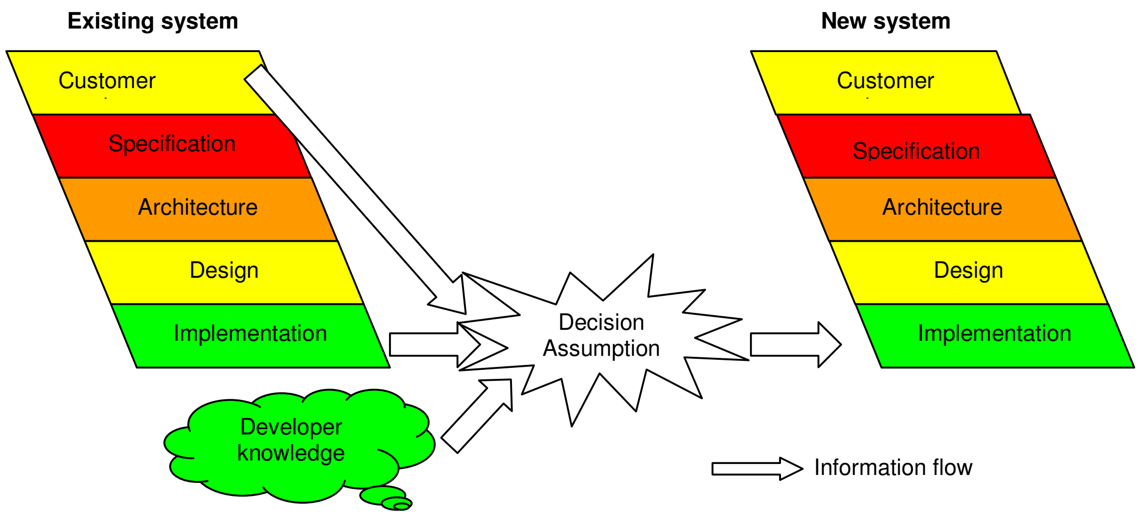

**Fig. 1: Actual situation when developing a new system from an existing one**

The knowledge thus remains implicit, and contradictory requirements or misunderstandings with the customer may only be revealed with difficulty. The effect of the new requirements remains unclear, and the risk for the project is incalculable.

## Reasons

Why is it that why requirements engineering is often neglected against the better judgment of those responsible for projects?

Firstly, the focus of attention when executing follow-on projects is often very much on the implementation, as many aspects are deemed to be known and are thus never questioned. Adapting existing software is thus carried out without complete knowledge of the original requirements, thus leading more often than not to problems.

Secondly, the process is seen “only” as the further development of an existing system. That means that neither time nor money is available for analyzing the previous system. It is simply too expensive to generate a comprehensive requirements basis prior to commencing the further development process.

A third factor is the lack of experience when it comes to requirements engineering. Deriving requirements from knowledge to date about customer requirements and solution strategies is initially unfamiliar to developers. If a complete specification is desired on top of this, then the hurdle is often too high to tackle.

The challenge in a new project lies, therefore, in extracting the requirements from the old system and using these as the basis for the new development. Only then can new features be incorporated in a targeted manner. How can the first step be taken, then, without holding up the project for weeks?

## Idea

An approach which promises success must address the problems outlined above in a targeted way. The idea presented here involves documenting the most important steps which need to be taken in developing the system based on the new customer requirements. From there, further requirements may be derived. This enables the set of requirements to be completed iteratively. The requirements specification remains manageable and can be synchronized with the customer. Deviations may then be corrected in the specification and must then, in turn, flow forward into the architecture, the fine design or the implementation.

This approach is based on the assumption that the crucial questions in realizing the system must have an answer even without needing to be explicitly asked. In viewing the important decision points during development based on the core functions that are required, it is possible to ask the relevant questions and to derive the requirements from this. An important factor for success is the analysis of the previous system. This is because important decisions and concepts may be identified which can then serve as a basis for the new development.

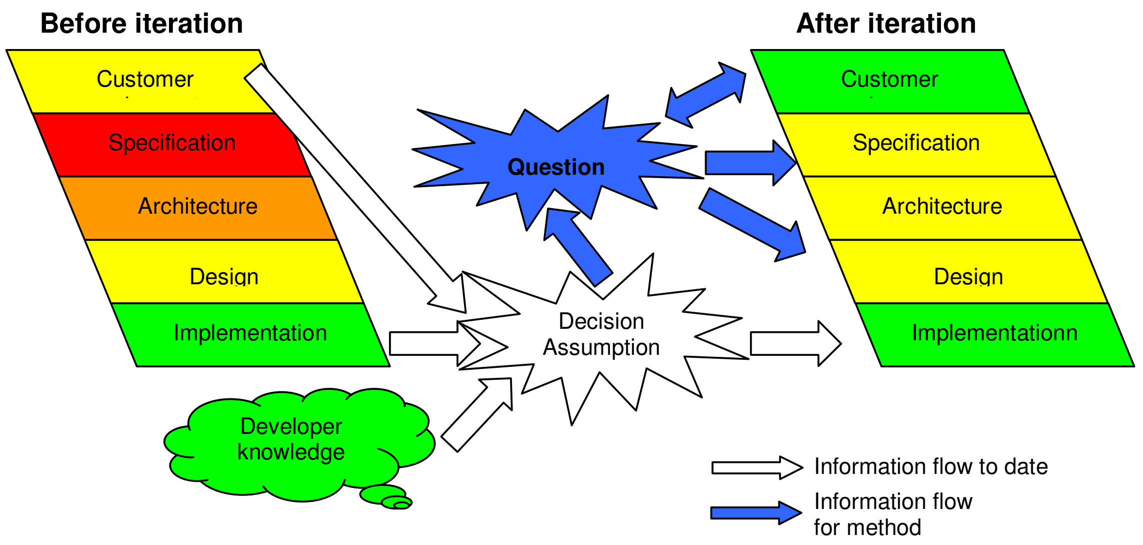

**Fig. 2: Iteration when developing the new system and application of the methodical approach**

Fig. 2 shows such a scenario for iteration during the development of the new system. In contrast to Fig. 1, the decision or assumption is explicitly questioned. The reasons which are unearthed during this process are documented as requirements in the specification, while fundamental decisions and the corresponding concepts are also documented in the system architecture. These measures ensure that the quality of the specification and architecture is improved at each iteration without having had to previously complete the analysis of the system. Iterative, with small steps and little effort – that is how the documentation is generated alongside the system.

## Prerequisites

In order to be able to document the requirements that have been determined in a targeted and efficient manner, a documentation structure needs to be laid down from the start. It must be clear from this structure where decisions on particular issues and the underlying assumptions are documented. How are requirements for a core function described? And how are universal, non-functional requirements described? There are several possibilities to choose from here depending on the type and structure of the project.

A further important step is to identify the contact person from the previous project. As knowledge is often stored in the minds of the developers, it is highly important for these developers to be available to clarify questions on the concepts used and the decisions made. Furthermore, they are able to provide important pointers towards additional sources of information.

## Detailed View of the Steps in an Iteration Process

### Step 1: identify the core functions which need to be considered

The point of departure for iteration lies in identifying the most important functions of the system which is to be created. The purpose is not to consider all functions, but to set priorities by selecting the most important ones. Each additional iteration then allows further functions to be added.

### Step 2: determining the critical aspects of the previous system

Here it is important to identify the influencing factors which were particularly crucial in implementing the previous system. These are typically those aspects which will also play an important role in the new project. First and foremost, they are factors which are not purely related to one function, but which influence the entire system. It is these quality attributes which often shape the realization of the system. In addition, it is also necessary to discuss critical points for realizing individual functions.

### Step 3: analyzing the concepts of the previous system

This step involves identifying the most important concepts of the previous system. What formed the basis of the solution concepts? What were the decisions based on? Following this, these concepts and decisions are evaluated with respect to the extended requirements for the new system. Is it possible to simply use them as they are or do they need to be adapted or completely replaced?

### Step 4: documenting the results

The result involves documenting these influencing factors, as well as the derived concepts and the decisions that were made. The influencing factors together with the identified core functions of the system form the basis for the initial version of the requirements specification. In addition, all the assumptions made are documented.

### Possible Synergies for the Documentation

Once these steps have been taken into account, some possible synergies should be revealed for the documentation:

- The documentation on realizing a function using system modules can serve both to describe the implementation of a function as well as the specification thereof using existing modules. The description is thus both the requirement and the solution. This mix is certainly permissible for the initial documentation.
- The documentation may be designed more efficiently if the dividing line between requirements engineering and testing is blurred. What is the difference between describing a function in requirements engineering and in the test specification for this function? Both aspects may be combined. This approach certainly has disadvantages, but it does allow the initial documentation to be more efficient.

## Benefits

Systematic requirements engineering represents the basis for better quality and the ability to recycle. A single project is not able to cover the initial efforts involved. The approach described thus provides benefits both for the project at hand and beyond. The project is provided with concise documentation which highlights only the most important aspects. This documentation allows incoming change requests to be more easily organized during the project. At the same time, the prerequisites for efficiently executing follow-on projects are created, as these may be built upon using the existing documentation. Furthermore, the documentation provides material which allows the project and the solution concepts to be evaluated at a later stage in order to learn from the insights gained.

## Embedding Documentation into an Overall Requirements Engineering Concept

The advantage of the approach presented, namely that changes and expansions to the system are documented in a sensible way, also represents its greatest drawback. The problem is that only parts of the system are taken into consideration, i.e. only those that require changes or expansions. Although the documentation does become more comprehensive and better with each iteration, it will never be sufficient to completely cover the entire system.

Additional efforts are thus necessary for achieving comprehensive requirements engineering. In order to achieve the actual goal of attaining high-quality documentation for the entire system, those parts must also be analyzed that are not affected by the changes incurred by current projects. This cannot be achieved by a single project for the reasons, e.g. time and cost pressures, outlined above.

These additional efforts may be justified, however, when seen as an evolutionary process, i.e. by improving individual projects to create a product line strategy. In order to attain this, not only the variable parts need to be recognized and documented but also the unchanged core of the system. By applying the method outlined here, such an introductory project no longer needs to begin right from scratch. Depending on the scope of the variable parts, the efforts involved may thereby be reduced to a considerable extent.

Furthermore, this approach makes it possible to build up the necessary experience and acceptance in small steps, thereby establishing requirements engineering within an organization.

## Summary

The approach described above details current problems relating to requirements engineering in real projects using the example of the automotive industry. It offers alternatives to classic procedures in order to achieve an initial consistent specification. Even under tight deadline pressure, this approach makes it possible to lay the foundations for carrying out targeted requirements engineering at the start of future projects. This approach does not, therefore, seek to replace comprehensive requirements engineering but rather provides a pragmatic solution to achieving it.

## Contact:

method park Software AG
Wetterkreuz 19a
91058 Erlangen
Tel. +49 (0) 9131-9 72 06-281
Fax +49 (0) 9131-9 72 06-280
Margit.Brendl@methodpark.de
www.methodpark.de

**Authors:**
Tel. +49 (0) 9131-9 72 06-309
Michael.Gerdom@methodpark.de

Tel. +49 (0) 9131-9 72 06-303
Uwe.Rastofer@methodpark.de