Requirement Engineering for a Small Project with Pre-Specified Scope

Thomas D. Nielsen[1], Sigve Hovda[2], Antonio Fernández[3],
Helge Langseth[2], Anders L. Madsen[4],
Andrés Masegosa[2] and Antonio Salmerón[3]
[1]Univeristy of Aalborg, [2]Norwegian University of Science and Technology,
[3]Universidad de Almeria, [4]Hugin Expert

Abstract

This paper describes the requirements engineering process developed for the EU-funded project Analysis of MassIve DataSTreams (AMIDST). The process adopts a use case based approach to requirements engineering (RE), that is tailored to the specific characteristics of the AMIDST project. These characteristics include a relatively small group of developers, a pre-defined project scope, stakeholders from different industries, and the development of a general software framework that can be instantiated according to the needs of the stakeholders. The resulting methodology is sufficiently general to be of relevance for similar development projects.

1 Introduction

The requirements engineering (RE) process adopted by a particular software project typically reflects intrinsic characteristics of the project in question. Thus, there is often no standard way to conduct a RE process [1], but one rather adapts existing processes to the needs and specifics of the project. This especially applies to smaller software projects [2, 3], where the RE processes follow more ad-hoc strategies and a best practice seems to be lacking.

In this paper, the RE process for the project entitled *Analysis of MassIve Data STreams* (AMIDST) is outlined. AMIDST is a project that is partially funded by the European Union's Seventh Framework Programme for research, technological development and demonstration, and falls in the category of a small development project. AMIDST has a pre-specified scope, because the grant agreement asserts that the result of the RE process must be in compliance with the *Description of Work* (DoW). The DoW defines the scope of the project as well as more general requirements to the system, both functional and non-functional, and e.g. includes a fixed deadline for the RE process. The latter restricts the project from, e.g., following a strict agile approach [4], where a product owner is continuously renegotiating the requirements and where the RE process is seen as a continuous process throughout the whole project [5]. Furthermore, the software framework developed in the project should be sufficiently general to accommodate

This paper was presented at the NIK-2014 conference; see http://www.nik.no/.

stakeholders and use case providers representing different industries. Thus, the RE process relates not only to the software framework, but also to the solutions to be developed for the use cases of the project. Specifically, for each use case provider the general AMIDST framework will be instantiated in order to meet the needs and requirements of that use case provider.

To the best of our knowledge, there exist no guidelines for the RE process for small projects with these characteristics. This paper therefore attempts to make a first step towards defining a best practice for such situations. The project's characteristics and challenges have formed the basis for the development of the AMIDST RE process. Based primarily on organizational characteristics of the project, the RE process is composed of selected components from existing RE processes that have been tailored to the identified AMIDST characteristics. However, we believe that the characteristics are sufficiently general to make them applicable for other small projects. In particular, it is expected that projects that share at least some of the characteristics of the AMIDST project may draw on the methodology that is outlined in this paper. This paper is build on two AMIDST reports [6] and [7].

The paper is outlined as follows. In section 2, the basic principles of requirement engineering are briefly outlined. Section 3 starts by describing the main characteristics of the AMIDST project, before the RE process is presented. In section 4, we have described the realization of the process, and the paper is concluded in section 5.

2 Basic principles in requirements engineering

An RE process typically ends up with a document containing a list of requirements for the system to be developed. This could, for instance, include what a software component must do or comply with. To date there is no common and agreed-upon definition of RE. Some definitions focus on the elicitation of requirements and therefore the interaction with the user, while others focus on the documentation or the specification. A definition that takes both foci into account is the IEEE standard [8], which defines requirements analysis as

1. *The process of studying user needs to arrive at a definition of system, hardware or software requirements.*
2. *The process of studying and refining system, hardware or software requirements.*

In the context of understanding the RE process, a possible definition of a requirement is given in the IEEE standard [8]:

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.*

This definition has a clear focus on the user, the system and also which contract, standard or specification is needed to be met. Notice, that the requirement is related to *what* a system can do and not *how* it is done. We will adopt the same perspective in the present document.

Activities involved in requirements engineering

The activities involved in RE vary widely, depending on the type of system being developed and the specific practices of the organization(s) involved [9]. These may include:

- Requirements elicitation
- Requirements analysis and negotiation; checking requirements and resolving stakeholders conflicts
- Requirements specification; documenting the requirements in a requirements document
- Requirements validation; checking that the documented requirements are consistent and meet stakeholders needs
- Requirements management; managing changes to the requirements as the system is developed and put into use

These activities are sometimes presented as chronological stages although, in practice, there are considerable interleaving between them.

Anyone who has a direct or indirect influence on the process or the system to be developed is identified as a stakeholder. Stakeholders include end-users that will interact with the system, the developers that will maintain the system, management, domain experts, union representatives, etc.. A challenge in the RE process is therefore to keep a smooth communication between the different stakeholders. The use-case driven approach to RE focuses on simplifying the communication between the end-users and the developers to improve the overall communication.

Use-case driven requirements engineering

It can often be a challenge for the future users of a software system to accurately communicate expectations about functionality to the software developers. Moreover, this communication can be further hampered by the users not being able to accurately express what functionality they want. To improve on this communication, a use-case driven approach to RE was developed in the nineties [1, 10, 11]. A use-case focuses only on the interaction between a user and the system to be developed, and forms the basis for the requirements engineering where each requirement is associated with a use-case. This means that when specifying the use cases and requirements, the user is requested to only focus on what he/she wants. This is an advantage, compared to the previous RE approaches, where requirements are listed in relation to components or sub-components in the software; an approach that often entails a degree of complexity that the user finds difficult to understand.

A use-case consist of a list of steps, typically defining interactions between an actor and a system component, with the aim of achieving a specific goal; the actor can be either a human or an external system. An overview on how to write effective use-cases is given in [11], where several templates are presented. Common for these templates is that the users are asked to describe the use-cases in natural language and addressing the following questions:

1. Who are the actors involved in the use-case? An actor is either a person or an entity that interacts with the software.
2. What is the main event that initiates the use-case? This could, e.g., be an external business event or a system event that causes the use-case to begin. It could also be

the initial step in a normal work flow.

3. What are the main user actions and system responses that will take place during the normal execution of the use-case?.
4. How can we evaluate the success of the use-case?

This dialog/question sequence will ultimately lead to accomplishing the goal that is implied by the use-case name and description.

Since different users may have different requirements to the functionality of the system, it is also common to define user groups that are expected to have similar types of interactions with the system. The users within a user group typically have the same profile and role within their organization and their set of competences are expected to be similar. This allows use cases and their associated requirements to be defined in relation to particular user groups.

In order to provide further insight into the use-case driven approach, it is useful to distinguish between functional and non-functional requirements. Functional requirements are those requirements that are directly related to the interaction between the user and the system. The non-functional requirements are hidden for the user and are more indirectly related to the overall success of the system. Non-functional requirements can, for instance, include scalability, trace-ability and test-ability. After use-cases are provided and functional requirements are identified, it is the requirement engineers responsibility to identify, document and communicate these non-functional requirements. The use-case driven approach to requirement engineering thus focuses on eliciting the functional requirements in collaboration with the users. This process supports and improves the communication between the users and the developers, because the focus is on what the users want and less on how the requested functionality can actually be achieved.

3 The AMIDST requirements engineering process

This section contains a description of the AMIDST RE process. We first give a brief overview of AMIDST and describe the main characteristics of the AMIDST project that has influenced and shaped the RE process. Based on these characteristics we present the AMIDST RE process, which is based on the RE processes described in Section 2, but tailored to the specific characteristics of AMIDST. Since the focus of the present RE process is on the functionality and documentation of the software products being developed, we will, e.g., not cover process-related requirements or non-functional requirements, c.f. Section 2. Non-functional requirements are defined by the aforementioned DoW.

The AMIDST project: a brief overview

AMIDST is a project funded by the European Union's Seventh Framework Programme for research technological development and demonstration. The objective of AMIDST is to develop a scalable toolbox capable of providing a framework that facilitates efficient prediction and data analysis in streaming data. The toolbox will be instantiated to target three distinct industries represented by the three industrial use case providers in the AMIDST consortium. The use case provider in the energy domain is Verdande Technology, the use case provider in the financial domain is the Cajamar Cajas Rurales Unidas, and the use case provider in the automobile domain is Daimler AG. In addition to the use case providers, there are also three academic partners and a fourth industrial partner in the software industry. The fourth industrial partner is Hugin Expert and the three academic partners are the Norwegian University of Science and Technology, Universidad de Almeria, and Aalborg University.

The development of the toolbox and its subsequent instantiation will be driven by functional requirements specified by the use case providers and elicited in accordance with the RE process described in the present document; the requirements supplement the non-functional requirements described in the DoW framework. If deemed effective, the instantiated toolboxes (referred to as AMIDST *solutions*) will be adopted by the use case providers.

The AMIDST project is structured around ten work packages. The first work package is concerned with requirements engineering and evaluation. Work packages 2–4 focus on methodological developments to be applied on the massive data streams. In particular, there is emphasis on probabilistic models as well as scalable inference and learning algorithms tailored to these models. Work package 5 is concerned with extensions to the AMIDST solutions to the Hugin software toolbox, which is already commercially available. In Work packages 6–8 the AMIDST solutions for the three industrial partners will be realized. Finally, Work packages 9 and 10 deal with dissemination and exploitation as well as management.

Characteristics of the AMIDST project

In this section we identify and describe the key characteristics of the AMIDST project that directly influence the requirements engineering process.

Characteristics one: Pre-specified scope of the project

The AMIDST project is funded by the European Union's Seventh Framework Programme for research, technological development, and demonstration. The overall scope and main developments in the project are therefore defined from the beginning of the project period and documented in the Description of Work. This will henceforth be referred to as the DoW framework. More detailed requirements pertaining to the functionality and documentation of the developed software should thus fit within the DoW framework, and their necessity in relation to AMIDST should be justified and demonstrated.

Characteristics two: Partners at different geographical locations

The AMIDST consortium consists of 7 partners/stakeholders, 4 industrial and 3 universities, which are situated in 4 different countries. This diverse consortium composition has at least a two-fold impact on the RE process.

First of all, although AMIDST targets the industrial stakeholders' common need for processing massive data streams, the more intrinsic aspects of the three industrial domains differ significantly. This, in turn, means that the partners will have different (possibly conflicting) requirements for the system being developed. To ensure that the requirements are comparable across domains and abide to the DoW, a unified formal framework for eliciting system requirements is needed. Such a framework may also provide transparency in the overall requirement engineering process and help prioritize requirements across different domains and thereby help resolve potential conflicts.

Secondly, with the project partners located in different countries, there is a need for a controlled and stringent requirements process in order to limit travel expenditures. This approach is supported by a unified formal requirements engineering framework. Consultancy and discussions in relation to the requirements will primarily be achieved through telecommunication conferences and by physical meetings only secondarily.

Characteristics three: Transfer of domain knowledge between partners
The industrial partners of the AMIDST project come from very different domains: the automotive, energy, and finance industry. To ensure the development, refinement, and completion of the unified formal requirements framework it is necessary with regular and structured communications among the project partners during the RE process. This not only relates to the specific requirements, but also to the software and user context in which the AMIDST framework should be deployed. The latter part, in particular, is required for a proper evaluation and validation of the elicited requirements.

Characteristics four: One framework for three different domains
The AMIDST toolbox should define a general framework that can encompass the diverse domains of the three industrial partners. Thus, the format of the unified requirements framework should be sufficiently general and flexible to allow for all relevant requirements to be elicited for the three domains. At the same time the framework should be appropriately structured and formalized enabling a controlled elicitation process (see also Characteristic two) with the requirements specified in a consistent manner making them comparable across domains. In order to also provide a basis for a controlled and balanced system development, the requirements should be linked to relevant project phases, work packages, and tasks. This, in particular, will provide the work package leaders with a clear overview of the requirements that are relevant for the activities in a specific work package.

Characteristics five: Potential refinement of project focus
AMIDST is an RTD project, where both the industrial and academic partners' understanding of the domains develop as the project unfolds. In order to support a potential refinement of the project's focus and goals, the requirements engineering process should allow for an internal (re)prioritization of the requirements that is transparent across application domains.

Project phases and AMIDST requirements identification

The overall project duration will be decomposed into different phases, each having distinct requirements. According to [12] a project's life cycle can be divided into three general phases: the design phase, operations phase, and disposal phase. The disposal phase is outside the scope of AMIDST and, thus, will not be considered in the present document. The design and operations phase, however, represent the temporal development of the project, and are initiated by the start of the project and ends with the testing of the deployed system, which will be adopted by the use case providers if deemed effective. Each phase can furthermore be described as a collection of distinct stages in the project. The overall process is illustrated in Figure 1.

In the design phase general functionality requirements for the system are specified, i.e., what the system should do and support. Figure 1 details key stages inside this phase. The first stage consists of the design of the general framework (models and algorithms) as well as the design and development of the software tools. These stages are primarily related to Work packages 1–5. In the second stage, the general framework and software is instantiated for each specific use case. Finally, initial tests of the use case instantiated frameworks are conducted. During this phase of the project, possible design requirements could, e.g., address

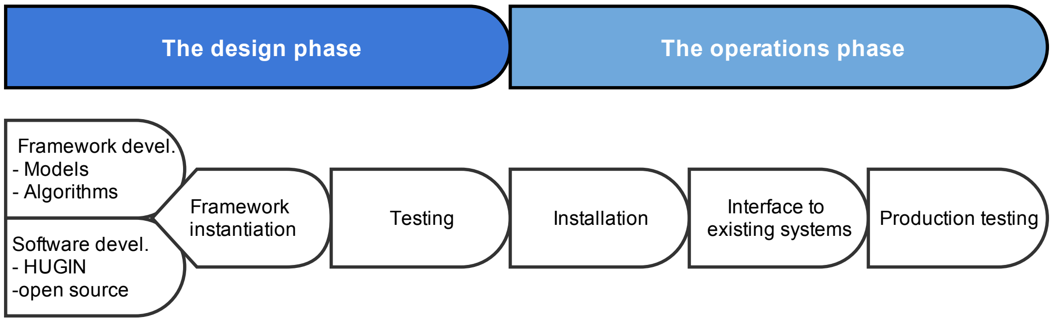

Figure 1: The figure shows the key stages in the design and operations phases. For the requirement specification, each requirement should be defined relative to one of these stages.

- the scope of the model
- the interpretability of the learned models
- the extent and type of domain knowledge that can be integrated into the models
- prediction accuracy of the developed models
- documentation

The requirements for the operations phase concern the functionality of the deployed system. In Figure 1, we decompose this phase into three stages: installation, interface to existing systems, and production testing. The requirements for this phase could, e.g., address

- hardware constraints
- interfaces to existing software or data base systems
- inference functionality, i.e., what queries the system should be able to answer
- response time

Use cases and user groups in the requirements engineering process

As discussed in Section 2, a use case driven approach to RE puts emphasis on the functional requirements of the system, by focusing on the interactions between actors (this being either persons or other hardware/software modules) and the system. This focus is consistent with the general objectives of the RE process in AMIDST, and the use case-based approach to RE is therefore adopted in AMIDST. However, to obtain more well-defined requirements and establish a closer connection between the use cases and the project stages/work packages (c.f. Characteristics four), we further require that use cases should be specified as indivisible scenarios. Specifically, when defining the use cases, the industrial partners were informed that:

> *. . . a use case should ideally be indivisible. If a use case can be decomposed into multiple sub-use cases, each with a well-defined sub-objective relevant for AMIDST, then these sub-use cases should be described separately.*

The requirements derived from a use case are typically specified in relation to a particular user or type of user (possibly another component of the system). In AMIDST the possible users have very diverse backgrounds, ranging from developers with an intimate knowledge of the key technologies embedded in the AMIDST framework to programmers and users working in marketing. In order to ensure that focus is on the future *users* of the system, the AMIDST requirements engineering process also adopts and identifies user groups (as described in Section 2), which will be explicitly linked to the relevant requirements.

The general AMIDST requirements engineering process

To ensure a sufficient amount of knowledge transfer between the partners (c.f. *Characteristics three*, Page 6), the overall RE process will be carried out in an iterative fashion that is expected to involve a high level of cooperation and interaction between the partners. In Figure 3, inspired by [13], an illustration of the RE process for AMIDST is given. The process contains five phases, which are discussed below.

Preparation I: This phase starts at the same time as Work package 1 and ends when the initial template of the RE process is finished, see Appendix A in [6]. In this template, the RE process is outlined, including definitions of use cases, user groups, and how to link requirements with the stages and WP/tasks in the development process. In order to meet characteristic three, four and five, the use case providers are asked to provide a detailed description of the system context that the AMIDST solution is expected to operate in, identify user groups, describe use cases and requirements. In order to meet characteristic four, the requirements are linked to their associated work packages and tasks.

Elicitation: The distribution of the above mentioned template marks the initialization of this phase. Its aim is to get an initial high-level description of the different use cases and their requirements. This information are specified by the use case providers in collaboration with the academic partners, thus addressing characteristic one, three and four. Once the use case providers return the document with the requested information, feedback and review sessions should be held to clarify and refine the information provided. These review sessions not only serve as a quality check and to align the expectations of the developers/designers and use case providers, but they also provide the end-users with an opportunity to give feedback to the developers. At the end of the elicitation phase, the aim is to have a first coherent description of the requirements for each use case provider.

Prioritization: In this phase the use case providers complete and refine the document template used in the previous phase. The provided template explicitly links each of the requirements to the relevant work packages and tasks in the AMIDST project, thus providing an initial consistency check with the DoW framework (see Characteristic one). Moreover, the template allows the use case providers to give a prioritization of the relevant requirements for the AMIDST framework. Specifically, the use case providers are asked to rate each requirement in terms of whether it is a must, should, or could requirement:

Must (be) These requirements are expected by the use case providers and include properties described in the AMIDST DoW framework.

Should (performance) These requirements are expected by the use case provider, but are not explicitly agreed upon.

Could (delighters) Optional requirements that will often be satisfying to have, but which have not been required by the use case provider in neither an explicit nor implicit manner.

This high-level prioritization scheme is inspired by the Kano model correlating product development with customer satisfaction, see Figure 2. Within each of these categories, the use case providers should also make a more fine-grained prioritization by numerically weighting the different requirements on a scale from 0 to 100.

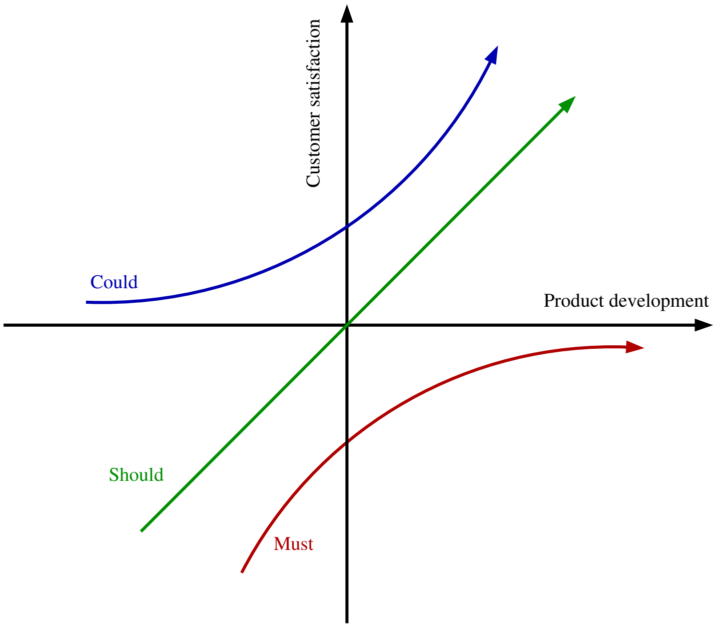

Figure 2: The Kano model.

Validation: In this phase, the requirements from all use case providers are collected to get the *big picture*. This includes a discussion among the members of the project science review group as to the extend to which the requirements can be accommodated and whether they collectively produce any potential conflicts, either internally or in relation to the DoW framework. Revisions and negotiations of the detailed requirements are therefore expected. In this phase, it is important to ensure that Characteristic one is met.

Evaluation and Testing: In this phase, the focus is on the elicitation of the evaluation and testing procedures in the AMIDST project. This phase starts with the distribution of a new document template, where the aim is to obtain a high level description of the evaluation and testing methods that are necessary to measure the performance of the AMIDST framework. This phase is not strictly part of the RE process, but will supplement the process by providing detailed specifications of how to perform specific tests and evaluations. Documentation of this phase is out of the scope of the present document, but will be included in the initial version of the AMIDST handbook (Deliverable D1.3).

4 Realization of the requirements engineering process

The RE process was organized by coupling each use case provider with an academic partner; Verdande Technology was paired with Norwegian University of Science and Technology, DAIMLER was paired with Aalborg University and Hugin, and CajaMar was paired with Universidad de Almeria. The particular partner associations were based on geographical as well as affinity considerations. It is important to stress that the RE process does *not* prescribe such a partner association, but it does bring distinct advantages. First of all, the academic partners can better assist the use case providers when completing the requirements template, and the ongoing internal communications and discussions

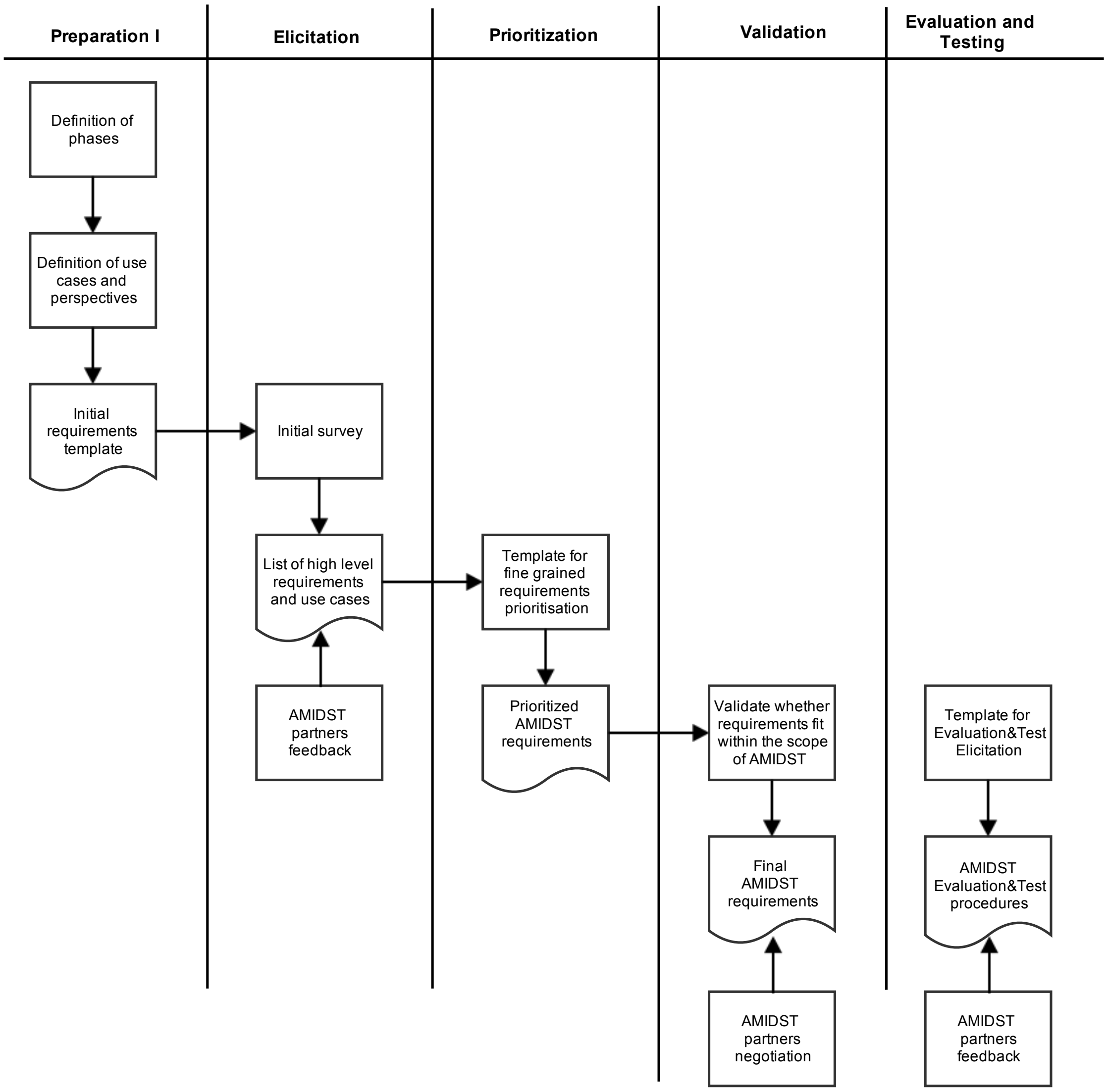

Figure 3: Description of the five phases in the RE process in AMIDST.

(both formal and informal) provide an opportunity for early feedback on drafts of the requirements specification. Secondly, this division of work also provides an increased knowledge transfer between industrial and academic partners.

As described in Section 3 one of the design considerations for the requirements engineering process was to base the requirements specification on a formal template that would be shared by all three use case providers. In addition to the information that the use case providers are requested to fill-in, the template also provides a description of the overall RE process as well as guidelines on how to complete the template.

The completion of the templates was conducted as an iterative process with a close collaboration between the use case providers and the paired academic partners. In addition to the more formal deadlines marking transitions between phases in the RE process, we also introduced several short-term deadlines, where the use case providers were given feed-back on draft versions of their completed templates. Not only did this serve as an instrument to ensure a continuous progression in the requirements specification, where misunderstandings and problems could be identified and mitigated at an early stage, but it also provided an early transfer of knowledge from the industrial partners to the academic partners in the project. Part of this (otherwise tacit) knowledge were documented for the benefit of the other partners, both current and future, in the consortium, and is expected to be included in the deliverables planned for Work packages 6–8. This, e.g., includes a description of the data characteristics for the use case providers.

The specified requirements together with their work package/task allocations and prioritizations will be summarized in tables at the work package level. These tables allow work package leaders to get a clear overview of the specific requirements that need to be taken into account in the different work packages. An example of a part of such a work package requirements table can be found in Table 1, which includes some of the presently collected requirements pertaining to Work package 2.

Req. ID.	Relevant subphase	Must/should/could	Points	Task
DAI.U5.D1	Framework devel. & instan.	Should	20	2.2
DAI.U5.D2	Framework devel. & instan.	Should	20	2.2
DAI.U5.D3	Framework devel.	Should	15	2.2
DAI.U5.D4	Framework devel.	Should	15	2.2
DAI.U5.D4	Framework instant.	Should	20	2.2
DAI.U7.D1	Framework devel.	Must	35	2.1
⋮	⋮	⋮	⋮	⋮

Table 1: The work package requirements table containing the presently collected requirements for Work package 2.

5 Conclusion, observations and reflections

In this paper we have introduced a general methodology for the RE process in software development projects with the following characteristics: relatively small group of developers, a pre-defined project scope, stakeholders from different industries, and the development of a general software framework that can be instantiated according to the needs of different stakeholders. The presented methodology adopts a use case-based approach tailored to these specific characteristics of the project. We also provide several key considerations for the RE process of this kind of project: division of the RE approach in different phases to ease the overall implementation of the process; structured prioritization of all the requirements to ease the agreement between the stakeholders; and the employment of a template based document to ease the elicitation of the requirements and the communication between stakeholders with different backgrounds, expectations, and locations.

In our opinion, the presented methodology is general enough to be applicable to a wide range of software development projects with similar characteristics. Concretely, we believe that the describe RE process could be of great help to technology transfer based projects between the academia and the industry.

Acknowledgement

The AMIDST project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 619209. This paper is based on the AMIDST project's Deliverable 1.1: *General methodology for requirement analysis* [6]. The full report, which also

includes the general template for the developed RE process described in Section 4, is available from the project's website amidst.eu.

References

[1] Pohl, K.: Requirement Engineering - Fundamentals, Principles and Techniques. Springer-Verlag (2010)

[2] Quispe, A., Marques, M., Silvestre, L., Ochoa, S.F., Robbes, R.: Requirements engineering practices in very small software enterprises: A diagnostic study. In: Proceedings of the 2010 XXIX International Conference of the Chilean Computer Science Society. SCCC '10, Washington, DC, USA, IEEE Computer Society (2010) 81–87

[3] Aranda, J., Easterbrook, S., Wilson, G.: Requirements in the wild: How small companies do it. In: Proceedings of the 15th IEEE Requirements Engineering Conference. (2007) 39–48

[4] Dingøyr, T., Dybå, T., Moe, N.B.: Agile Software Development: Current Research and Future Directions. Springer (2010)

[5] Kavitha, C.R., Thomas, S.M.: Requirement gathering for small projects using agile methods. IJCA Special Issue on Computational Science - New Dimensions & Perspectives (3) (2011) 122–128

[6] Fernández, A., Hovda, S., Langseth, H., Madsen, A.L., Masegosa, A., Nielsen, T.D., Salmerón, A.: Deliverable 1.1 of the amidst project, general methodology for requirement analysis. http://amidst.eu/ (July 2014)

[7] Borchani, H., Fernández, A., Gundersen, O.E., Hovda, S., Langseth, H., Madsen, A.L., Martinez, R., Masegosa, A., Nielsen, T.D., Salmerón, A., Sørmo, F., Weidl, G.: Deliverable 1.2 of the amidst project, requirements for the automotive, oil and financial data domains. http://amidst.eu/ (August 2014)

[8] The Institute of Electrical and Eletronics Engineers: IEEE standard glossary of software engineering terminology. IEEE Standard (September 1990)

[9] Sommerville, I.: Software Engineering. Addison-Wesley (2011)

[10] Jacobson, I.: Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley (1992)

[11] Cockburn, A.: Writing Effective Use Cases. Addison-Wesley Professional (2001)

[12] Eigner, M.S.: Product Lifecycle Management. Springer-verlag (2009)

[13] Ebert, C.: Systematisches Requirement Engineering. Dpunkt Verlag (2010)