Challenges in Market-Driven Requirements Engineering -
an Industrial Interview Study

Lena Karlsson¹, Åsa G. Dahlstedt², Johan Natt och Dag¹,
Björn Regnell¹, Anne Persson²

¹Department of Communication Systems, Lund University, Sweden,
(lena.karlsson, johan.nattochdag, bjorn.regnell)@telecom.lth.se
²Department of Computer Science, University of Skövde, Sweden,
(asa.dahlstedt, anne.persson)@ida.his.se

Abstract. Requirements engineering for commercial off-the-shelf software packages
entails special challenges. This paper presents preliminary results from an empirical study
investigating these challenges through a qualitative approach using semi-structured inter-
views. The survey is exploratory with the objective of eliciting relevant topics for further
research. Seven employees at five software companies with a market-driven development
focus were interviewed. The areas of interest include process-related issues on release plan-
ning, requirements quality and decision support, as well as artefact-related issues regarding
requirements as discrete entities and their representation. The paper also contains a charac-
terization of each company, regarding aspects such as products, processes and customers. A
number of challenging issues were elicited, including communication gaps between mar-
keting and development, the problem of balancing the influence between marketing and
development on requirements decisions, as well as the limited value of monolithic require-
ments specifications and the problem requirements overloading.

1. Introduction

This paper reports on preliminary results from the first stage of an industrial survey,
focusing on current practice and challenges concerning requirements engineering (RE)
in Swedish software industry. Before problems related to inefficient RE can be prop-
erly addressed, more research is needed to better understand the challenges that the
software industry is currently facing. This survey aims at discovering hypotheses for
future research and complementing existing surveys in the area of RE.

The study focuses on market-driven development, which is currently gaining
increased interest compared to development of customer-specific systems. This is due
to the emergence of the market for off-the-shelf or packaged software [17, 4]. Market-
driven RE differs from customer-specific in several ways. Sawyer [17] concludes that
the major differences between market-driven and customer-specific RE are the charac-
teristics of stakeholding and schedule constraints. In market-driven projects, the main
stakeholder is the developing organization and therefore requirements are invented by
the developers, since there are no discrete set of users who can articulate their require-
ments. There is also a major pressure on time-to-market for these software products.
Other characteristics of market-driven RE are release planning and prioritization, and
managing the constant flow of new requirements [4, 10, 15].

There are several surveys that concern or include RE issues [5, 6, 7, 8, 9, 13, 14].
However, none of these have a primary focus on market-driven development. Further-
more, in most of these surveys the participating projects and organizations are fairly large, in terms of number of persons and requirements involved, as well as in terms of the duration of the projects. This survey provides characteristics from a number of small, and fairly new market-driven development companies.

The survey presented in this paper aims at understanding the conditions and practices that characterize modern software industry, particularly with regard to requirements identification, requirements specification, and product management. In this initial survey, seven persons at five different companies participated. The forthcoming stages of the study include workshops with RE experts and an extended interview survey based on these initial results.

The paper includes a characterization of the companies involved, regarding company facts, typical projects and development processes. The result of the study is a set of interesting issues that may indicate the direction of further research. However, the small number of interviews implies that a larger survey needs to be performed in order to generalize the results and to be able to formulate relevant research hypotheses. A larger survey could e.g. make it possible to compare smaller organisations with larger ones, and compare RE for different types of products.

The remainder of the paper is organized as follows. In Section 2 the research method is described. Section 3 presents the results uncovered through the study. Section 4 concludes the paper and discusses future work and further research issues.

2. Research Method

The study was carried out using a qualitative interviewing approach. This approach is useful to explore an area of interest, to obtain an overview of a complex area, and to discover diversities rather than similarities [16]. As the purpose of the study is to gain an improved understanding of the nature of requirements engineering within market-driven software companies, the qualitative approach is considered suitable. Furthermore, interviewees in different roles from both large and small companies were interviewed in order to collect different viewpoints and perspectives on the nature of RE.

The research procedure is illustrated in Figure 1. The initial stage of the study involved a brainstorming and planning meeting to identify different areas of interest and plan the study.

In order to gain an insight into the area of market-driven requirements engineering, we aim at interviewing a large number of software-developing companies. However, it
was concluded that this initial stage of the study would benefit from selecting a handful of companies before adjusting the interview instrument and carry out the full study. Therefore seven interviewees within five Swedish software companies were asked to participate. The interviewees were selected among our industrial partners. The companies all have a market-driven development focus. They have had at least one market release of a software product or are just about to release their first.

The interview instrument was designed with respect to the different areas of interest and with inspiration from other requirements engineering surveys [13, 7]. The questions in the interview instrument were divided into three groups in order to give a structure to the interview:

• **Characterization** – facts about the company, its products, and the interviewee.

• **Process issues** – questions on the development procedure, requirements activities, and decision support.

• **Artefact issues** – questions on requirements as entities, how requirements are documented, tool usage, and requirements interdependencies.

To test the interview instrument, two pilot interviews were carried out. Some questions were clarified and the structure was improved before proceeding. A summary of the interview instrument is available in Appendix A.

The study has a semi-structured interviewing strategy, where the set of questions were the same for all interviewees [16]. However, the order of the questions varied depending on the interviewees’ knowledge and role in the company. This meant e.g. that some issues were discussed more in-depth with certain interviewees. In order not to steer the interviewee, additional questions were asked depending on the interviewee’s answers until all areas of interest were covered.

All interviews involved one interviewee and three interviewers. One of the three interviewers ran the interview process while the other two posed additional questions. The duration of the interviews was 90 to 150 minutes. All interviews were recorded on tape and extensive notes were taken in order not to lose information.

Afterwards, the interviews were transcribed in order to facilitate and improve the analysis. The size of the transcripts was 7 to 23 pages of text depending on the length of the interview.

The analysis was performed mainly through marking and discussing interesting sections in the transcripts [16]. The interviewers examined the transcripts from different perspectives and searched for explicitly stated or concealed RE challenges. Another researcher, who did not attend the interviews, also analysed the transcripts to ensure the quality of the study and to enhance confirmability [12].

### 2.1 Validity

It is almost impossible to achieve an unbiased interview analysis; all interviewers have expectations and interpretations affecting the analysis of the interviewees’ answers. Furthermore, notes and transcripts may be insufficient, since accentuations and gestures are difficult to reproduce. The analysis hence relies on the researchers’ perceptive
skills and good memory. Recording the interviews on video-tape would cover these details and probably improve the analysis. In our study, using three interviewers with different background and experience has reduced the risk of bias and misinterpretations.

The results presented in the paper are first and foremost issues and suggestions for further research, which have surfaced during the analysis of the interviews. However, there may be several other important issues in the transcripts, not yet discovered.

Evidently the interviews only capture the subjective opinion of each interviewee. A larger number of employees should be interviewed to capture the general view of each company. We have chosen companies of different size and maturity to manage the issue of transferability [12]. However, we do not attempt to generalize to a larger population, but merely to discuss some interesting issues discovered during the interviews, and present some hypotheses for future research.

3. Analysis and Results

This section presents some interesting issues discovered during the analysis of the interviews. Each issue is concluded by a conjecture to summarize our current views on the subject. These conjectures may be topics of further research.

Section 3.1 corresponds to the characterization questions in the interview, and includes other facts to widen the description of the companies. Section 3.2 corresponds to the process issues and discusses some issues concerning the development process. Section 3.3, finally, deals with the artefact issues in the interview instrument.

3.1 Company characterization

Table 1, which summarizes the characteristics of each company, is inspired by [18]. The companies’ age range from 1 to 20 years. The companies also vary in size, from 12 employees to 1200, which means that the results include challenges for both large and small companies.

Two managing directors and one product leader were interviewed to gain the managers’ view, and three project leaders and one head of developers were interviewed as development representatives. The responsibilities of the project leaders differ. In the small companies, project leaders tend to have more overall responsibilities such as process introduction and business analysis. In contrast, the large company uses a project leader to co-ordinate resources and work with requirements. All companies use natural language on a high level of abstraction to specify requirements. Two companies use UML at a later stage for modelling the system, one of which also creates state charts. In three of the five companies, databases are used to collect suggestions from users and developers.

Only two of the companies have a well-defined, elaborate process with defined phases and regular evaluation and improvement efforts.
3.2 Process issues

This section addresses questions on the development process and development decisions. It also focuses on the requirements engineering procedures and activities. These questions revealed some interesting issues concerning the process and the challenges companies face in requirements engineering.

3.2.1. Living with changing requirements. Market-driven projects often do not have a single well-defined customer. However, customer representatives and retailers may be involved to validate the product and users of the current versions of the product may provide feedback and state new requirements. Challenges these companies face are for example that the market changes, competitors improve and customers are not certain of their requirements. Company C and D use a strategy to develop functions to only 60-90% and then release a beta-test, because, as the project leader at Company D states, “customers will always change their minds”. In this manner, it may be possible to receive customer opinions earlier and with less effort spent on the project.

Conjecture: Requirements are volatile and it is necessary to find methods to cope with changes. Although these market-driven companies do not have a single well-defined customer, it is important to obtain feedback from potential customers. Beta-test releases make it possible to receive customer feedback at an early stage and can be used to manage changing requirements.

3.2.2. Market-driven vs. technology-driven requirements. The interviewees at Company B and C estimate their requirements sources to be 50% users and market department and 50% internal developers. The project leader at Company B concludes that some developers see themselves as “artists” who enjoy coding but do not reflect on whether the code solves customer problems or not. The same interviewee identifies a problem in the company focus; since the company is new and inventive the focus has been on development and coding instead of marketing and sales. Therefore there have been difficulties when introducing the products to an open market, since the products do not live up to the users’ expectations. On the other hand, too much marketing influence may result in unrealistic requirements, which are not possible to implement with the resources available, and new and creative requirements may be ignored.

Conjecture: It is necessary to find a good trade-off between requirements corresponding to perceived user needs and new, inventive ones that may provide a competitive advantage. The decision regarding which aspect to focus on may depend on the maturity of the market. To succeed on a stable market it may be necessary to create technical inventions, while on an immature market it is important to satisfy customer needs.

3.2.3. Gap between marketing staff and developers. When interviews were held with two persons at the same company, some different views were discovered. For example, in Company C, the managing director claims that it is not difficult to understand requirements written by different people, while the head of developers regards that particular issue to be the greatest challenge in his daily work.

There are also different views with regard to what a “good requirement” is. From the company perspective, represented by the managing director or marketing, a good requirement is something that makes money for the company. In contrast, project lead-
<table>
<thead>
<tr>
<th>Table 1</th>
<th>Company A</th>
<th>Company B</th>
<th>Company C</th>
<th>Company D</th>
<th>Company E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of employees</td>
<td>1200</td>
<td>65</td>
<td>13</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>Age</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Product in focus</td>
<td>Software development tool.</td>
<td>Embedded software with focus on image processing. 3 product lines.</td>
<td>Software development support tool based on requirements management.</td>
<td>Communication tool for distributed groups working with software development.</td>
<td>Software development and visualization tool for integration of electronic business processes.</td>
</tr>
<tr>
<td>Number of employees involved in the product in focus</td>
<td>80</td>
<td>65 (in all 3 product lines)</td>
<td>13</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Customer</td>
<td>Software developing companies, mostly in the telecommunication industry.</td>
<td>Consumer electronics retailers and distributors.</td>
<td>Software developing companies.</td>
<td>Safety critical software developing companies.</td>
<td>Large companies that need an improved business process.</td>
</tr>
<tr>
<td>End user</td>
<td>Developers at software-developing companies.</td>
<td>Bankers, students, office personnel.</td>
<td>Developers and managers at software developing companies.</td>
<td>Managers and developers at safety critical software developing companies.</td>
<td>Software developers that integrate electronic business processes</td>
</tr>
<tr>
<td>Interviewee(s)</td>
<td>1. Requirements engineer and project leader</td>
<td>1. Project leader 2. Product manager</td>
<td>1. Founder and managing director 2. Head of developers</td>
<td>1. Project leader</td>
<td>1. Founder and managing director</td>
</tr>
<tr>
<td>Role/responsibility of the interviewee(s)</td>
<td>1. Co-ordinate resources and clarify, define and prioritize requirements.</td>
<td>1. Introduce process and break down the requirements specification. 2. Coordinate between marketing, development and production. Suggest and write requirements.</td>
<td>1. Release planning, marketing and sales. 2. Allocate work between developers. Suggest document and time-estimate requirements.</td>
<td>1. Process responsibility and business analysis.</td>
<td>1. Marketing, customer contacts, and release planning.</td>
</tr>
<tr>
<td></td>
<td>Company A</td>
<td>Company B</td>
<td>Company C</td>
<td>Company D</td>
<td>Company E</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Projects</strong></td>
<td>New release every 6 months to include new requirements and correct detected errors. Involves all 80 employees.</td>
<td>Includes 25-30 employees for a new product line and 5-15 employees for a new version of existing product. Initiated because of new requirements or when many errors are detected.</td>
<td>New release when enough requirements or errors have been added in the database, approximately every 6 months. Divided into functions involving 1-3 employees for 1-10 weeks.</td>
<td>3 parallel projects involving all 7 employees at 2 locations. At least one release every month.</td>
<td>No regular releases, puts together a project including all employees when many bugs or requirements have been identified.</td>
</tr>
<tr>
<td><strong>Requirements documentation</strong></td>
<td>Natural language. Support system database where customers and developers add requirements. A web site with the ones for the current release written on a high abstraction level.</td>
<td>Natural language, state charts and UML in the requirements specification. Changes to the specification are rarely documented.</td>
<td>Natural language. Database included in the tool where customers and employees can add requirements.</td>
<td>Natural language. Customers’ requirements suggestions on virtual story cards, which can follow through the process.</td>
<td>Natural language in the requirements specifications and UML at a later stage.</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Defined but not documented, followed thanks to the experienced staff.</td>
<td>Introducing process. The ad hoc process works thanks to the experienced staff.</td>
<td>Elaborate, documented and integrated in the tool they develop and use. Based on requirements status.</td>
<td>Elaborate. Extreme Programming, pair programming excluded.</td>
<td>Not documented, but the experienced staff knows the activities by heart.</td>
</tr>
<tr>
<td><strong>Project or process evaluation</strong></td>
<td>Not regularly. Have increased requirements awareness. Earlier prioritization in categories Must, Wish, and Postpone. Decreased database size.</td>
<td>Not yet in use. Have increased systems thinking and awareness of requirements and testing.</td>
<td>Included in the process. Occurs after each release. Have adjusted the status range.</td>
<td>Weekly meetings to sum up the week’s problems. Individual time surveillance.</td>
<td>Strong process focus in the company thanks to the process focus in the tool.</td>
</tr>
</tbody>
</table>
ers and developers focus on issues regarding the way requirements are described, for example completeness and understandability.

Furthermore, the project leader at Company A describes a gap between marketing staff and developers. The marketing department’s task is to write requirements, which in their opinion means writing down ideas for the next release. Concurrently, developers expect the requirements to be written down clearly enough to start coding.

Conjecture: There is a gap between marketing and developers concerning the views on requirements engineering. Better communication and collaboration between these groups are needed, in order to increase the requirements quality and thereby the quality of the final product.

3.2.4. Elaborate vs. elementary processes. Two of the new and small companies express no need for an elaborate process. In fact, the product leader at Company B expressed a wish for “a simple tool”, which may be interpreted as a wish for “a simple process”. The project leader at the same company mentioned that many of the developers are reluctant to introduce a process, because they believe that it would limit their freedom and creativity. Others, working with an elaborate process, for example the head of developers at Company C, find the process valuable since it structures the work and all employees are aware of their respective responsibilities.

Conjecture: It is difficult to balance between elaborate and elementary development processes. Opinions differ on whether the process limits or simplifies development. The necessary degree of elaboration depends on the maturity of the company. Immature and small companies may succeed with an elementary process while more mature companies need an elaborated process to accomplish their goals.

3.2.5. Organizational stability and market turbulence. Companies that use an ad hoc process still manage to get sufficient results, to a large extent thanks to their staff. A low staff turnover and skilful developers are needed to survive. “The project’s success depends on the individuals” was mentioned in Company B, which had recently gone through a large down-sizing process. Companies without a defined process take a significant risk if key persons leave the organisation, since they lack the necessary documentation and structure. “Some projects would die if certain people leave” is another quotation from this interviewee.

Conjecture: A low staff turnover will help companies succeed even in times of instability and business environment turbulence, since the knowledge remains with the skilful developers in the company.

3.2.6. Relation between time estimates and release plans. Release planning decisions are mostly taken by marketing or management departments. The project leader at Company A claims that the release plan is very much dependent on accurate time estimates, since the estimates affect how many of the requirements that are selected. This is especially important in companies with regular releases, since under-estimation may result in an exceeded deadline while over-estimation may exclude valuable requirements. Time estimates are mostly performed by developers, since the requirements have to be carefully examined and partitioned in order to make a reasonable estimate.

Conjecture: The better time estimates, the more accurate release plans. Improved communication between marketing and developers may improve release planning quality, which in turn increases the quality of the product.
3.3 Artefact issues

Artefact issues deal with requirements as entities and include questions on the documentation of requirements, requirements interdependencies and tools for requirements engineering. This section discusses some artefact issues that emerged during the interviews.

3.3.1. Traditional requirements specifications. Market-driven development companies have to deal with a steady stream of new requirements as well as select an optimal part of these requirements to implement in the next version [2]. Among the interviewed companies, only two use a traditional requirements specification, i.e. a single document including all requirements relevant for the project. These specifications are used to structure and store the requirements prior to every new project. It is interesting to note that these two companies are also the ones with the least elaborate and structured RE processes.

The companies with more elaborate RE processes manage their requirements as single units, one by using Extreme Programming story cards [1] and the other by using their own requirements management tool. In both cases, the requirements are continuously managed, time estimated and prioritized, and attributes are used to store additional information about each requirement. Based on this information, the requirements to be implemented in the next release are chosen.

Conjecture: In the market-driven situation the traditional, monolithic requirements specification is of limited value when managing a steady stream of incoming requirements of varying quality. Instead, new ways of dealing with requirements are developed and in order to increase the control of the requirements, they are managed as single units.

3.3.2. Requirements overload. Company A has had substantial problems with an overload in the requirements database, since more requirements were added than could be managed [11]. This resulted in difficulties when prioritizing requirements for the next release, since there were thousands of requirements to consider. The problem has been temporarily solved by the requirements manager through creating a “TOP 10-list” with the ten most important requirements for the next release. There is an apparent risk that the selected requirements are not the most important ones, and that some of the requirements left in the database should have been included instead.

There is also a risk that customers and developers, using the database, feel neglected since they are not given any feedback on their requirement suggestions. The project leader at Company D stresses the importance of customer feedback, because “if they are neglected they never come back”.

Conjecture: Requirements suggestions from developers and customers are essential. However, too many requirements suggestions complicate release planning. Furthermore, the important feedback to those who suggest requirements may be overlooked. There is a need for a method to prevent databases from being flooded with requirements in order to facilitate release planning.

3.3.3. Simple techniques for basic needs. All the interviewed companies use natural language to define their requirements. Two of the companies use UML at a later stage and one of these creates state charts. The smaller companies have fewer requirements to handle, and therefore do not express the need for CASE tools to manage the
requirements. The number of requirements is small enough to be handled manually, typically a few hundred. The main challenges at the small and immature companies are at the moment to understand the requirements, to make all requirements complete and easy to understand, as well as to follow a certain standard – not to manage a massive set of requirements. The product leader at Company B requests a simple tool, since the ones considered would have a too large introduction overhead and a too steep learning curve. The head of developers at Company C wants a checklist to control that certain items or attributes are considered when different stakeholders write requirements in different ways.

Conjecture: There is a lack of tools and techniques simple enough to be introduced in small and immature companies. Such simple techniques and tools could increase requirements completeness and understandability.

3.3.4. Requirements bundling. Requirements relate to and affect each other in various ways, which affects for instance release planning [3]. All companies acknowledge this, but few have routines to document and take these relations into account. Instead, they tend to group related requirements, i.e. by bundling requirements which should be implemented together. Bundling is usually made based on the fact that the requirements deal with the same part of the system, e.g. the user interface or the same part of the software code. Requirements bundling is used to facilitate release planning and implementation. Interdependencies, such as conflicting requirements or duplicates, are managed manually when the requirements are planned for a new release.

Conjecture: Dependencies between requirements are managed through bundling. This could imply that the complexity of requirements interdependencies is not well understood. On the other hand it could imply that bundling is sufficient enough to make the decisions needed and that a deeper understanding of requirements interdependencies is not needed.

3.3.5. Living with design in requirements. The nature of requirements differs; some concern the user, some concern the developer. Requirements are expressed at many levels of abstraction, ranging from abstract ones such as “usable” to more detailed ones concerning, for example, memory capacity. Since requirements specifications also often concern the design, it is difficult to draw a clear line between the phases.

Conjecture: Requirements specifications often partly concern design issues and there is a risk that the specification is not flexible enough to create a good design. However, the line between requirements and design is thin and design issues will always concern requirements. The key is to make requirements management flexible enough to handle it.

4. Conclusions and Further Research

This paper reports on interesting issues discovered in the interview study on market-driven requirements engineering. These are divided into three sections: Characterization, Process issues and Artefact issues complying with the interview structure. The issues discussed in this paper are preliminary results of a continuing study, and will be used to find hypotheses for our future research.
Although Curtis et al. [6] performed their survey on large systems development and that it was performed 14 years ago, the presented study indicates that some problems remain. Communication is still a cornerstone in software development and the project’s success depends on the staff. Another recent survey [8] also confirms that organizational problems, for example lack of skilled personnel and high staff turnover, have a larger impact than technical problems on requirements engineering.

In agreement with Lubars et al. [13], Company C and D manage requirements changes by tracking them through the development to locate the effect of each requirements change. Furthermore, the same companies have a strategy to develop functions to less than 100% in order to get customer feedback earlier.

As a complement to other surveys the presented study describes a communication gap between marketing and developers, resulting in insufficient time estimates and requirements quality. The balance between marketing and developers’ requirements decisions is also recognized as a dilemma. The use of a requirements database rather than a traditional, monolithic requirements specification is also salient, as well as the urge to group requirements into bundles to ease requirements structuring and work partitioning.

This paper concludes the first stage of a research project, which is planned in four stages. The next stage is one or more workshops where the results from the interview study are discussed and further validated with a selection of RE experts. Then an extended interview study will be carried out, involving additional companies. And, finally, in order to even further validate, quantify and generalize the results, a quantitative survey is planned, involving a larger number of companies.

Acknowledgement

This work is partly funded by the Swedish Knowledge Foundation (KK-stiftelsen) and the Swedish Agency for Innovation Systems (VINNOVA) under grant for The Centre for Applied Software Research at Lund University (LUCAS), Sweden. We would like to direct a thank to participating interviewees, without whom this paper would not have existed. We would also like to thank Pär Carlshamre for his support during the first stages of this study and Carina Andersson for reviewing the paper.

References


## Appendix A. The interview instrument

<table>
<thead>
<tr>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Tell us about the company (number of employees, age, business area, etc.)</td>
</tr>
<tr>
<td>1.2 Tell us about the company’s product/products (time on the market, typical customer/end-user, size of product projects, etc.)</td>
</tr>
<tr>
<td>1.3 Tell us about your position in the company (role, daily tasks, responsibility, etc.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 What is the procedure when developing a product? (kind of process, activities performed, documentation developed, evaluations performed, etc.)</td>
</tr>
<tr>
<td>2.2 What is a “requirement” to you?</td>
</tr>
<tr>
<td>2.3 In what way are requirements handled? (requirements process, activities, etc.)</td>
</tr>
<tr>
<td>2.4 What challenges do you face when working with requirements? What has been successful regarding requirements engineering?</td>
</tr>
<tr>
<td>2.5 How much resources are spent on requirements engineering? How much would be optimal?</td>
</tr>
<tr>
<td>2.6 What is a “good requirement” to you? And to the company? Is the quality of the requirements assessed? How?</td>
</tr>
<tr>
<td>2.7 What kinds of decisions are taken during the development of a product? What kind of support is needed in those decisions?</td>
</tr>
<tr>
<td>2.8 Is it possible to make decisions too late? What can be the effect in that case?</td>
</tr>
<tr>
<td>2.9 How is it decided what to include in the product? How are the requirements prioritized? What is difficult when deciding what to include in the product?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Artefact issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 How are requirements documented? What information and attributes are documented about the requirements?</td>
</tr>
<tr>
<td>3.2 What support and what tools do you use to document your requirements? What pros and cons does these tools have?</td>
</tr>
<tr>
<td>3.3 How many requirements are handled in a typical project? Who suggests the requirements?</td>
</tr>
<tr>
<td>3.4 What kinds of dependencies between the requirements have you come across? Are dependencies documented? Are dependencies actively looked for?</td>
</tr>
<tr>
<td>3.5 How do dependencies affect product development? How is it handled?</td>
</tr>
</tbody>
</table>