JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE
J. Softw. Maint. Evol.: Res. Pract. (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/smr.535

Introduction of a process maturity model for market-driven product management and requirements engineering

Tony Gorschek[1,*,†], Andrigo Gomes[2], Andreas Pettersson[3] and Richard Torkar[1]

[1]*Blekinge Institute of Technology, Karlskrona, Sweden*
[2]*Spidexa Technologies AB, Ronneby, Sweden*
[3]*Anoto AB, Lund, Sweden*

SUMMARY

The area of software product development of software intensive products has received much attention, especially in the area of requirements engineering and product management. Many companies are faced with new challenges when operating in an environment where potential requirements number in thousands or even tens of thousands, and where a product does not have a customer, but any number of customers or markets. The development organization carries not only all the costs of development, but also takes all the risks. In this environment traditional bespoke requirements engineering, together with traditional process assessment and improvement models fall short as they do not address the unique challenges of a market-driven environment. This paper introduces the Market-driven Requirements Engineering Process Model, aimed at enabling process improvement and process assurance for organizations faced with these new challenges. The model is also validated in the industry through three case studies where the model is used for process assessment and improvement suggestion. Initial results show that the model is appropriate for process improvement for organizations operating in a market-driven environment. In addition, the model was designed to be light weight in terms of low cost and thus adapted not only for large organizations but suitable for small and medium enterprises as well.

Received 30 December 2007; Revised 4 January 2011; Accepted 6 January 2011

KEY WORDS: market driven; process improvement; technology product management; requirements engineering; process assessment; maturity model; software product management

1. INTRODUCTION

The development of software intensive products is changing focus, moving from a traditional bespoke (producer–customer) to a market-driven (producer–marketplace) perspective [1–5]. The overall implication being that the product development cost as well as producer revenues is not linked to one specific customer but to a market, comprising any number of potential buyers [6].

Requirements engineering (RE) is also very much affected by the change in perspective. Traditionally RE is described as 'what services a product should provide and under what constraints it should operate' [3], and in the bespoke perspective RE consists of the systematic process of eliciting, understanding, analyzing, documenting, and managing requirements throughout a product's life cycle [7]. The focus here is on the development instance (project) itself, i.e., the RE effort is project initiated and part of a pre-study or focused on the initial stages of development.

An organization operating in a market-driven requirements engineering (MDRE) context has a continuous flow of requirements and the RE effort is not limited to a development instance but

*Correspondence to: Tony Gorschek, Blekinge Institute of Technology, Karlskrona, Sweden.
†E-mail: tony.gorschek@bth.se

is a part of product management as a whole [8, 9]. In this environment requirements come from several sources both internal (e.g., developers, marketing, sales, support personnel, bug reports, etc.) and external (e.g., users, customers and competitors, often gathered via surveys, interviews, focus groups, competitor analysis, etc.) [10–13]. This can give rise to very large amounts of requirements, and all of them need to be caught, specified, analyzed, and managed continuously as the product evolves over time through releases. In MDRE, requirements selection (Release Planning (RP)) activities are central, i.e., the decision about which customers get what features and quality at what point in time. This makes the accuracy of RP a major determinant of the success of a product [14]. In MDRE the focus is on the product and requirements, and development efforts (projects) are initiated by requirements, packaged to a release.

The importance of having an adequate RE process in place that produces good enough requirements can be considered as crucial to the successful development of products, whether it be in a bespoke or market-driven development effort. There are, however, clear indications that RE is lacking in the industry since inadequacies in requirements is a major determinant in project failure [15–20].

Many of the challenges facing the area are relevant for both bespoke RE and MDRE. For example, problems (inadequate quality) in requirements filter down to design and implementation [3]. Davis published results indicating that it could be up to 200 times as costly to catch and repair defects during the maintenance phase of a system, compared to the RE phase [21], and several other sources indicate that inadequate requirements are the leading source for project failure [15–20].

This is further compounded in the case of MDRE as the initial selection of requirements is crucial, in particular since large volumes of requirements from multiple sources risk overloading companies. It is vital that incoming requirements can be handled in a structured way, dismissing irrelevant ones at an early stage, thus expending as little effort as possible in order to save resources for refining requirements that will actually be selected and allocated to development instances [22]. This initial process has been dubbed requirements triage [23, 24].

Furthermore, emphasis has been put on basing the selection of requirements triage on product strategies, business goals, and the overall vision of an organization as it enables optimizing both long-term and short-term perspectives as well as aligning the whole organization towards the same goals [4, 13, 25–29].

There exist many RE best-practice guides and software process improvement (SPI) frameworks (see CMMI [30], REPMG [20], and others [31–37]) that are targeted at finding challenges/issues and improving RE practices. However, most of them are adapted to suit mainly a bespoke environment with traditional, project-focused, customer–developer relationships, and there exist no best-practice guides or process improvement frameworks targeted at MDRE that address the unique challenges facing companies in a market-driven situation. This paper presents such a model, the Market-Driven Requirements Engineering Process Model (MDREPM), as well as its validation in the industry at three leading telecom companies. In addition, MDRE as a field is described and the unique challenges are listed as this gives a background to what MDREPM is aimed at addressing.

MDREPM incorporates many practices from the bespoke perspective, but it specifically targets the unique challenges facing product development organizations existing in a market-driven environment. The model can be seen as both a best-practice guide (like the REPMG [20]), and also as a process assessment framework that can be used for process assessment, and step-wise improvement planning by an organization. The incremental nature of the model coupled with a relative low assessment cost makes it suitable for both larger organizations and small- and medium-sized enterprises (SMEs).

The paper is structured as follows. Section 2 gives an introduction to market-driven product management and RE, and lists unique challenges and characteristics. It also briefly explores other best-practice models as the related work. Section 3 gives an overview of the research approach used in the development and validation of the MDREPM model. Section 4 presents the model itself, summarizing the structure and major characteristics. Section 5 gives an overview of the main validation of the model, namely its usage in the industry, as well as the feedback and experiences collected during usage in the industry. Section 6 presents the conclusions and future work.

DOI: 10.1002/smr

2. BACKGROUND AND RELATED WORK

2.1. The market-driven perspective

The main distinguishing feature that separates MDRE from bespoke RE is the fact that there is no customer, but rather a market(s) consisting of any number of customers. This fact influences all other aspects of MDRE including elicitation practices, analysis, and management. Figure 1 gives an overview of a 'generic' MDRE process based on several sources in the literature [9, 38–43]. Each part of the process is described below.

2.1.1. Requirement sources (Elicitation). The *initiation* of an MDRE effort is not governed by an external request or customer order, but is continuous in nature, where development efforts (projects) are initiated as needed as per release plan (following a product roadmap). Saying that a *market(s)* is the customer is true, if somewhat simplified. There are of course certain entities within the market(s) which can be identified. Key customers are a part of the market but due to their importance they usually have the possibility to put forward requirements directly to the developing organization in addition to going through indirect channels like marketing/sales/support.

Figure 1 shows additional potential external sources of requirements, e.g., distributors and partners who often have a partnership relation with the developing organization. This is especially the case for developing organizations offering their products to a market indirectly through a vendor or an overall larger product offering. An example could be development organizations offering software and hardware components for integration on motherboards. In this case the market consists of all motherboard manufacturers, repair shops, and other companies using the development organization's products as a part of their own product offering. In addition, end-users can be the direct customers, for e.g., software upgrades (firmware and drivers).

The use of external sources (e.g., key customers) for direct elicitation activities, as opposed to getting their input through surveys and other marketing efforts, has been proposed and tested by companies in the form of Customer Working Groups (CWG) [6]. Although this may generate new requirements and give good insight into individual customer priorities there is often a risk that multiple preferences (between different customers, and between customers and the development organization) conflict [6].

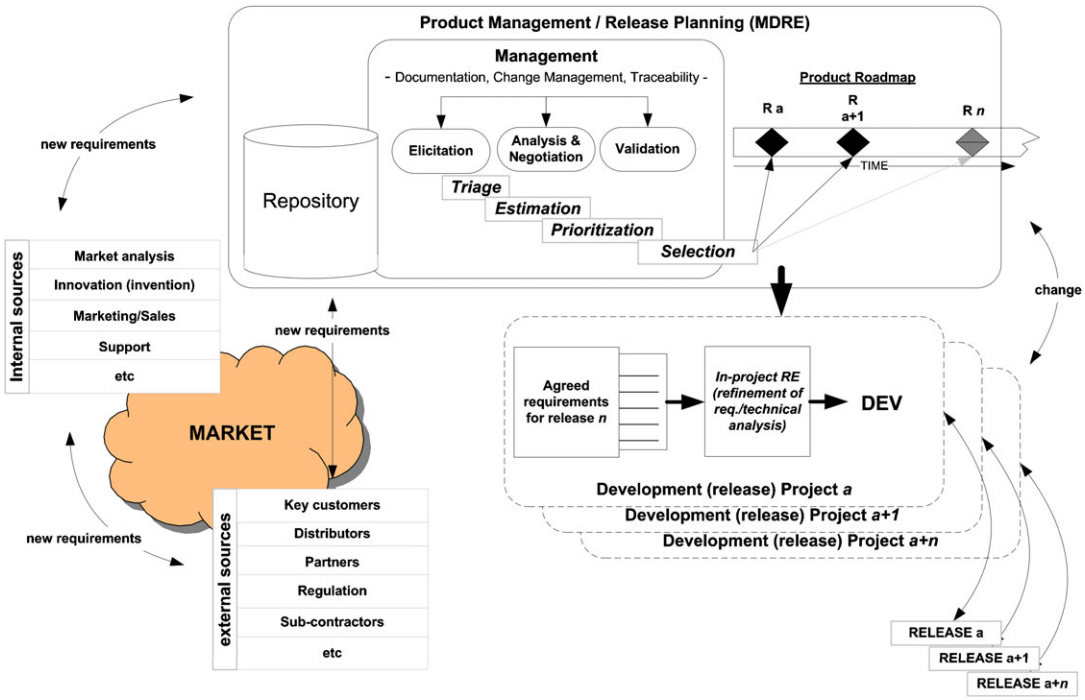

Figure 1. Overview of a generic MDRE process.

DOI: 10.1002/smr

Many requirements come from internal sources as requirements in MDRE are often *innovations* (invented by the development organization) attempting to add new and unique features and/or characteristics to a product. Development organizations often possess extensive *domain knowledge* making innovation not only possible but often profitable, allowing for differentiation of a product in comparison to competitors by offering unique or better features and/or quality [12].

2.1.2. Requirements analysis. All requirements are *documented* in a central repository independent of source and subsequently analyzed. This first involves performing early dismissal/acceptance of the incoming requirements (often called triage [23]) in order to avoid requirements overload, a major issue in MDRE [9, 22, 24].

Following early triage *analysis* is performed where implementation costs and resources are *estimated* for requirements going in for comparison and prioritization. Having accurate estimates is directly related to the ability to perform RP activities and selecting the requirements of high priority to be allocated to a certain release project [43–45]. This is mainly due to the fact that time-to-market is often central. Fixed delivery/release dates are generally rigidly enforced, postponing requirements to a later release instead of going over time [40, 44]. Regarding estimation techniques, attempts to estimate costs and schedules using requirements employ function points or feature points [46]. However, the use of non-structured informal techniques for requirements estimations seems to be pre-dominant, where professionals rely on expert judgment [47].

An important factor to be investigated during analysis is requirements interdependencies which can influence requirements selection. For example, a biometric sensor (e.g., fingerprint identification) on a laptop may decrease the value of having a card reader. Another case could be that a requirement stating '128-bit RSA encryption' would make network performance requirements more expensive to realize. In the market-driven context value-based dependencies, directly related to customer value and development cost seem to be predominant. Carlshamre *et al.* [48] report that only about 20% of all requirements are relatively singular, making dependencies a highly relevant factor in requirement analysis.

2.1.3. Requirements prioritization. Requirements *prioritization* has to be conducted taking several aspects into account. The central notion is that all requirements should be compared and prioritized independent of source. The *objective* of the prioritization is to get input for requirements selection for subsequent RP. The overall goal is to deliver the right product to the market at the right time and thus selling and generating revenues. Success is defined by outperforming competitors in the market and delivering a high perceived benefit to customers. From this perspective customer satisfaction is central and optimally customers (and potential customers) should perform prioritizations. Regnell *et al.* [45] report on attempts with distributed prioritization in a market-driven context involving distributed marketing departments in the prioritization of requirements. However, scalability of having large amounts of requirements prioritized by several stakeholders can be an issue. Potential scalability issues also apply to the prioritization technique chosen [38].

Internal considerations regarding technical aspects (e.g., architecture and maintainability), business aspects (e.g., strategic decisions regarding focusing on new market segments), and implementation aspects (e.g., dependencies) are also input to prioritization and selection. Several methods for attaining requirement priority exist, including AHP [49], the 100-point method [50], attainment [51], and the planning-game [52].

Common for any prioritization activity is that the requirements being prioritized need to be comparable with regard to abstraction level, otherwise there is a risk that requirements of higher abstraction level gets higher priority than requirements on a low level of abstraction. For example, a light in the glove compartment of a car is probably prioritized lower than the car having a boot/trunk [38, 53].

2.1.4. Requirements selection (RP)

Product strategy and roadmaps. After initial triage requirements are analyzed (estimated, prioritized, and crucial dependencies mapped) the actual selection and release allocation takes place. A product roadmap (Figure 1) can be used as an approach to document and communicate plans for

DOI: 10.1002/smr

future releases [54]. There are many types of roadmaps described in the literature, in fact sometimes any forward looking document is called a roadmap [55]. The one used for the purposes of MDRE RP is described as Product-Technology Roadmap [55], and has the purpose of 'mapping' and aligning efforts and resources towards common goals. A roadmap should convey several aspects, for example:

- themes of a certain product release (e.g., a theme could be offering a certain type of functionality, concentrating on improving quality, security, and so on)
- restrictions (e.g., what are the restrictions in terms of risk, time, resources available, internal technical considerations, and so on)
- goals (what are the overall product goals and what are the goals for every release)
- milestones (for releases and goals)

A roadmap can be seen as an explicit concretization of product strategies depicting the long-term plans of a product. Product strategies are in turn a concretization of company and business strategies pertaining to an individual product [56].

Product strategies should reflect not only current market knowledge and customer priorities (attained through market analysis, from internal experts, etc.) but also the long-term goals set for a certain product. The requirements selection should be done within the boundaries set by product strategies. For example, if a certain release has the theme of 'security', requirements pertaining to security should be prioritized over other requirements for that release. Ignoring product strategies (and only looking at current priorities) may mean that a product is successful short-term, but at the expense of the long-term goals [13, 29]. For example, security requirements may be deemed less important at present, but the long-term plans for the product is to eventually break into new market segments where security is deemed crucial. One of the fundamental aims of a product strategy is to explicitly plot the goals and limits of a product—focusing all efforts and aligning them in one deliberate direction [13, 27]. On the other hand, constraining development too much to a business strategy can cause a developing organization to miss new, out-of-the-box, business opportunities that can leverage business technology strengths [26, 57]. Thus another aspect which must be included in the formulation of product strategies (and thus specified in roadmaps) is the balance between technology-push and market-pull when selecting requirements for a release. There is always a risk that one perspective dominates, predominantly market-pull (business perspective) as reported by Wohlin and Aurum [58].

It should be noticed that requirements selection within the boundaries of product strategy following a roadmap does not guarantee success, as the strategies followed can be flawed. However, having up-to-date product strategies, which reflect all vital knowledge—from both technology and marketing perspectives—will most certainly increase the chance of developing a successful product [13, 28, 29].

For more information about roadmaps see Kostoff and Schaller [54] and Kappel [55], for information about how to construct roadmaps for SMEs see [59].

Performing RP. The actual 'mechanics' of allocating requirements to a certain release has been described by Carlshamre and Regnell [60] in the use of the REPEAT process (Requirements Engineering Process At Telelogic). REPEAT is centered on having fixed release dates and intervals, allowing for requirements to be allocated to Select-lists with a 'must' part and a 'wish' part. Together the must and wish requirements are estimated to take 1.3 times the available resources (must part taking 0.7 and the wish part 0.6), allowing for 30% requirement estimation error while still realizing the must part. For a description of the entire REPEAT process see Regnell *et al.* [40].

Greer and Ruhe [61] describe the EVOLVE approach based on a genetic algorithm where customers prioritize and reprioritize requirements for incremental development and delivery. Although the exemplification of EVOLVE assumes continuous customer involvement and is only exemplified using a limited amount of candidate requirements (20) it can be used for decision support generating a limited amount of candidate solutions.

Both processes/methods used as examples here are dependent on relative accurate effort estimations, priorities, guiding strategies (roadmaps), and clear release dates.

DOI: 10.1002/smr

Requirements selection quality. Being able to measure the success of the requirement selection process is crucial as it allows for continuous improvement. Regnell *et al.* present a model that allows for principal reasoning about both the quality and the current capacity of the requirement selection process [62]. Karlsson *et al.* present the PARSEQ method which focuses on post-release analysis of requirements selection quality, examining the requirements actually selected for realization [63]. This can be described as a postmortem of sorts for requirements selection quality.

The perceived customer value pertaining to a product can also be used to gauge selection quality (as an indirect indication). GAP analysis can be used to measure positive and negative 'gaps' between what the product offers and what the customer perceives. Features and characteristics of the product are identified and their fulfillment of customer needs is mapped. A positive gap represents when a product delivers more than is expected, a negative gap the opposite. One of the earliest descriptions of the need to measure this 'gap' was described in [64–66]. Customer Value Analysis (CVA) is similar to GAP analysis but also includes the perspective of using competitor products in the analysis and the evaluated product is graded with regard to the value of a need in comparison to alternatives [67]. Both GAP and CVA can be used to measure selection quality post-release, but the results can also be used actively as input to the next round of requirements selection for the product in question. The relationship between requirements change, development time, and customer satisfaction is presented in [68].

2.1.5. Requirements validation. Requirements validation is traditionally performed in close cooperation with the customer (bespoke RE), but in MDRE this is complicated for obvious reasons. There is the possibility to cooperate with key customers but their input may be hard to obtain, and in addition they may not be representative of all requirements in a release [44, 69]. This is further complicated by the fact that some requirements are invented to suit an imaginary customer [39]. Validation can be performed internally using the knowledge of the development organization; this is particularly relevant in the case of invented requirements. Traceability to requirements source is important in this case [43].

An additional method for requirements' validation is using beta-releases of upcoming releases, and having real customers (or potential ones) test the product.

2.1.6. Requirements Management (RM) (Specification). The requirements repository (e.g., an RE tool or database with a front-end) can act as a central storage for all requirements. For reasons of scalability, traceability, distributed work (e.g., prioritization), and overall management document based (e.g., word editors) tool is not appropriate.

The use of a tool also enables the use of attributes. Attributes can be a good way of structuring requirements in a repository. Examples of attributes can be seen in Table I. Some can be mandatory (have to be stated) others optional (stated if specifier deems it necessary), and some can be auto-generated by the tool used.

Table I. Example of requirement attributes (Inspired by Regnell *et al.* [40]).

Attribute	Description
ID	Unique identifier, e.g., auto number
Title	Title for the requirement
Description	Free description of the requirement
Rationale	A description of the rationale/benefit of the requirement from the req. source's perspective
State	What state the req. is in at present, e.g., new, dismissed, specified, planned for release, released, etc.
Version	Version of the requirement (maybe with the possibility to view different versions and differences btw. versions)
Source	The source of the requirement, that is the one who champions it
Estimation	Cost/time for implementation
Dependency	Dependency and type of dependency
Priority	The priority of the req. on a scale of 1–5 where 5 is more
Test	Links to test cases
etc.	etc.

DOI: 10.1002/smr

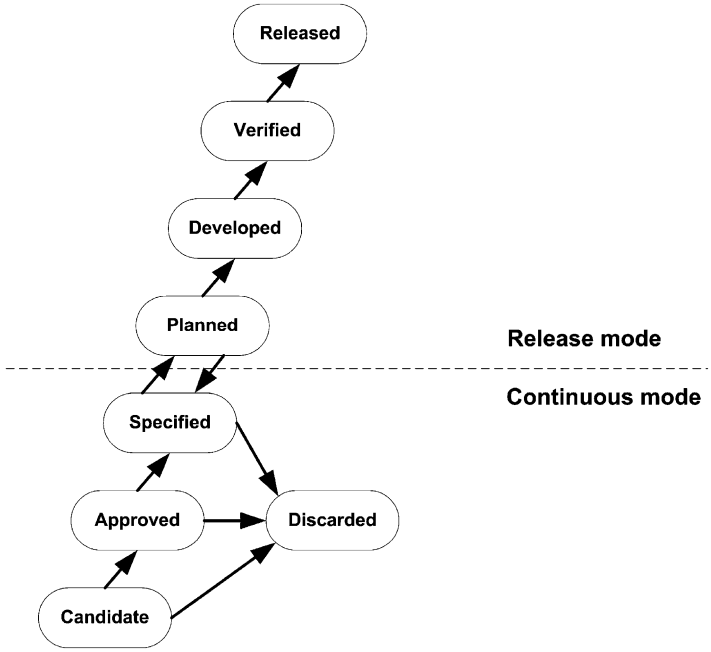

Figure 2. Example of Requirement State model [6].

Clear benefits of using attributes are:

- separation of information regarding a requirement enabling filtering and different views, sorting according to a specific attribute, etc.
- allowing for requirements to be specified in a similar manner by several people as attributes steer what is to be specified
- assuring that a minimum amount of information (mandatory attributes) is specified independent of who is specifying the requirement
- connecting attributes to the overall process and requirement states enables control and traceability (for example to be eligible for prioritization a requirement has to have the 'Estimation' attribute specified).

It should be noted that different development organizations may need different attributes. More information and examples on the use of attributes can be found in Regnell *et al.* [40].

Figure 2 gives an example of a requirement state model called the 'salmon ladder' showing the upward road a requirement has to take in order to be implemented (described in Regnell and Brinkkemper [6]).

The use of state models in MDRE enables not only control but also the ability to sort requirements enabling different views. For example, a manager performing allocation of requirements to a release can choose to view only the 'Specified' requirements (avoiding the ones not yet specified/analyzed). Requirements in the states 'Discarded' and 'Released' may be filtered out entirely when viewing requirements for the purpose of planning, and so on. This can also be seen as one way of battling RM overload. For examples of other state models see Regnell *et al.* [40] and Carlshamre and Regnell [60].

Requirements change in MDRE just as in the case of bespoke RE. In case of bespoke RE negotiations can be performed with the customer allowing for deadlines to be renegotiated. In MDRE changes to requirements prior to release allocation can be due to the emergence of new high-priority requirements, market changes can lower the priority of an already prioritized requirement and so on. Changes can be handled through general change management processes (see [70] for examples). However, the issue is complicated if changes occur after a requirement has been allocated to a release. Changes may give the need for re-prioritization, re-selection, and

DOI: 10.1002/smr

Table II. Overview of bespoke RE vs MDRE.

	Bespoke RE	MDRE
Initiation	RE process is initiated and terminated in accordance to a development project	RE process is continuous and serves the evolution of a product(s), projects are initiated as needed
Objective	Contractual fulfillment, adherence to requirements specification	Right product, right time, take market shares
Success criteria	User acceptance, customer satisfaction	Sales, market share, product reviews, customer satisfaction (many customers)
Life cycle	Development → maintenance	Evolution through continuous releases
Elicitation	One organization is customer T: Interviews, observation, etc.	Market is customer (might include key customers) often represented by internal product management. T: Innovation (invention of req.), market analysis, focus groups, competitor analysis, etc.
Domain knowledge	The development organization and the customer can cooperate to ensure that the domain is understood	The development organization has to be an expert in the domain, or at least have internal experts
Analysis and negotiation	Analysis and negotiation with customer regarding what to implement, conflict resolution, etc.	Early triage, estimation, prioritization, selection, release planning for fixed time
Validation	Continuously with customer/stakeholders	Late product tests with key customers, beta releases, focus groups, surveys, etc. Internal validations against marketing and other req. sources
Specification (management)	Finite amount of requirements, specification technique depending on need and amount of requirements T: NL, formal, modeling, etc.	Large amount of requirements continuously growing T: NL (scalability (cost) of modeling and formal specifications is an issue)
Change (management)	Requirements change handled in communication with customer, new requirements alternatives allow for more resources and/or extended time delivery time, i.e., renegotiation	Requirements change handled by the developing organization in communication with internal parties (development, management, marketing, etc.). Generally rigid demands on time-to-market (fixed release dates), new requirements can be allocated to next release

re-allocation of all requirements to a certain release in the worst case, due to dependencies (Table II).

The work described can be performed by a steering committee [44] or through the use of a traditional Change Control Board (CCB) [70]. The main parties involved in change management in the case of MDRE are internal to the development organization as external stakeholders (e.g., key customers) seldom possess the big picture needed for requirements selection. An important factor to remember is that the time-to-market (release dates) are generally set, thus the selection of requirements for a release has to adapt to this fact [44, 71]. In MDRE time aspects are even crucial enough to be prioritized over quality aspects [72].

Owing to the potentially large amount of requirements (which is continuously growing) traceability issues are crucial in this context [43]. For example, traceability to source enables sources to be elicited for input regarding re-prioritization, or if the source is external they can be informed about delays if a requirement is reallocated to a later release. In addition, traceability to the actual specifier of a requirement can also be crucial, especially in a large organization where any number of people can specify the requirements.

2.1.7. MDRE—Challenges. There are several challenges/issues that can be identified in relation to MDRE. Some are evident from the description of the generic MDRE process while others are based on the industry experiences described in the literature [9, 24, 38–40, 44, 48, 60, 71, 72].

DOI: 10.1002/smr

Some of these issues can also be applicable for bespoke RE, although for the purpose of the work presented here a market-driven perspective is adopted.

Challenge 1—Requirements Elicitation. The research in bespoke RE has produced many techniques for requirements elicitation [7]. One common thing across most techniques is that they rely on access to customer stakeholders. This is the case with interviews, brainstorming sessions, observation, investigation of customer documents, and so on. However, these techniques are not very useful in MDRE, since the customer role in this case is not clearly known. Access to stakeholders may be limited, or stakeholders may just be too many. Even when some customers are available (e.g., key customers), which can be considered during elicitation, there is always a doubt as to how accurately they represent the rest of the market. This poses a challenge to market-driven organizations, since they need to resort to other means to elicit requirements. In this case, the marketing organization has an important role, since it is in charge of market and competitor analysis, and can be seen as a replacement for the customer stakeholder.

Challenge 2—Requirements overload. Large amounts of requirements are not only a potential threat (overload) but also an opportunity as they can be seen as input to the MDRE process containing information about the needs and wants of different stakeholders. The implication being that curtailing the inflow of requirements is not an option, but rather that the MDRE process needs to be able to handle large amounts of data continuously. The heavier the load an MDRE process can handle the greater the chance of catching more relevant requirements (more input gives more information). However, more input does not equal better product offering if the requirements overload the organization.

Challenge 3—Abstraction level and contents of requirements. Large amounts of requirements from multiple and diverse sources generate requirements that can be described as everything from abstract 'one-liners' and goal-like statements from marketing channels, to detailed technical solution proposals from technically adept customers. In traditional RE these raw 'requirements' would not even be considered as proper requirements until they were 'filtered' through a specification and refinement process, involving elicitation activities performed by a requirements engineer getting all information needed to formulate the information as 'proper' requirements. In market-driven development requirements come in the raw form, and any MDRE process needs to take this into consideration as it influences all aspects of later processing, whether it be early triage, analysis and refinement, estimation, prioritization, and ultimately selection. An MDRE process needs to be flexible enough to handle multiple types of requirements.

Challenge 4—Requirements capture and specification. Given the characteristics of various requirement sources, the need for continuous requirements handling, and the potential for high requirements influx, there needs to be a way of capturing all requirements and specifying them properly in order to allow for activities, such as requirements triage and RP, not to mention the creation of requirements-based test cases. The traditional and widespread document-based approach to requirements specification poses several problems for MDRE [53]:

- It is hard to separate and store requirements attributes
- There is no easy way for the many requirements sources to issue requirements (especially when personnel is geographically dispersed)
- Tracking requirements status and version is cumbersome
- It is difficult to handle requirements during RP (e.g., how to assign a requirement to a release or to remove from a release back to the set of available requirements)
- It is not possible to baseline a subset of requirements within a document; rather, the whole document is baselined, a solution that lacks precision
- In MDRE products often evolve over time and reuse of requirements and other artifacts can be advantageous, this is hindered by having a flat document-based approach

These challenges imply that traditional, monolithic, document approaches to requirements specification are of little value for MDRE [26]. A call for a database-driven approach to requirement

DOI: 10.1002/smr

specification is necessary, which should allow for a practical way to issue requirements (so that no idea is missed from any of the potential sources), and allow for easier manipulation of requirements.

Challenge 5—Requirements dependencies. Requirements dependencies (a.k.a. interdependencies) influence primarily requirements selection and RP. An MDRE process needs to enable professionals to take dependencies into account when handling large amounts of requirements. The predominant type of dependencies identified as crucial for the market-driven development involves value-based dependencies, directly related to customer value and development cost [48, 73]. The main characteristic of value-based dependencies is that the development of multiple products and features in a company risk spawning dependencies that cannibalize on customer value. For example, a company that has developed extensive biometric identification software may compete with itself by offering keycard solutions.

Challenge 6—Selection/release planning, Fixed releases (time-to-market), Estimation, Prioritization. In order for estimation and prioritization activities to be possible (prerequisites for selection/release planning taking time-to-market into account) requirements stated need to be good enough for estimation and prioritization. Good enough implies that the consequence of a specific requirement needs to be known. For example, a too abstract requirement may need additional analysis and refinement (break-up and/or more information) prior to estimation efforts. Accurate estimates are especially important in the market-driven context, i.e., crucial for planning activities.

The same can be said for prioritization purposes, although comparability is paramount here. Requirements that are to be compared to each other and prioritized need to reside on a similar abstraction level in addition to giving a good enough view of what they imply.

Many factors influence RP, for example, the value of a requirement for customers and associated cost of implementation, strategic importance, architectural importance, specific customer or specific market [14], market window, market size and market penetration [74], specific markets and/or stakeholders, release theme, and requirements dependencies [14], as well as time, budget, and resources available for implementation [24]. These factors dictate if requirements selected for a release will result in a competitive product, and if it will be possible to implement the requirements by the desired delivery date. In MDRE a fixed release date is often paramount as marketing efforts and release windows demand fixed release dates.

Very few companies consider them all when doing RP, as the knowledge needed is spread over a multitude of roles and across the organization.

Challenge 7—Gap between marketing/Management and technical management/Development. Organizations are often grouped by functional areas, each with specialized personnel who do not always speak the same language, thus incurring communication problems. This fact has been acknowledged in the literature [9, 24, 74]. This gap in communication is accentuated by the different interpretations each party has regarding what requirements are. This is closely related to the discussion about different levels of abstractions of requirements. For example, for marketing, a good requirement is one that brings profit to the company; for developers, it is one that is understandable and complete [26].

Requirements can be seen as the least common denominator on which decisions regarding *what* to include in the product offering is decided. This makes communication between marketing/management and technical management/development crucial. All parties should be able to participate in the decision process from early triage to final requirement selection. In addition, all parties are substantial requirement sources in themselves. For these reasons an MDRE process needs to offer relevant decision support material to multiple roles in order to encourage and support cooperation and joint work. The same situation exists in bespoke product development, however here, the major challenge is often associated with the involvement of the external customer.

Challenge 8—Market pull vs Technology push. Requirements can be categorized into two main types: requirements stemming from aspirations of creating (technical) innovations

DOI: 10.1002/smr

(technology-push) and requirements based on requests/wishes/needs identified in the market environment (market-pull) [38]. Premiering either over the other can have severe consequences. For example, prioritizing innovation may leave current customers unsatisfied, and prioritizing market needs may result in missing the development of innovations that could generate great value and competitive advantages in the future.

The two aspects need to be balanced by way of enabling communication of requirements and their implications between market and technical propagators. In addition, the use of product strategies (roadmaps) in requirement selection activities can help the balancing of these two aspects.

Challenge 9—Requirement change. Requirements change. This is true for both bespoke RE and MDRE. The main difference is that in a market-driven scenario there is no renegotiation with a customer (who probably was the originator of the change), but release time has to be enforced. This may imply that a release is changed, requirements selected are postponed to the next release or even less time is devoted to quality assurance to allow for inclusion of the new requirement [58]. In addition, the risk of getting key customers demanding a certain feature mid project presents challenges as ongoing projects and long-term plans may be influenced by any significant change. In other words, in a bespoke situation, the change initiator is often the organization paying for the development effort, thus renegotiations are possible. In a market-driven situation renegotiation is not possible as the market does not negotiate, and no one customer bears the cost of development.

Challenge 10—Organizational Support (OS). An important factor for successful MDRE is OS. The activities of process areas, such as requirements elicitation, analysis, management, and RP, are dependent on this. This is probably true for bespoke RE as well, but in the MDRE case, the OS becomes even more important when considering continuous handling of requirements, and that good interaction among different roles is paramount. For example, activities such as RP demand good communication, understanding and respect between finances, development and marketing to be successful [74]. In addition, OS is necessary to promote the role of MDRE as the bridge between management and business concerns on one hand, and engineering concerns on the other hand [26]. This has to do with the ability of an organization to develop products which fulfill market and/or specific customers' needs and that are synonymous with successful sales, thus generating good revenues and positive return on investment.

A suitable line organization to address MDRE challenges is needed, with defined responsibilities for product management and marketing. Moreover, concepts, such as product roadmaps and organizational strategies, are important to provide a long-term view for the organization and its products, which will guide MDRE efforts in finding the most proper requirements in order to develop successful products [26, 75].

2.2. *Related work—The current approaches and MDRE*

There exist many good practice guides and SPI frameworks, both specific ones targeting RE, e.g., REGPG [20] and R-CMM [35], and general ones looking at process improvement in general like CMMI [30] and SPICE [76]. These guides and frameworks have several things in common. All of them focus on the development project itself, and see RE as a part of the development, i.e., RE is project initiated. In addition, all of them primarily take the aspect of bespoke development into account when suggesting practices to perform.

Market-driven software development needs an outward view towards markets, where business aspects such as strategic planning play an important role [77]. This affects how requirements are elicited. Elicitation techniques of bespoke RE, such as interviews, brainstorming sessions, and observation, are no longer useful, since the customer role is not always clearly defined for market-driven organizations. Even if key customers can fill a similar role as the traditional customer, there is nothing to say that the key customers are representative of the targeted market segment. Thus there is a need for additional elicitation techniques not really covered by bespoke RE practices.

DOI: 10.1002/smr

In addition, new areas of concern appear in market-driven development. For example, products are evolved throughout a series of releases rather than developed once and maintained like in bespoke development [78]. The activity of RP, thus, becomes very important in order to weigh sometimes conflicting factors, such as available resources, time, product strategies, market analysis, innovations, and key customer wishes [14, 74]. In addition, continuous handling of requirements happens throughout the product life cycle, and release projects are initiated to refine requirements further (in-project RE), before implementing them, which is unlike the practice of project-initiated development in bespoke development where everything happens within the confines of projects [4].

Many organizations developing products in a market-driven context are faced with challenges that are not addressed by the main process improvement frameworks and best practice guides, leaving them without the possibility to change their practices to suit their situation.

3. RESEARCH METHOD AND APPROACH

The MDREPM was created as a response to the challenges listed above. The model was refined in several steps, each involving different research partners. Figure 3 depicts this in five steps, inspired by the technology and knowledge transfer model presented by Gorschek *et al*. [79].

Step 1. *Gap identification*: The need for developing a new process model was identified through the industry experience reports as well as through the industry collaborations (see Steps 3 and 4). Challenges for MDRE were identified and the current best practice guides and SPI frameworks were studied.

Step 2. An initial version of the MDREPM was created based on the current best practice models and practices were added to reflect the challenges identified in Step 1 (see Section 3.1).

Step 3. Static validation of the model implied conducting interviews with researchers in the field to get an initial feedback on the model. This feedback was used to refine the model prior to the next validation (see Section 3.2).

Step 4. The dynamic validation involved testing the model in an industry setting by using it to perform process assessments of three major telecom and software companies. The result from each of these assessments was the assessment results themselves, but also feedback on the model from senior practitioners in the industry (see Section 3.3).

Step 5. Following refinements from Step 4, the model was released in Version 1.0.

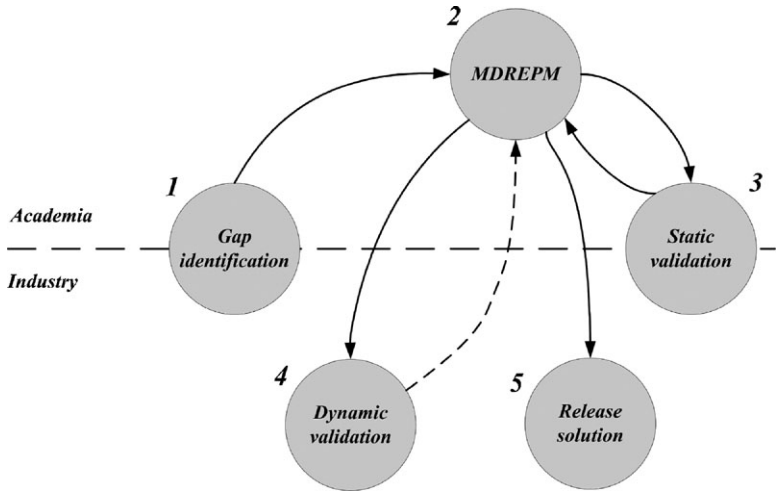

Figure 3. Overview of model creation and validation.

DOI: 10.1002/smr

3.1. Model creation and evolution

An extensive literature survey was the basis for (i) identifying the gaps between the industry needs and the current bespoke best-practice models and (ii) creating the model in its first draft version. This resulted in the study of the fields of strategic management and marketing as well since these were important areas in relation to the market-driven perspective and had to be included in any market-driven best-practice framework.

The MDREPM is based on the Requirements Engineering Process Maturity (REPM) model developed and first presented by Gorschek *et al.* [80]. The REPM model was created as a best-practice guide that could be used for low-cost and fast process assessment and improvement planning, but primarily angled towards bespoke environments, not suitable for the market-driven context. The REPM model was inspired by several of the well-known frameworks, such as CMMI, REGPG, and ISO TickIT guide [31], and was validated in the industry through usage in process evaluation and improvement initiatives [81].

The REPM model (just like the MDREPM) included innovative features *model-lag* enabling practitioners in the industry to use it to perform an assessment to choose what parts of the best-practice framework was inapplicable for their particular organization. This made the *one-size-does-not-fit-all* concept to be included in the model. In addition, the inclusion of graphical representation of process evaluation results proved to be very useful for practitioners using the REPM model [80], and this too is included in the MDREPM. REPM was and is used in the industry today, to perform lightweight process assessments. The MDREPM, similar to the REPM model, is designed to be lightweight in nature, for example a process evaluation using the REPM takes about 40–50 person hours, including both the assessment and post-assessment analysis. One of the main points of the model is to enable any organization to perform a quick snapshot evaluation of the current practices, and suggest additional practices and thus enable process improvement planning without the demand that the organization expend large resources. The main idea behind the lightweight nature of the MDREPM was to be a counterpart to many other SPI frameworks which many times are considered too large and bulky to get an overview of and to implement [82–84], and if implemented return on investment can take anything from 18 to 24 months [85]. This makes it difficult for organizations, in particular SMEs, to initiate and perform assessment and improvement activities as cost and time are crucial considerations [82–84].

The MDREPM can be said to build on the REPM model with regard to both structure and features, and also with regard to the included practices. This means that the MDREPM builds on previous experiences from the REPM model. With this as a base the MDREPM can be seen as a separate model as the practices within the model have undergone a fundamental upgrade and the model has been angled towards the market-driven perspective (including new practices and updating old), which to a large extent was missing from the REPM model.

3.2. Static validation

The static validation of the model was conducted to analyze the completeness, effectiveness, and usefulness of the model prior to the industry validation/piloting. It involved utilizing two experts in the field of MDRE. Both subjects were researchers working in close cooperation with the industry.

The static validation was qualitative in nature and consisted of reviews of the model using checklists and subsequent semi-structured interviews [86].

The two experts contacted were chosen to participate based upon their research as well as their industry experience in MDRE, namely (i) Dr Patrik Berander—Blekinge Institute of Technology and (ii) Mr Richard Berntsson Svensson—Lund University.

Dr Berander's research was focused on RM. More specifically, the research is concerned with studying the decision process and measurements in RM and change management, in order to better understand and improve decision making. For that, requirements prioritization was also a subject of concern, as decisions on what requirements to implement are made during that process.

The second interview was conducted with Mr Richard Berntsson Svensson. Mr Svensson was a researcher at Lund University. He was in the first year of his PhD studies, focusing on MDRE, more specifically on non-functional requirements handling.

DOI: 10.1002/smr

The results gathered during the reviews and interviews were used to refine the model and included everything from changing practice descriptions to discussions about presentation and structure. Further details regarding the static review will not be covered in this paper as it is outside the scope. A detailed description of the static validation, including results can be found in [87].

3.3. Dynamic validation

The second validation was performed in the industry. It involved letting practitioners use the MDREPM to assess their own organizations, and in doing so giving active feedback on the model itself. This was done through letting the practitioners perform the assessment by answering assessment questions, and suggesting improvements to the model itself. Below, the participating organizations are listed with a short introduction. The assessment results themselves are presented in Section 5, and the analysis of feedback on the model itself is presented in Section 5.4.

3.3.1. Industry participants. Three organizations participated in the dynamic validation of MDREPM, (i) Ericsson AB, (ii) Telenor Sweden AB, and (iii) UIQ Technologies AB.

Ericsson AB: Ericsson is a provider of telecommunications equipment and services related to mobile and network operators. Their network equipment is part of over 1000 networks spanning over 140 countries. They also offer through joint venture with Sony Ericsson Mobile Communications a variety of mobile devices. In Karlskrona—Sweden, where the interview took place, the focus is software development, of charging systems, pre-paid, and mobile positioning services among others. The contact persons at Ericsson were Therese O-Starheim and Eva Kristoffersson. They work as Requirements Managers, breaking down requirements from product level to project level, as well as managing tool usage.

Telenor Sverige AB: Telenor offers mobile and fixed communication services for both companies as well as for private customers. The office consulted for this research is situated in Karlskrona—Sweden. However, Telenor is present in 12 countries and their services are used by approximately 1 500 000 customers. The contact person at Telenor was Mr Peter Johansson. He is a project manager, and works with methodologies and tools within Telenor Sweden AB for the department planning and projects.

UIQ Technology AB: UIQ creates and licenses the open software platform, UIQ, to mobile phone manufacturers. Mobile phones from Sony Ericsson, Motorola, BenQ, and Arima use the UIQ platform. UIQ is situated in Ronneby—Sweden and has about 154 employees. The contact person at UIQ was Mr Young Fögelström. He is a project leader, and is also involved with requirements handling.

4. THE MDREPM

The MDREPM is a collection of good practices in MDRE, as well as a process assessment framework, benchmarking the organization under assessment against the recommended practices (the complete model can be downloaded from www.bth.se/mdrepm)[‡]. This is done through a questionnaire that evaluates whether the good practices in the model are fulfilled by the organizations or not, and by graphical representation of the assessment results. In addition, the MDREPM also intends to provide software organizations with a step by step process improvement path towards a better RE process. This is done by organizing the good practices in the model on different levels, and also indicating dependencies between practices. The dependencies help practitioners implement practices in an appropriate order.

[‡]To view and/or download the entire MDREPM including assessment questionnaires please refer to www.bth.se/mdrepm. Observe that this also includes summary tables of practices in both continuous and staged representations.

DOI: 10.1002/smr

The model is structured in five main process areas, (i) OS, (ii) RP, (iii) RM, (iv) Requirements Elicitation, and (v) Requirements Analysis, each holding related practices. The main process areas are described briefly below.

OS: Market-driven product development has a strong market focus, which demands organizations to have an outward look towards markets, competitors, as well as opportunities and threats that may arise in relation to them [6, 77]. On the other hand, an inward look towards the organization itself is also important, for example to foster innovative thinking [88]. Ideas of new products or new product features that are created within the organization can potentially become a success and assure competitive advantages through innovation. Assuring the roles and responsibilities to conduct activities related to marketing, product management, and RE are crucial to ensure that ideas for product features get translated into actual software products by development personnel [40, 53, 74].

These organizational aspects are needed to support the execution of a requirements process. Therefore, the process area OS has been identified for MDREPM. It contains several practices that software organizations can perform in order to ensure a strong market and strategic orientation to their businesses. In addition, this process area is also concerned with practices needed to set up the foundations for a structured and repeatable requirements engineering and development process.

RP: Market-driven software development is characterized by constant product evolution throughout continuous releases [78]. Planning a new release involves deciding which requirements to select for implementation within a project.

These decisions, though, are complicated due to a series of factors. For example, product development is usually constrained by time-to-market aspects, which often clashes with resources available to implement the desired product features [60]. In addition, the process of deciding which features to implement and when can be difficult to perform given the sometimes conflicting interests of different stakeholders [61]. In addition, there are several factors that can be considered when deciding which requirements to implement in a certain release; examples of this can be seen in Figure 4.

Organizations need to define which of the factors to consider, and have clear approaches to handle them. This will enable conscious and explicit decisions on how to get from a list of candidate requirements to a list of selected requirements for a certain release project.

RM: Market-driven organizations are usually faced with high requirements influx from a multitude of sources [26]. These requirements are not only of interest for developers who will implement them, but also for many other roles. For example, a product manager will be interested in a high-level product feature when deciding whether it should be selected for implementation in a release.

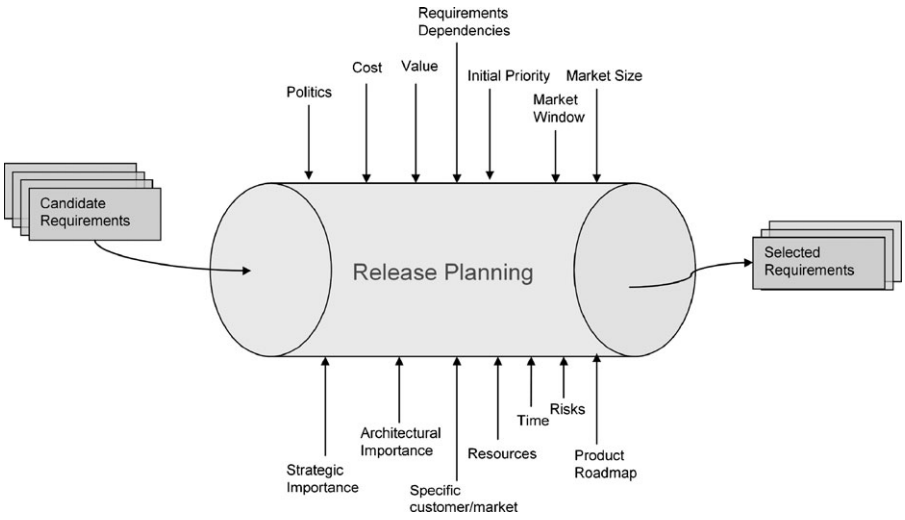

Figure 4. Factors influencing Release Planning.

DOI: 10.1002/smr

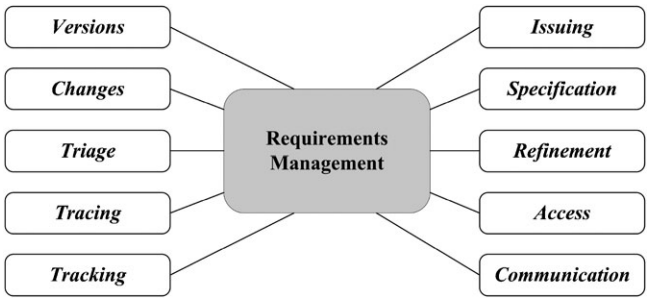

Figure 5. Requirements Management overview.

On the other hand, developers will need more details about the functionalities that compose such features in order to develop them properly, not to mention being able to verify them. Therefore, the way requirements are specified will define their audience [4, 53].

RM covers the procedures to specify requirements that they may be understood and interpreted correctly by different stakeholders. In addition, as illustrated in Figure 5, it is also concerned with controlling changes to requirements, controlling their versions, and providing proper tool support for managing requirements attributes, their life cycle, as well as the relationships between them [7, 53]. In MDRE, several unique aspects may be of relevance. For example, the access rights and views of requirements might be controlled as many roles are involved in the continuous handling of requirements. Offering different views of requirements, as well as ensuring access rights may be relevant not only from a security perspective, but also from a perspective of protecting a certain role from information overload.

Requirements Elicitation: In a market-driven situation, potential customers can be as few as a dozen of known key customers, to as many constituents in a mass market [6]. These customers or market segments need to be considered individually when eliciting requirements but they have to be considered in combination with other sources of requirements as well, e.g., standards, trends, and regulations. Therefore, finding out which requirements should be implemented in a certain product is challenging. The Requirements Elicitation process area in MDREPM contains practices that can be used to identify what requirements sources can be considered, and techniques for eliciting requirements from them.

Requirements Analysis: As requirements are elicited, they can, and usually are, specified at varied levels of detail and quality [7]. The requirement mass may span everything from badly specified requirements with easily caught defects to requirements that look fine at a first glance but hide ambiguities with potential for causing misinterpretations [7]. Requirements Analysis is the process area of the MDREPM that is concerned with the issues mentioned. It contains practices to aid organizations in assuring the quality of their requirements, and to also help in managing a high requirements influx.

4.1. Model structure

The MDREPM has two representations: (i) Representation by Level (similar to REPM [80] or CMMI-staged representation [30]) and (ii) Representation by Process Area (similar to CMMI continuous representation). Figure 6 gives an example of Representation by Process Area[§]. Every main process area can have practices of its own, and subprocess areas with practices. The model can be seen as a tree structure where the leaves are the practices. In addition every practice can have related practices, giving indication to the user that there may be pre-requisites to the practice or dependencies between practices. It should be noted, however, that MDREPM focuses mainly

[§]To view and/or download the entire MDREPM including assessment questionnaires, please refer to www.bth.se/mdrepm.

DOI: 10.1002/smr

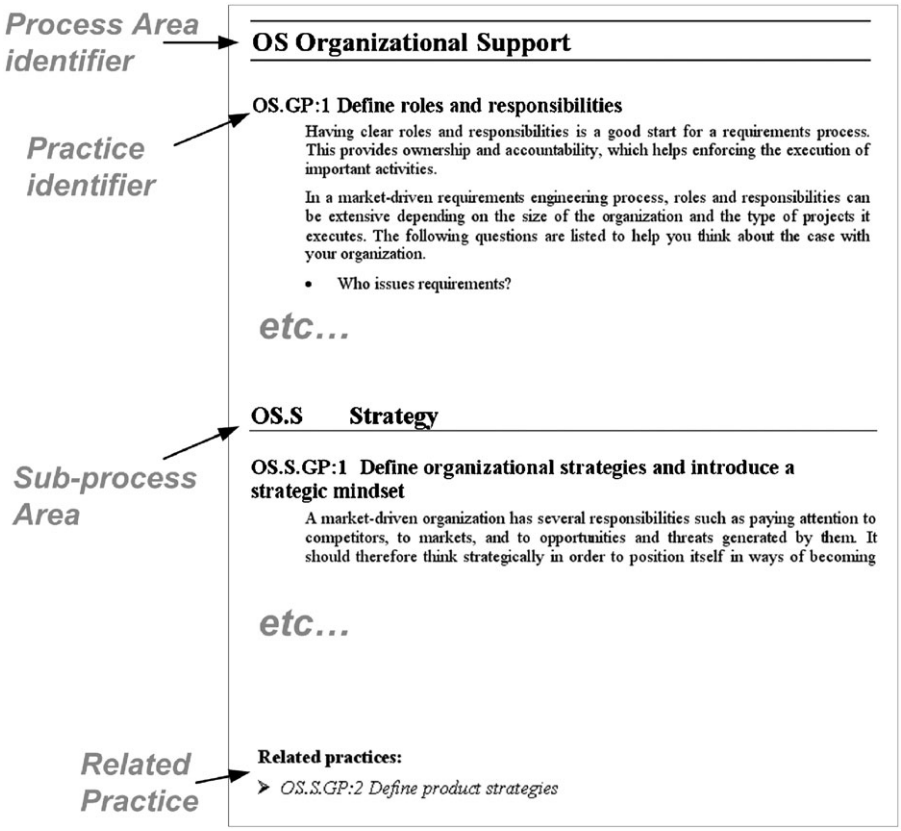

Figure 6. MDREPM structure example.

on the process dimension, unlike CMMI or ISO15504 which have both a process and capability dimension. The main motivation for this was keeping MDREPM lightweight and smaller.

The representation by level structures MDREPM into five levels of maturity and is primarily used for process assessment purposes while the representation by process area can be seen as a good-practice guide that can be used to study an area. Figure 7 gives an example of representation per level, while the staged representation is clarified under Section 4.1.1.

4.1.1. Staged representation and maturity levels An alternative view shows practices according to the level of process maturity they belong to, also known as staged representation [20, 89]. The MDREPM organizes practices at five different levels. The rationale behind the initial placement of practices on maturity levels was inspired by previous models and the validations they underwent in the industry [80], as well as related maturity frameworks [20, 89]. Then, as the model was validated (see Section 5) the placements of practices were also a part of this validation. The goals and main features of each level are described below:

Level 1: The goal of Level 1 is to lay out the ground for the creation of an MDRE process. This is done by introducing practices from the process areas of OS, RM, and Requirements Elicitation. By implementing the practices of level 1, organizations are expected to acquire a strategic and market orientation to their businesses. In addition, they also commit to implement a requirements process by introducing tool support for RM and assigning roles and responsibilities to ensure that requirements-related activities get performed. Some basic elicitation techniques are also introduced at this level.

Level 2: builds upon level 1 and assumes that the organization attains a deeper awareness about its own strengths and weaknesses. This includes an analysis of the company's relation and standing relative to internal and external stakeholders as well as competitors. The development

DOI: 10.1002/smr

Level 3
OS Organizational Support
OS.S Strategic
OS.S.GP:4 Spread strategic thinking throughout middle-management
OS.S.GP:5 Define a roadmapping process
OS.S.GP:6 Let requirements affect product strategies when applicable
OS.M Marketing
OS.M.GP:6 Identify critical success factors for specific markets and/or key customers
OS.M.GP:7 Identify the basis of the competitive advantage
RP Release Planning
RP.GP:3 Involve different perspectives in release planning
RP.GP:4 Use product strategies and product roadmap to guide release planning
RP.GP:5 Plan more than one release at a time
RP.P Prioritization
RP.P.GP:2 Prioritize requirements based on their abstraction levels
RP.P.GP:3 Prioritize requirements based on cost, value and risk

Figure 7. Example of practices by level (extract from Level 3 seen here).

of product strategies is also recommended as part of the practices in OS at this level, as is the utilization of explicit practices in RP. Moreover, RM builds upon practices of level 1 by introducing version and change control for requirements and basic traceability control. Requirements Elicitation introduces more advanced techniques for eliciting requirements.
Level 3: The main goal of level 3 is to take the strategic orientation acquired with the definition of organizational and product strategies in previous levels a step further. This is done by provoking the organization to better understand what customers and/or markets judge as important for their products. Strategies are also used to guide RP activities at this level.
Level 4: Its main contributions are the introduction of advanced RP practices. These are related to advanced prioritization of requirements which consider dependencies and different stakeholders' importance. OS contributes by introducing a practice for product portfolio management at this level.
Level 5: Advanced practices in RP and elicitation are introduced. These consist of taking a pro-active approach to improve decisions on requirements selection by making postmortem analysis of previous decisions in RP. Elicitation activities are now guided by product strategies, and OS is improved by motivating the organization to foster the development of a creative environment which supports innovation [90].

Two main criteria were used to ascertain on which level a specific practice should reside. One was using previous models as input to judge maturity level, as mentioned previously, however, dependencies between practices were also used as a strong indicator of maturity level. That is, practices on advanced levels can depend on practices on more base levels, but not the other way around. This presents the model and the staged representation with a problem as a perfect practice—maturity match is impossible. However, the process assessment and the step-wise improvement scheme, presented to companies being assessed are dependent on a staged representation. Thus, we are fully aware of the possible limitations in terms of mapping maturity (or dependencies for that matter), and this is communicated in the model. Further, during the validation of the model (see Sections 3.2 and 5) maturity level placement and dependencies were also a part of the validation.

For a complete list of practices, as well as a summary of practices per level see MDREPM v.1.0, pages 13–17, downloadable at: www.bth.se/mdrepm.

4.1.2. Continuous representation and maturity levels. The MDREPM can also be used as a model with continuous representation. This is done through the selection of a process area, and using the subprocess areas and GPs (Good Practices) associated with the process area. Under each area the

DOI: 10.1002/smr

overall purpose and goals of the said process area are clarified and can be used as an indicator of what should be achieved. However, unlike CMMI continuous representation, MDREPM does not offer a differentiation into specific or generic goals and practices, but rather focuses on specific GPs for each area. It should be observed, however, that MDREPM is mainly designed with staged representation in mind, and is used as such in this paper. Also, the process evaluation examples presented in this paper are based on the staged representation of MDREPM.

For a complete list of practices see MDREPM v.1.0, page 18, downloadable at: www.bth.se/mdrepm.

4.2. Process assessment with the MDREPM

The MDREPM can be seen as a good-practice guide, but it can also be used for process assessment and result presentation. For this, two additional parts are included in the model, namely a questionnaire and a graphical representation of the results giving an easy overview of the process status for the assessed organization.

The assessment questionnaire is composed of several questions (or groups of questions), aiming to assess each process area in the model. Questions can be answered in one of the following three ways, 'YES', 'NO', or 'SATISFIED/EXPLAINED'. If a practice is fulfilled by the organization, the answer to the question should be YES, if not, NO.

On the other hand, if the practice is judged not applicable to the organization, the question can be answered as SATISFIED/EXPLAINED. That is, the practice is satisfied for assessment purposes, but it is explained why it is not applicable for the organization performing the assessment. This allows the model to be useful for a broader range of organizations, as it does not demand them to fulfill practices that are not relevant to their domain and/or organization. For example, a start-up might not have any customers yet, and the product is an innovation that is not present on any market. In this case a practice instructing to identify representative key customers is not relevant at this stage of the company's development. By enforcing a YES and NO answer there would be no room for taking inapplicability into account for every individual case. The organization doing the evaluation makes the distinction when an action is to be considered SATISFIED/EXPLAINED. It is important to notice that a practice should not be deemed SATISFIED/EXPLAINED for reasons, such as lack of time, lack of money, lack of know-how, or just 'did not think of it'.

As a result of SATISFIED/EXPLAINED a new concept can be seen, namely *model-lag*. Model-lag is a summation of the practices deemed SATISFIED/EXPLAINED and can be seen as the distance between the model and the organization performing the evaluation. A high model-lag may indicate that a large part of the practices are not applicable for the organization, in which case the model might be inappropriate. On the other hand, it might also indicate that the assessor is justifying the lack of utilized practices. Either way, a high model-lag should be accompanied by an evaluation as to the source of the model-lag.

The assessment results, as well as the model-lag, can be visualized in a graphical representation, exemplified in Figure 8. The solid 'MaxMDREPM' line depicts the total number of actions per maturity level, for example for level two there are 20 practices. The gray 'Fulfilled' line depicts

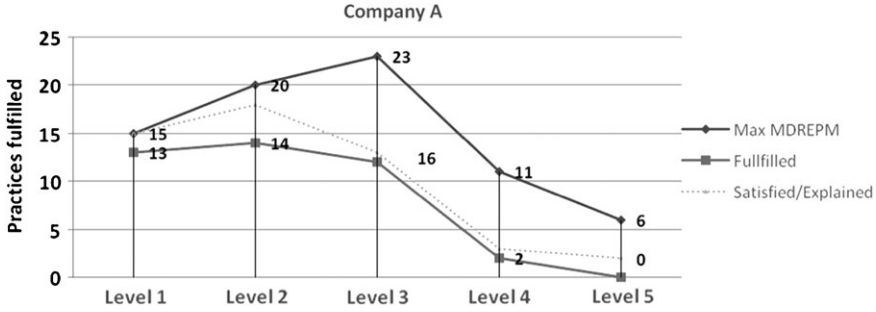

Figure 8. Example of graphical representation of assessment results.

DOI: 10.1002/smr

the number of practices performed/used by the company, and the dashed line represents the number of Fulfilled practices including the ones deemed to be SATISFIED/EXPLAINED. This kind of diagram can be drawn for the total process assessment, or per process area, depending on analysis need.

From the companies' perspective, following the example in Figure 8, the area between the Fulfilled line and the SATISFIED/EXPLAINED line represents model-lag, i.e., to what extent the model is not applicable to the organizations' situation. However, the area between SATISFIED/EXPLAINED and MaxMDREPM represents the possibility of improvement the company has, by its own admission as the 'irrelevant' practices have been sorted under SATISFIED/EXPLAINED.

From the models perspective, high levels of model-lag might indicate that the model needs to be updated or is inapplicable for a certain domain.

5. DYNAMIC VALIDATION—USING THE MDREPM IN INDUSTRY

The dynamic validation, testing the model in the industry, consisted of performing process assessments in three different companies: Ericsson AB, Telenor Sweden AB, and UIQ Technologies AB. This had the purpose of evaluating the model's usefulness and applicability in the industry, as well as collecting feedback regarding how the model could be improved. Each assessment took 2–4 hours to perform.

This section presents the results of these validations (case studies) in each company, and the following structure is adopted. Each case is presented and contains three subsections. The first subsection presents the assessment results, the second discusses the results with respect to the model's practices and finally, the third discusses the results with respect to the challenges of MDRE presented earlier in Section 2.1.7. The assessment result presentations are kept to only a sample of the data gathered from each company for reasons of brevity.

It should be highlighted, though, that the order in which the assessment results are presented does not in any way relate to the order of the presentation of the organizations above. The choice to anonymize the data has been chosen since the purpose of the dynamic validation was not to compare the maturity of RE processes of different organizations. Rather, the purpose was to demonstrate how the MDREPM could be used in an industry, and how the presentation of assessment results could be used to target improvements. The participating organizations received a complete analysis of the assessment results, which also contained suggestions of practices to include and/or improve. Based on the process assessment and analysis the companies then gave explicit feedback on the model which is detailed in Section 5.4.

5.1. Case study A

5.1.1. Company A—Assessment results. Two process areas are chosen for discussion in the case of Company A, namely OS and RP. These were chosen given the high level of dependencies between the process areas and the discrepancies in maturity level between them.

Figure 9 shows that Company A fulfills a significant number of practices in OS—50% or more in most levels. However, looking at Figure 10 it lacks many practices in RP, as revealed by the large area between the curves Max MDREPM and Fulfilled. Another point revealed in Figure 10 is a small deviation of the model from the company's perceived reality, which can be seen by the dashed line stating that one practice (out of two total for the area) is considered Satisfied/Explained. In other words, the model is shown to have a slight deviation since it has a practice that is not useful for Company A.

Some other points worth mentioning are the relatively high level of fulfilled practices on higher levels in OS compared to lower levels in the same process area. For example, on levels 2 and 3 60% or more of the practices are fulfilled which can be considered quite a lot when more basic practices on level 1 are unfulfilled. This shows that the company implements practices that in the

DOI: 10.1002/smr

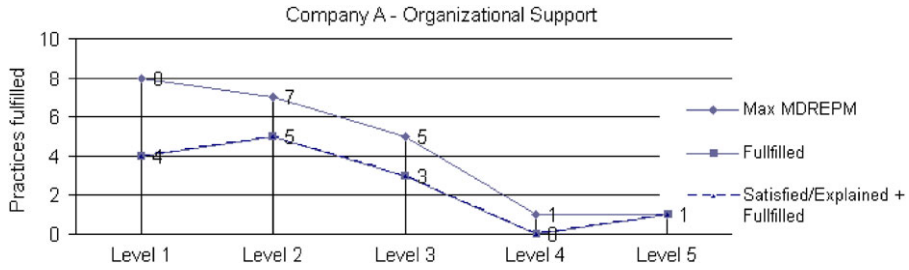

Figure 9. Organizational Support—Company A results.

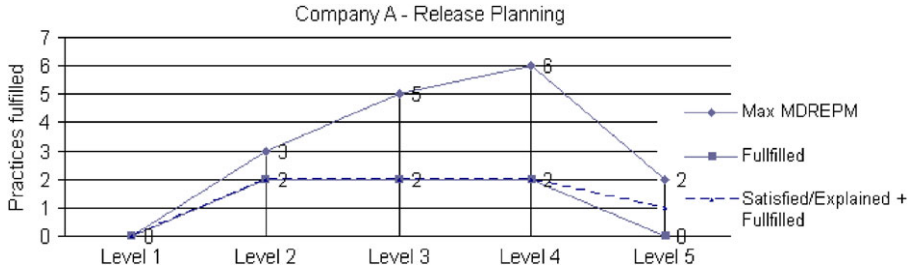

Figure 10. Release Planning—Company A results.

model are considered more advanced, although it does not implement practices considered on a basic level.

5.1.2. Company A—Analysis of results. The RP process area is strongly built upon basic OS practices. This is shown by the shape of the curves Max MDREPM in the graphs above. The OS graph shows that curve starting at a high number in lower levels, and then going down as levels advance. The RP graph, on the other hand, shows the curve with the opposite behavior. In other words, RP relies on basic OS practices to be in place before its practices are put in place.

The detailed assessment results, not discussed here, reveal that practices not implemented in levels 2–4 of RP are related to missing practices on level 1 of OS. For example, Company A does not implement the strategy-related practices '*OS.S.GP:2 Define product strategies*', '*OS.S.GP:5 Define a roadmapping process*', and the marketing-related practice '*OS.M.GP:5 Identify stakeholders and map their influence*'. As a consequence, the practices '*RP.GP:4 Use product strategies and product roadmap to guide release planning*' and '*RP.P.GP:5 Consider multiple stakeholders during requirements prioritization*' are not implemented in RP.

The findings above indicate that the MDREPM can be useful in revealing why certain practices are not implemented as a consequence of dependencies between process areas. The model indicates relationships between practices, and organizes them in a way that helps companies visualize the big picture of what is needed to have a fully fledged MDRE process. By having graphs like the ones above, in association with tables showing the answers to each question of the assessment, it is possible to get an understanding of the missing basic practices that need to be in place before the benefits of more advanced practices are maximized.

5.1.3. Company A—Model discussion. Previously, many challenges related to MDRE were discussed in Section 2.1.7. From the perspective of this case challenges *#2 Requirements overload* and *#3 Abstraction level and contents of requirements* are relevant. These challenges are related to each other in that handling the overload can be helped by defining requirements in different abstraction levels. When requirements are in a high level of abstraction, they can be compared against product strategies [4]. Requirements that do not align with strategies can be discarded early, thereby preventing them from contributing to overload, enabling triage at an early stage [4, 91].

DOI: 10.1002/smr

Company A could gain the benefits of better handling requirements overload by implementing the missing practices revealed in the assessment. For example, by first defining its product strategies, the company could use those to not only guide its RP efforts, but also to foster requirements triage. This should, of course, be accomplished by also implementing the practice '*RM.RS.GP:4 Specify requirements in multiple abstraction levels*', in the RM process area (not mentioned above).

It should also be observed that since Company A fulfills several advance level practices, they also missed several basic level practices. This can have several implications. One, that the company does certain things unofficially or implicitly (e.g., define product strategies). In this case the practices on higher levels (e.g., RP activities based on product strategies) might be improved if explicit base practices were satisfied.

Another implication might be that the MDREPM model itself and the division of practices across levels needs to be further validated, to rule out the model itself as a factor. This risk cannot be dismissed, however, it is not confirmed by the other cases below.

5.2. Case study B

5.2.1. Company B—Assessment results. Similar to Company A, the assessment results for Company B also reveal a mature requirements process. RM is the process area where the company has the most number of fulfilled practices, whereas it lacks practices in OS and Requirements Elicitation (RE), as can be seen in Figures 11 and 12.

The graphs show that the company has some potential for improving both process areas, as the area between the Max MDREPM and the Fulfilled curves is big, especially in Figure 12. A small model-lag is also noted in Figure 11, at a level, where one practice is considered Satisfied/Explained.

5.2.2. Company B—Analysis of results. An interesting finding during the analysis of Company B's results is that the company lacks the practices *OS.GP:3 Assign a person to manage and own the requirements process* and *OS.GP:4 Train people in requirements engineering* (*both in level 1 of OS; notice the missing practices in the first graph*). This in turn has an effect on other process areas, one of which is Requirements Elicitation (RE).

The effect of not having a clear definition of process ownership is that the implementation of certain practices might be compromised. For example, it could be difficult to implement practices like *RE.GP:2 Identify requirements sources* and *RE.GP:5 Create elicitation channels for requirements sources* without a dedicated role to manage the requirements process. In addition, other parts

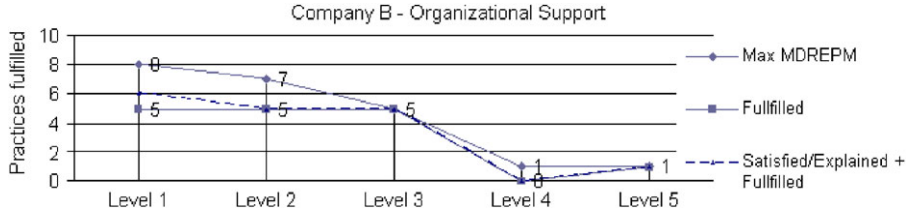

Figure 11. Organizational Support—Company B results.

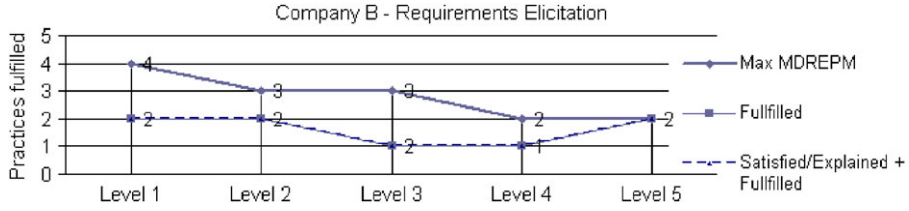

Figure 12. Requirements Elicitation—Company B results.

DOI: 10.1002/smr

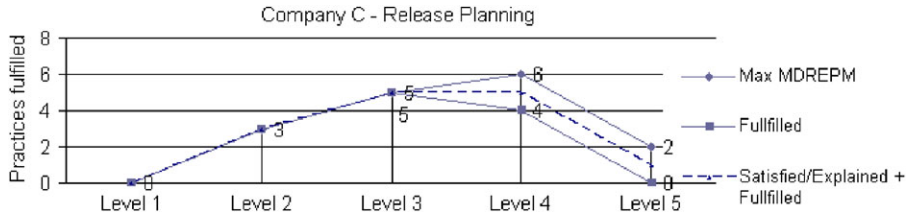

Figure 13. Release Planning—Company C results.

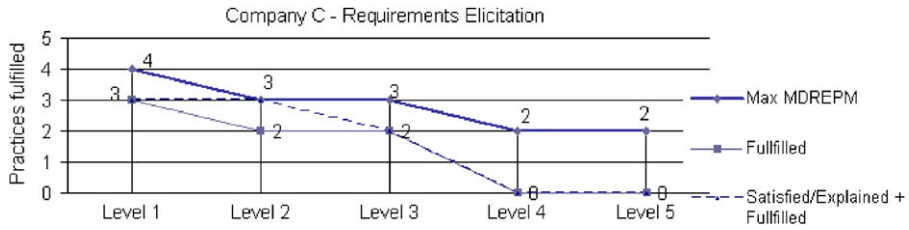

Figure 14. Requirements Elicitation—Company C results.

of the process may result in low quality assurance of the process itself, resulting in low quality decisions as a consequence of lacking requirements quality.

5.2.3. Company B—Model discussion. The results shown in this case study indicate a good match between the practices in the MDREPM and what the company considers applicable to their reality. This is illustrated by a small model-lag (only one practice was specified as Satisfied/Explained).

Further, the analysis indicates that the company has some challenges to overcome in Requirements Elicitation, as it is the process area where the company fulfills the least number of practices. This is connected to challenge *#1 Requirements Elicitation*, which describes the need for different techniques for eliciting requirements in a market-driven situation. For example, the company could consider implementing a number of techniques suggested in the MDREPM, such as

- *RE.T.GP:2 Elicit requirements through user groups*
- *RE.T.GP:4 Elicit requirements through personas*
- *RE.T.GP:6 Elicit requirements through customer value analysis*.

5.3. Case study C

5.3.1. Company C—Assessment results. The last case study, performed with Company C, reveals that the company fulfills a significant number of practices in the process areas OS and RM. This is very positive, since RM and OS are the process areas which enable the basic foundation for enabling practices in other process areas.

However, the company lacks a few practices in RP and Requirements Elicitation (RE), as can be seen in Figures 13 and 14.

5.3.2. Company C—Analysis of results. The company fulfills all practices on levels 1–3 of RP. However, practices on levels 4 and 5 can be used to further improve RP activities. One of them is '*RP.P.GP:6 Let stakeholders' importance be a factor during prioritization*'. This practice is a good candidate for implementation since the company already implements the practice '*OS.M.GP:5 Identify stakeholders and map their influence*'. When the preferences of a new release are driven by different interests, a systematic approach to prioritizing requirements, taking stakeholder importance into consideration, can be helpful to achieve a good selection of requirements.

Looking at Requirements Elicitation, the company misses an important practice, namely '*RE.GP:1 Distinguish between end-user and customer*'. Implementing such practice could help the company improve elicitation accuracy, given that customers and end-users may be two different

DOI: 10.1002/smr

groups. For example, focusing too much on the customers may lead to usability requirements being overlooked (mainly a concern for end-users or the customers). On the other hand, focusing only on usability may cause other requirements to be overlooked that are important for the direct customer.

Company C fulfills the practices to specify both product and organizational strategies, but does not use them as they do not fulfill the practice of '*RE.GP:7 Align elicitation activities with product strategies*'. The implementation of this single practice would enable the strategies already defined to be used directly in requirements elicitation and for triage.

5.3.3. Company C—Model discussion. Company C is well prepared to face many of the MDRE challenges presented in Section 2.1.7, as it fulfills a great number of practices stated by the MDREPM. For example, it addresses challenge *#8 Market Pull vs Technology Push* by implementing the practices:

- *OS.M.GP:4 Identify and analyze competitors*
- *OS.M.GP:6 Identify critical success factors for specific markets and/or key customers*
- *OS.M.GP:7 Identify the basis of competitive advantage*
- *OS.GP:7 Create an environment that fosters innovative thinking*
- *OS.GP:6 Establish a teamwork mindset*.

The practices above allow the company to be aware of what the market is in need of (market pull), in addition to also enabling innovation and teamwork, which together can contribute to ideas that can be pushed into the market, creating new product demands.

In addition, the company addresses challenge *#7 Gap between Marketing/Management and Technical Management/Development* by implementing the following practices:

- *OS.S.GP:4 Spread strategic thinking throughout middle management*
- *OS.GP:5 Create a cross-functional team to analyze and specify requirements*
- *RP.GP:3 Involve different perspectives in release planning*.

These practices contribute to lessening the gap between technical and managerial perspectives, as they foster the interaction between them (e.g., by having cross-functional teams in activities such as RP).

The findings above indicate that the MDREPM can be of help in addressing typical market-driven issues, which have been absent from previous models that focused on bespoke RE. The model defines the practices related to marketing and strategic planning like the above, which is new to models of the kind. Moreover, it also adapts traditional process areas to also address market-driven issues. An example is Requirements Elicitation, which now counts on practices, such as *RE.GP:6 Consider strengths and weaknesses of competitor's products* and *RE.GP:7 Align elicitation activities with product strategies*, which are typically market-driven.

5.4. Feedback collected during dynamic validation

The case studies presented above were used to test (validate) the MDREPM model in the industry, and the potential improvements to the model collected from the practitioners is presented in this section.

Answer types. During the case studies, one of the company representatives raised the idea of using a different approach to the assessment questionnaire. Rather than using Yes/No questions, it was suggested that a scale of 1–5 could be used instead. The scale would be used to rate the extent to which a practice is implemented.

Initially, during the development of the model, there was an idea of using such a scale for the assessment questionnaire. However, this would make the computation of the assessment results more complicated, and the interpretation of these results as well. For example, if a company rates most of its RE practices as being 3 on a scale from 1 to 5, are those practices in such a state that they do not need to be looked at for improvement? If they are considered lacking, how do you

DOI: 10.1002/smr

evaluate what is missing in the practice? Although the scale would make it easier for interviewees to answer the assessment questions, we still think that Yes/No questions are a better approach, and encourage companies to answer *No* in case they do not see a practice as completely fulfilled. The reason for this is that answering a *No* will make the assessment results in the graphical representation clearly show that there is room for improvement in some point of the requirements process.

It should be stated that the main objective of MDREPM is not to rate companies on a certain level of maturity. The model does not encourage the pursuit of a higher level just for the sake of it. Rather, it encourages companies to acknowledge their challenges, even if they are partially solved.

The model then tries to help companies to solve such problems by offering a description of a good practice in MDRE that can be implemented to improve their requirements process.

Suggestions of practices. Some practices that were not present in the model have been suggested as being useful from the perspective of some of the companies. The following is a summary of these suggestions.

- How to write good requirements

One interviewee reported that requirements come at very different levels of detail, and that people who specify requirements are not always aware of the right level of detail to write for a given situation.

MDREPM does contain a practice for encouraging companies to specify requirements at different levels of abstraction, as a way to solve the issue reported above. However, MDREPM does not contain detailed directions on how such requirements should be written. The model is focused on suggesting good practices addressing the market-driven perspective, and does not contain any detailed instructions on how these practices should be implemented. This is usually the nature of process assessment models. A future version of the model could have appendices suggesting how practices might be implemented, but these should cover all practices and is out of the scope of the current model.

A Requirements Analysis lead role. One of the interviewees made a suggestion to include a practice pointing at having a defined role for a *Requirements Analysis lead* role. According to the interviewee, it is important that a dedicated person be responsible for overseeing the whole analysis process, evaluating requirements and the decisions based on them, eliminating all assumptions before the requirement was implemented.

MDREPM contains a general practice for defining roles and responsibilities for the requirements process. However, we do recognize the value of detailing this practice further, and this suggestion will be considered for the next version of the model.

Care for supplier deliveries. Another suggestion coming from a company representative related to caring for supplier deliveries. The reasoning was that in some situations, the market-driven organization may be part of a larger supply chain, where dependencies on software components from other organizations may cause delays, or even obsolete requirements, depending on delivery times.

Although dealing with time aspect is more of a project management responsibility, the fact that some organizations have to plan their requirements based on other organizations' software requirements is indeed an issue. This could be taken care of by practices related to managing software requirements for COTS (commercial off-the-shelf) software. This suggestion will, therefore, be considered for a future version of the model.

5.5. Applicability of the model

The general opinion of the model based on feedback from all three companies was that it provided a comprehensive coverage of what is involved in defining a MDRE process. This can be seen

DOI: 10.1002/smr

as the improvement suggestions described above are relatively moderate in quantity. Moreover, company representatives were unanimous in confirming that the practices in the model are, or could be, useful for their cases. This was further confirmed by the low number of Satisfied/Explained answers used during the assessments. In model terms, having few Satisfied/Explained answers is an indicator of low model-lag.

It should be observed that the dynamic validation of MDREPM, holds a potential flaw, that is, given the results and suggestions given by using the model, would changes to the organization under evaluation give rise to actual improvements? This cannot be answered fully through application of the model, rather it needs to be checked post-improvement, and through measurement of the improvement effects. The use of expert opinion as in the case of the dynamic validation presented in this paper is thus the first substantial manner in which the model needs to be evaluated.

5.6. Model size and assessment time

When using a process model for assessment and improvement suggestion, the cost of an assessment and obtaining results should always be considered. The three case studies conducted during the dynamic validation occurred without any problems, and within the expected time of 2–3 h for the assessment interview. It should be noted that during this time, 76 questions were posed, one for each practice in the model. As the questions were of Yes/No type, it usually took little time to obtain an answer, although in some cases the company representatives found it difficult to use such a black and white approach to evaluate their requirements process.

One of the goals of MDREPM was to provide a fast assessment, whereby problem areas in the requirements process could be identified expending limited resources. Given the results obtained with the three case studies, we believe MDREPM has achieved this goal, as explained below.

The overall assessment for any of the three companies did not take more than 3–5 h, computing the interview time plus the analysis of the answers in a spreadsheet in order to generate the result graphs. In the case studies, answers were noted down on paper and then transferred to a spreadsheet. However, if tool automation was used, it could be possible to obtain the result graphs immediately after the interview.

In addition to 3 h for each interview and result compilation, a further 15–20 h was needed to write a report for each company with the result presentation and improvement recommendation. This was the most time-consuming task as it involved analysis of all result graphs, along with individual answers to the questionnaire. The results were then checked against the model in order to suggest improvements through implementation of missing practices.

6. CONCLUSIONS

MDREPM was motivated by the many challenges that organizations face when developing software in a market-driven environment, and the lack of good-practice and assessment models taking these challenges into consideration. In addition, the model offers assessment capabilities that take the unique situation of an organization into consideration, not forcing practices, but allowing for the statement and usage of model-lag which takes the model's imperfections into account. One size does not fit all.

The use of graphs depicting fulfilled practices, areas of potential improvement, and model-lag gives organizations the possibility to get a fast overview of the current situation and possibilities for improvement. On a more detailed note, the stated dependencies between practices (in and cross-process areas) offers support for understanding the challenges of interacting practices, and also gives indication as to order for improvement and how an organization can maximize the benefit of already realized practices by implementing subsequent ones that take advantage of work already being performed.

The model was developed in cooperation with, and subsequently validated in the industry in an attempt to assure usability and usefulness. One perspective is the model's coverage in terms of practices, which can be considered adequate, although room for improvement can definitely

DOI: 10.1002/smr

be found looking at the industry feedback post validation. Another perspective is the cost of assessment. In total the cost of using the model can be estimated to under 30 h, including the assessment itself, subsequent result generation and analysis and recommendation. This cost must be considered very moderate.

There are many possibilities for improving the MDREPM model, based both on the industry feedback obtained thus far, but also in terms of further investigating the market-driven context adding practices not yet identified. In addition, the limited validation of the model (three case studies) should be expanded in order to collect further feedback for improvement. Optimally, the MDREPM model should be compared to other best-practice and assessment models, but to our knowledge there are no such models taking the market-driven context into account. A very important part that has to be evaluated in detail is the issue of practice dependencies.

The main contribution of this paper is the presentation of MDRE and the characterization of the area in contrast to bespoke RE, highlighting the unique challenges facing organizations developing products in a market-driven context. Using this as a backdrop the MDREPM model is presented, containing over 70 practices and their interdependencies divided over five process areas.

The MDREPM model presented should be considered as a starting point. The future research and investigation of MDRE will offer opportunities for both refinement of the model itself, and also for identification of new challenges and addition of practices enabling organizations to optimize their processes.

7. FUTURE WORK

The MDREPM model needs further refinement, as any maturity model it must evolve over time. This includes further validation in the industry. In addition, the model at this stage is fairly generic, thus possibilities of tailoring for different domains will be explored.

The dependencies between the practices within the model like AND, OR, REQUIRES, and also value-based need to be explored. This requires the collection of metrics from many companies over time.

A central issue of any validation of a maturity model is the evaluation of its impact, that is if an organization uses the model for improvement, what is the return of investment, this is the ultimate test of any maturity model. This requires long-term use and measurement, work that is on the way.

Any changes and updates to the model will be possible to follow on the model home pages: www.bth.se/mdrepm or http://www.gorschek.com/doc/REPM_Project.html.

REFERENCES

1. Ruhe G, Greer D. Quantitative studies in software release planning under risk and resource constraints. *Proceedings International Symposium on Empirical Software Engineering* (*ISESE*). IEEE: Los Alamitos CA, 2003; 262–271.
2. Butscher SA, Laker M. Market-driven product development. *Marketing Management* 2000; **9**(2):48–53.
3. Sommerville I. *Software Engineering*. Addison-Wesley: Essex, 2001.
4. Gorschek T, Wohlin C. Requirements abstraction model. *Requirements Engineering Journal* 2006; **11**(1):79–101.
5. Fricker S, Gorschek T, Myllyperkiö P. Handshaking between software projects and stakeholders using implementation proposals. *International Workshop on Requirements Engineering*: *Foundation for Software Quality*, (*Lecture Notes in Computer Science*, vol. 4542), 2007; 144–159.
6. Regnell B, Brinkkemper S. Market-driven requirements engineering for software products. *Engineering and Managing Software Requirements*. Springer: New York NY, 2005.
7. Kotonya G, Sommerville I. *Requirements Engineering*: *Processes and Techniques*. Wiley: New York NY, 1998.
8. Wieringa R, Ebert C. Guest Editors' Introduction. Re'03: Practical requirements engineering solutions. *IEEE Software* 2004; **21**(2):16–18.
9. Karlsson L, Dahlstedt Å, Natt och Dag J, Regnell B, Persson A. Challenges in market-driven requirements engineering—An industrial interview study. *Proceedings of the Eighth International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ'02*), 2003.
10. Kotler P, Armstrong G. *Principles of Marketing*. Prentice Hall: Upper Saddle River NJ, 2001.

DOI: 10.1002/smr

11. Gorschek T, Davis A. Assessing the quality of requirements process changes. *Proceedings of the 11th International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ'05*), 2005; 101–112. Available at: http://gorschek.com/doc/publications.html [10 January 2011].
12. Lehmann DR, Winer RS. *Product Management*. McGraw-Hill: Boston MA, 2002.
13. Mintzberg H, Ahlstrand BW, Lampel J. *Strategy Safari*: *A Guided Tour through the Wilds of Strategic Management*. Free Press: New York NY, 1998.
14. Carlshamre P. Release planning in market-driven software product development: Provoking an understanding. *Requirements Engineering* 2002; **7**(3):139–151.
15. Bray IK. *An Introduction to Requirements Engineering*. Addison-Wesley: Dorset, 2002.
16. Glass RL. *Software Runaways*. Prentice Hall: Upper Saddle River NJ, 1998.
17. Connolly TM, Begg CE. *Database Systems*: *A Practical Approach to Design*, *Implementation and Management*. Addison-Wesley: Harlow, 1998.
18. Jirotka M, Goguen JA. *Requirements Engineering Social and Technical Issues*. Academic Press: London, 1994.
19. van Buren J, Cook DA. Experiences in the adoption of requirements engineering technologies. *Crosstalk—The Journal of Defense Software Engineering* 1998; **11**(12):3–10.
20. Sommerville I, Sawyer P. *Requirements Engineering*: *A Good Practice Guide*. Wiley: Chichester, 1999.
21. Davis AM. *Software Requirements*: *Objects*, *Functions*, *and States*. Prentice Hall: Englewood Cliffs NJ, 1993.
22. Weber M, Weisbrod J. Requirements engineering in automotive development: Experiences and challenges. *IEEE Software* 2003; **20**(1):16–24.
23. Simmons E. Requirements triage: What can we learn from a 'Medical' approach? *IEEE Software* 2004; **21**(4):86–88.
24. Davis AM. The art of requirements triage. *IEEE Computer* 2003; **36**(3):42–49.
25. Wohlin C, Aurum A. Aligning requirements with business objectives: A framework for requirements engineering decisions. *Proceedings of Requirements Engineering Decision Support Workshop*, 2005.
26. Gorschek T, Davis A. Requirements engineering: In search of the dependent variables. *Information and Software Technology* 2008; **50**(1–2):67–75.
27. Teece DJ. *Managing Intellectual Capital*: *Organizational*, *Strategic*, *and Policy Dimensions*. Oxford University Press: Oxford, 2000.
28. Sasaki T, Nagata A, Toyama R, Hirata T, Hasegawa K. Coevolution of patent strategy and product strategy. *Portland International Conference on Management of Engineering and Technology* (*PICMET'01*). IEEE: Los Alamitos CA, 2001; 481–484.
29. Ross SA, Westerfield R, Jordan BD. *Essentials of Corporate Finance*. McGraw-Hill: Boston MA, 2001.
30. CMMI-PDT, *Capability Maturity Model Integration* (*CMMI-Dev*), *Version 1.3. CMMI for Development—Version 1.3* (*CMU/SEI-2010-TR-033*). Carnegie Mellon: Pittsburgh PA, 2010.
31. http://www.tickit.org/ [1 October 2002].
32. IEEE, *Recommended Practice for Software Requirements Specifications* (*Standard 830-1984*). IEEE Press: New York NY, 1984.
33. Sawyer P, Kotonya G. *Software Requirements. Guide to the Software Engineering Body of Knowledge Swebok*, *Trial Version 1.0*. IEEE: Los Alamitos CA, 2001.
34. Kautz K, Hansen HW, Thaysen K. Applying and adjusting a software process improvement model in practice: The use of the ideal model in a small software enterprise. *Proceedings International Conference on Software Engineering*. IEEE: Los Alamitos CA, 2000; 626–633.
35. Beecham S, Hall T, Rainer A. Defining a requirements process improvement model. *Software Quality Journal* 2005; **13**(3):247–279.
36. Gorschek T, Wohlin C. Packaging software process improvement issues—A method and a case study. *Software*: *Practice & Experience* 2004; **34**(14):1311–1344.
37. Pettersson F, Ivarsson MTG. A practitioner's guide to light weight software process assessment and improvement planning. *Journal of Systems and Software* 2008; **81**(6):972–995.
38. Wohlin C, Aurum A. (editors). *Engineering and Managing Software Requirements*. Springer: New York NY, 2005.
39. Potts C. Invented requirements and imagined customers: Requirements engineering for off-the-shelf software. *Proceedings Second IEEE International Symposium on Requirements Engineering*. IEEE: Los Alamitos CA, 1995; 128–130.
40. Regnell B, Beremark P, Eklundh O. A market-driven requirements engineering process—Results from an industrial process improvement programme. *Requirements Engineering* 1998; **3**(2):121–129.
41. Balm GJ. Benchmarking and gap analysis: What is the next milestone? *Benchmarking for Quality Management and Technology* 1996; **3**(4):28–33.
42. Loucopoulos P, Karakostas V. *System Requirements Engineering*. McGraw-Hill: New York NY, 1995.
43. Soffer P, Goldin L, Kuflik T. A unified reapproach for software product evolution: Challenges and research agenda. *International Workshop* (*SREP'05*) *in conjunction with the IEEE 2005 International Requirements Engineering Conference*. IEEE: Los Alamitos CA, 2005.
44. Dahlstedt A, Karlsson L, Persson A, NattochDag J, Regnell B. Market-driven requirements engineering processes for software products—A report on current practices. *International Workshop on COTS and Product Software* (*RECOTS*) *in Conjunction with the 11th IEEE International Requirements Engineering Conference*. IEEE: Los Alamitos CA, 2003.

DOI: 10.1002/smr

45. Regnell B, Host M, Natt och Dag J, Hjelm T. An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. *Requirements Engineering* 2001; **6**(1):51–62.
46. Albrecht AJ, Gaffney JE. Software function, source lines of code, and development effort prediction. *IEEE Transactions on Software Engineering* 1983; **SE-9**(6):639–647.
47. Hihn J, Habib-agahi H. Cost estimation of software intensive projects: A survey of current practices. *Proceedings 13th International Conference on Software Engineering*. IEEE: Los Alamitos CA, 1991; 276–287.
48. Carlshamre P, Sandahl K, Lindvall M, Regnell B, Natt och Dag J. An industrial survey of requirements interdependencies in software product release planning. *Proceedings Fifth IEEE International Symposium on Requirements Engineering*. IEEE: Los Alamitos CA, 2001; 84–92.
49. Saaty TL, Vargas LG. *Models*, *Methods*, *Concepts and Applications of the Analytic Hierarchy Process*. Kluwer Academic Publishers: Boston MA, 2001.
50. Leffingwell D, Widrig D. *Managing Software Requirements*: *A Unified Approach*. Addison-Wesley: Reading MA, 2000.
51. Feather MS, Menzies T. Converging on the optimal attainment of requirements. *IEEE Joint International Conference on Requirements Engineering*. IEEE: Los Alamitos CA, 2002; 263–270.
52. Karlsson L, Berander P, Regnell B, Wohlin C. Requirements prioritisation: An experiment on exhaustive pair-wise comparisons versus planning game partitioning. *Proceedings Eighth International Conference on Empirical Assessment in Software Engineering* (*EASE 2004*) (*Proceedings ICSE 2004*). IEEE: Los Alamitos, 2004; 145–154.
53. Wiegers KE. *Software Requirements*. Microsoft Press: Redmond WA, 1999.
54. Kostoff RN, Schaller RR. Science and technology roadmaps. *IEEE Transactions on Engineering Management* 2001; **48**(2):132–143.
55. Kappel TA. Perspectives on roadmaps: How organizations talk about the future. *Journal of Product Innovation Management* 2001; **18**(1):39–50.
56. Gorchels L. *The Product Manager's Handbook*: *The Complete Product Management Resource*. NTC Business Books: Lincolnwood IL, 2000.
57. Gorschek T, Svahnberg M, Borg A, Börstler J, Eriksson M, Lonconsole A, Sandahl K. A controlled empirical evaluation of a requirements abstraction model. *Information and Software Technology* 2007; **49**:790–805.
58. Wohlin C, Aurum A. What is important when deciding to include a software requirement in a project or release? *International Symposium on Empirical Software Engineering*, 2005; 237–246.
59. Vähäniitty J, Lassenius C, Rautiainen K. An approach to product roadmapping in small software product businesses. *Seventh European Conference on Software Quality*, *Conference Notes*, 2002; 12–13.
60. Carlshamre P, Regnell B. Requirements lifecycle management and release planning in market-driven requirements engineering processes. *Proceedings 11th International Workshop on Database and Expert Systems Applications*. IEEE: Los Alamitos CA, 2000; 961–965.
61. Greer D, Ruhe G. Software release planning: An evolutionary and iterative approach. *Information and Software Technology* 2004; **46**(4):243–253.
62. Regnell B, Karlsson L, Host M. An analytical model for requirements selection quality evaluation in product software development. *Proceedings 11th International Conference on Requirements Engineering*. IEEE: Los Alamitos CA, 2003; 254–263.
63. Karlsson L, Regnell B, Karlsson J, Olsson S. Post-release analysis of requirements selection quality—An industrial case study. *Proceedings of the Ninth International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ'03*), 2003; 47–56.
64. Redling TJ. A methodology for developing new product line requirements through gap analysis. *Proceedings 22nd Digital Avionics Systems Conference*. IEEE: Indianapolis, 2003; 10.A.1–101-11.
65. Zairi M. *Best Practice*: *Process Innovation Management*. Butterworth-Heinemann: Boston MA, 1999.
66. Hill N, Brierley J, MacDougall R. *How to Measure Customer Satisfaction*. Gower, Aldershot: Brookfield, 1999.
67. Kotler P, Armstrong G, Saunders J, Wong V. *Principles of Marketing*. Prentice Hall: New York NY, 2002.
68. Davis A, Bersoff E, Comer E. A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering* 1998; **14**(10):1453–1461.
69. Deifel B. A process model for requirements engineering of ccots. *10th International Workshop on Database and Expert Systems Applications*. IEEE: Los Alamitos, 1999; 316–321.
70. Hass AMJ. *Configuration Management Principles and Practice*. Addison-Wesley: Reading MA, 2003.
71. Höst M, Regnell B, Nattoch Dag J, Nedstam J, Nyberg C. Exploring bottlenecks in market-driven requirements management processes with discrete event simulation. *Proceedings of PROSIM'00*, 2000.
72. Yeh AC. Requirements engineering support technique (Request): A market driven requirements management process. *Proceedings Second Symposium on Assessment of Quality Software Development Tools*. IEEE: Los Alamitos CA, 1992; 211–223.
73. Dahlstedt Å, Persson A. Requirements interdependencies—Molding the state of research into a research agenda. *Ninth International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ'03*), 2003; 71–80.
74. Davis AM. *Just Enough Requirements Management*: *Where Software Development Meets Marketing*. Dorset House Publications: New York NY, 2005.
75. Rautiainen K, Lassenius C, Pyhajarvi MA, Vanhanen JA. A tentative framework for managing software product development in small companies. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002; 3409–3417.

DOI: 10.1002/smr

76. http://www.sqi.gu.edu.au/spice/ [11 September 2003].
77. Berry M. Strategic planning in small high tech companies. *Long Range Planning* 1998; **31**(3):455–466.
78. Gorschek T. *Requirements Engineering Supporting Technical Product Management*. Blekinge Institute of Technology: Karlskrona, 2006.
79. Gorschek T, Garre P, Larsson L, Wohlin C. A model for technology transfer in practice. *IEEE Software* 2006; **23**(6):88–95.
80. Gorschek T, Svahnberg M, Tejle K. Introduction and application of a lightweight requirements engineering process evaluation method. *Proceedings of the Ninth International Workshop on Requirements Engineering*: *Foundation for Software Quality* (*REFSQ'03*), 2003; 101–112. Available at: http://gorschek.com/doc/publications.html [10 January 2011].
81. Gorschek T, Svahnberg M. *Requirements Experience in Practice*: *Studies of Six Companies. Engineering and Managing Software Requirements*. Springer: New York NY, 2005.
82. Calvo-Manzano Villalón JA, Cuevas Agustín G, San Feliu Gilabert T, De Amescua Seco A, García Sánchez L, Pérez Cota M. Experiences in the application of software process improvement in Smes. *Software Quality Journal* 2002; **10**(3):261–273.
83. Kuilboer JP, Ashrafi N. Software process and product improvement: An empirical assessment. *Information and Software Technology* 2000; **42**(1):27–34.
84. Reifer DJ. The CMMI: It's formidable. *Journal of Systems and Software* 2000; **50**(2):97–98.
85. Zahran S. *Software Process Improvement*: *Practical Guidelines for Business Success*. Addison-Wesley: Reading MA, 1998.
86. Robson C. *Real World Research*: *A Resource for Social Scientists and Practitioner-Researchers*. Blackwell Publishers: Oxford, 2002.
87. Gomes A, Pettersson A. Market-driven requirements engineering process model—MDREPM. *Master Thesis in Software Engineering*, Blekinge Institute of Technology, Ronneby, 2007. Available at: http://www.bth.se/fou/cuppsats.nsf/ [10 January 2011].
88. Johnson G, Scholes K. *Exploring Corporate Strategy*: *Text and Cases*. Prentice Hall: New York, 1993.
89. CMMI-PDT. Capability Maturity Model Integration (CMMI), Version 1.1. CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing Version 1.1 (CMMI-SE/SW/IPPD/SS, V1.1), Pittsburgh PA, 2002.
90. Gorschek T, Fricker S, Palm K, Kunsman S. Star search—A light-weight innovation process for software intensive product development. *IEEE Software* 2010; **27**(2):72–80.
91. Gorschek T, Garre P, Larsson S, Wohlin C. Industry evaluation of the requirements abstraction model. *Requirements Engineering Journal* 2007; **12**(3):163–190.

AUTHORS' BIOGRAPHIES

Tony Gorschek is a professor of software engineering at Blekinge Institute of Technology (BTH) with over ten years of industrial experience. He also manages his own industry consultancy company, works as a CTO, and serves on several boards in companies developing cutting edge technology and products. His research interests include requirements engineering, technology and product management, process assessment and improvement, quality assurance, and innovation. Contact him at tony.gorschek@bth.se or visit www.gorschek.com.

Andrigo Spall Gomes received his BSc in Computer Science from Federal University of Santa Maria, Brazil in 2000, and an MSc in Software Engineering—Specialization Management from Blekinge Institute of Technology, Sweden in 2007. He has broad experience in the software industry, having worked as a developer and architect in several projects in Brazil, Finland and Sweden. In the past 4 years his industry focus has been on requirements engineering and software processes, where his consultancy work has helped several companies improve their practices. His areas of interest are agile requirements engineering, lean software development and product management.

DOI: 10.1002/smr

Andreas Pettersson received his MSc in Computer Science and Software Engineering from Blekinge Institute of Technology, Sweden, in 2007.

After briefly working as a consultant during the last four years he has worked as a Product Manager for innovative market leading software product & service companies. His areas of interest include product management, strategic product management, product marketing management, technical product management, and requirements engineering. In 2010 when he started working as Product Manager for Milestone Systems, Denmark, his main responsibilities shifted from a technical focus to a strategic, business, and market focus.

Richard Torkar is an Associate Professor at the School of Computing at Blekinge Institute of Technology. His focus is on quantitative research methods in the field of software engineering. He received his PhD in software engineering from Blekinge Institute of Technology, Sweden, in 2006. He is a member of ACM and IEEE.

DOI: 10.1002/smr