Volume 2, Issue 12, December 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Research and Reflection Report on MDRE Problems

Kuda Nageswara Rao *
Dept. of CS&SE, College of Engineering(A) Andhra University, Visakhapatnam, India

Praneeth Chakka
Dept. of CS&SE, College of Engineering(A) Andhra University, Visakhapatnam, India

N. Thirupathi Rao
Dept. of CS&SE, College of Engineering(A) Andhra University, Visakhapatnam, India

Thota Srikrishna
Dept. of CS&SE, College of Engineering(A), Andhra University, Visakhapatnam,India

Abstract— Market driven software development differs from the traditional Bespoke in the context that there is no fixed customer for whom the company develops the software product. Instead there is a market. This brings various challenges to the market driven software development. The report focuses on the challenges in Market driven Requirements Engineering (MDRE). Overall the report deals with the following.

- ***Characterization of the challenges (refereed hereon as problems) and motivation.***
- ***The context of the problems.***
- ***Solutions proposed in the academia and industry to solve the problems.***
- ***Validation of the solutions proposed.***

Keywords: Release Planning (RP), Requirements Prioritization, ReqSmileTool, Requirements overload

I. INTRODUCTION

Distinguishing differences between MDRE and BESPOKE are the following: Stake holding, Requirement Elicitation Analysis, Management, etc. It can be noted that the challenges in MDRE arise from these distinguishing differences.
In this report, we deal with challenges, otherwise known as problems, in MDRE. Some of the existing problems in MDRE [3] are Stakeholding, Requirements Overload, Requirements Abstraction, Requirements Dependencies, Release Planning, Gaps in various Departments, Market Pull vs Technology Push, Requirements Management etc. It can be seen that these problems and their causes are intertwined and therefore it is important to focus not only one specific problem but consider the other reasons causing the problem. In this report we consider two problems of MDRE. These are – Release Planning, and Requirements selection from the perspective of product Management. In the subsequent sections we give the motivation behind considering these problems. The first is a direct problem while the latter is a combination of various problems like Requirements Abstraction, Continuous flow of requirements and Requirements Management. The report is divided into sections. The problems are discussed in separate sections for the purpose of exclusiveness and clarity. However it should be noted that the report is written after a literature survey and since we did not have prior industrial experience out reflections on the problems are limited in scope.

II. RELEASE PLANNING

Release Planning (RP) is an important activity in MDRE. In MDRE a product is developed in releases and released in the market. A software release consists of delivery of new or changed requirements (or features). These requirements form the basis of what the identified stakeholders get in a particular release. That means, it is concerned with "which customers get what features and quality at what point in time" [6]. Moreover, RP is "where requirements engineering for market-driven software product meets the market perspective" [8]. RP is selection of requirements/features in sequence of product releases, henceforth it is crucial for the overall success [8]. Accuracy of RP is an important factor for the determinant of the product's success [7]. Moreover, in MDRE due to the continuous inflow of large number of requirements, companies face the challenge of selecting a subset of requirements and plan their delivery in RP [9]. RP has got a strategic importance in requirements elicitation. “Developing and releasing small increments of requirements, in order for customers to give feedback early, is a good way of finding out exactly what customers want, while assigning a low development effort” [7].

2.1 *Release Planning as a Problem*

Effective and well done RP gives a strategic boost to the launching of the software product. As pointed by Ruhe et. al. [8], a good release plan contains the following.

- Reflect existing dependencies between features.
- Be feasible with available resources.
- Satisfy the identified customers' needs.
- Provide high return of investment by delivering the right set of features/requirements.

Aligning the release plan activities with the above mentioned bullets and aspects of RP is challenging. By focusing resources and effort on the development of a right set of features in a particular release, there is an increase chance of

good return of investment, as understood from [10]. It can be noted that 50% of the costs is required to produce 80% of the value [10]. Therefore it is essential to focus on the value generating features in the software release and make right decisions regarding the necessary trade-offs of the functionalities. A product must find customer's satisfaction in order to generate profits. Also, profitability will not be reached if the product is not released at the right time with a high level of quality when compared to the competitors in the market [10]. This is the economics side of the delivering software in releases. To gain an edge in terms of profits, it is necessary to understand how technical decisions and business strategy affect each other [10]. There are many factors that need to be taken under consideration before doing a systematic RP. Some of them are – needs of the stakeholders, requirements dependencies, prioritization, market analysis, competitors etc. Since RP centrally is about selecting features/requirements, it is also vital to prioritize them. This needs a good understanding and estimation of customer needs and the features the competitors might develop in their subsequent releases. One common theme in RP is balancing resources needed for a fixed set of features. Another major constraint in RP is balancing the factors – Time-to-market, needs and costs. For example, there is a high anticipation from the customers regarding the features/functionalities in the product, it takes more resources to release on the estimated release date. This also demands that the requirements are prioritized and estimated on two aspects. Firstly, Time and resources are spent on right amount of requirements and secondly, the right selection of requirements [7].

2.2 *Implications of the problem*

For the RP activities to be successful, requirements have to be elicited, analyzed before entering the RP phase [10]. The parameters that are needed for the RP activities are time horizon, objectives/goals, involvement of the stakeholders, short and long term planning. Time-to-market and scope [11]. It should be noted that these parameters are multi-dimensional and they are dependent on various factors. In [7] RP is decreed as a "wicked problem". One of the ways of understanding a "wicked problem" is the description given by the author. It follows: "*There is no definitive formulation of a wicked problem. To define the problem is the same as defining the solution*" [7]. Due to this non-definitive nature of RP it is considered wicked in nature [12]. Also, RP is partly supported by tool support [7]. Mostly in industry RP is done in an ad-hoc mane [8, 13]. Ad-hoc way of doing RP lacks structure and systematization and relies on expert's judgment[13]. It is a challenge to shift the paradigm of RP from an ad-hoc way of doing to a structured way of doing RP activities. In selection of the requirements/features in a release, there are various technical and non- technical factors involved. Technical factors include tool support, features to include (many of non-functional like security performance etc.), amount of re usability, requirements inter -dependencies etc. The non-technical factors include business and product strategies, Prioritization of stakeholders' requirements, time of release, cost estimation etc. [10]. Other implications include: It is very difficult to know alternate solutions to compare with the current release plan [11], aligning product and business strategies with the set of features to be released, estimating resources when the requirements suffer from volatility etc. These factors are challenges and in turn imply that they need to be attended.

In essence it can be understood that, a bad RP will have severe consequences. For example, the company may lose the market share by not delivering the product on right time, or not including the features that suit the customers' needs. And also wrong estimates of the resources needed may result in delays effecting time-to market.

2.3 *Context of the problem*

RP is naturally difficult because of the various aspects that need to be considered. The combination of Cost and Value is a complex one in performing RP [7]. Cost is the amount of resources that would be needed in developing the requirements. Cost estimation is a major problem in many companies [7]. Moreover, in MDRE, thee is a tendency that requirements change. So cost estimation always runs the risk of volatile requirements. Knowing and calculating value is more difficult. The perception of the value of the requirements varies among different stakeholders. The fore, there is a need to balance the values of stakeholders. This is the case among the departments in the companies [7]. Balancing the stakeholders' needs might affect the time-to-market [8]. Value of the requirements or a feature changes all the time and therefore, it is critical to respond to the changing 'value-trends' and requirements [10].

Requirements suffer from dependencies. Some of the types of dependencies are [7] Customer Value, Implementation Cost, And relations (requirements cannot exist individually but together), and Requires (one requirements/feature requires another). It means, requirements coming in for RP should not have the problem of dependencies. A clear overview of dependencies is difficult to obtain and it requires tool support for effectiveness [7].

RP can be little ambiguous if the company is unsure of the market it wants to capitalize, or increase returns. One problem with RP is it is limited to one release [7, 8]. It is suggested in [7] that it is necessary to plan for two releases so that the sets of features not already released could be delivered in the subsequent release. This is to ensure that the stakeholders' interest in maintained. It can be understood that all features cannot be delivered in one release. Lack of planning of what to release and what to reserve for the next release will be problematic. Not knowing the right time to release is a major problem of RP. This is because in order to gain high return of investment, it should be release at a certain time, or it looses the value [8]. RP can be a problem when the company does not have enough resources to perform within budget and capacity [8]. And finally lack of good tool support can be a problem to perform RP. This can be accounted to the large number of requirements.

2.4 *Solutions Proposed/Tested*

In the previous sub-sections we described about the RP problem. This subsequently deals with the solutions that have been proposed to perform RP. It is understood that for effective RP requirements have to be prioritized. So, we first point out the existing prioritization techniques and then proceed to the RP models that have been proposed.

2.4.1 ***Requirements Prioritization***

There are several techniques that have been proposed. Some techniques are based on determining the absolute importance of the candidate requirements [15], by e.g., assigning priorities like essential, conditional or optional [15]. Other techniques are relative and require a person to determine which requirement is of higher importance [15]. Relative techniques are more accurate and informative [16]. Examples of relative techniques are 100$-test [17] and Pairwise comparisons (PWC) [18]. It should be noted that these techniques are not discussed here in detail as the primary purpose is to bring out the various solutions that were proposed and used.

Developing a product that does not meet customers' satisfaction is a major threat to to developers. In article [19] it is concluded that "the set of requirements selected for implementation is a primary determinant of customer satisfaction". Also it was concluded to prioritize according to the implementation cost/customer value. In the article [18] a solution was proposed to what has been concluded in [19]. The idea was to let somebody from the marketing department prioritize according to the customer value and somebody from technical background prioritize the development cost. Both the groups use AHP (Analytic Hierarchy Process) [20] as the prioritization technique.

Planning Game (PG) [21] is another useful technique for the purpose of prioritization. The idea of planning game is borrowed from Extreme Programming.

A major advantage of PWC is its time consumption [24]. To decrease the time consumption factor, a requirements management tool called Focal Point has been developed.

2.4.2 ***Release Planning***

There have been over 20 models that have been proposed for RP. It should be noted that a well detailed discussion of the models is not provided here since the primary aim of the report is to give information regarding the problem itself and the solutions that have been proposed. Some of these models are extensions of existing models (for e.g. The EVOLVE family of RP models, mentioned below). Very few models are being used in the industry. There are different requirements selection factors that are used in these models. Almost all models have many factors in common but it should be noted that stakeholders' opinion is considered in a very few models. Various factors that are considered in these models are the following: Requirements Dependencies, Stakeholders' opinions/influences, Resources and Effort, Budget constraints, Time-to-market, Return of investment for each requirement, Quality of the requirements, Risk and value. The models do not classify these factors, i.e., categorization or taxonomy of these factors is not present.

EVOLVE [25] is the first RP model from the Evolve family. The factors this models takes into account are: Stakeholders Priorities, Effort required, Requirements dependencies, Maximum total benefit, Minimum penalties [25]. This is a mathematical model using the strength of Genetic Algorithms [25].

Other models from the Evolve family are Evolve+ [26], Evolve-star [27], Evolve Extended [11], Financial– Evolve-Star [28], Evolutionary Evolve+ [29], System-Evolve-Star [30].

Evolve-Star [27] is implemented in the form of a tool called Release Planner Tool [29].

Fuzzy logic has been used in two models. The purpose in these models is to model the uncertainties in the identifying the structural dependency constraints between the requirements [31]. In [32] the work has been extended to aid the decision making process. The study basically measures dependency constraint satisfaction in RP by using a set of fuzzy graphs. Both of these models are based on mathematical modeling. These models are not named.

[33] is a Value-based approach, taking stakeholders' views into consideration. This model provides a framework for RP and configuration method assisting decision making for small development teams in analyzing and prioritizing requirements and find the set of candidate requirements that are finally developed for the release. The candidate requirements are developed considering the constraints like time, quality and functionality.

[34] provides an agent based solution to the class of wicked problems, with RP being one of them. It is known as an explanation and interactive supported approach for RP.

RP has been a challenge in Extreme Programming (XP) also. A risk-driven approach has been developed in [35] for XP. This approach guides the developers to decide a suitable release plan and control the development process [35], and also analyze the technology risks involved.

The next study [36] deal with optimizing cost and value of the requirements in their analysis. The study is a mathematical modeling of RP that uses an optimization tool in determine the next release of the software product. The assumption used in the model is that the release's best set of requirements is the maximum revenue generating set of requirements [36]. The technique is enhanced using managerial steering mechanism (as termed by the authors of the study) using what-if analysis, team transfers, deadline extensions etc.

The research and problem solving has been extended to the non-functional requirements and quality aspect of the product. The following two studies [37, 38] deal with this. In [37], Quality Performance Model, otherwise known as QUPER is presented. In QUPER, quality aspect is taken along with cost and value dimensions in prioritizing approaches in functional requirements [37]. This model considers cost and value of non-functional requirements. In [38], QUPER has been applied to six domains and the study primarily talks about the lessons learned from authors' experiences. QUPER combines cost and benefit views in the form of a roadmap to give a better of product development.

Study [39] uses retrospective analysis, as the authors termed. The study presents a tool that supports retrospective analysis of RP decisions. The tool [40] was applied in a company and yielded positive results. The method described in [39] is called as PARSEQ (Post-release Analysis of Requirements Selection Quality) and also aims at finding process improvement proposals for RP activity [39].

To improve RP process the QIP [41] model is presented. QIP deals with Release Planning process improvement.

2.5 Validation of the Solutions

2.5.1 *Requirements Prioritization*

In the article [21] PG is compared against AHP and found that PG is easier to use. Other conclusions from this article include both the techniques are useful and have their own strengths but it recommended to use PG to sort the most important techniques and then use AHP prioritize among the sorted techniques. Another study [22] concludes that PG is used extensively in various extreme programming projects but the use of PG can be improved.

In the article [23], the study on AHP resulted in the claim by the authors that this is a good technique with high potential but is very difficult to adapt when there are multiple stakeholders.

Article [24] contains two experiments done on the effectiveness of three prioritization techniques PG, PWC and TPWC. Focal Point tool is used in the experiment. The conclusion is that TPWC is superior to PG and PWC when considered the time-consumption factor in prioritizing the requirements. PG was found to be easier than PWC but it could not be determined if PG or TPWC was easier although a majority included in the experiment found TPWC easier.

2.5.2 *Release* Planning

Almost all models/solutions have been validated either in industry or academia. These models are validated through case studies and interviews, or experiments on a limited scale. In academia, these models are validated using case projects, taking a sample to understand the results of the proposed solution/model.

EVOLVE [25], the first of its kind from the Evolve family has been validated through a case example and in Industry through a controlled experiment. The experiment involved different stakeholders and the number of requirements used in the experiment are 20. EVOLVE however assumes that requirements are delivered in increments. The study concludes that the method has great advantages like considering "inherent precedence and coupling constraints" [25], offering greater flexibility by slowing changes in requirements and their priorities. The future work included conducting an experiment in a complex industrial setting. The model is not being used currently.

Evolve+ [26], an extension of [25] has been validated by an industrial case-study and also by experiments in academia. The experiment included a set of 20 requirements. The number of stakeholders is 5. Two case studies were performed (although the information/design of the case studies is not given in the study). The results of the case studies indicate that Evolve+ is able to solve the problems effectively and efficiently with hundreds of requirements and multiple stakeholders.

Evolve-star [27] has been validated through two Industrial case studies. In the first case study the model is statically validated and in the second one, it is tested at Tema Labs, and implemented in an organization. The experiment was conducted using 30 requirements and 3 stakeholders. The model gives maximum stakeholder satisfaction "under the competing criteria of time, benefit and quality" [27]. This model is implemented as a tool, Release Planner and is currently being used in industry.

Evolve Extended [11] is validated using an experiment in academia, but not in industry/industrial setting. 63 students were involved in the experiment.

Financial–Evolve-Star [28] has been validated through an industrial case study. The model is an extension of [27] where the financial value of proposed features in considered. The results show that the model can be used "to determine which features generate the highest returns, with the shortest development time" [27]. But the model is not currently used in the industry. Evolutionary Evolve+ [29] has been validated through an industrial case study using 50 requirements and 6 stakeholders. The model is a promising one when there is uncertainty in the problem data. Some parts of the method are implemented in the tool Release Planner but not fully, for e.g. factors like risk and resources constraints are not implemented in the tool.

System- Evolve-Star [30] has been validated through an industrial case study involving 49 requirements and 6 stakeholders. But the flip side of the model it needs historical collection of requirement data.

Both studies [31, 32], models based on fuzzy logic have been validated. The first one was validated using an example. The results show that it can be applied to support decision making in RP. This model is not being used in industry. The latter study. [32], has been validated in academia using 10 requirements. But the study does not say about how better is this approach than other models and concludes by adding empirical studies and experiments in industry setting as a direction in future research.

The consensus -driven value-based approach given in the study [33] has been validated in academia. The model was developed from the experiences gained in small and medium sized enterprises. The model is not being used in industry. The model is simple, as said in the study itself and avoids in "analysis paralysis" [33] situations that arise due to inconsistencies in decision making. The study [34] is not validated.

The study [35], a Risk-driven method for XP, is validated through an industrial case study on a web-based application project. The results of the case study are: the method is suitable for XP practice, improvement in negotiation between customers and developers, and identification and assessment of risks. The model is not being used in industry but has good potential as the conclusion of the study suggests. The weakness of the model is risk data involved in the method can be used in the risk analysis phase but the model does not offer any reuse scheme, as said in the study.

The study [36], a mathematical modeling of RP, has been validated through an experiment in academia. The model is not being used in industry. The study concludes further research in industry setting using business cases and scheduling activities in time.

Studies [37, 38] are about the same model called QUPER. The QUPER model has been validated in industrial setting using interviews and is implemented in Sony Ericsson company. This is being used. Interviews were conducted using experts from six different mobile domains.

Study [39], PARSEQ model and the tool based on the model have been validated through two industrial case studies. The model suggests and gives valuable insights in RP. The PARSEQ method and the tool have been found to be useful and useable.

Finally, the study [41], an RP improvement process called Quality Improvement Paradigm (QIP) has been validated and is currently being used in industry to improve the release planning process improvement, as intended by the model.

2.6 Analysis

The models/solutions presented in the literature are validated in academia while some are validated in industry. It can be seen that some of the solutions are mathematical in nature. We believe, there should be tools developed using techniques that are mathematical. This is because using tools is relatively easier than manual work using mathematical formula. It can also be seen that some of the models are not implemented as tools. This we understand as a future step in solving RP issues. In the Release Planner Tool factors like risk and resource constraints are not implemented. This can also be seen a future step. The fuzzy logic based models are not being used in industry. We believe such methods can be put as a pilot study in industrial setting, or real data and know the drawbacks. Ans also, it is logical, we believe, to know the factors necessary to implement a certain model. For example, prerequisites and other environmental settings that might be essential in implementing a model in a real setting should be known. Most of the models given in the report are not specific to MDRE or BESPOKE while some are specifically made for MDRE. We believe more case studies will be useful in knowing the effectiveness of the RP models, or to if any of the models found in literature are being used. We also believe that researchers should develop models by coming in close contact with the industry, identifying their needs, and tailoring ideas and models to their needs. This is because each company has their own set of problems and some tend to be unique pertaining to their business needs. Many of the models presented in the report are not being used. Therefore, we believe it makes sense to know why the models AR not being adopted – are the models complex in nature? - or there is a lack of knowledge transfer of these models to industry. Also the models have been developed based on certain requirements factors like cost of implementation, risk, resources constraints, budget, benefit, return of investment etc. But most of the models do not address the factors like schedule and level of expertise needed to perform RP.

III. Requirements overload

This section of the report deals with the Requirements Overload problem in MDRE. Various aspects of requirements overload are discussed. Also, the solutions to the problems identified in industry are given. But the models are not discussed in detail as the primary purpose of the report is to explicitly mention solutions that have been prescribed.

3.1 *Requirements Overload as a problem*

In MDRE objectives of Requirements Engineering (RE) press should be in such a way that it ensures envisioning and fostering of new requirements and should also ensure competitiveness in marketplace [42]. This can partly be attributed to the emergence of markets for off-the-shelf or packaged software [43]. When developing products for market place instead of a specific customer, one goal of the RE process is to invent requirements, based on the customer behavior and select requirements based on their value generation in the marketplace [42]. Market driven products are vended in an open market and therefore there exists a large range of potential customers and diverse requirements are to be considered [44]. Therefore an MDRE process must handle such inflow of requirements and manage them. In the same study [44], the challenge of requirements overload is described. In MDRE, developers have to deal with constant flow of product requirements. Valuable ideas and requirements are gathered/elicited from various sources, for e.g. Customers, users, developers, feedback, reviews. There are various requirements sources and many requirements coming into the RE team. Some requirements appear like goals while some are detailed [45]. It can be noted that requirements gathering is a continuous process. The result is such a continuous flow of requirements leads to overload of requirements in the development organization [44]. The result is a large amount and continuous flow of requirements that threaten to overload the development organization [3]. Moreover, in MDRE products are delivered in releases/increments. It follows that these requirements have to be prioritized for the software team to decide which set of requirements best suit for the software release. The emphasis on the selection of requirements based on product strategies, business goals, and overall goal of the developing organization [46, 47, 48]. The work in subsequent sections deal with problems that arise from requirements overload emerging due to continuous flow of requirements in MDRE.

3.2 *Implications/Consequences*

It is evident from the above exemplification of the RE challenge that large number of requirements flow in causing requirements overload. Requirements elicitation becomes more difficult since consumers usually do not explicitly know what they want [48]. So it is not guaranteed that right requirements come in and therefore right requirements set is delivered as part of release as the number of requirements are heavy. Requirements overload in turn hampers good product development. If there is an efficient method to select appropriate set of requirements for the software release,

inflow of large number of requirements will not be a problem. But industrial survey shows there is a lack of good model to handle the problem. Here we can draw two sides of requirements overload. Firstly, handling such large number of requirements, i.e. Requirements Management, and secondly selecting the right set of requirements. It is suggested to select requirements based on product strategies, business goals etc. There is an industrial evidence that says the product managers lack time and resources to handle all the requirements [49]. The situation of requirements overload affects RP as it is all about delivering an appropriate set of requirements/features. At the same time, quality requirements have to be selected. As mentioned earlier, of these requirements some appear like goals, while some are explicit in content and nature [45]. Requirements coming from various sources have different meanings/perspectives [44], thus affecting the overall understanding of requirements abstractions. Some requirements emerge due to market trends, or the latest technology trends and some due to customer needs. It can be understood that requirements abstraction is another byproduct of the requirements overload challenge. Also, such large number of requirements might contain inter-dependencies and conflicts among them, thus causing the challenge if requirements dependencies. But the challenge of requirements dependencies is not discussed in this report. Requirements often have the trait of being volatile in nature. Requirements can change anytime during the entire life cycle. The situation gets work when the requirements change after a long and heavy process of capturing large number of requirements.

In an industrial survey Karlsson et al. [44] presented their work exemplifying the challenges/problems. For example, for a company that is constantly involved in developing software products pertaining to the customer and market trends, having a continuous and large number of requirements is one of the most problems. This is because a product development starts with requirements and for the final product product to be good, requirements have to be good. Most importantly, the requirements should be good enough to get a good understanding, and good enough to be implemented. Requirements overload becomes a problem when the company is able to manage such large number of requirements. Therefore, lack of good requirements management process will be a drawback as far as product development in MDRE is concerned. The managers lack time and resources to manage all the requirements [49]. As seen in the above sub-section, requirements have different meanings/perspectives. Failing to understand the requirements abstractions will be a problem. Requirements are prioritized as per certain criteria to implement. The prioritization/selection mechanism takes place following certain strategies, for e.g. Business goals, product goals [46, 47,48]. Understanding what requirements achieve what goals of the company is an important aspect. Failing to select requirements that result in not realizing business goals is unwanted. Therefore, in that case selection becomes a challenge. The following part of the report is written in these contexts.

There are other cases when requirements overload becomes a problem. For example, requirements overload affects RP. It is practically not feasible to select all the important requirements to include in a software release. The management has to select a set of important requirements and at the same time the management must estimate what requirements set will fetch maximum return. In selection of requirements requirements might suffer from dependencies. As seen above, Some of the types of dependencies are [7] Customer Value, Implementation Cost, And relations (requirements cannot exist individually but together), and Requires (one requirements/feature requires another). For example lets consider the management has two sets of requirements for a certain release. And when some of the requirements in the first set depend on the other set, it might lead to dependency problem. The management will have to reconsider their selection of requirements that are slated to be developed.

3.3 Solutions Proposed

There are many solutions that have been proposed for the requirements overload problem emerging due to the inflow of large number of problems. The solutions cover the various aspects of the problem itself. This section deals with the solutions and the context it caters to.

The first set of solutions concern with the early selection of requirements, otherwise known as Triage. Triage is a process to optimize resources usage by balancing the need for treatment with the likelihood of a successful outcome [50]. The risk of requirements overload can be alleviated by performing triage [46]. Early work on Triage was suggested by Simmons [50]. It was to measure various aspects of the requirements like implementation cost, volatility, schedule slips, domain experience etc. In order to perform Triage. Another suggestion came from Davis [49] and it was to consider estimated effort, inter-dependencies etc in order to perform Triage. Davis suggests a list of things to do to perform Triage. These are more of recommendations than a list that has to be followed. This list was based on 3 product development case studies.

Khurumm et al. [46] presented a model for performing early requirements triage. This is called as MERTS. This is a very good model, as its advantages from the practical point of view can be understood from the validation section of the report. MERTS builds on what Davis's [49] recommendations lack. utilizing business goals and product strategies in the selection of requirements. MERTS utilizes product strategies in selecting requirements enabling knowing their importance in the product to achieve market share, and also acts as a roadmap, i.e. various strategies that will be needed in the product development. Prerequisite of MERTS is that the requirements have to be comparable to the overall product strategies. If they are not comparable, MERTS is not efficient [46].

For the requirements comparison to be possible they have to been at the same level of abstraction. As seen above, requirements abstraction is a major concern emerging from requirements overload. Requirements Abstraction Model (RAM) [47] attempts to solve this problem. The objective of RAM is to give support to the product managers with a requirements engineering model that enables taking requirements of various abstraction levels, in a continuous effort.

This yields in working up requirements by breaking abstract, goal like requirements into detailed ones by making comparisons with the overall product strategies and vice-versa [47].

Laurent et al. [58] presented their work in automating the requirements triage process. This model uses a probabilistic traceability model and a clustering algorithm. Clustering algorithm is used to cluster incoming stakeholder requests. Other factors that are considered are architecturally significant requirements and business goals. This method is named as Pirogov. This is a partially automated technique.

The study [51] presents an experiment with a 'Linguistic support Tool' aimed at the consolidation of the requirements sets. The experiment attempts to answer the requirements overload problem where the new requirements are compared with the already existing requirements and compliment these requirements with the new ones. Linguistic engineering techniques give automatic assistance in finding similar requirements thus saving time, effort and resources. The tool used in the experiment is ReqSmileTool [52].

REPEAT [54] is a model proposed to address the area of requirements management when there is a coniferous inflow of requirements. REPEAT cover the activities elicitation, documentation and validation. REPEAT manages by controlling a product pipeline in which 3 releases are developed in parallel [54]. REPEAT gives a strong focus on requirements selection.

The study [53] presents that shows how discrete simulation can be used to explore overload situations in industrial software requirements process. This addresses the requirements management task in MDRE. Simulations of the process REPEAT [54] are used. The simulations give recommendations in order to avoid overload. These are the following: One, by improving the capacity of the management by adding more employees, or increasing productivity. Two, decreasing the workload on requirements management by decreasing the rate of issue of new requirements.

A method was presented in the study [55] to support different views on product development in order to avoid requirements overload. The method attempts to giving rich requirements and at the same time provide detailed design information designers would need.

Study [56] presents an analytical model of requirements process. The model considers both quality and capacity of the selection process thus taking care of overload situations. The model also addresses to estimate the highest product quality that can be achieved. The model is based on Queuing Theory [57]. The main focus is on requirements selection and quality as part of continuous effort in MDRE. Study [57] addresses the issue of balancing commercial requirements based on market pull and interval quality requirements in handling large number of requirements. Prospect Theory [57] has been applied to understand what affects product managers in making decisions regarding the selection of the requirements. The study indicates that greater risk is taking when selecting internal quality requirements while risk is avoided in the case of commercial requirements. The study [40] addresses the concept of establishing links between business decisions and RE. The paper recommends 5 key practices to enable this.

These are the following, as defined in the study: explicate planning levels and time horizons, separate planning of products' business goals from R&D resource allocation, plan open-endedly with a pre-defined rhythm, emphasize whole-product thinking and make solution planning visible [40]. These practices address identifying value creating customers, solutions that create value etc. The practices mentioned in the study [40] are not yet validated. The authors look forward to carry few case studies to know the defectiveness of the practices. But the practices have been framed based on close contact with industry.

3.4 Validation of the Solutions

MERTS [46] has been validated in the industry. The concepts of usability and scalability were investigated. The model was appreciated as promising and yielding as the method offers decision making support and structure in clarifying business goals. The validation primary took place based on interviews. It was also understood that it might take a lot of time and effort in detailing the strategies and aggregate perspectives of business goals, but once it is done selecting requirements becomes easier and faster.

The use of MERTS [46] has been validated in another instance. In the study [59], an experiment in academia was conducted to examine the effectiveness of using MERTS format and Natural Language format in doing requirements triage utilizing product strategies. The experiment was not conducted with product managers but with graduate students from Blekinge Institute of Technology, Sweden. The experiment was conducted to validate and carry a pilot study in the industry. The experiment results are positive and it was found that MERTS format is far superior to Natural Language formal in formulating product strategies and utilizing them in requirements triage.

RAM [47] has been validated. The experiment [60] is validated in academia in 3 Swedish universities involving nearly 180 subjects. The results point out towards the usefulness and usability of RAM. The subjects involved in the experiment performed well in using the RAM model in placing the requirements in the right abstraction level based on product perspective. The results point out towards the usefulness and usability of RAM. The RAM model has been validated in industry also [14]. The RAM model has been validated in two companies. The model is tailorable and it was done to suit the company needs. The results show that in both companies the use of the model yielded an increase of accuracy of requirements quality, practices in product management and RE. The model is also expected to save overload conditions of requirements by allowing early dismissal of requirements. The model is currently being used in industry.

The study [51], experiment with Linguistic Support Tool has been validated in a laboratory setting. Subjects found the tool support efficient and more correct in consolidation of the requirements, and may save time in industry purposes. There is an increase in speed and correctness. Industrial experiment is not yet done.

The method presented in [55], supporting different views on product development, is in its trial phase and validation is not yet done. The analytical model or requirements process [56] using Queuing Theory has been validated in an industrial survey through interviews involving experts and the results show the model is ready for industrial trials.

The study [57], balancing commercial requirements using Prospect Theory has been validated through an experiment in academia involving 71 students. Industrial validation has not been done yet but the authors of the study look forward to it as a part of future direction. The results show that decisions makers are more risk adverse when considered requirements in terms of revenue and less in terms of cost.

The practices mentioned in the study [40] are not yet validated. The authors look forward to carry few case studies to know the effectiveness of the practices. But the practices have been framed based on close contact with industry.

Study [53] is a simulation to explore bottleneck situations in RE. The model uses REPEAT [54]. REPEAT has been validated is being used in industry, at Telelogic, Sweden.

Automating requirements triage approach, Pirogov described in [58] has been validated through a case study using a case example. But the number of requirements used in the study are relatively smaller.

3.5 Analysis

We feel there has been substantial work done in the area of requirements overload and its various contexts. Interesting point is that some of the models have been developed with close contact with industry. For example, the MERTS and RAM models were developed by understanding the problems in industry through case studies. There is a study about automating the requirements triage. We believe it is a good step as part of future research as it marginally decreases the effort and consumption of resources. This is because the requirements are large in number, doing manually takes time. Some models have been implemented as tools thus improving the process of selecting requirements and increasing the rapidity of the whole process. It should noted that the work presented here addresses various contexts of the overload problem. But we believe more field work, case studies will yield insights into the latest challenges as we believe challenges or problems evolve with the market and technology trends.

IV. Conclusions

The practices mentioned in the study [40] are not yet validated. The authors look forward to carry few case studies to know the effectiveness of the practices. But the practices have been framed based on close contact with industry. But still MDRE is continuous and evolving, the solutions will need to evolve/tailor to the evolving circumstances. It is also seen from the report that the problems in MDRE are inter-related and the solutions should also provide a holistic view, i.e. should address the issues arising from the main problem. Performing what have been suggested in various solutions are little time and resource consuming but they are promising in terms of accuracy of the results as part of Requirements Engineering.

References

[1] Chair: P. A. Freeman, "Software Engineering Body of Knowledge (SWEBOK)," Software Engineering, International Conference on, p. 0693, 23rd International Conference on Software Engineering (ICSE'01), 2001.

[2] Sommerville, I. and Kotonya, G. 1998 *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, Inc.

[3] Aurum, A. and Wohlin, C. 2005 *Engineering and Managing Software Requirements*. Springer-Verlag New York, Inc.

[4] Carmel, E.; Becker, S., "A process model for packaged software development," *Engineering Management, IEEE Transactions on* , vol.42, no.1, pp.50-61, Feb 1995

[5] Saliu, O.; Ruhe, G., "Supporting Software Release Planning Decisions for Evolving Systems," *Software Engineering Workshop, 2005. 29th Annual IEEE/NASA* , vol., no., pp.14-26, 7-7 April 2005

[6] Gorschek T (2004) Requirements Engineering Supporting Technical Product Management. PhD Thesis No. 2006:01, ISBN 91-7295-081-1. Blekinge Institute of Technology, Ronneby.

[7] *Carlshamre, P*., "*Release Planning in Market-Driven Software Product Development*: *Provoking an Understanding*". Proceedings of the International Conference on software maintenece (ICSM'02), p.122, October 03-06,2002.

[8] Ruhe, G., Saliu, M.O., "The art and science of software release planning," Software, IEEE, vol.22, no.6, pp. 47-53, 2005.

[9] B. Regnell and J. Brinkkemper, "Market-Driven Requirements Engineering for Software Products," *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, eds., Springer, 2005, pp.287–308.

[10] Barney, S., Aurum, A., and Wohlin, C. 2008. A product management challenge: Creating software product value through requirements selection. *J. Syst. Archit.* 54, 6 (Jun. 2008), 576-593.

[11] Ruhe, G. and Momoh, J. 2005. Strategic Release Planning and Evaluation of Operational Feasibility. In *Proceedings of the Proceedings of the 38th Annual Hawaii international Conference on System Sciences - Volume 09* (January 03 - 06, 2005). HICSS. IEEE Computer Society, Washington, DC, 313.2.

[12] Ngo-The, A. and Ruhe, G. 2007. A systematic approach for solving the wicked problem of software release planning. *Soft Comput.* 12, 1 (Aug. 2007), 95-108.

[13] Du G., McElroy, J., Ruhe, G., "*Ad Hoc Versus. Systematic Planning of Software Releases - A Three-Staged. Experiment*", Proceedings PROFES 2006

[14] Gorschek, T., Garre, P., Larsson, S. B., and Wohlin, C. 2007. Industry evaluation of the Requirements Abstraction Model.*Requir. Eng.* 12, 3 (Jul. 2007), 163-190.
[15] L. Karlsson. Requirements Prioritization and Retrospective Analysis for Release Planning Process Improvement. PhD thesis, Lund University, Department of Communication Systems, Lund Institute of Technology, 2006.
[16] Karlsson, J. 1996. Software Requirements Prioritizing. In *Proceedings of the 2nd international Conference on Requirements Engineering (ICRE '96)* (April 15 - 18, 1996). ICRE. IEEE Computer Society, Washington, DC, 110.
[17] Leffingwell, D. and Widrig, D. 2000 *Managing Software Requirements: a Unified Approach*. Addison-Wesley Longman Publishing Co., Inc.
[18] Karlsson, J. and Ryan, K. (1997) "A Cost-Value Approach for Prioritising Requirements", IEEE Software, Vol. 14, pp. 67-74.
[19] Karlsson J., Ryan K., "Supporting the Selection of Software Requirements", Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD '96), 1996
[20] Saaty, T.L. (1980) The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill, New York.
[21] Karlsson L., Berander P., Regnell B., Wohlin C., "Requirements Prioritisation: An Exhaustive Pair-Wise Comparison versus PG Partitioning ", EASE '04 Empirical Assesment in Software Engineering, 2004.
[22] Rumpe B., Schröder A., "Quantitative Survey on Extreme Programming Projects", 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering, 2002
[23] Regnell B., Höst M., Natt och Dag J., Beremark P., Hjelm T., "An Industrial Case Study on Distributed Prioritisation in Market -Driven Requirements Engineering for Packaged Software", Requirements Engineering, Feb 2001, vol. 6, number 1, pp 51-62
[24] Karlsson, L., Thelin, T., Regnell, B., Berander, P., and Wohlin, C. 2007. Pair-wise comparisons versus planning game partitioning--experiments on requirements prioritisation techniques. *Empirical Softw. Engg.* 12, 1 (Feb. 2007), 3-33.
[25] Greer, D., and Ruhe, G., "Software Release Planning: An Evolutionary and Iterative Approach", Information and Software Technology, no.1, 2004.
[26] Ruhe, G., and Des, G., "Quantitative Studies in Software Release Planning under Risk and Resource Constraints", Proceedings of the 2003 International Symposium on Empirical Software Engineering, pp. 262-270, 2003.
[27] Ruhe, G., and Ngo, A., "Hybrid Intelligence in Software Release Planning", Int. J. Hybrid Intell. Syst. vol.1, no.1-2, pp.99-110, 2004.
[28] Maurice, S., Ruhe, G., Saliu, O., and Ngo-The, A., "Decision Support for ValueBased Software Release Planning", Journal of Value-Based Software Engineering, pp. 247-261, 2006.
[29] Ngo-The, A., and Ruhe, G., "A Systematic Approach for Solving the Wicked Problem of Software Release Planning", Soft Comput, vol. 12, no.1, pp. 95-108, 2007.
[30] Saliu, O., Ruhe, G., "Supporting Software Release Planning Decisions for Evolving Systems", 29th Annual IEEE/NASA Software Engineering Workshop, pp. 14-26, 2005.
[31] Ngo-The, A., and Saliu, O., "Fuzzy Structural Dependency Constraints in Software Release Planning", The 14th IEEE International Conference on Fuzzy Systems (FUZZ '05), pp.442-447, 2005.
[32] Ngo-The, A., and Saliu, O., "Measuring Dependency Constraint Satisfaction in Software Release Planning using Dissimilarity of Fuzzy Graphs", Fourth IEEE Conference on Cognitive Informatics, (ICCI 2005), pp. 301-307, 2005.
[33] Ziemer S., Falcone Sampio, R. P., and St alhane, T, " A decision modelling approach for analyzing requirementes configuration trade-off's in time-constraned web application development", Eighteenteth International Conference on Software Engineering and Knowledge Engineering (SEKE'2006), pp.144-149,2006.
[34] Du, G., Richter, M. M., and Ruhe, G., "An Explanation Oriented Dialogue Approach and Its Application to Wicked Planning Problems," Journal of Computing and Informatics, vol. 25, pp. 223-49, 2006.
[35] Mingshu, L., Meng, H., Fengdi, S., and Juan, L., "A Risk-Driven Method for Extreme Programming Release Planning", Proceedings of the 28th International Conference on Software Engineering, pp., 2006.
[36] Van den Akker, M., Brinkkemper, S., Diepen, G. and Versendaal, J., "Software Product Release Planning through Optimization and What-if Analysis," Information and Software Technology, vol.50, issue.12, pp.101-11, 2008.
[37] Rengell, B., Host, m, and Svesson, R.B., "A Quality Performance Model for Cost-Benfict Analysis of Non-functional Requirements Applied to the Mobile Handset Domain," 13th International Working conference on Requirements Engineering (REFSQ07), pp. 277-91, 2007.
[38] Regnell, B., Svensson, R.B., and Olsson, T., "Supporting Road mapping of Quality Requirements", IEEE Software, vol.25, no.2, pp.42-47, 2008.
[39] Karlsson, L., and Regnell, B., "Introducing Tool Support for Retrospective Analysis of Release Planning Decisions", Conference on Product-Focused Software Process Improvement, pp. 19-33, 2006.
[40] Laura Lehtola, Marjo Kauppinen, Jarno Vähäniitty, Marko Komssi, Linking business and requirements engineering: is solution planning a missing activity in software product companies?, Requirements Engineering, Volume 14, SpringerLink, 2009.
[41] Amandeep, A., Ruhe, G., and Stanford, M., "Intelligent Support for Software Release Planning", 5th International Conference on Product Focused Software Process Improvement, pp. 248-262, 2004.
[42] A Market-driven Requirements Engineering Process Results from an Industrial Process Improvement Programme. Regnell, Björn, Beremark, Patrik and Eklundh, Ola. s.l. : Springer, 1998. pp. 121-129.

[43] K. El Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development," in Proceedings of the Second IEEE InternationalSymposium on Requirements Engineering, 1995, pp. 68-80.
[44] Karlsson, Lena, et al. Requirements engineering challenges in market -driven software development - An interview study with practitioners. Information and Software Technology. 2007, Vol. 49, 6, pp. 588-604.
[45] Gorschek, T.; Wohlin, C.; Carre, P.; Larsson, S., "A Model for Technology Transfer in Practice," Software, IEEE , vol.23, no.6, pp.88-95, Nov.-Dec. 2006.
[46] Khurum, M.; Aslam, K.; Gorschek, T., "A Method for Early Requirements Triage and Selection Utilizing Product Strategies," *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific* , vol., no., pp.97-104, 4-7 Dec. 2007
[47] T. Gorschek and C. Wohlin,"Requirements Abstraction Model," Requirements Engineering, vol. 11, pp. 79-101, 2006.
[48] Tuunanen, T.; Rossi, M., "Market driven requirements elicitation via critical success chains," *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International* , vol., no., pp. 367-368, 8-12 Sept. 2003
[49] Davis A (2003) The Art of Requirements Triage. IEEE Computer, Volume 36, Issue 3, PP 42-49.
[50] E. Simmons, "Requirements Triage: What Can We Learn from a "Medical" Approach?," IEEE Software, vol. 21, pp. 86-88, 2004.
[51] Natt Och Dag, J., Thelin, T., and Regnell, B. 2006. An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development. *Empirical Softw. Engg.* 11, 2 (Jun. 2006), 303-329.
[52] Natt och Dag J, Gervasi V, Brinkkemper S, Regnell B (2005), "A linguistic-engineering approach to large-scale requirements management," IEEE Software 22(1):32–39.
[53] *Exploring bottlenecks in market-driven requirements management processes with discrete event simulation*. Journal of Systems and Software 59(3): 323–332.
[54] Regnell, P. Beremark and O. Eklundh , A market-driven requirements engineering process – results from an industrial process improvement programme. Journal of Requirements Engineering 3 2 (1998), pp. 121–129
[55] Tuunanen, T.; Rossi, M., "Market driven requirements elicitation via critical success chains," *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International* , vol., no., pp. 367-368, 8-12 Sept. 2003
[56] Tuunanen, T.; Rossi, M., "Market driven requirements elicitation via critical success chains," *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International* , vol., no., pp. 367-368, 8-12 Sept. 2003
[57] Wu, G. and Markle, A. B. 2008. An Empirical Test of Gain-Loss Separability in Prospect Theory. *Manage. Sci.* 54, 7 (Jul. 2008), 1322-1335.
[58] Laurent, P.; Cleland-Huang, J.; Chuan Duan, "Towards Automated Requirements Triage," *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International* , vol., no., pp.131-140, 15-19 Oct. 2007.
[59] Khurum, M., Gorschek, T., Angelis, L., and Feldt, R. 2009. A Controlled Experiment of a Method for Early Requirements Triage Utilizing Product Strategies. In *Proceedings of the 15th international Working Conference on Requirements Engineering: Foundation For Software Quality* (Amsterdam, The Netherlands, June 08 - 09, 2009). M. Glinz and P. Heymans, Eds. Lecture Notes In Computer Science, vol. 5512. Springer-Verlag, Berlin, Heidelberg, 22-36.
[60] Gorschek, T., Svahnberg, M., Borg, A., Loconsole, A., Börstler, J., Sandahl, K., and Eriksson, M. 2007. A controlled empirical evaluation of a requirements abstraction model. *Inf. Softw. Technol.* 49, 7 (Jul. 2007), 790-805.

