A. Aurum and C. Wohlin, "The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process", Information and Software Technology, Vol. 45, No. 14, pp 945-954, 2003.

The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process

Aybüke Aurum[1], Claes Wohlin[2]

[1] School of Information Systems, Technology and Management, University of New South Wales,
Sydney, 2052 NSW Australia
[2] Department of Software Engineering and Computer Science, Blekinge Institute of Technology,
Box 520, SE-372 25 Ronneby, Sweden

Abstract

The requirements engineering (RE) process is a decision-rich complex problem solving activity. This paper examines the elements of organization-oriented macro decisions as well as process-oriented micro decisions in the RE process and illustrates how to integrate classical decision-making models with RE process models. This integration helps in formulating a common vocabulary and model to improve the manageability of the RE process, and contributes towards the learning process by validating and verifying the consistency of decision-making in RE activities.

Keywords: Requirement engineering activities, Requirements decisions, Classical decision-making model, Project management,

1. Introduction

Requirements engineering (RE) has evolved into a key activity in the field of software engineering. RE is both an organizational activity and a project activity. It is an organizational activity in terms of deciding what sort of requirements will go into products, and the final requirements that will be released. It is also a project activity when it comes to actually implementing them. This dualism of RE involves a range of decisions that have to be made to ensure effective organizational as well as project decisions. As decision-making lies at the heart of RE, it therefore makes sense to try to use decision-making models to understand the nature of RE activities. This paper looks at requirements activities from the classical decision-making point of view, where the decision process is assumed to be unstructured and consequential, and many of the decisions are intertwined and interdependent. The objective of this paper is to map two of the accepted decision-making models to the domain of RE for software systems, and to show how these models can help us to understand some of the intricacies of RE.

Motivation

Researchers agree that the last area of software development to be understood is management. Software technology is changing too fast for software managers to manage effectively on an ad-hoc basis. In other words, management methodologies have not improved as much as the software technologies. Poor strategic management and related human factors have been cited as a major cause for failures [18, 29].

The quality of the software product is largely controlled by the quality of the development process used to create it. Managing the RE phases is crucial to the successful development of software products. There are several phases in an RE model, and each phase contains several activities. The effective management of the RE process mandates procedures and tools to support the phases of RE process model. Andriole [1] points out that while most of the RE process models and specification methods have been created by people who like precision and diagnostics and who also care about software design and development, in reality, requirements management is a political process. Andriole argues that most of the software projects come from preferences of managers, from intuition about a value, or from their pet projects they all carry around with themselves. Strigini [35] also emphasizes that in the software industry, in many cases, important decisions are subjective decisions. In some cases, subjectivity is unavoidable but it is important that developers recognize the dangers and take precautions against the consequences.

The process of RE is essentially a complex communication and negotiation process involving customers, designers, project managers and maintainers. The people or stakeholders involved in the process are responsible for deciding what to do, when to do it, what information is needed, and what tools need to be used [10, 34]. In many situations conflict is inherent in requirements, thus they need to be negotiated between stakeholders. Some tools, such as WinWin Groupware, have been developed to support stakeholders throughout the negotiation process [7]. The discussion between stakeholders is generally informal and unstructured and involves many decisions and review points with iterations and design implementation activities. These decisions are generally continuous and in numerous cases uncertainties in the final cost, schedule, performance, and functionality are inevitable [12].

Since the RE process is a decision-rich problem solving activity, it is valuable to look at the classical theories of decision-making to better understand RE decisions. The decision-making process in management basically involves the following stages: identification of a problem, search for solutions, evaluation and selection of a solution and commitment. The process is iterative in nature and very similar to the software development cycle. Several researchers have looked to the classical decision-making theory when explaining the design process in engineering, including software engineering. Vliegen and Van Mal [37] report that in many cases, designers focus on the creation of proper solutions rather than spending time on redoing design because of poor decision-making. These researchers propose using the decision-making cycle in the design process to obtain methodological design and progress controls. In a similar way, Ashrafi [3] expands Simon's [33] decision-making model to show how this model can provide a framework to embed quality throughout the software development cycle and facilitate the total quality management of software production.

A number of researchers have taken a methodological approach when studying decision-making process. For instance Wild *et al.*, [39] present a methodology where the software development process is modeled as a set of decisions in which each decision relates to a problem solving activity. In the F3 project [13] a generic decision trace mechanism was developed to record significant decisions, which concerns every part of the RE process. Rolland *et al.*, [31] introduce a decision-oriented process meta-model that aims to capture not only how activities are performed during the RE process but also why and when these activities are performed (the decisions and the decision context). In the CREWS project [9], Tawbi and Souveyet [36] develop a tool called L'ECRITOUIRE, by combining goal-oriented modeling and scenario authoring. They present an intention-oriented process map, which includes a non-deterministic ordering of intentions and strategies that flow from one intention to another. Later, Rolland [30] proposes a requirements map that focuses on systematic selection and assembly of COTS components that meet the requirements of an organization, where the selection process is restricted not only for individual components but also as a whole.

Evans *et al*., [12] emphasize that it is important to recognize requirements as design decisions in order to achieve a fully integrated software system. Aurum and Martin [4, 5] point out the resemblance between the activities involved in organizational decision-making and those in the RE process by referring to classical decision-making theory and they present an approach that facilitates problem solving activities for requirements engineers. Regnell *et al.* [28] also discuss the descriptive and prescriptive research issues for understanding and supporting the RE decision-making process. In general, researchers agree that the RE process is a semi-structured or unstructured complex decision-making process [5, 23].

Extensive research is being undertaken to improve the RE process. Growing interest in developing effective approaches to RE has motivated a more in-depth analysis of RE activities. Insights into the RE process can be achieved through a closer analysis of the decision-making processes involved in RE activities. We believe that this approach is valid as decision-making and RE processes involve decisions that are both continuous and iterative in nature. Thus, we take this opportunity to analyze the RE activities in the light of two well-accepted models of decision-making.

To interpret RE decisions it is necessary to appreciate the various aspects of the decision-making process. This paper provides an alternative view to RE activities in which major organizational and project influences are considered. In order to better understand the RE process, this paper (a) overviews decision-making models, (b) examines the decision-making process in RE activities, and (c) combines the decision-making models with RE activities, which aims to bring about a more structured approach to the RE process.

The outline of this article is as follows. Section 2 provides an overview of decision-making models from management science with an emphasis on two decision-making models: Anthony's organizational decision-making model [2] and Mintzberg's process model [26]. Section 3 reviews RE process models, with a focus on Macaulay's RE process model. Section 4 examines RE decisions by mapping Anthony's model and Mintzberg's model on RE activities. Based on this mapping, Section 5 provides a discussion on elements of micro and macro decisions in RE activities. Finally, Section 6 concludes the paper.

2. An Overview of Decision-making Models

The objective of this section is to give a brief overview of decision-making models to provide background analysis of the complex decision-making processes inherent in RE activities. Thus, it is important to introduce some concepts from managerial decision-making. These concepts are useful for providing an organized framework for structuring and analyzing decision-making, which have been examined broadly in psychology, organizational behavior, and decision sciences. Researchers have extensively studied the individual elements of complex decisions where the decision process is unstructured, consequential and preference based. They have defined the ideal complex decision-making process and incorporated these insights into decision support system models. In other words, the following classical views are so powerful that they lay the foundation for practical implementations and they have been widely used as a basis for the design of decision support systems in the IT industry [16, 40].

Problem solving is in essence a decision-making activity. It involves the *intelligence*, *design* and *choice* phases [33]. There are various strategies that a decision maker can employ to organize their efforts when making decisions such as optimizing, satisficing, or using multi-criteria decision-making methods such as, for example, the analytical hierarchy process [32]. A particular decision maker in a particular decision-making process may make use of all of these strategies, a combination, or an entirely different strategy [16].

Organizations face different types of decisions that are related to, for example, technology, products and customers. Some decisions are repeated several times during a day or week while others may occur less frequently. According to Simon [33], based on the way in which managers deal with the problems that confront them, decisions that are repetitive and where the process is clearly identified, are called *structured decisions*. Decisions that are novel and where the process is ambiguous are referred to as *unstructured decisions*. These terms refer to extremes on a continuum. That is to say that there may be decisions that are completely structured or unstructured. Other decisions may change from novel situations to repetitive situations, depending on the degree of ambiguity in the solution to the problem.

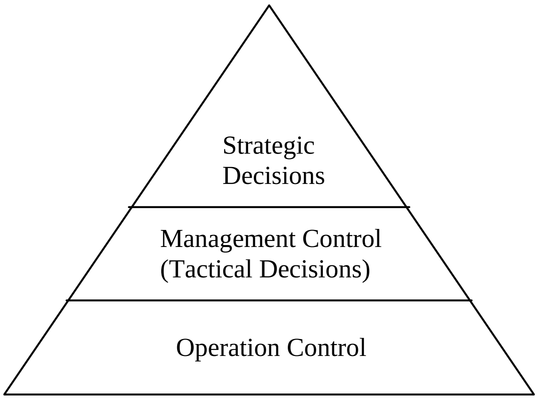

Figure 1: Managerial Decision-Making Framework [2]

Various parts of decision-making activities have been studied in detail by researchers in the field of management science. Anthony [2] notes three types of decision-making activities (Figure-1) in organizations based on the purpose of the management activity: *strategic planning, management control and operational control*. The boundaries between these categories are not distinct. However, in terms of information requirements, they differ from one another. According to this model, *strategic planning* deals with decisions that are related to organizational goals and objectives. The information concerning such decisions is usually incomplete and the decision-making process may extend over a considerable period of time. On the other hand, *management control* deals with decisions related to the identification and use of resources. In the case of *operational control*, decisions deal with assuring effectiveness when performing operations within the organization. In this study, we refer to these decisions as '*macro decisions*'. It is important to note that macro decisions can be structured or unstructured decisions. For instance, an unstructured strategic decision may have some familiar components in its structure. On the other hand, there may be an instance where a seemingly routine operational control decision may end up as an unstructured decision. In each case, a decision is subject to the intelligence, design and choice phases. In short, Anthony’s model is concerned with organization-oriented macro decisions.

Another group of studies focus on how managers actually go about their decision-making within organizations. These models focus on more detailed activities in management. In this study, we refer to these decisions as ‘*micro decisions*’. Our literature review revealed there have been several models put forward to explain micro decisions. One well-known descriptive model was developed by Mintzberg, based on a field study of 25 organizations [26]. This model (Figure 2) attempts to explain organizational problem solving activities and addresses process-oriented micro decisions. The researchers found patterns in the decision process and described a model for complex and unstructured decisions. First, they categorized the decisions according to the stimuli that evoked them along a continuum: opportunities, crises and problem decisions. Then they proposed a model which is more complex than normative models, and which has a structure. The main phases in this model are (1) problem identification, (2) development, and (3) selection. According to this model, the *problem identification phase* concerns decisions and has two sub phases: ‘recognition’ and ‘diagnosis routine’. The *development phase* contains two sub phases: a ‘search routine’ for locating ready-made solutions and ‘design routine’ to modify those solutions that have been found or to develop custom-made solutions. The *selection phase* contains 3 sub phases: ‘screening’, ‘evaluation and choice’ and ‘authorization’. The unstructured and open-ended decisions can be influenced by environmental attributes *e.g.* complexity and uncertainty. The internal power structure of the organization, the degree of rationality of decision maker(s), the extent of individual or group involvement, and previous policies or strategies may all have a significant impact on decisions.

Mintzberg *et al.*’s Model	Problem Identification		Development Phase		Selection Phase		
	Recognition Rt.	Diagnosis Rt.	Search Rt.	Design Rt.	Screen	Evaluation and Choice	Authorization

Figure 2: Decision-Making Framework [26]

In this research, we take the view that the software development process involves decision-rich activities and it is subject to organization-oriented macro decisions as well as to process-oriented micro decisions. However, before studying the decision-making process in RE activities, it is worth examining RE process models. It is important to note that the issues that affect individual and group decisions, such as negotiation, consensus building, prioritization, judgment of choice, rationality or bounded rationality are outside the scope of this paper.

3. Requirements Engineering Process Models

In order to study the decision-making process in requirements activities we needed to pick one of the existing RE process models. In this section, we briefly review existing RE process models from literature and illustrate the rationale behind the selection of the RE process model that was studied.

Background

There are several normative models that explain the RE process by describing how the RE process *should* work, rather than how it *does* work [10, 22]. The strength of these models lies in delivering the 'best' process results for activities that make up the process, and the order in which issues are attended to. These models take into account organizational goals and objectives. Sommerville and Sawyer [34] point out that, normative models do not transfer well from one organization to another. They emphasize that in order to define a good RE process, it is important to involve stakeholders who are actually involved in the RE process itself.

A review of existing RE process models in the literature showed that although these models encompass common RE activities, they differ in nature, ranging from linear and incremental to cyclical and iterative in structure [25]. While some studies tend to portray the RE process as a linear progression of activities, or a repetition of the same series of activities [19, 22], non-linear models have also been put forward [20]

RE field studies have gathered conflicting results as to the status of RE process standards in organizations. Kotonya and Sommerville [19] put forward that not many organizations have a standard RE process definition. Consistent with this, Hofmann and Lehner [15] examined the RE processes of 15 RE teams in industry and found that most participants saw RE as an ad hoc process, and only some of the projects used an explicitly defined RE process or tailored a company standard RE process to better suit the project. Furthermore, studies of RE in web development projects have shown that RE is occurring in an ad hoc manner [21]. In contrast to these findings, El Emam and Madhavji [11] concluded that organizations tend to use standard RE processes, as they are thought of as being best practice.

Results from studies of RE processes in practice have indicated that the systematic and incremental RE models presented in the literature do not reflect the RE processes in current practice. For example, Nguyen and Swatman [27] found that the RE process in their case study did not occur in a systematic, smooth and incremental way, but rather was opportunistic, involving the sporadic simplification and restructuring of the requirements model when it reached points of high complexity. Furthermore, Houdek and Pohl [17] performed a field case study, but failed to produce a comprehensive RE process model, as RE activities were heavily intertwined, and participants took a more holistic view of the RE process, rather than being aware of individual specific components.

There are only a few organizations that have explicitly defined the RE processes, and even fewer have modeled RE activities at both the organizational and the project (or process) levels. In many cases, the process is an unstructured problem solving activity. In unstructured decisions, the alternatives are generally vague, difficult to compare and contrast, prioritize, or cannot be easily evaluated with respect to organizational goals and objectives.

As illustrated above, although existing RE models do not reflect well how RE processes are implemented in practice, they have the potential to provide insights into RE activities in a case-by-case basis. Moreover, the above RE models involve both organizational and process oriented decisions. Although a number of RE process models have been put forward, in this study we will focus upon Macaulay's RE descriptive process model as an example of an RE process model. We believe that the selection of the RE process model is not of vital importance as the objective of this research is to understand the fundamental nature of the decision-making process in requirements activities.

Macaulay's RE Model

Macaulay [22] argues that the model of the RE process depends to a large extent on the customer-supplier relationship and offers seven different process models for different situations. Macaulay also describes a generic model as follows: (1) concept, (2) problem analysis, (3) feasibility and choice of options, (4) analysis and modeling, and (5) requirements documentation. Macaulay points out that each phase is subject to validation, which serves to focus attention on unresolved issues, which then need to be reviewed with stakeholders.

According to this model as illustrated in Figure 3, the product *concept*, which might be an improvement or enhancement of the product, triggers the RE process. In many cases, it triggers the organizational level RE process, and as the concept moves into a specific project it triggers the RE process on the project level (or development process level). During the *problem analysis phase*, an understanding of the nature of the problem is developed. Another key aspect is generating an appropriate representation of the problem, which can help requirements engineers to identify the set of alternative solutions. The third phase, *feasibility and choice of options*, is concerned with evaluating the costs and benefits of alternative solutions and negotiations. The fourth phase, *detailed analysis and modeling*, is concerned with a more detailed analysis of the requirements. Once this process is finished, the *requirements specification document* can be completed. As pointed out earlier, a validation process takes place at the end of each phase in this model.

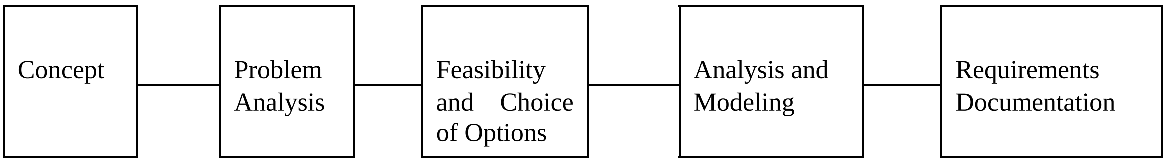

Figure-3: Requirements Engineering Model [22]

Being an effective software developer does not imply that the person necessarily understands their decision-making process. In order to improve the overall level of effectiveness of the RE process it is important to understand the nature of the decisions made. One approach is to record past cases of activities in the RE processes, and then to analyze the decisions in each activity. Another approach is to decompose each process into sub-processes and study the decisions within these sub-processes. In each case, it is important to consider the stakeholders' point of view when comparing, generalizing and classifying decision models into different categories. The following section uses Macaulay's RE process model to take the latter case and examine decisions involved in the RE process in light of a management context. Recording each activity and decision provides structure, for what is often an unstructured process in the early stages of RE activities. For instance, if a problem is resolved and the decision is recorded, then the project manager (PM) will have a better idea of project progress. If the problem is not resolved, then the PM will focus on allocating human and non-human resources to solve the problem. This way, the PM will have increased control over the scheduling of activities (Wild et al., 1994). In other words, this approach provides a more efficient structure for managing RE processes. This structure has the potential to be used for requirements control throughout the software development life cycle.

4. Combination of Models

In this section we examine RE decisions. In order to better understand RE decisions, elements of both micro and macro decisions (*i.e.,* Anthony's organizational decision-making model and Mintzberg's managerial decision-making model) are projected onto Macaulay's RE process model, as illustrated in Figure 4. Section 4.1 discusses the mapping of *macro* decisions onto the RE process and Section 4.2 presents the mapping of *micro* decisions onto the RE process. Macro and micro decisions are of course not independent and in many cases they will be interleaved. However, here we present them separately to illustrate the emphasis of the decision-making model's mapping to the activities in a RE process.

<table>
<tr><td><b>Model</b></td><td colspan="7"><b>Classification</b></td></tr>
<tr><td><b>Macaulay's Model</b></td><td colspan="2">Concept
(Problem Recognition)</td><td colspan="2">Problem Analysis</td><td>Feasibility & Choice of Options</td><td>Detailed Analysis &amp; Model</td><td>Requirements Document</td></tr>
<tr><td><b>Anthony's Model</b></td><td colspan="2">Strategic Planning</td><td colspan="2">Management Control (Tactical)</td><td colspan="3">Operational Control</td></tr>
<tr><td rowspan="2"><b>Mintzberg <i>et al.</i>'s Model</b></td><td colspan="2">Problem Identification</td><td colspan="2">Development Phase</td><td colspan="3">Selection Phase</td></tr>
<tr><td>Recognition Routine</td><td>Diagnosis Routine</td><td>Search Routine</td><td>Design Routine</td><td>Screen</td><td>Evaluation & Choice</td><td>Authorization</td></tr>
</table>

Figure-4: Mapping the RE Process Model to Classical Decision-Making-Models

4.1 Macro Decisions in the Requirements Engineering Process

Anthony's model [2] is organization-oriented and demonstrates macro decisions. Each phase in the model has components that may have structured-to-unstructured decisions. More specifically, we believe that in the RE process 'strategic decisions' are primarily an organizational issue, 'management control decisions' involve a project management view, whereas 'operational control' decisions are about the implementation of requirements. In the following subsections macro decisions in RE activities are addressed in terms of the decision type, information type and challenges that project managers face.

<table>
<tr><td><b>Model</b></td><td colspan="5"><b>Classification</b></td></tr>
<tr><td><b>Macaulay's Model</b></td><td>Concept
(Problem Recognition)</td><td>Problem Analysis</td><td>Feasibility & Choice of Options</td><td>Detailed Analysis &amp; Model</td><td>Requirements Document</td></tr>
</table>

Anthony's Model	Strategic Planning		Management Control (Tactical)		Operational Control		
Mintzberg *et al.*'s Model	Problem Identification		Development Phase		Selection Phase		
	Recognition Routine	Diagnosis Routine	Search Routine	Design Routine	Screen	Evaluation & Choice	Authorization

Figure-4: Mapping the RE Process Model to Classical Decision-Making-Models

4.1.1 Strategic Decisions

The activities involved in the first phase of RE, known as 'concept', include components of strategic decisions. The decision-making process is generally unstructured and related to organizational policy and models. Strategic decisions typically defy structure or examination of requirements and involve human intuition.

Strategic decisions in requirements activities require creativity and opportunistic inputs, and should be based on an accurate understanding of the current business process and a detailed understanding of the software product. It is important to align the RE process with the business and technical goals of the organization [14]. An example of strategic decisions in requirements activities include packaging requirements for a specific release of a software product before the requirements are actually put in a project. In the early stages of product development, developers need to decide which requirements to include in the next release and which ones to postpone to the following release. This is a strategic decision from a product point of view as some crucial features of the product, i.e. requirements, may signify the difference between gaining and loosing market shares.

At this stage, the actual level of risk from requirements is uncertain. It is essential to identify the potential risk from an initial set of requirements and to assess the likelihood and potential impact of risk. Incoming information is generally external, ambiguous and predictive. It is crucial to seek information about potentially important aspects of the problem. This leads to a way of relating pieces of information to each other in order to better understand the product in a business context. As a result, developers have improved understanding of the company business, as well as business opportunities. Accordingly, developers analyze, evaluate and select strategies to eliminate, manage or mitigate the potential risk. In summary, risk management is a fundamental organizational activity where uncertainty needs to be handled with care in this phase.

Developers need decision aids, such as decision trees or simulation models, to reduce error and uncertainty in the requirements risk identification process. Project mangers involved in strategic decisions need a vast amount of information in order to make effective decisions in this area. At this level, as part of initial preparations information is gathered from the stakeholders for developers to gain a better understanding of the problem and to draw the system boundaries. Decisions are typically performed in an intuitive, experience-based manner (Strigini, 1996). Project managers need to also be sensitive to organizational and political considerations.

4.1.2 Management Control

Anthony [2] describes management control as a process by which managers assure that resources are obtained and used effectively and efficiently in the accomplishment of the organization's objectives. The second phase of Macaulay’s RE process model [22], known as ‘*problem analysis*’ corresponds with management control activities at the organizational level. The decision-making process changes from structured to unstructured decisions. The management control level information is integrated, more internal and historical. The objective of this phase is to decide the best way to implement the strategic decisions.

Problem analysis in RE involves clarifying an initial set of fuzzy requirements and generating some possible solutions. Thus, understanding and developing solutions require the identification of alternatives, and assessment of the relative costs and benefits of each alternative. The existence of sufficient organizational resources is crucial for software developers during RE activities. The decision maker is constrained by the availability of key resources, *i.e.* human and non-human resources. Examples of decisions may involve the following questions: What kind of development and implementation strategy will be taken? What sort of plans exists for project management? What are the planned benefits? What are the resource reductions? What are the product enhancements? What priorities are placed upon the achievement of these benefits?

During the problem analysis stage, the project manager interested in knowing the ideal resource allocation, how this may be modified to fit the organization, the effect of various tradeoffs, and what changes should be made in the methodology to minimize resources expenditure. It is important that the project manager has a good understanding of the software development process and is able to control the distribution of key resources such as computer time, personnel, and the cost.

4.1.3 Operational Control

According to Anthony [2] operational control is the process of assuring that specific tasks are carried out effectively and efficiently. Decisions in this stage are structured and have a solution (or a reaction) to a given situation. Accuracy of information is an important issue in decision-making at this level.

The activities involved here are concerned with the implementation of requirements. The information involved in this stage is quite detailed and arises from sources of organization. Decision makers may consider several tools and decision aids when making decisions. In operational control project manager is concerned with the software product delivery and budget and handles decision-making with less ambiguous information.

4.2 Micro Decisions in Requirements Engineering Process

Mintzberg’s model [26] is a descriptive model and identifies micro decisions made in a managerial context. In the following sections micro decisions in RE activities are addressed in terms of the decision type, information type and the types of stakeholders that are involved in decision-making. The following points are also identified from Figure 4.

4.2.1 Problem Identification

It can be easily seen from Figure 4 that there is almost a one-to-one relationship between the activities of the first phase of the RE process model, '*concept*' and the first phase of Mintzberg's model, '*problem identification*'.

The problem identification phase involves scanning the environment and searching for information to identify the different aspects of the software product. Decisions mainly involve the nature of the product, the reason building the product, the scope of the product and the qualities of the product. The information is collected and aggregated mainly from stakeholders. This information heavily involves the stakeholders' point of view, goals, objectives, constraints and agendas. Identifying and integrating different stakeholders' points of view is a critical process and affects the success of subsequent activities of the RE process. If one perspective is neglected then the system may be viewed as a failure by the stakeholders. Once developers have a good understanding about the scope of the software product, they need to decide how much work is to be done for the product.

Negotiation is one of the roles of the requirements engineers and always involves multiple stakeholders. For instance, requirements elicited from customers may overlap or conflict. Thus, requirements need to be negotiated and validated before they find their way to the requirements document. Customers and developers need to work together to decide a mutually agreeable list of requirements for the system. Some decisions may be not be negotiated, these may involve a case where a requirement is imposed upon the development team by upper management, or where the development team may decide on the requirements of the system among themselves.

The flow of information within this phase is very erratic, and important information may be overlooked or ignored which can result in poor project evaluation decisions. An example of this is cost estimation at the early stages of the project. Carr and Wagner [8] argue that project managers do not estimate cost first, but instead prioritize maintenance of the project, trying to determine which project to carry out (first) within fixed budgets and resource capabilities. This essentially means that 'cost estimation' is done qualitatively, before formal cost estimation techniques are employed.

4.2.2 Development Phase

The development phase in Mintzberg model corresponds to *problem analysis* phase in Macaulay's model. There are two sub-phases in the development phase: search routine and design routine. The search routine involves searching information to clarify the problem and issues related with the problem and to discover existing solutions. The design routine involves identification of potential solutions.

In RE activities, to better understand the scope of the software product, developers also go through the similar activities i.e. they search information and analyze the problem. During the search they may identify more requirements. The *problem analysis* may have both quantitative and qualitative dimensions. The quantitative analysis may involve the development of models for forecasting. The qualitative analysis is usually performed by pooling the experiential and tacit knowledge possessed by stakeholders regarding each of the alternatives and their likely consequences. The example of decisions includes the following questions. How should the requirements be structured? How should the requirements be classified? How important is the requirement? Who is interested in the requirement? How to prioritize requirements? How does the requirement depend on other requirements? There are several tools available for software developers to analyze and model the existing system. These tools quantitatively analyze decisions with uncertainty, for example, by using expected utility theory [38]. Once an appropriate set of alternatives has been generated, developers typically examine the possible consequences of each option and the likelihood of each occurring. A cost-benefit assessment of the requirements is made then.

In this phase, the dialog mainly involves customers and developers. Project manager's involvement in requirements analysis is not significant.

4.2.3 Selection Phase

The selection phase, in Mintzberg's model, has three sub-phases: Screen, Evaluation and Authorization. There is almost a one-to-one correspondence between these three phases and the last three phases of Macaulay's RE process model i.e. feasibility and choice of options, detailed analysis and modeling, and requirements document.

The ability to make effective decisions under uncertainty is a critical aspect of management. During the phase of feasibility and choice of options, decisions on requirements entail challenging questions concerning the description of alternatives and the estimation of outcomes, both of which require inference and prediction. In this phase, software developers choose one of the alternatives from the set generated and analyzed in the development phase. This is typically based on a number of criteria, some of which are clear and precise while others are somewhat intangible and ambiguous. There is again the question of the degree of uncertainty and risk attached to each of the alternative. The trade-offs among the alternatives is an important aspect of the selection phase. The chosen solution needs to be validated and verified by a variety of stakeholders and carried over time. Information type in this level needs to be accurate and clearly defined. The main decision makers are generally developers and the project manager.

Examples of decisions may include the following question: What are the costs and benefits of collecting additional information? During detailed analysis and modeling, decisions are concerned with individual values and preferences as well as the process of choice. There is no simple model that describes how preferences of individuals are determined, but there are few decision aid tools.

5. Discussion

Several comments about the mapping we generated in Section 4 are in order. An understanding of decision-making in RE activities is a challenge in the advancement of decision-making in software project management. The cycle of the RE activities is somewhat complex. RE activities do not involve a single decision but rather series of complex decisions. Furthermore, making particular decisions in each phase is itself a complex decision-making process.

In Section 4 we aim to illustrate both macro- and micro-elements of decisions in RE activities. Macro decisions are related to the purpose of the management activity at the organizational level, whereas micro decisions are related to how decision makers actually go about their decision-making. Both macro and micro decisions are intertwined and interdependent. Furthermore, macro decisions may also overlap considerably with micro decisions or vice versa. For instance, there might be some strategic decisions not only at a conceptual level but also in the problem analysis stage. Second, requirements decisions will be played out against the context of the organization and its norms, structures and procedures. Third, several stakeholders involvement with diverse expectations and interests adds further challenge for understanding the decision-making process in RE activities. Moreover, decision-making is a knowledge-intensive activity, where cognitive limits substantially restrict developers' decision-making efficiency and effectiveness.

In many cases decision activities include various types of comparisons at some stage *e.g.* comparisons between alternatives, comparisons between decision situations and comparison between the attributes of different alternatives. It is possible to provide decision aids to software developers which support their decision activities. Hence, information acquisition plays an important role in identification of decision alternatives in RE activities. There is likely to be certain types of decision behavior that are always going to be present in unaided decision-making in RE activities. It might also be useful to identify them to help in the development of more appropriate prescriptive decision-making models in RE activities.

March [24] argues that each phase of a product life cycle requires different management styles and no manager is well suited to manage a product through more than two consecutive phases of its life cycle. In a similar way, software development is a team-work and involves several stakeholders, including product manager, project manager, customer and developers.

In order to be able to support and improve the RE activities, the following points need to be addressed.

- Keep track of RE decisions, their rationale, and their effect on software product. Thus, explicitly identify the RE process and ensure that RE activities within the process are explicit, measurable and it can be repeatable.
- Identify the stakeholders who participate in each RE activity and accordingly consider specific decision aids for each stakeholder.
- Identify the decision types involved at each RE phase as well as the meaningful actions or options that each decision maker carries out for each decision type.
- Identify the information type (or knowledge) needed at each phase. Make sure that there is an access to adequate knowledge.
- Provide decision support tools to project managers as well as to development team.

The quality of the software product is largely controlled by the quality of the development process used to create it. Managing the RE phases is vital to the successful development of software products. By focusing on improving the decision-making in the RE process, software developers are more likely to achieve success in the development process. In other words, if software developers' understanding of the decisions they make is improved, and if these decisions are well made, the developers can proceed with confidence, and efficiently design a quality product that will meet stakeholders' requirements.

6. Conclusion

The manageability of the requirements process is important as the quality of the process affects the quality of the product. By studying the decision-making process in RE activities in more detail, it is possible to conduct an analysis of the RE process and its underlying decision-making processes. We believe that decision-making experiences in literature are highly relevant for anyone working with RE, hence the lessons-learned in another field ought to be brought into RE to a larger extent. In a preliminary study [6], we briefly reviewed RE decisions, provided decision taxonomy, and identified a subset of stakeholders and decision types based on a simple case study. In this paper we take a novel approach to illustrate the fundamental nature of decision-making in requirements activities and exemplify several opportunities to the RE community. This paper addressed the major organizational and project decisions in RE activities and explored the nature of requirements activities in detail from the decision-making point of view. The main contribution of this paper was its mapping of a requirements process model to two classical decision-making models and illustrating how these models help us to understand some of the complexities of RE. We believe that mapping categories of decision-making in RE activities can contribute towards the (organizational) learning process through better understanding of current decision-making patterns and changes in those patterns over long periods of time. In addition to this, the classification is very useful to stakeholders, and particularly valuable to software teams in terms of validating their decisions and verifying the consistency of their decision-making.

In addition to this the following points are illustrated:

- It presents an approach to understanding RE decisions by using classical decision-making models.
- It emphasizes the importance of several aspects of decision-making in the RE process.
- It illustrates how to integrate decision-making models with RE process models to improve the manageability of the RE process.

We need a better understanding of what it takes to generate adequate management support and stakeholders' participation in the RE process. A question that arises from the new understanding of RE decisions is 'how can we best manage the RE activities as a decision-making process?' The complexity of the activities involved in the RE process call for a need for organizations to coordinate the decision-making process and make the decisions and the roles played with respect to decision-making in RE more visible. By mapping the RE process model to the decision-making models, we are able to pinpoint the different roles and responsibilities and hence support an organization in structuring their decisions in the RE process. Once an organization has identified the relevant RE activities, the suggested 'mapping' acts as a guideline for deciding what sorts of decisions need to be made. This mapping also indicated the key areas on which the different stakeholders involved in the RE process should focus. It is also important to document the discussions and decisions related to both the organizational and project levels. Finally, there is a need for a continual tracking of RE decisions to ensure that they are conducted in a way that supports the business goals. The further research will study the decision-making process for customer based, market-driven and Internet enabled complex systems.

Acknowledgement

We would like to thank Adrian Gardiner, Christine Van Toorn and Irem Sevinç at the School of Information Systems, Technology and Management, University of New South Wales, Australia, for reviewing this paper and providing very useful comments.

References

[1] S. Andriole, The Politics of Requirements Management, IEEE, Software, November-December, 1998, pp. 82-84.

[2] R.N. Anthony, Planning and Control Systems: A Framework for Analysis, Harvard University, Boston, USA, 1965.

[3] N. Ashrafi, A Decision-making Framework for Software Quality Management, International Journal of Technology Management, 16(4-6), 1998, pp. 532-543.

[4] A. Aurum, E. Martin, Managing both Individual and Collective participation in Software Requirements Elicitation Process, 14th International Symposium on Computer and Information Sciences, (ISCIS'99), Kusadasi, Turkey, 1999, pp. 124-131.

[5] A. Aurum, E. Martin, Requirements Elicitation using Solo Brainstorming, 3rd Australian Conference on Requirements Engineering, (ACRE'98), Melbourne, Australia, 1998, pp. 29-37.

[6] A. Aurum, C. Wohlin, Applying Decision Making Frameworks in Requirements Engineering, 8th International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ'02), Essen, Germany, September, 9-10, 2002.

[7] B. Boehm, P. Grünbacher, R.O. Brigges, Developing Groupware for Requirements Negotiation: Lessons Learned, IEEE, Software, May/June, 2001, pp. 46-55.

[8] M. Carr, C. Wagner, A Study of Reasoning Processes in Software Maintenance Management, Information Technology and Management, 3(1-2) 2002, pp.181-203.

[9] CREWS, Cooperative Requirements Engineering with Scenarios, <http://www-i5.informatik.rwth-aachen.de/lehrstuhl/projects/crews/> Accessed on 1st March, 2003.

[10] A. Davis, Software Requirements – Analysis and Specification, Prentice Hall, UK, 1990.

[11] K. El Emam, N.H. Madhavji, A Field Study of Requirements Engineering Practices in Information Systems Development, Second International Symposium on Requirements Engineering, IEEE CS Press, Los Alamitos, Calif., 1995, pp. 68-80.

[12] R. Evans, S. Park, H. Alberts, Decisions not Requirements: Decision-Centered Engineering of Computer-Based Systems, Proceedings of the International Conference on Engineering and Computer-Based Systems, 1997, pp. 435-442.

[13] F3 (From Fuzzy to Formal). <http://www.newcastle.research.ec.org/esp-syn/text/6612.html> Accessed on 1st March 2003.

[14] D. R. Goldenson, A. Gopal, T. Mukhopandhyay, Determinants of Success in Software Measurement Programs: Initial Results, Proceedings of 6th International Software Metrics Symposium, 1999, pp. 10-12.

[15] H.F. Hofmann, F. Lehner, Requirements Engineering as a Success Factor in Software Projects, IEEE Software, 18(4), July-August, 2001, pp. 58-66.

[16] C.W. Holsapple, A.B. Whinston, Decision Support Systems: A Knowledge-Based Approach, West Publishing Company, USA, 1996.

[17] F. Houdek, K. Pohl, Analyzing requirements engineering processes: a case study Proceedings of the 11th International Workshop on Database and Expert Systems Applications, 2000, pp. 983-987.

[18] T.D. Korson, V.K. Vaishnavi, Managing Emerging Software Technologies: A Technology Transfer Framework, Communications of the ACM 35(9), 1992, pp. 101-113.

[19] G. Kotonya, I. Sommerville, Requirements Engineering – Processes and Techniques, John Wiley & Sons, UK, 1998.

[20] P. Loucopoulos, V. Karakostas, System Requirements Engineering, McGraw-Hill Book Company Europe, 1995.

[21] D. Lowe, J. Eklund, Development issues in specification of Web systems 6th Australian Workshop on Requirements Engineering AWRE'01, 22-23 November, Sydney, Australia 2001, pp. 4-13.

[22] L.A. Macaulay, Requirements Engineering, Springer-Verlag Series on Applied Computing, London, UK, 1996.

[23] S. Mallick, S. Krishna, Requirements Engineering: Problem Domain Knowledge Capture and the Deliberation Process Support, Proceedings of 10th International Workshop on Database and Expert Systems Applications, 1999, pp. 392-397.

[24] L. March, How to Think Strategically, IEE Colloquium, Management and Design Division, 1991, pp. 3-18.

[25] S.R. Martin, A. Aurum, R. Jeffery, B. Paech, Requirements Engineering Process Models in Practice, 7th Australian Workshop on Requirements Engineering, AWRE'02, Melbourne, Australia, 2002, pp. 141-155.

[26] H. Mintzberg, D. Raisinghani, A. Theoret, The Structure of Unstructured Decision Process, Administrative Science Quarterly, June, 1976, pp. 246-275.

[27] L. Nguyen, P.A. Swatman, Managing the Requirements Engineering Process, to appear in Journal of Requirements Engineering, Springer-Verlag London Ltd, England, 2002.

[28] B. Regnell, B. Paech, A. Aurum, C. Wohlin, A. Dutoit, J. Natt och Dag, Requirements Mean Decisions! – Research Issues for Understanding and Supporting Decision-Making in Requirements Engineering, 1st Swedish Conference on Software Engineering Research and Practice, SERP'01, Ronneby, Sweden, 2001, pp. 49-52.

[29] A.G. Rodrigues, T.M. Williams, System Dynamics in Software Project Management: Towards the Development of a Formal Integrated Framework, European Journal of Information Systems. 6(1), 1997, pp. 51-66.

[30] C. Rolland, Requirements Engineering for COTS Based Systems, Information and Software Technology, 41, 1999, pp. 985-990

[31] C. Rolland, C. Souveyet, M. Moreno, An Approach for Defining Ways-Of-Working, Information Systems, 20(4), 1995, pp. 337-359.

[32] T.L. Saaty, L.G. Vargas, Models, Methods, Concepts & Applications of the Analytic Hierarchy Process, Kluwer Academic Publishers, Dordrecht, the Netherlands, 2001.

[33] H.A. Simon, The New Science of Management Decisions. Prentice Hall, Inc. Englewood Cliffs, New Jersey, USA, 1960.

[34] I. Sommerville, P. Sawyer, Requirements Engineering – A Good Practice Guide, Lancester University, John Wiley and Sons, 1997.

[35] L. Strigini, Limiting the Dangers of Intuitive Decision-making' IEEE, Software, January, 1996 pp. 101-103.

[36] M. Tawbi, C. Souveyet, Guiding Requirements with a Process Map'. Proceedings of MFPE'99: 2nd International Workshop on the Many Facets of Process Engineering, Gammarth, Tunisia, 12-14 May 1999. Also published in the Crews Report 98-33, 15p.

[37] H.J.W. Vliegen, H.H. Van Mal, Rational Decision-making: Structuring of Design Meetings, IEEE Transactions on Engineering Management, 37(3), August, 1990, pp. 185-190.

[38] J. Von Neumann, O. Morgenstein, Theory of Games and Economic Behavior, 1953 edition, Princeton University Press, Princeton, NJ, USA, 1944.

[39] C. Wild, K. Maly, C. Zhang, C. Roberts, D. Rosca, T. Taylor, Software Engineering Life Cycle Support – Decision Based Systems Development, IEEE Region 10's 9th International Conference on Computer Technology, TENCON'94, 2, 1994, pp. 781-784.

[40] J.R.G. Wood, A.T. Wood-Harper, Information Technology Support of Individual Decision-making, Journal of Information Systems, 3, 1983, pp. 85-101.