December 2002

Improving the Requirements Engineering Process: a process oriented approach

Mahmood Niazi
University of Sydney

Improving the Requirements Engineering Process: a process oriented approach

Mahmood Khan Niazi

School of Information Technologies
University of Sydney
Sydney, Australia
mkniazi@cs.usyd.edu.au

Abstract

The Requirements Engineering (RE) process plays an important role in the software development process. In order to produce quality software greater attention must be given to the improvement of RE process. In this paper five key process areas (KPAs) have been identified from the research literature in order to improve the RE process. Firstly to support a goal-based approach in the RE process; secondly to support the incremental and cyclical behaviours in the RE process; thirdly to encourage stakeholders involvement in the RE process; fourthly, to support the management of RE process and fifthly to define a planning phase for the RE process. This research project aims to show that better results will follow when the RE process supports these five KPAs. To address these KPAs, a requirement elicitation, analysis and validation method (REAVM) is proposed. A case study has been conducted in order to test and evaluate the REAVM in the real world environment.

Keywords

FB03, FB0302

INTRODUCTION

A complete understanding of software requirements is essential to the success of a software development effort. Inaccurate, inadequate, or misunderstood requirements are the most common causes of poor quality, cost overruns and late delivery of software systems (El Emam and Madhavji, 1995). A well defined RE process is not included in most system's lifecycles. Normally the life cycle begins with the writing of requirements. Most existing requirements methods and techniques concentrate on the later phase of the RE process which focuses on specification, i.e. the documenting of the requirements. In contrast, the early stage of the RE process aims to elicit, analyse and validate the requirements from different stakeholders. The early stage is often haphazard and ill defined, which increases the chance of failure. Attention to the early stage of the RE process is crucial in order to achieve better results in the RE process.

Requirements problems are widely acknowledge to reduce the quality of software and to reduce the effectiveness of the software development process (Sommerville, 1996). Despite the importance of requirements engineering, little work has been done on developing ways to improve requirements process. Existing standards for software process improvement (SPI), i.e. Capability Maturity Model (CMM) (Paul *et al*., 1993; 1994) and ISO 9000 (Johnson, 1993) series standards do not address the requirements engineering adequately. There is no specific section referring to requirements engineering in these standards and they consider requirements engineering as a single activity in the development process. While "The importance of requirements engineering demands that it be recognised as a complex process in its own right and not simply as a phase of the software life-cycle" (Sommerville *et al*., 1997:23).

In this paper five KPAs have been selected from research literature in order to improve the RE process. This research investigates the research question that: 'Better results will follow when the RE process supports the five KPAs. In order to address this question a Requirement Elicitation, Analysis and Validation Method (REAVM) is proposed. A process-oriented approach has been used as REAVM is divided into five major processes and each process is an organised set of activities that transforms inputs to outputs. The objective is to achieve the better results in the RE process and enable requirements engineers to develop incrementally a more complete version of the requirements document. A case study has been conducted in order to see the behaviour of REAVM in the real world environment.

In the paper, motivation and background is provided; different KPAs are defined; REAVM is described; REAVM is evaluated through case study; followed by a conclusion.

MOTIVATION AND BACKGROUND

Many software projects have failed because they contained a poor set of requirements (El Emam and Madhavji, 1995). No software process can keep delivery times, costs and product quality under control if the requirements are poorly defined (Sommerville *et al.*, 1998). In order to produce software, which closely matches the needs of an organisation, an application domain and the stakeholders, great attention must be given to the RE process (Niazi, 2000). The RE process plays an important role in the software development process. The objective of a RE process should be to develop a set of necessary, verifiable and attainable requirements, which are acceptable to all the relevant stakeholders (Kotonya and Sommerville, 1998).

Requirements engineering is an important process of the software life-cycle. It has been observed that one can achieve better quality in software and systems development process if the RE process is properly defined (Sommerville *et al.*, 1998). Normally the RE process is started without any planning, which results in poor quality requirements and less control over the management of the whole RE process. A mismatch has been observed between the problems experienced by industry and the techniques developed from research in requirements engineering (Sommerville *et al.*, 1997). It is also observed that many analysts have limited knowledge of the problem domain, which also results in poor quality requirements and cost overruns (Kotonya and Sommerville, 1998). Some examples of fairly common problems with the RE process are as follows (Sommerville *et al.*, 1997) and (Kotonya and Sommerville, 1998):

- Lack of stakeholder involvement.
- Business needs are not considered.
- Lack of requirements management.
- Lack of defined responsibilities.
- Stakeholders' communication problems.
- The requirements do not reflect the real needs of the customers.
- Requirements are inconsistent and/ or incomplete.
- It is expensive to make changes to requirements after they have been agreed.
- There are misunderstandings between customers and software engineers.

The fundamental problems in requirements engineering have been identified by many researchers (e.g. Siddiqi and Chandra, 1996; Nuseibeh and Easterbrook, 2000; Lubars *et al.*, 1993; Nikula *et al.*, 2000; Morris *et al.*, 1998; Kamsties *et al.*, 1998; El Emam and Madhavji, 1995). To highlight a few of these, Siddiqi and Chandra (1996) and Nuseibeh and Easterbrook (2000) outlined the ongoing research in requirements engineering and its future directions. Siddiqi and Chandra (1996) mentioned a gap between current research and practice and in order to reduce this gap they suggested a continuous discussion between researchers and practitioners. Lubars *et al.* (1993) and Nikula *et al.* (2000) analysed the requirements engineering practices in different organizations. Lubars *et al.* (1993) interviewed ten organizations to find out how they defined, interpreted, analysed and used requirements. They concluded that organizational solutions were favoured over technological ones and general-purpose tools were more common than special purpose tools in requirements engineering. Nikula *et al.* (2000) conducted a survey with twelve small and medium enterprises in order to get some numerical data on the knowledge on current requirements engineering practices and the desire to improve them. They presented the results of an empirical survey showing that the problem is not in the practitioners' lack of desire for improvement but in the management not knowing that many requirements engineering issues can be solved with standard practices that are well documented in the literature. El Emam and Madhavji (1995) described a field study and the results indicate that there are seven key issues of greatest concern that must be addressed in a successful RE process improvement effort: package consideration, managing the level of detail of functional process models, examining the current system, user participation, managing uncertainty, benefits of case tools and project management capability.

The above problems have been addressed to some extent in different proposals and methods in the requirements engineering community (Macaulay, 1993; Gonzoles, 1997; Gonzoles and Wolf, 1996) but these proposals/methods are far from being completely satisfactory. Some of these are still in the early stages of use, and as such have not been widely accepted by the software engineering industry. A method may be conceptually good but when it is applied to the real system, it does not satisfy the requirements of the system (Sommerville *et al*., 1998). Conventional methods usually support only parts of the RE process or help identify only specific kinds of requirements. Methods widely used in industry have serious weaknesses, both in the modelling paradigms and in the preciseness of their definition. Academic research on the other hand focuses primarily on formal specification techniques. However, most of these techniques are too complicated for broad industrial application. Therefore, work on methods lies at the very core of requirements engineering research.

Some people have worked on software process improvement, i.e. Capability Maturity Model (CMM) (Paul *et al*. 1993; 1994) and ISO 9000 (Johnson, 1993) series standards. The CMM and ISO 9000 series of standards share a common concern with quality and process management. There is no specific section referring to requirements engineering in these method and they consider requirements engineering as a single activity in the development process. The CMM is a valuable method for the software process improvement but it is very hard to gain benefits when it is applied to the requirements process. Only requirements management is treated in details and is identified as a KPA for level 2 (repeatable) processes. But requirements management is only one area of the requirements process. CMM does not provide any specific section for the other areas of the RE process, i.e. requirements elicitation, requirements analysis, requirements negotiation and requirements validation. There is also no particular section to requirements engineering in ISO 9000 series standards and they do not say much about the activities involved in eliciting, analysing, negotiating and validating the requirements

Sommerville *et al*. (1997; 1998) have published the RE process maturity model which has been derived from the existing standards and has three levels, i.e. Level 1-Initial, Level 2-repeatable and Level 3-Defined. This model can be used to assess current RE process and it provides a template for requirements engineering practice assessment. This model does not provide any general methodology for the improvement of the RE process. It also does not provide KPAs like CMM but rather it organizes different requirements practices with various deliverables in the RE process.

Requirements engineering is an important process of the software life-cycle. As no current SPI methods adequately address the issues of RE process and they broadly treat requirements engineering as a single activity in the overall development process, therefore, research in the area of RE process improvement lies at the very core of requirements engineering research.

IMPROVING THE REQUIREMENTS ENGINEERING PROCESS

This research addresses some of the important issues in requirements engineering. This is applied research rather than theoretical research, and its objectives are the solution of the real world problems. The major objective is to improve and structure the RE process because if the RE process is improved, better results can be achieved and the real needs of the stakeholders can be reflected.

Like CMM (Paul *et al*., 1993; 1994), the following five KPAs have been selected from research literature. This research project aims to show that better results will follow when the RE process supports the following:

To support a goal-based approach in the RE process

Goals are the high level objectives of the business, organisation or system which provide a framework for the desired system (Anton, 1997). Goals denote the objectives a system must meet (Nuseibeh and Easterbrook, 2000). Goals are useful for organising and justifying requirements. Goals have been introduced into requirements engineering for a variety of reasons, i.e. requirements acquisition, relating requirements to the organisational and business context, clarifying requirements, dealing with conflicts, assisting the management of change and driving the initial design (Yu and Mylopoulos, 1998). Goals set an agenda by which requirements are discovered, analysed and documented (Sommerville *et al*., 1998). Normally it is difficult for the stakeholders to fully understand the requirements of the organisation or application domain but with clear goals a good understanding can be obtained. By focusing on goals initially instead of broad requirements, analysts enable stakeholders to communicate using a language based on concepts with which they are both comfortable and familiar (Anton, 1997).

To support the incremental and cyclical behaviours in the RE process

Although there is no universal requirements process but several studies (Potts *et al*., 1994; Boehm *et al*., 1994; Sommerville *et al*., 1997; 1998) strongly suggested that the requirements process is cyclical. Potts *et al*. (1994) have proposed a cyclical model, called the Inquiry Cycle that consists of three iteratively repeated activities: expression, discussion and commitment. Boehm *et al*. (1994) have proposed a requirements process model based on its spiral model of software development (Boehm, 1988), which establishes stakeholders' 'win' conditions and includes steps in order to facilitate identification and negotiation of requirements trade-offs. Sommerville *et al*. (1997; 1998) have also proposed a spiral model that consists of three iterative activities: requirements elicitation, requirements analysis and validation and requirements negotiation.

The incremental behaviour is regarded as the most realistic approach to software development for large-scale systems (Pressman, 1997). It uses an evolutionary approach to development and contains the systematic and the 'development in steps' approach of the traditional project life cycle (Sommerville, 1996). Using this behaviour the functionality of the system is produced and delivered to the customers in small increments which avoids the 'Big Bang' effect, i.e. for a long time nothing happens and then, suddenly, there is a completely new situation (Vliet, 1993).

To encourage stakeholders involvement in the RE process

In most cases the concerned stakeholders are not involved in the RE process and their real needs are not considered in the system (Sommerville *et al*., 1997). Involving the stakeholders in the development process can reduce their fear for example that the development of a software system will result in loss of jobs. It is also possible that if a new system is installed in an organisation without consulting the stakeholders, who would be affected by the system, then they may feel that a new system is unnecessary and therefore they tend to not co-operate in its specification. Stakeholders involvement in the RE process is one of the most important factors that contribute to the success of the project (Rauterberg and Strohm, 1994; DeBillis and Haapala, 1995). With the stakeholders involvement less rework of the documentation items is required, real requirements can be gathered, political conflicts are reduced and chances of destruction are reduced (El Emam and Madhavji, 1995).

To support the management of RE process

During the RE process new requirements emerge and existing requirements change at all stages of the system development process. It is often the case that more than 50% of system's requirements will be modified before it is put into service (Kotonya and Sommerville, 1998). The RE process is a learning process, and ideas generated at one point may change at another point. This evolution of requirements throughout the whole software development life cycle has to be managed in order to ensure high-quality specifications. The management includes issues such as information storage, organization, traceability and documentation. Requirements management may look like an overhead in the RE process, but it is usually rewarded by better customer satisfaction and lower system development costs.

To define a planning phase for the RE process

Effective management of a software project depends on thoroughly planning the project (Sommerville, 1996). Normally the RE process is started without any planning and the requirements engineers inevitably wish to start very quickly as in (Gonzoles 1997; Gonzoles and Wolf, 1996). The RE process will be an unproductive exercise if started haphazardly and without planning. Particular attention should be paid to the planning of the RE process.

REQUIREMENTS ELICITATION, ANALYSIS AND VALIDATION METHOD (REAVM)

In order to depict the five KPAs a "Requirements Elicitation, Analysis and Validation Method (REAVM)" has been developed This method has been derived from the cyclical and incremental models and has an iterative and feedback nature. The reason for the development of a method is that a method is a systematic way of working by which one can achieve a desired result (Wieringa, 1996). "A method provides a prescription for how to perform a collection of activities, focusing on how a related set of techniques can be integrated, and providing guidance on their use" (Nuseibeh and Easterbrook, 2000:39). All the identified KPAs are incorporated in this method.

The development of a method is heavily dependent on a thorough definition of its processes, roles, activities and interactions. Further recent trends, focusing on process technology, have confirmed that a quality product can only be the result of a quality process (Aliee, 1996). Thus a process-oriented approach to method definition has been selected as the basis for this research project. REAVM is divided into five major processes and each process is an organised set of activities which transforms inputs to outputs. Each phase takes an input, adds value to it and provides an output. The output of a phase is used as an input for the next phase and so on.

The incremental behaviour of REAVM

This method assumes that the requirements for large systems are incrementally gathered, analysed and validated using multiple builds as shown in Figure 1. The initial planning for build-1 is performed at the beginning of the project. Further planning is performed as and when required, as new goals can emerge during different phases. The next three phases, elicitation, analysis, and agreement are performed once for each build. The last phase, validation, is performed after each build. This shows the incremental behaviour.

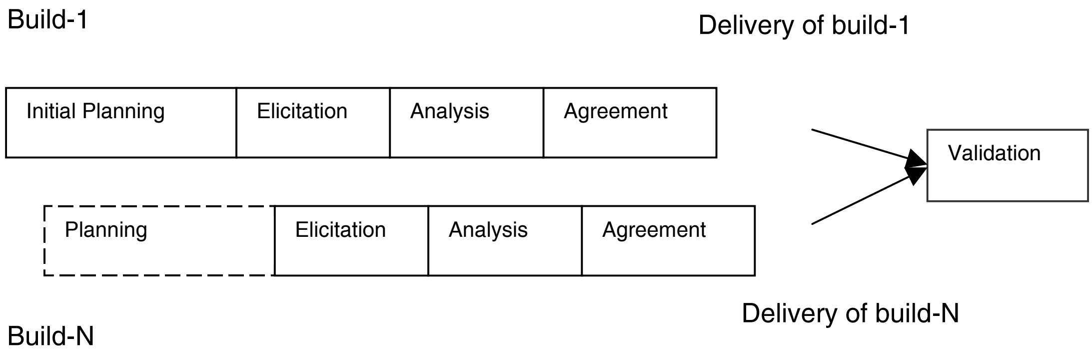

Figure 1: Incremental behaviour of REAVM

The cyclical behaviour of REAVM

Several studies (Potts *et al.*, 1994; Boehm *et al.*, 1994; Sommerville *et al.*, 1997; 1998) strongly suggested that the requirements process is cyclical. Figure 2 illustrates proposed cyclical model that has been abstracted from these studies. It is cyclical in that requirements become apparent from successive iterations in the context of the requirements which emerge from previous iterations. Hence requirements which emerge in the later iteration may limit requirements which emerge in the previous iteration. Therefore, requirements may need to be modified in the light of information that emerges later.

In this cyclical model, five activities are repeated each iteration of the REAVM cycle. This model works at two levels: Firstly, only one goal set [see later] is considered for REAVM cycle. After the first cycle of REAVM, if sufficient information is not collected or some conflicts are still not resolved then the same goal set is re-considered for the second cycle of REAVM and so on. Through this cyclical behaviour requirements will become apparent and it is possible that the requirements generated in the later iteration may limit requirements generated in the previous iteration. Secondly, after the completion of first goal set then the second goal-set is considered for REAVM cycle and as mentioned earlier requirements which emerge in the iteration of second goal set may limit requirements which emerged in the iteration of first goal set. Hence requirements elicited in each cycle of REAVM are validated with the previous elicited requirements for consistency, completeness and feasibility.

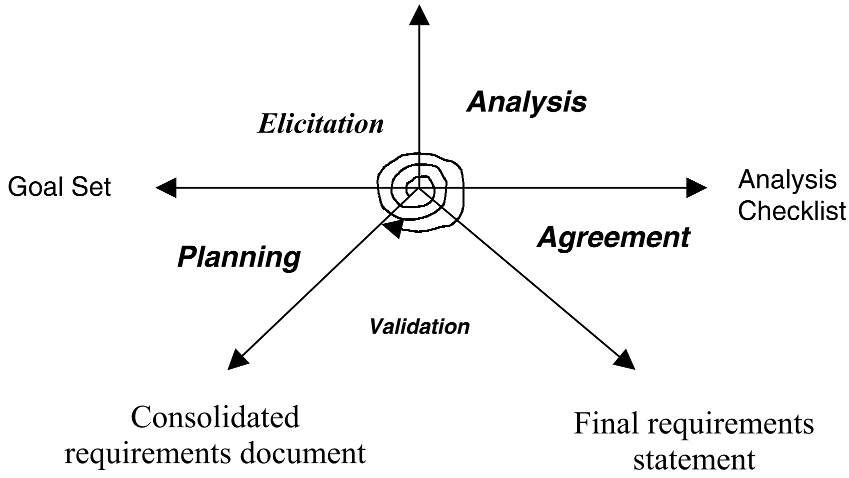

Figure 2: Cyclical behaviour of REAVM

The structure of REAVM

REAVM is divided into five phases (as shown in Figure 3):

- Initial planning.
- Requirements elicitation.
- Requirements analysis.
- Requirements agreement.
- Requirements validation.

Initial planning phase

This is the first phase of REAVM. The aim of this phase is to provide some planning for the subsequent phases of REAVM. The following key activities are performed in this phase:

- Identifying stakeholders.
- Introducing the Goal Proforma (GP).
- Developing the Initial System Model (ISM).
- Performing customisation

There are four types of stakeholders in REAVM, i.e. the executive sponsor, the analyst, the domain experts (DEs) and the user representatives (URs). The executive sponsor is the manager or executive who is responsible for making executive level decisions and commitments. The analyst is responsible for the production of ISM and different tasks of REAVM. A domain expert is a person who can provide detailed information on a narrow, well-defined topic. They have the best available view of a particular domain area.

Goals are the high level objectives of the business, organisation or system that provide a framework for the desired system (Anton, 1997). Goals are considered in REAVM because it is difficult for the stakeholders to fully understand the requirements of the organisation or application domain but with clear goals a better understanding can be obtained. The GP is introduced in REAVM to establish the goal and the flow of that goal from each DE and UR. According to (Sommerville, 1996) simple diagrams, supplemented by descriptions of the system entities, are the appropriate starting points for describing system contexts. The GP is constructed by assuming that the stakeholders have goals in their minds. The format of the GP is shown in Table 1.

Flowchart is a graphical representation used to show the flow of control and the sequence of processes of a particular system. The flowcharts are used in GP in order to provide a preliminary graphical view of goals containing necessary information about input and output documents; different processes of the system; their relationship and flow. It is possible that some DEs and URs may need assistance during the creation of the flowcharts. If there are no flowcharts then the information, which is provided by the DEs and Urs, should be converted into an appropriate graphical representation by the analyst, however, it is still necessary that these flowcharts should be owned by the DEs and URs.

The analyst creates the ISM using his understanding of the system and by merging all the flowcharts and information provided by the DEs and URs in different GPs. It is important that only flowcharts notations should be used during the creation of the ISM. All the flowcharts developed in the GPs are merged into one ISM. The interconnections, inconsistencies and the deficiencies should be checked carefully during creation of the ISM. Unclear areas must be reviewed. All the processes, inputs and outputs of the system should be clarified.

The final ISM contains necessary information about input output documents, different processes of the system, their relationship and flow. It helps the analyst to become familiar with the existing system and organisational structure

The customisation is an important and necessary stage in the initial planning phase. In this customisation, different essential tasks are performed by the analyst, i.e. goals serialisation, creation of goal sets, team organisation, assigning of responsibilities and preparation of different materials to be used in different phases of REAVM. To support the incremental behaviour it is important to serialise the goals so that an incremental approach could address these goals in some priority order. For this purpose, goals serialisation process is performed on the basis of information provided by different DEs and URs in their specified GPs and the goals serialisation list created by the analyst from the ISM. This serialisation gives the work sequence through which different goals will be processed. It is not feasible to tackle all the goals at one time, because it is possible that important information may not yet be elicited and this process thus becomes a fruitless exercise. To cope with this problem and to support the cyclical behaviour, goals are organised into goal sets and only one goal set is considered at one time for each cycle of REAVM. To be manageable, the number of goals per set should be small; it is therefore important that each set should not contain more than five goals. Teams are organised according to the goal-sets. Responsibilities are assigned to different stakeholders for elicitation, analysis, agreement and validation phases. Also some materials are prepared which can be used during different phases of REAVM. This material also contains the question lists to be used during the elicitation phase. After customisation a high priority goal-set (a set which has goals of low serial numbers) is forwarded to the REAVM cycle then second priority goal-set and so on.

Goal Number (To be completed by the Analyst): __ Date:____ Goal Name: ____________
Description of Goal: ________Sources of Goal: __________ Function of Goal: __________
Problems:____
Serial Number of Goal (1-10): __ (To be completed after the development of ISM)
Flowchart of Goal:
Name and Designation________________
Signature_________________

Table 1: The Goal Proforma

Requirements elicitation phase

This phase of REAVM is derived from JAD (Raghavan *et al.*, 1994). The goal-sets, which are created in the initial planning phase, are taken as input. The goal-set which has the highest priority number is considered first for elicitation and so on. The following steps are performed in this phase:

- Defining high-level requirements.
- Bounding the scope of the requirements.
- Generation of initial requirements statements.

The analyst facilitates the group discussions that elicit the requirements. First of all the analyst explains the purpose and objectives of the meeting. The participants bring different ideas and views about different problems. Participants are invited to express their viewpoints about any of the problems. Different questions are asked from each DE and UR using the lists of questions prepared in the initial planning phase. Every participant is allowed to present his viewpoint, if required. The following topics are addressed and the acceptable solutions are provided:

- What objectives the system should meet.
- Problems with the existing system.
- Requirements collection of the organisation, the application domain and the stakeholders.

Through carefully facilitated discussions, the ideas and views about the above topics are presented, examined and refined, so that by the end of the elicitation phase everyone is in agreement. If necessary, interviews can be conducted with those who are not participating in the meeting but they have some relation with the goal-set under consideration.

At the end of the elicitation phase an initial requirements statement (as shown in Table 2) is generated for each goal by the analyst and is given to each DE and UR for analysis and discussion.

Initial Requirements Statement
Set No: ________ Goal Number: _________ Serial Number of Goal: ________ Date: _________
Goal Name: ________ Description of Goal: ____________Sources of Goal: _________________
Function of Goal: ____________ Problems: ____
Elicited Requirements: Requirement 1: _____________________________________
Requirement n: _____________________________________

Table 2: Initial requirements statement

Requirements analysis phase

The goal of this phase is to find problems in the initial requirements statements generated in the requirements elicitation phase of REAVM.

In the requirements analysis phase the following types of checking is performed using the analysis checklist (available from the author):

- Completeness checking.
- Necessity checking.
- General comments.

The requirements are analysed for completeness. Completeness means that no requirements that are needed have been omitted, i.e. whether the elicited requirements have covered all of the needs and objectives of the organisation, application domain and stakeholders? An initial requirements statement can be considered as complete when all of its parts are present and no postponed decision or no ‘to be defined’ statements, still exist. In completeness checking the incomplete requirements are pinpointed.

The analysis of the necessity checking is to see if the elicited requirements contribute to the business goals of the organisation, i.e. whether the elicited requirements satisfy the needs and objectives of the organisation, application domain and stakeholders. It is also analysed to ascertain whether the elicited requirements are in fact necessary and solve the specific problems. In necessity checking, unnecessary requirements are pinpointed.

At the foot of the analysis checklist the stakeholders have to provide general comments about the initial requirements statement. In general comments the stakeholders give their point of view about the elicited requirements and mention whether or not they agree with the elicited requirements, or want further modification. If they want further modification then that modification is specified. They can also mention new goals, if any have emerged during the elicitation and analysis phases.

Requirements agreement phase

The agreement phase is the process of discussing the issues/problems pointed out by the DEs and URs in the requirements analysis phase of REAVM and finding some agreement with which all of the stakeholders can live. All the analysis checklists are discussed individually and the objective of discussion is to solve the issues in particular checklist. All the stakeholders are encouraged to give comments on the problems identified and the recommendations made by them in different analysis checklists. Solutions are identified and issues are resolved to the satisfaction of the parties involved. Generally, this will involve deletion of some requirements and making changes to some of the requirements in order to improve them.

In many cases, it is possible that some questions may be raised which cannot be answered, and for which the stakeholders may not agree with the proposed solutions. This means that the information available for the agreement is insufficient. In such cases, the unresolved issues are forwarded again to another round of REAVM.

If some new goals emerge then it is decided in the agreement phase whether these newly emerged goals require some planning or not. If these goals do not require planning then these goals are considered for elicitation. If these goals require planning then they are considered for planning separately, i.e. these new goals should not be mixed up with those goals whose initial planning has been performed. It is also possible that some explicit steps of planning are performed for these new goals e.g. team organisation, assigning of responsibilities etc., according to the state of emerged goals.

This phase is concluded by reviewing with the participants the information collected and the decisions made. At the end of this phase, the final requirements statements (as shown in Table 3) and the agreement checklists (available from the author) are generated and forwarded to the validation phase for validation and discussion.

Final Requirements Statement
Set No.________________ Goal Number.________ Serial Number of Goal.________________ Date.___________ Goal Name.__________
Final Elicited Requirements:
Requirement 1: ___________________________________
Requirement n: ___________________________________

Table 3: Final requirements statement

Requirements validation phase

This is the final phase of REAVM. In REAVM the goals are divided into related sets and only one goal-set is considered for elicitation, analysis and agreement at any one time. It is therefore possible that some infeasibility, inconsistency and incompleteness may exist when all the goals are consolidated into one document. It is also possible that some previous requirements may change because the customers can change their minds, or even the environment of the system, laws or regulations might change. Therefore, the objective of this phase is to check and remove such infeasibility, inconsistency or incompleteness and to modify the changed requirements to the new requirements.

The following steps are performed in the validation phase:

- Consolidation.
- Pre-review checking.
- Review by the DEs and URs.

- Review by the analyst.

The final requirements statements generated in the agreement phase of REAVM, are consolidated into one requirements document after each cycle or build. This document acts as a channel of communication amongst the stakeholders, thus it should be carefully constructed. Each goal has its own serial number, so using this serial number all the final requirements statements are consolidated sequentially into one document. This document should clearly and unambiguously be understandable to all the stakeholders. It should be sufficiently precise and easy to modify. It should be representative of all the stakeholders who contribute to it.

Requirements review technique is used in REAVM to minimise the work of the reviewers and to remove the avoidable errors from the requirement document. Therefore, before distributing the requirements document for general review one person should do a quick check to remove the avoidable errors, e.g. spelling mistakes etc. Then the requirements document is distributed to the DEs and URs for review.

The DEs and URs read and analyse the requirements document and look for different problems e.g. changed requirements, inconsistencies, incompleteness and infeasibility. They note different problems in the validation checklist-1 (available from the author). Each reviewer notes the different problems identified by him in a separate validation checklist. They also give recommendations for the solution of identified problems. After the completion of validation checklist-1, each checklist is forwarded to the analyst with requirements document for cross checking.

The analyst reads and analyses the requirements document and each validation checklist-1. By using his knowledge and understanding of the system the analyst looks at different problems and recommendations given in the checklists. He gives comments on them using validation checklist-2 (available from the author). He also develops interaction matrix to find conflicts between various requirements.

Finally a meeting is held in which these validation checklists and intersection matrix are discussed, and agreed actions are performed. If some requirements are incomplete then for those specific requirements the elicitation, analysis, agreement and validation phases can be performed again. If some requirements are inconsistent then meetings are held between the stakeholders, whose requirements are inconsistent, in order to reach agreement and to remove these inconsistencies. If some requirements are infeasible then those requirements are modified or eliminated and if some requirements are changed then those requirements are modified according to new requirements.

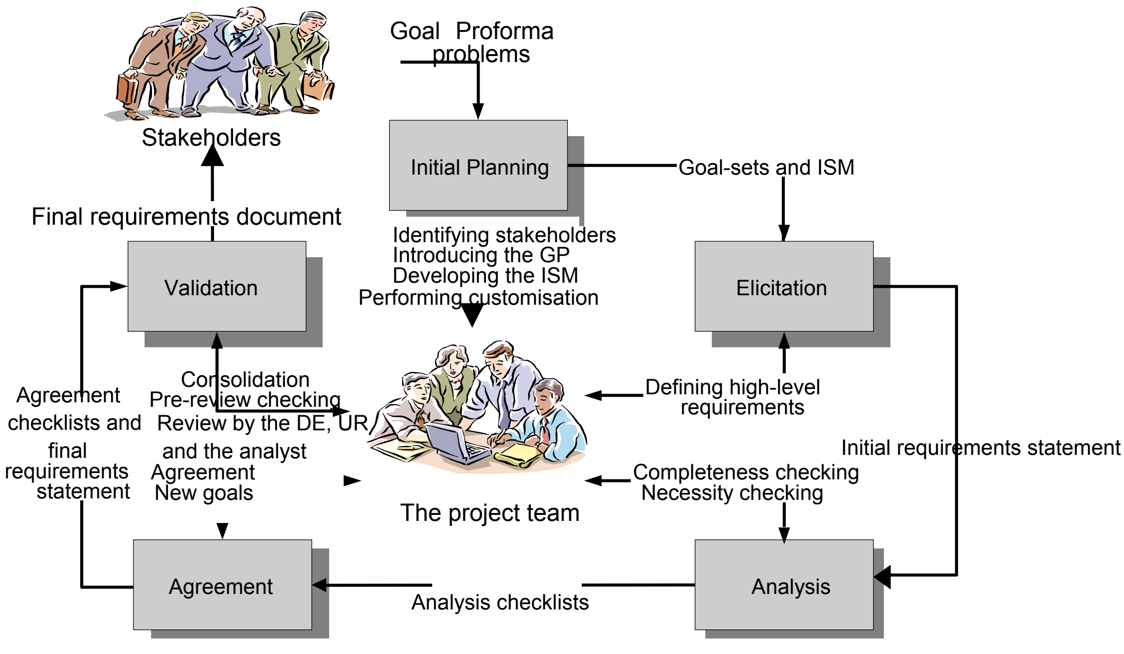

Figure 3: The structure of REAVM

EVALUATION OF REAVM USING CASE STUDY

A case study was conducted at the XYZ Company. The main purpose of the company is to enhance the efficiency and effectiveness of the Information Systems prevailing in public and private sectors. This study was conducted in order to test and evaluate the REAVM in the real world environment. The case study was carried out with three main objectives. Firstly, to test the validity of REAVM. Secondly, to highlight areas where the REAVM has deficiencies. Thirdly, to show the practicality of REAVM in use. In this case study, 8 goals were collected from the stakeholders. Then these goals were serialised and put into sets. Three sets of these eight goals were made; set number one and two contains three goals respectively while set number three contains two goals. Then teams were organised and responsibilities were assigned to different stakeholders. Priorities were given to goal-sets. Sixty-six requirements were generated from these 3 goal sets.

At the end of this case study, a requirements review process was carried out by the stakeholders who were involved in the case study in order to compare the REAVM with the method used by XYZ Company. The author worked as an observer in this process. This process contained four checklists, i.e. requirements elicitation, requirements modelling, requirements verification and requirements management (available from the author). These checklists have been developed using different literature (El Emam and Madhavji, 1995; Sommerville *et al*., 1997; 1998; Kotonya and Sommerville, 1998). Each checklist were jointly completed by all the stakeholders who were involved in the case study and at the end of this process a report was produced which compare REAVM with the standard method used by the XYZ Company. The assessment criteria were adapted from Sommerville *et al*. (1997; 1998). In these checklists 4 points were given to the guidelines which were very well defined, 3 points were given to the guidelines which were adequately defined, 2 points were given to guidelines which were less than adequately defined, 1 point was given to guidelines which were not defined very well and zero points was given to guidelines which were not applicable.

As a whole, the REAVM did not perform exceptionally well when compared with the method used by XYZ Company. A column chart is shown in Figure 4 where REAVM satisfied 70.23% of the criteria and XYZ Company satisfied 55.95% of the criteria. Although REAVM did not perform exceptionally well when compared with the method used by XYZ Company but it was observed that the method followed by XYZ Company has some deficiencies, which have been overcome in REAVM. Some of the important properties which have not been considered in the methodology followed by XYZ Company are: consideration of sources of requirements for the traceability, consideration of goals for the derivation of requirements, management of new goals, unique identification of requirements for effective management, classification of requirements and use of checklists and interaction matrix for the verification of collected requirements. These deficiencies have been overcome in REAVM where sources of requirements have been recorded in REAVM using GPs, a goal-based approach has been used in REAVM in order to derive, analyse and document different requirements, new goals have been managed by the use of checklists, requirements have been classified into related goals, different checklists have been used in order to validate different requirements and the interaction matrix has also been used in order to check requirements conflicts. In addition to all these improvements stakeholders actively participated in all the phases of REAVM. A planning phase has helped in the management of the whole REAVM process. Incremental and cyclical behaviours have helped in the generation of requirements in steps and avoided the ‘Big Bang’ effect. Requirements management has given more control to the monitoring and effectively generating different kinds of REAVM statements.

CONCLUSIONS

For the improvement of RE process five KPAs were considered and it was believed that if these KPAs have been considered then the RE process will be improved. To check whether RE process has been improved or not it is important to check the results, which have been achieved by conducting the case study. When the results of the case study are considered then REAVM satisfied 70.23% of the criteria and XYZ Company satisfied 55.95% of the criteria. It is clear that REAVM did not perform exceptionally well when compared with the method used by XYZ Company.

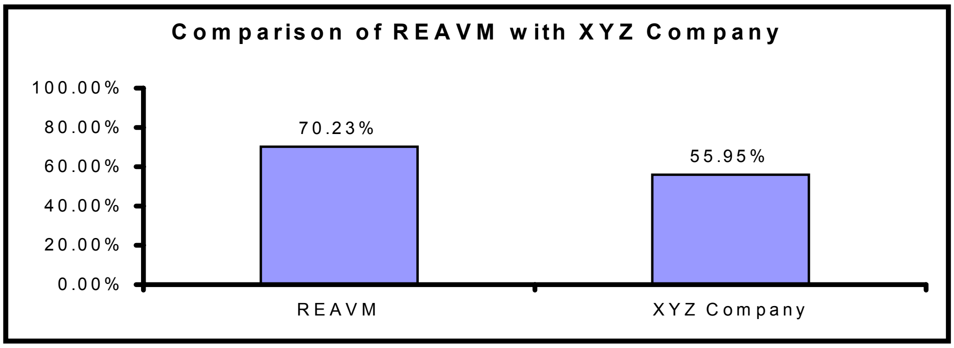

Figure 4: Comparison of REAVM with XYZ Company

As a whole REAVM did not perform exceptionally well. It is believed that the KPAs selected for the improvement of REAVM are the best cluster in order to enhance the capability of the RE process but the way these KPAs are structured into REAVM is inadequate and needs further refinement and improvement.

REFERENCES

Aliee Shams Fereidoon (1996), *Modelling the Behaviour of Processes using Collaborating Objects*, PhD thesis, University of Manchester, Computer Science Department.

Anton Annie (1997), *Goal Identification and Refinement in the Specification of Software-Based Information Systems*, PhD thesis, Georgia Institute of Technology.

Boehm B. W., Bose P., Horowitz E. and Lee M-J (1994), Software requirements as negotiated win conditions, *First International Conference on Requirements Engineering,* 74-83.

Boehm B. W. (1988), A Spiral Model of Software Engineering Development and Enhancement, *IEEE computer*, 21(5), 61-72.

DeBillis M., and Haapala C. (1995), User-Centric Software Engineering, *IEEE Expert*, 10(1), 34-41.

El Emam Khalid and Madhavji H. Nazim (1995), A Field Study of Requirements Engineering Practices in Information Systems Development, *Second International Symposium on Requirements Engineering,* 68-80.

Gonzoles M. Regina (1997), A Model Based Requirements Elicitation Method, *The High Integrity Software Conference*, Albuquerque, N.M.

Gonzoles M. and Wolf L. Alexander (1996), A Facilitator Method for Upstream Design Activities with Diverse Stakeholders, *Second International Conference on Requirements Engineering,* 190-197.

Johnson P. (1993), *ISO 9000 Meeting the New International Standards*, McGraw-Hill.

Kamsties Erik, Hormann Klaus and Schlich Maud (1998), Requirements Engineering in Small and Medium Enterprises, *Requirements Engineering*, 3(2), 84-90.

Kotonya Gerald and Sommerville Ian (1998), *Requirements Engineering Processes and Techniques*, John Wiley & Sons.

Lubars Mitch, Potts Colin, Richter Charles (1993), A Review of the State of the Practice in Requirements Modelling, *First International Symposium on Requirements Engineering*, 2-14.

Macaulay Linda (1993), Requirements Capture as a Co-operative Activity, *First International Symposium on Requirements Engineering*, 174-181.

Morris Philip, Masera Marcelo and Wilikens Marc (1998), Requirements Engineering and Industrial Uptake, *Third International Conference on Requirements Engineering*, 130-137.

Niazi Mahmood (2000), A Method to Improve the Requirements Engineering Process, *Fifth Australian Workshop on Requirements Engineering*, 23-36.

Nikula Uolevi, Fajaniemi Jorma and Kalviainen Heikki (2000), Management View on Current Requirements Engineering Practices in Small and Medium Enterprises, *Fifth Australian Workshop on Requirements Engineering*, 81-89.

Nuseibeh Bashar (1994), *A Multi-Perspective Framework for Method Integration*, PhD thesis, Imperial College of Science, Technology and Medicine, University of London, Department of Computing.

Nuseibeh Bashar and Easterbrook S. (2000), Requirements Engineering: a roadmap, *Proceedings of International Conference on Software Engineering (ICSE-2000)*, June 2000.

Paul Mark, Weber Charles, Curtis Bill and Chrissis Mary (1993), *Capability Maturity Model for software*, Version 1.1, CMU/SEI-93-TR-24, Software Engineering Institute USA.

Paul Mark, Weber Charles, Curtis Bill and Chrissis Mary (1994), *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addision-Wesley.

Pressman S. Roger (1997), *Software Engineering - A Practitioner's Approach*, The McGraw-Hill Companies, Inc.

Potts C., Takahashi K., and Anton A. (1994), Inquiry-Based Requirements Analysis, *IEEE Software*, 11(2), 21-32.

Raghavan Sridhar, Zelesnik Gregory and Ford Gary (1994), *Lecture Notes on Requirements Elicitation*, CMU/SEI-94-EM-10.

Rauterberg M. and Strohm O. (1994), About the Benefits of User-Oriented Requirements Engineering, *Proceedings of the First International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'94)*.

Siddiqi Jawad and Chandra Shekaran (1996), Requirements Engineering: The Emerging Wisdom, *IEEE Software*, 13(2), 15-19.

Sommerville I., Sawyer P. and Viller S. (1998), Improving the Requirements Process, *Fourth International Workshop on Requirements Engineering: Foundation of Software Quality*, 71-84.

Sommerville I., Sawyer P. and Viller S. (1997), Requirements Process Improvement Through the Phased Introduction of Good Practice, *Software Process-Improvement and Practice*, 3, 19-34.

Sommerville I. (1996), *Software Engineering Fifth Edition*, Addison-Wesley.

Vliet Van Hans (1993), *Software Engineering – Principles and Practice*, John Wiley and Sons.

Wieringa R. J. (1996), *Requirements Engineering Frameworks for Understanding*, John Wiley & Sons.

Yu E. and Mylopoulos J. (1998), Why Goal-Oriented Requirements Engineering, *Fourth International Workshop on Requirements Engineering: Foundation for Software Quality*.

COPYRIGHT