CHAPTER 1: Introduction

1.1 Requirements Analysis

Software engineering, as defined by Ian Sommerville, is an engineering discipline where software engineers use methods and theories from computer science and apply this cost-effectively to solve difficult process (Sommerville, 1996).

Requirements play a vital role in the software development process. Before making a costly decision of "what to build", it is important that the requirements are understood completely (Sodhi, 1992). This process evolves from an initial statement of requirements for software engineering product to be completed. There is always a need to engineer system software that meets user requirements and expectations within available resources and to accommodate changes throughout the software life cycle.

1.2 Requirements Analysis Definitions

Institute of Electrical and Electronics Engineers (IEEE) Standard 610 (1990), defines software requirements as

- A condition of capacity need by a user to solve a problem or achieve an objective
- A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
- A documented representation of a condition or capability stated earlier.
According to Kramer (1988), requirements analysis is the most important step in system development. It is considered as the most critical task in software engineering. Another general definition of requirements and requirements analysis as defined by Leite (1987), requirements analysis is a process of elicitation and modeling of "what is to be done". This process has to deal with different viewpoints, and it uses a combination of methods, tools, and actors.

In a typical Software Development Life Cycle (SDLC), the requirements analysis process takes place after initial feasibility studies. It is a major stage in requirements engineering. Sridar (1994) has divided requirements engineering into four specific processes:

- Requirements Elicitation

  "The processes through which the customers, buyers or users of a software system discover, reveal, articulate and understand their requirements." (Sridar, 1994)

- Requirements Analysis

  "The process of reasoning about the requirements that have been elicited; it involves activities such as examining requirements for conflicts or inconsistencies, combining related requirements and identifying missing requirements." (Sridar, 1994)

- Requirements Specification

  "The process of recording the requirements in one or more form, including natural language and formal, symbolic or graphical representations; also, the product that is document produced by that process." (Sridar, 1994)
• Requirements Validation

"The process of confirming with the customer or user of the software that the specified requirements are valid, correct and complete." (Sridar, 1994)

Sommerville(1996) had defined requirements analysis as a process of deriving the system requirements through observations of existing systems, discussions with potential users and/or task analysis. He highlighted the important aspects of requirements analysis, which are:

• Requirements Analysis Sizes the Problem

Requirements analysis enables the software or system engineers to specify software functions and performance. It also assists them to identify interfaces between software and other system elements. Moreover, it helps to establish software design constraints

• Requirements Analysis Is a Process of Discovery

In addition to the previous mentioned characteristics, requirements analysis helps in recognizing problems, evaluating and synthesizing solution.

• Requirements Analysis Is a Process of Refinement

In requirements analysis process, the software scope is refined in detail in which alternative solutions are analyzed and allocated to software elements. A complete specification of software requirements is developed, as it is essential to a project's success
CHAPTER 1: Introduction

Generally, all software requirements analysis irrespective of methods or architecture practiced encompasses of four areas, which are (1) problem recognition, (2) evaluation and synthesis, (3) specification and (4) review.

After running through the different definitions and explanations of requirements analysis by various experts in the field, one cannot help but to conclude that if the requirements are done well, the software design flows logically and smoothly. Conversely, if the requirements are done poorly, the resulting design is awkward and the coding is more difficult. Usually, errors identified in the requirements stage are the fastest and least expensive to correct, while those found in later stages are increasingly more time-consuming and expensive to correct.

Consequently, requirements analysis is a process of developing clear, complete, agreed upon and feasible requirements for a product generally. The acceptability of the system after it has been delivered depends on how well it meets the customer needs and supports the business process of the client's organization. If the analyst does not discover the customer's real requirements, the delivered system is unlikely to meet the expectations.

1.3 Requirements Analysis Objective

The tangible result of a requirements analysis is a set of requirements that can be used by the software development team. Therefore, the objective of requirements analysis is to produce a well-defined set of requirements through involvement of various parties or participants in this process.
1.4 Requirements Analysis Benefits

It is a norm for clients or users of a software system that is to be developed to have a vague idea of what they really need and with little idea of what software technology might offer. This leads the developers to many assumptions. A good analysis process helps them to explore and fully understand their requirements.

Looking into the economic and financial importance of the requirements analysis, Boehm and Pappacio (1988) pointed out that high number of faults attributed at the design stage could have derived from requirements error. Earlier, Basili and Periccone (1984) reported that 48% of the faults observed in a medium scale software project could be avoided if not for the incorrect or misinterpreted functional specifications or requirements. Beizer (1990) in his publication cited that slightly over 8% of the faults in his samples could have been minimized if not for the incorrect, illogical, unreasonable, ambiguous, over specified, unverifiable, poorly presented and changed requirements. He also added that it is not unusual for a faulty requirement to get through all development testing, beta testing and initial field use, only to be caught after hundred of sites have been installed.

Thus, by active participation of both developer and end users, the requirements analysis process allows them to have a good understanding of the implications of the decisions they have made in developing the requirements. This results in fewer surprises when the system is built and delivered. At the end of the day, participants of a good requirements analysis process feels a sense of ownership of the product, satisfied with the process, feel informed, educated and believe their risk is minimize therefore commitment towards the success of the project is relatively higher (Sridhar, March 1994).
1.5 Requirements Analysis Difficulties

The requirements phase is not considered an easy process. There are several reasons for this. Among them are requirements volatility, requirements elicitation problems, language problems and requirements traceability problems. Sommerville (1996) have further elaborated the problems. According to him,

- Stakeholders often do not really know what they want from the computer system except in the most general terms. Even if they have a clear idea of what they want the system to do, they may find this difficult to articulate. They make unrealistic demands because they are unaware of the costs of their requests.

- Stakeholders in a system naturally express requirements in their own terms with implicit knowledge of their own work. Engineers, without much experience in the customer's domain, must understand these requirements and translate them to an agreed form.

- Different stakeholders have different requirements and they may express these in quite different ways. Engineers must discover all potential sources of requirements and discover commonalities and conflict.

- Analysis takes place in an organizational context. Political factors may influence the requirements of the system. These factors may not be obvious to the system end-users. They may come from higher management influencing the system procurement in ways that satisfy their personal agenda.
CHAPTER 1: Introduction

- The economic and business environment in which the analysis takes place is dynamic. It inevitably changes during the analysis process. Hence, the importance of particular requirements may change. New requirements may emerge from new stakeholders who were not originally consulted.

![Year 1982: Nine Contracts Totalling $6.8 Million](image)

*Figure 1.1: Results of GAO survey of software contracts (Sridhar, March 1994)*

Figure 1.1 indicates the importance of understanding the user’s requirements. Generally, almost none of the software purchased under these contracts could satisfy the user’s need.

A survey of 23 development organizations showed that projects in many applications have similar problems with requirements (Lubars, 1993). Among the problems are,

- Organizational solutions are more pertinent than technology solutions.
 Organizations under invest in support and education so that analysts cannot effectively use the tools they have.

 Many requirements are invented during product design. There is no definite source of customer requirements.

<table>
<thead>
<tr>
<th>Type of Failure</th>
<th>Process</th>
<th>Interaction</th>
<th>Expectation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of Systematic Process</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Poor communication between people</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of appropriate knowledge and understanding</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Inappropriate, incomplete or inaccurate documentation</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Poor management of people and resources</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is important for the requirements analyst to be aware of the possible causes of failure and must use techniques that help to avoid failure. Table 1.1 briefly outlines the link between possible causes with types of failure.

1.6 Easing Requirements Analysis Difficulties

Davis (1993) concludes that a perfect software requirements analysis does not exist and a perfect requirements analysis produces a document very large that it loses its conciseness. However, the requirements analysis methods must be able to generate, at
least, an acceptable set of requirements. Otherwise, developers do not know what to construct, customers do not know what to expect and there is no means to validate the system (Hsia, 1993).

Requirements analysis methods should be independent and provide structuring facilities to obtain more modifiable, traceable, annotated and organized requirements. Additionally, they should have a notation that favors the communication within the stakeholders.

According to Vanwelkenhuysen (1996), a quality or highly effective requirements analysis group decision-making must be implemented. Essential to group decision-making is that each participant has beliefs about (1) why a group decision is relevant to their concerns (2) how their actions influence the rest of the group and (3) the extent her concerns are addressed by the decisions.

In summary, requirements analysis is a total teamwork process. It requires active involvement of the stakeholders. In order to support their active involvement, groupware application or defined as a platform that provides the tools to support team activities is needed and the science that concentrates on building such tools is called Computer Supported Collaborative Work or CSCW for short.

1.7 Groupware

In a competitive environment that is global, intense and dynamic, the development of new products and processes is increasingly becoming a focal point of competition (Clark, 1993). Organizations earn more profit by getting to the market faster and more efficiently with products that are well matched to the needs and expectations of the end user. In order to cope with this competitive environment, many organizations are attempting to transform their structures and processes through teamwork, global integration and networking (Ciborra, 1993; Orlikowski et al., 1995).
CHAPTER 1: Introduction

This is where groupware comes into action. Groupware is a technology designed to facilitate the work of groups. The technology is used for communicate, cooperate, coordinate, solve problems, compete and/or negotiate in attending or completing a task. While traditional technologies like the telephone qualify as groupware, the term is ordinarily used to refer to a specific class of technologies relying on modern computer networks, such as email, newsgroups, videophones, or chat (Brinck, 1998).

CSCW refers to the field of study, which examines the design, adoption, and use of groupware. Groupware applications are computer-based system that supports teamwork in a common task or goal thus providing an interface to a shared environment (Ellis, 1991). Despite the name, this field of study is not restricted to issues of "cooperation" or "work" but also examines competition, socialization, and play. The field typically attracts those interested in software design, social and organizational behavior, including business people, computer scientists, organizational psychologists, communications researchers, and anthropologists, among other specialties.

One might ask how groupware design is different from traditional design. Groupware design involves the understanding of teamwork and how people behave in groups. It involves having a good understanding of networking technology and how aspects of that technology, for instance, delays in synchronizing views, affect a user's experience.

Many aspects of groups require special consideration. In other words, not only do million-person groups behave differently from 5-person groups, but also the performance parameters of the technologies to support different groups vary. Ease-of-use must be better for groupware than for single-user systems because the pace of a conversation often drives the pace of use of an application. System responsiveness and reliability become issues that are more significant. Designers must have an
CHAPTER 1: Introduction

understanding of the degree of homogeneity of users, of the possible roles people play in cooperative work and of who key decision-makers are and what influences them.

1.8 Research Motivation

Requirements analysis is a process involving stakeholders, from developer to end-user. A common platform that supports the activities is required to execute the processes. In other words, stakeholders involved in a requirements analysis process need a ground to get them involved, provide ideas, receive feedbacks, share opinions, review and make decisions. Therefore, a research on groupware supported requirements analysis tool was carried out. Being web-based, it fulfills the tools characteristic of being in a common ground and groupware supported to fulfill tasks on working together.

1.9 Research Objective

This research provided an environment for conducting requirements analysis process utilizing the web technology and platform. The objective of the project is to build a web-based requirements analysis tool that supports teamwork activities. The system enables requirements analysis process activities conducted via their terminals through the Internet. With these characteristics, time and space constraint is reduced if not avoided. The two main aims of this research are:

1. To build a Groupware Supported Requirements Analysis Tool that distributes the requirements analysis process among the team members. In the same time, it should also provide flexibility on time and place. This tool provides

   - A repository for projects as this would help to keep track of the projects
CHAPTER 1: Introduction

- Requirements traceability to see the owner of the requirements and the changes made on the requirements.
- Collaborative activities like e-mailing, sharing ideas and discussions
- Ability to express disagreements or raise issues
- Negotiating to resolve issues
- Traceability of requirements decisions
- Checking the completeness and consistency
- Sharing information
- Supports brainstorming session
- Supports classifications of the requirements

2- To evaluate the implemented system. The system was tested against the above objectives and against its features that was identified from this research.

1.10 Research Importance

This research carries its importance in providing to the needs of the software engineers to support the phases of requirements analysis. It is important for software engineers to interact with other stakeholders, achieving traceability as from the first document presented and managing more efficiently the problem of changing user requirements.

1.11 Research Scope

This research covers the application of web technologies and TCP/IP platform as a communication medium for the entire requirements analysis process, taking into
CHAPTER 1: Introduction

consideration for facilities for team cooperation and decision making. In addition, the flexibility of time and space is taken into account.

1.12 Methodology

A number of requirements analysis methods or architectures exists and being widely implemented. This research as mentioned earlier is to build a Groupware Supported Requirements Analysis Tool. The system was called as GRAT or Groupware Supported Requirements Analysis Tool. In pursuit of reaching this objective and scope, the following procedures are implemented.

- A literature review is carried out to evaluate current requirements analysis methods that are being implemented. This is done to get a proper understanding on importance of various processes in requirements analysis that varies from one to another. From here, it assists in selecting a proper requirements analysis method to be implemented in the tool. This step is considered the most crucial as it enables to determine positive and negative aspects of requirements analysis. An in depth study on groupware in general and how groupware is implemented in current requirements analysis tool was carried out too. The outcome of this literature review had assisted in successfully developing a platform that supports groupware activities in requirements analysis. Moreover, the technologies that are implemented are also reviewed in terms of capability and reliability.

- From the findings, the GRAT features are identified and the development process is executed. Analysis and design based on the requirements are done and documented.
Finally, the GRAT system is implemented and tested. The results of the testing is also captured and documented in order to be able to identify useful features and areas to be improved in future.

Figure 1.2 shows the methodology applied in this research.

The limitation of this methodology is only one requirement analysis approach will be selected from the literature review. GRAT will be built to support this approach. The tool might be too rigid to support other approaches in future.
CHAPTER 1: Introduction

Figure 1.2: Methodology applied for GRAT

1.13 Thesis Organization

Chapter 2: Review On Requirements Analysis Tools And Groupware

Groupware Supported Requirements Analysis Tool
CHAPTER 1: Introduction

This chapter discusses about the various requirements analysis methods and tools that are widely being used. Second part of the chapter looks into groupware and relates its support to some of the tools mentioned earlier. Finally a framework for GRAT is identified.

Chapter 3: Groupware Support For A Requirements Analysis Model

This chapter introduces the requirements analysis model. Looking into groupware: strategy and support, a groupware blueprint for GRAT is defined. The groupware activities and users and their roles for GRAT are discussed.

Chapter 4: GRAT Analysis and Design

This chapter describes the analysis and design of GRAT. It includes the requirements analysis of GRAT and object-oriented analysis and design of GRAT. For the sake of simplicity, the detailed analysis and design of GRAT in shown in Appendix A.

Chapter 5: Implementation and Execution

This chapter presents the implementation and how GRAT executes the requirements analysis process. It is divided by the different phases of GRAT. The image capture of GRAT is presented in Appendix B.

Chapter 6: GRAT Evaluation and Results

This chapter looks into the evaluation process and the results recorded from the pilot study.

Chapter 7: Conclusion
CHAPTER 1: Introduction

This chapter summarizes the content and contribution of the research and the thesis and also identifying some of the aspects for future work.