A Novel Framework for Requirements Elicitation in a Military Setting



Fernando Martins Muradas

St. Cross College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2012

A Novel Framework for Requirements Elicitation in a Military Setting

Fernando Martins Muradas
DPhil candidate, St. Cross College
Trinity 2012

Abstract

Due to their unique characteristics, military domains contain various peculiarities that directly or indirectly and favourably or unfavourably impact the development of software products. Observations of systems development in the Brazilian Navy for many years have shown that systems are developed and delivered usually with many delays, and during development there are many changes to the requirements initially elicited. Since many authors in the software engineering literature agree that problems in requirements account for more than 70% of total system development failures, it seemed to be logical that any effort to solve the problems encountered in the military environment should start focusing on the requirements area. Currently several techniques and approaches already exist that support the execution of activities in this phase. With this abundance of techniques, it becomes a difficult task for the analysts to select the best technique in a given circumstance. To support the selection of these techniques, several frameworks were also created, each one guided by a respective group of influencing factors. This study examined, in a detailed manner, these techniques and frameworks, and noticed that there are still issues to be addressed to guide the selection of elicitation techniques, especially in a complex environment such as the military one. To elicit these issues an exploratory qualitative research was performed. The results showed that social issues rather than technical ones were the main concern in the domain under study. Issues such as hierarchy and high staff turnover interfere in the requirements process but are yet not addressed. The findings of the qualitative research are the first contribution of the thesis. Based on these results this research proposed a novel framework for requirements elicitation within the context of a military environment taking into account social and technical factors, which is the second contribution of the thesis. Such a framework was developed following Conflict Theory from sociology. This framework organized a selection of techniques based on possible solutions to conflicts. Finally, the solution was evaluated to assess its efficacy. This evaluation was based on qualitative and quantitative research. Based on the evaluation results the framework was updated.

Acknowledgements

This research would not be possible without all the support that I received along the way. I would like to thank sincerely:

My wife Cristiane and my beloved step daughter Lara for all the support and the understanding of my absence.

My parents Alfredo and Carmen for always being there for me.

My brother Alfredo for being such a good example.

My supervisor Prof Marina Jirotka for the support, guidance and sympathy.

My assessors Prof. Janet Smart, Prof. Jim Davies and Prof. Gareth Digby for the support and guidance.

The Brazilian Navy for the opportunity.

All team members involved in the research at CASNAV (Brazilian Navy), CISCEA (Brazilian Air Force), CTEx (Brazilian Army) and participants from other countries.

The Brazilian Naval Attachés in the UK: CMG Kouri and CMG Jorge Antonio, and all their staff for the support in the UK and friendship.

My friends that supported me and helped me to get to the end.

My professors along the way for the grounding.

The Oxford University staff for the support and sympathy.

Finally, GOD for putting all these people on my way.

Contents

Chapter 1 - Introduction 1

Part I - Understanding the Problem 7

Chapter 2 - Literature Review 8

2.1 Requirements Engineering 8

2.2 Evolution of Elicitation Methods 16

2.3 Selection of the Elicitation Method 31

2.4 Summary 38

Chapter 3 – The Military Environment 40

3.1 Introduction 40

3.2 Characterisation of the Military Environment 41

3.3 Software Development in the Military Domain 45

3.4 Existing Military Frameworks 58

3.5 Summary 61

Chapter 4 - Exploratory Qualitative Research 63

4.1 Introduction 64

4.2 interview Results 67

4.2.1 Interview Analysis 81

4.3 Observation Results 83

4.3.1 Hierarchy 84

4.3.2 Multicultural Teams 86

4.3.3 Turnover Rate 88

4.4 Documentation Review Results 88

4.4.1 Hierarchy 88

4.4.2 Multicultural Teams 92

4.4.3 Turnover Rate 93

4.5 Other Institutions 95

4.6 Foreign Institutions 96

4.7 Conclusions of Part I (Chapters 2, 3 and 4) 99

Part II - Building the Solution 102
Chapter 5 - Conflicts in Requirements Engineering 103
5.1 Introduction 104
5.2 Groups and Conflicts 105
5.3 issues Related to Conflicts 124
5.3.1 Communication 124
5.3.2 Collaboration 127
5.3.3 Power and Politics 130
5.4 Existing Literature in Requirements Engineering Based on Conflicts 133
5.5 Summary 140
Chapter 6 - The Framework 142
6.1 Framework Structure 143
6.2 The Framework Provided for the Teams 165
6.3 Summary 173
Chapter 7 - Evaluation of the Framework 175
7.1 Qualitative Evaluation 176
7.2 Quantitative Evaluation 187
7.2.1 Quantitative Evaluation - Results 193
7.3 Revision and Extension of the Framework 199
7.4 Summary 206
Chapter 8 - Conclusions and Future Work 208
References 216
Appendices 233
Appendix A – interview Questions (Exploratory Qualitative Research) 234
Appendix B – interview questions (Evaluation Case Study) 237
Appendix C – Guidelines for Elicitation Techniques 239
Appendix D – Metrics Specifications and Procedures 269
Appendix E - Complexity Calculation (Use Case Points) 283
Appendix F - Informed Consent Statement 289

List of Figures

Figure 1 – Comparison between CMMI and MPS.Br maturity levels	**2**
Figure 2 - Waterfall Model, Spiral Model, Agile Model	**11**
Figure 3 - Increasing costs of repair	**12**
Figure 4 - A spiral model of the requirements engineering process	**14**
Figure 5 - Membership status of facilitator	**23**
Figure 6 - Communication challenges	**34**
Figure 7 - Details of elicitation activities	**35**
Figure 8 - The power culture	**42**
Figure 9 – The task culture	**43**
Figure 10 – The person culture	**43**
Figure 11 – The role culture	**43**
Figure 12 – Summary of the qualitative research methodology	**66**
Figure 13 – Client in a prominent position to reinforce the hierarchical difference	**84**
Figure 14 - Client discussing a questionnaire presented by the requirements team	**85**
Figure 15 – Requirements workshop with multiple ranks	**86**
Figure 16 – Multidisciplinary Team in a Project	**92**
Figure 17 – Turnover – Rate in a Project	**94**
Figure 18 – Main issues to be addressed and their consequences	**101**
Figure 19 – Factors involved in group interaction	**107**
Figure 20 – Classification of Group tasks	**108**
Figure 21 – Behavioural modes of tackling conflict	**118**
Figure 22 – Social network with sources of power	**132**
Figure 23 – Parallel Viewpoints of Easterbrook model	**134**
Figure 24 – The Dynamics of a conflict episode	**138**
Figure 25 – Focus of the existing research on conflicts and requirements	**139**
Figure 26 – Comparison between the focus of the existing research and the focus of this work according to the phases of conflict	**145**
Figure 27 – The framework - Classification of requirements elicitation techniques	**151**
Figure 28 – Requirements stages and activities, and the recommended facilitator	**154**
Figure 29 – Protocol analysis and observation performed in System 4	**182**

Figure 30 - Workshop with multiple ranks **185**
Figure 31 – GQM applied to the current research **189**
Figure 32 – Updated Framework **204**

List of Tables

Table 1 – Existing frameworks and the respective influencing factors **37**
Table 2 – Issues mentioned in the interviews / quantity of respondents that mentioned the issue **74**
Table 3 – Main issues mentioned by the executives in descending order of percentage **75**
Table 4 - Main issues mentioned by the managers in descending order of percentage **76**
Table 5 - Main issues mentioned by the analysts in descending order of percentage **77**
Table 6 - Main issues mentioned by the representatives in descending order of percentage **78**
Table 7 – Interview issues and the main related groups of issues **82**
Table 8 – Classification of courses in the Brazilian Navy **93**
Table 9 – Stakeholder analysis **148**
Table 10 – Group analysis **150**
Table 11 – List of stakeholders **166**
Table 12 – Stakeholder characteristics **167**
Table 13 – Group characteristics **168**
Table 14 – Summary of the guidelines **173**
Table 15 – Results of metric 1 **193**
Table 16 – Results of metric 2 – System 04 **194**
Table 17 – Results of metric 2 – System 05 **195**
Table 18 – Results of metric 3 – System 04 **196**
Table 19 – Results of metric 3 – System 05 **196**
Table 20 – Results of metric 4 **197**
Table 21 – Updated guidelines **206**

GLOSSARY

ACRE - **AC**quisition of **RE**quirements (Framework for requirements elicitation method selection)

Administrative System – System developed for administrative purposes

Agile - Agile methodologies (set of methods for system development)

CASNAV - Centro de Analises de Sistemas Navais (Naval Systems Analysis Centre - Brazilian Navy)

CISCEA - Comissão de Implantação do Sistema de Controle do Espaço Aéreo (Comission for the Implementation of the Airspace Control System – Brazilian Air Force)

CMG - Capitão-de-mar-e-guerra – (Rank equivalent to Captain in the Brazilian Navy)

CMM - Capability Maturity Model

CMMI - Capability Maturity Model Integrated

COTS - Commercial-off-the-shelf

CSCW - Computer supported cooperative work

CTEx - Centro Tecnológico do Exército (Army Technological Centre – Brazilian Army)

DARPA - United States Defense Advanced Research Projects Agency

Delphi - Method that allows a group of experts to make a decision

DGA - Direction Générale de l'Armement – Ministère de la Défense (French Ministry of Defense)

DoD - US Department of Defense

DODAF – US Department of Defence Architecture Framework

GPS - Global Positioning System

GQM - Goal Question Metrics paradigm

GSD - Global Software Development

HCI - Human-computer interaction

IBM - International Business Machines Corporation (American multinational technology and consulting corporation)

IEEE - Institute of Electrical and Electronics Engineers

ISO - International Organization for Standardization

IEC - International Electrotechnical Commission

ISO/IEC 15504 - A set of technical standards documents for the computer software development process

IT - Information Technology

JAD - Joint Application Development

LET - Lesson Elicitation Tool

LLS - Lessons Learned Systems

LTO - Laboratoires technico-operationel (Techno-operational laboratories – French Ministry of Defense framework)

MIL-STD-498 - Military Standard 498 (United States military standard to establish uniform requirements for software development and documentation)

MODAF - The UK Ministry of Defense Architecture Framework

MPS.Br - Melhoria de Processo de Software Brasileiro (Brazilian Software Process Improvement - the Brazilian Quality Model)

MUST - Requirements elicitation technique

NEPTUNE – Specific project in the Brazilian Navy

NGT - Nominal Group Technique

NIST - National Institute of Standards and Technology

Officer - One who holds a commission in the armed forces

OGSA - Ordenanca Geral para o Servico da Armada (General Ordinance for Navy Service - Brazilian Navy)

Operational System – System developed for military purposes

PCOM - Plano de Carreira para Oficiais (Career Plan for Navy Officers - Brazilian Navy)

PD - Participatory Design

PREview - An approach for requirements elicitation

RAD - Rapid application development

Rank - An official position or grade in the military

RDM - Regulamento Disciplinar para a Marinha (Disciplinary Regulation for the Navy - Brazilian Navy)

REng - Requirements Engineering

RUP - Rational Unified Process

SEI - Software Engineering Institute

Sergeant - a senior enlisted military rank

SSM - Soft System Methodology

ULRC - User led Requirements Construction (a method for requirements elicitation)

UML - Unified Modeling Language

XP - Extreme Programming

Chapter 1

Introduction

During my ten years working in system development in the Brazilian Navy it has been possible to observe at firsthand several problems with the military environment, initially as a developer and a system analyst. On many projects, difficulties have been experienced in eliciting requirements. Issues that consistently arise include keeping requirements updated; requirements changing without reasonable cause; strong disagreements among stakeholders; hierarchy influencing technical decisions, and representatives changing many times during development. The outcome was usually delay and sometimes even project failure. My interest in this area grew after taking a part-time MSc in Software Engineering, specifically in the software process area. After finishing the MSc, I became the manager of the Software Engineering Process Group in the Brazilian Navy. As the main task was to guarantee that all projects were following correctly the mandated development process, it was possible to observe closely all projects being developed. The Brazilian Navy is currently MPS.Br level F. The MPS.Br is the Brazilian Quality Model. It is based on ISO/IEC 15504 and also on the Capability Maturity Model

Integrated (CMMI). This means that the Brazilian Navy's development process is compliant with CMMI level 2, which equated to the MPS.Br levels F and G (Figure 1). The MPS.Br G level includes Requirements Management and Project Management while the MPS.Br F level includes Acquisition, Project Portfolio Management, Configuration Management, Quality Assurance, and Measurement.

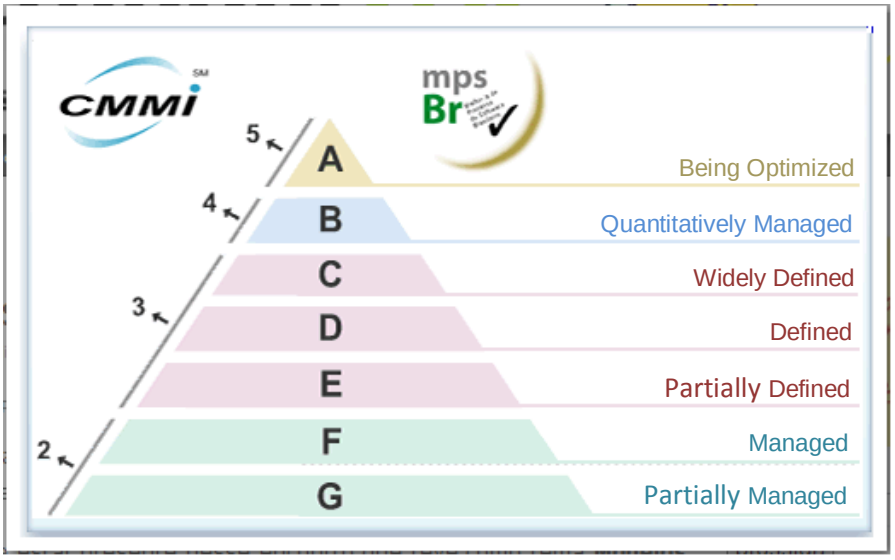


Figure 1 – Comparison between CMMI and MPS.Br maturity levels (Campos, 2012)

During this period I also became an MPS.Br Lead Assessor which allowed me to monitor other military and civilian institutions, including the Brazilian Air Force, and observe the problems that they encountered in system development. It became apparent that problems within this area were a common issue shared by all institutions observed. When comparing system development in civilian and military institutions, I became aware of how

some military characteristics strongly influence this activity. Consequently, I decided to undertake this research in order to search for a solution that could resolve these problems or at least minimize their influence in system development.

The problems observed encompassed all phases of the development which would be difficult to include in the period of the thesis. Thus, the decision was taken that the thesis should focus on a specific part of the development to make it feasible to cover within the period of research. Requirements engineering was the first phase and according to the literature is responsible for more than 70 percent of the problems faced by system development teams. Thus, if problems were to be solved in system development it seemed to be natural that the first step would be to reorganize the requirements area. Again even the requirements phase encompasses several tasks, which would still be unfeasible to cover fully within the available period. In this phase there is an extremely complex task that if not well performed can lead to the development of the wrong system or a system for the wrong purposes: requirements elicitation. Consequently, this thesis is focused on improving the specific activity of requirements elicitation in a military environment with the supposition that in doing so a large percentage of system development problems could be solved.

The first step was to study the requirements engineering area in general, and more specifically the requirements elicitation methods and selection frameworks that exist in the literature to support this task. A second step would be to understand the domain. The military environment was studied in accordance with the existing literature in this area. Thus, it was possible to categorise this specific domain and understand its general characteristics. Finally, it would be necessary to delve further into the military domain to understand which specific

issues related to system development should be addressed by a possible solution. To accomplish this, an exploratory qualitative research was performed. The focus was mainly on the Brazilian Navy, but also included Brazil's other armed forces, as well as personnel involved with system development in armed forces all over the world. Once these three steps were accomplished the understanding of the problem was complete, and a set of issues were defined to be addressed by the solution. These issues are the first contribution of the thesis.

The next step would be to start building a solution, the second contribution of the thesis. As long as several of the issues were social rather than just technical it would be important to ground the solution in some kind of social framework. The solution came from sociology. As it was observed that many of the problems were related to conflicts between people from different departments, it was logical that the thesis should use Conflict Theory as its base. The solution structured some of the existing methods around this theory. Thus, a framework was built and a guideline for the teams was developed. Once the solution was proposed, its efficacy needed to be evaluated. Consequently, a second qualitative research was performed. To deepen this evaluation as well as to assess the framework efficiency a quantitative research was also performed.

The research is organized into two distinct parts and nine chapters as follows. The first chapter is a general overview of the research as well as the organisation of the chapters. After the first chapter, the first part is presented composed of the second, third and fourth chapters, focusing on the issues to be addressed by the solution, thus searching for gaps and problems in the area and in the environment. The second part composed of chapters five, six

and seven discusses the proposed solution. Finally, the concluding chapter presents conclusions and future work.

A more detailed description of each chapter is given below:

The first chapter is a general overview of the research and the organisation of its parts and chapters.

In **the second chapter,** there is a literature review studying existing requirements elicitation approaches and techniques as well as the existing frameworks to support the selection of the appropriate elicitation techniques. This review aimed to discover which features the frameworks currently use to select the elicitation technique in each circumstance. Thus it was possible to conclude if military issues were already being addressed or not.

The third chapter presents a literature review of the military environment in order to characterise it. Furthermore, the chapter presents existing problems in this domain regarding software development according to the literature as well as the existing military frameworks. Studying the military frameworks it was possible to detect why those frameworks would not solve the problem identified by this research. However, literature in this domain is rare and often classified, especially in regard to problems.

Chapter four presents the first contribution of the thesis: an exploratory qualitative research performed in the Brazilian Navy in order to understand the main concerns of the military teams regarding system development and specifically requirements elicitation in a military environment. This research could fill the gap left by the lack of literature available. This qualitative research followed grounded theory (Corbin and Strauss 1990) and

encompassed interviews, observations and document analysis. To generalise and confront the issues gathered in the Brazilian Navy, personnel related to other military institutions, Brazilian and foreign, were interviewed. The results from this research crystallised the issues that needed to be addressed by the solution to be developed.

In **the fifth chapter,** a literature review presenting the grounding for the solution is presented. The concept of conflicts, Conflict Theory (Poundy, 1967), and how it is related to requirements are presented throughout this chapter as well as the existing research in the area.

In **the sixth chapter,** the framework itself is presented, the second contribution of the thesis. The framework was developed according to the Action Research approach (Reason, Bradbury, 2007), wherein opinions and comments from team members involved in the research are considered as inputs for the solution. First, the idea behind the framework is presented and later the framework as it is provided to the teams.

Chapter seven presents a new research performed to evaluate the efficacy and efficiency of the framework. The qualitative part of this research is based on interviews conducted with the teams after using the framework. To support the qualitative data and provide an idea of the solution effectiveness, metrics were defined and collected by the teams before and after the use of the framework. In addition, chapter seven presents an updated framework based upon the results of the evaluation case study.

Finally, **chapter eight** presents conclusions and future work.

Part I

Understanding the Problem

Chapter 2

Literature Review

This chapter presents a literature review looking at the area of requirements. The first section will present a general overview of the area. First, it will present the importance of the area to system development. Then, it will expatiate the process of requirements engineering and its steps from different perspectives. Finally, it will discuss the complexity of the area presenting some of the usual problems. The second section will discuss which methods and approaches exist to elicit requirements as well as their characteristics. Section 2.3 will present the current approaches to support the selection of elicitation techniques showing which factors guide this selection.

2.1 Requirements Engineering

"The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended. Broadly speaking, software systems requirements engineering (REng) is the process of discovering that purpose" (Nuseibeh, Easterbrook,

2000). Software systems are planned and anything that is planned has an intentional objective. According to Easterbrook and Nuseibeh (2004), "if a computer system is unsatisfactory, it may be because the system was designed without an adequate understanding of its purpose, or because we are using it for a purpose different from the intended one." Their study states that both issues can be attenuated by a meticulous diagnosis of objective during the system lifecycle. REng supports the understanding of the goals of a software system and how it can be used. Furthermore, REng takes the analysts from a blurred understanding of a problem that can be solved by a software system, to a clear view of a computer solution that will solve the problem (Easterbrook, Nuseibeh, 2004). There are several different definitions and perceptions of Requirements Engineeering. Davis (1990) suggests that "REng is the analysis, documentation and the ongoing evolution of both user needs and the external behaviour of the system to be built." Zave (1997) provides a broader definition: "REng is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families." The 'real-world goals' are the motivation for the system explaining why it must be built and what should be built. 'Precise specifications' are the guidance for building the system, also providing a way to validate it. It will guarantee that client needs are understood, developers will know what to do, and lastly it will be a way to prove what was done according to the needs elicited. Finally, it is mentioned that the "specifications evolution over time and across software families", show that requirements naturally will evolve during the development and there is the possibility of

using parts of previous specifications to support the modelling of any system (Nuseibeh, Easterbrook, 2000).

As with Zave (1997), Westfall (2005) discusses requirements in terms of What, Why, Who, When, and How. Requirements will describe the 'what' of a software product:

- *What the software must do* to add value for its stakeholders. These functional requirements define the capabilities of the software product.
- *What the software must be* to add value for its stakeholders. These non-functional requirements define the characteristics, properties, or qualities that the software product must possess. They define how well the product performs its functions.
- *What limitations there are on the choices* that developers have when implementing the software. The external interface definitions and other constraints define these limitations.

Business requirements describe the 'why' of the solution. They are organisational issues or possibilities to be addressed by the new system. The 'Who' are described as the stakeholders directly or indirectly involved in the system. Developing requirements includes several activities of collecting, understanding and validating the requirements, besides 'when' they should happen. Requirements will evolve and be modelled in various levels of abstraction in order to clarify 'how' each goal should be achieved (Westfall, 2005).

"Software development is a very complex, often distributed process with a variety of process and product attributes, as well as numerous functional and non-functional objectives and constraints" (Ruhe et al, 2002). According to Jirotka and Goguen (1994), usually several computer-based systems exceed the budget and schedule, and fail to meet all stakeholders' expectations. These issues led to a need of a more rigorous approach as an attempt to

increase the efficacy of the development process (Jirotka, Goguen, 1994). Due to this need systematic, disciplined and quantifiable approaches for development, operation, and maintenance of software were developed, and to evolve and better understand these approaches, there was the application of engineering to software. The Institute of Electrical and Electronics Engineers (IEEE, 2004) states that software engineering can be divided into ten subdisciplines: "software requirements; software design; software development; software testing; software maintenance; software configuration management; software engineering management; the software development process; software engineering tools; software quality".

With the aim of using engineering to develop software, several different approaches to the development process were proposed from the "Waterfall Model" defined by Royce (1970), the simplest and widely accepted software development model, to the "Spiral Model" and the "Agile Model" (Figure 2).

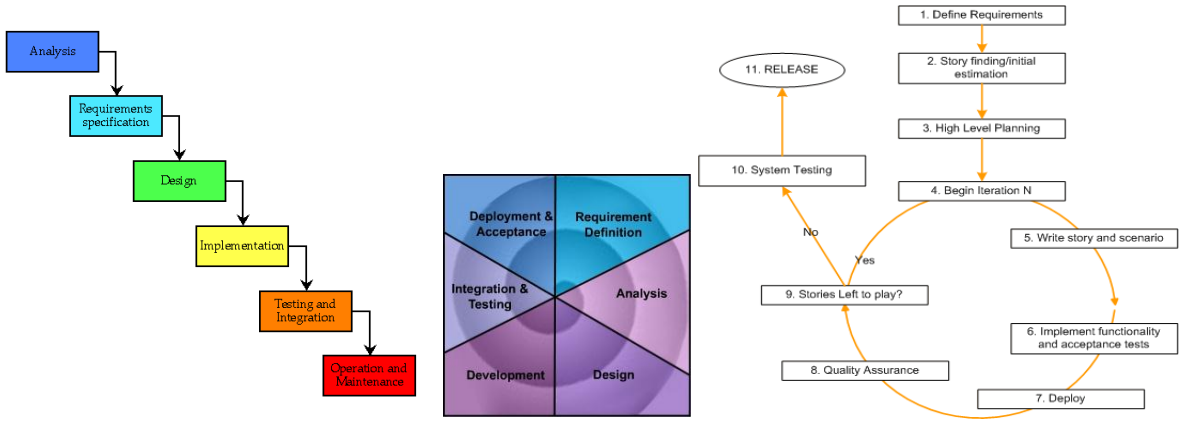


Figure 2- Waterfall Model, Spiral Model, Agile Model

Although there was an improvement on development processes, requirements engineering is still extremely complex and hard to manage, remaining one of the greatest sources of risk (Smite, 2006). Thus, naturally, the efforts to improve the development then

focused on requirements elicitation (Jirotka, Goguen, 1994). In 2006, Davis et al stated that "failure to accurately capture system requirements is the major factor in the failure of 90 percent of large software projects", echoing earlier study by Lindquist (2005) who concludes "poor requirements management can be attributed to 71 percent of software projects that fail; greater than bad technology, missed deadlines, and change management issues." Furthermore, "fixing mistakes made at the requirements elicitation stage accounts for 75 percent of all error removal costs" (Davey, Cope, 2008). Because all system developments start with requirements, any errors not discovered in this phase will be propagated to other phases. This inevitably leads to an increased cost of repair as noted by the National Institute of Standards and Technology (NIST, 2002) (Figure 3). Hickey and Davis (2003) go even further and state that "poor execution of requirements elicitation is a virtual guarantee of complete project failure."

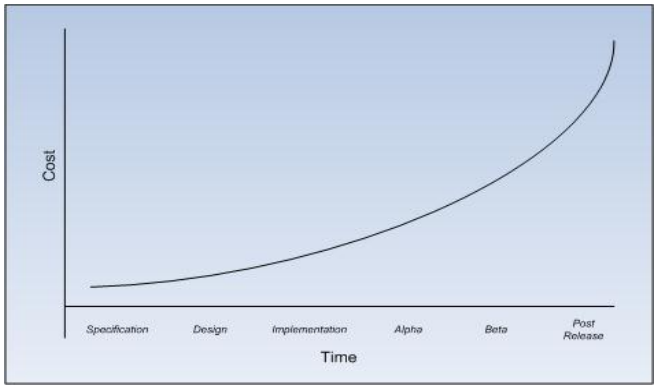


Figure 3 - Increasing costs of repair (NIST, 2002)

Usually, in the beginning of software development, requirements are vague and inaccurate, evolving continuously to more detailed and accurate requirements throughout the lifecycle (Ruhe et al, 2002). According to Ebert and Wieringa (2006), "REng is both a problem-oriented and a solution-oriented discipline. As a problem-oriented discipline, REng

interfaces with systems engineering in that it analyses the software problems that exist in the socio-technical system in which the software is to play a role." REng has to manage individuals with different expectations, as well as finding a solution within organisational needs and limitations. To achieve this goal, many times REng relies on other disciplines such as psychology and product management. Requirements analysts have to deal with different views of the problem as well as distinct prioritisations. Furthermore, usually clients do not have the right notion of the effort needed to implement each function of the system, besides all schedule and resources constraints (Ruhe et al, 2002). Cooper Jr. et al (2009), for example, suggest the use of visualization techniques such as bar graphs, pie charts, and hierarchical structures to get to a common view of the solution. "As a solution-oriented discipline, REng interfaces with software engineering in that it specifies the desired functions, quality attributes, and other properties of the software that is to be built or assembled" (Ebert, Wieringa, 2006). "The challenge is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints and preferences of the critical stakeholders are fulfilled, and the overall business value of the product is maximized" (Ruhe et al, 2002). "Because both views of REng (as both a problem- and solution-oriented discipline) are valid, REng is a discipline that maps needs to solutions" (Ebert, Wieringa, 2006).

The REng process is composed of several activities, including requirements elicitation, requirements analysis and negotiation, and requirements validation (Figure 4).

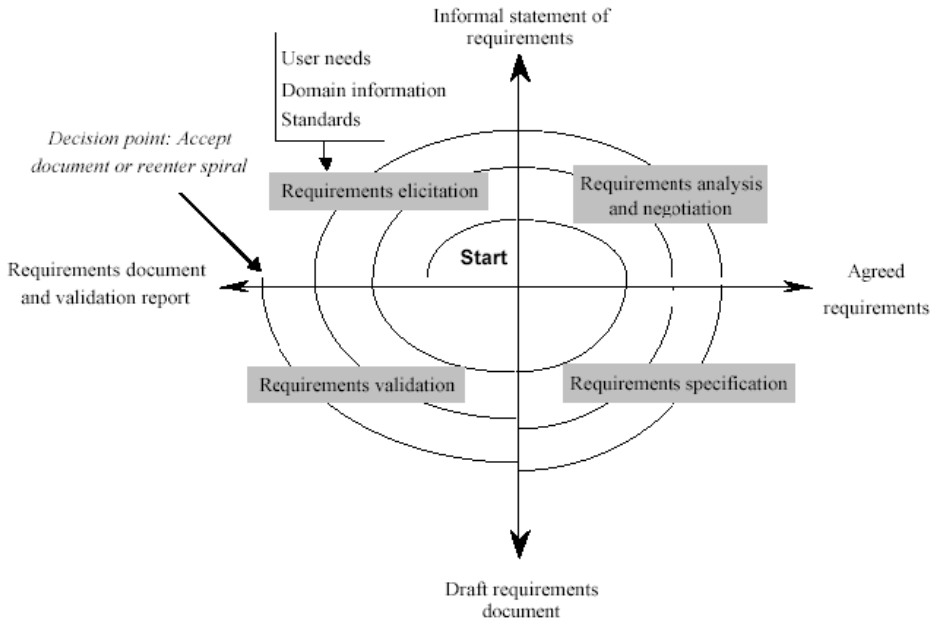


Figure 4 – A Spiral Model of the requirements engineering process (Lopes et al, 2006)

The complexity of this area can also be seen in the list below that shows some of the problems that may be encountered (Christel, Kang, 1992):

- problems of scope
    - the boundary of the system is ill-defined;
    - unnecessary design information may be given;
- problems of understanding
    - users have an incomplete understanding of their needs;
    - users have a poor understanding of computer capabilities and limitations;
    - analysts have poor knowledge of the problem domain;
    - users and analysts speak different languages;
    - ease with which 'obvious' information may be omitted or overlooked;
    - conflicting views of different users;
    - often vague and untestable requirements, such as 'user friendly' and 'robust';
- problems of volatility;
    - evolving nature of requirements over time.

To perform the vital system development process of REng while trying to avoid the extensive list of problems mentioned above, some novel techniques and methods have been developed. These methods are described and defined in theory, but it is still a challenge to put them into practice. Button and Sharrock (1994), for example, describe a certain project "Archer" that attempted to use "Yourdon Methodology" and during the development noticed that adaptations should be made to fulfil all project needs. The context in which REng takes place is usually a human activity system. Therefore, REng should be concerned about individuals' perceptions of the environment, personal interactions and to what extent the environment influences their activities (Nuseibeh, Easterbrook, 2000). That is, "the problems of requirements elicitation cannot be solved in a purely technological way, because the social context is much more crucial at this stage of development than in programming, specification and design phases" (Goguen, Linde, 1992).

The complexity of the area is highlighted by Button and Sharrock (1994) when they make the following observation: "in our attempts to familiarise ourselves with the work of software developers, we have become aware of how difficult, and often intractable, are the problems with which they contend." After studying the project Archer, in which several problems were faced by the development team, they compiled seven maxims intended to illustrate the difficulties in the area: "requirements are enmeshed in the organisational process; requirements are not objective, they come from a point of view; ending the specification of requirements for any development is a practical matter; any methodology is always a trade-off; no method is genuinely foolproof; the term 'a requirement' is effectively a swarm of changing contingencies; and that capturing a requirement is like capturing a

butterfly, once it's pinned down it's no longer what you chased for, it is now dead" (Jirotka, Goguen, 1994).

From what we have seen so far, most notably the strong influence of social issues, it is apparent that requirements elicitation is extremely complex and must therefore be approached systematically if a successful outcome is to be reached. This work will set out to examine in detail this task as it is performed on the system development process employed by the Brazilian Navy. It will take into account characteristics such as multidisciplinary teams and hierarchy that have an impact on this phase and influence teams in their choices of methods and tools. The study will focus on the task of Requirements Elicitation with the assumption that this is a critical first step in improving REng and consequently, the software development process.

2.2 Evolution of Elicitation Methods

All developers, when developing a software product, follow some kind of formal or informal process composed of a sequence of steps from the specification to the final code. This process is a set of procedures, methods and tools. This makes constructing software similar to constructing any other type of product (Wiegers, 1996). Processes are needed to develop quality products at each stage of development, not only in the coding, providing a systematic and a more predictable way of working.

Software appears increasingly as an integral part of all areas of human knowledge, either as an auxiliary or principal activity. With this growing profile, it needs some

formalization to support managers and analysts in building software solutions. Software engineering emerged in the mid-70s in an attempt to solve the software crisis and to apply the ideology and formality of engineering to the development of complex software systems (Leite, 2009). Once an engineering process was introduced in the design and coding phases it became clear that some formality was also needed in the requirements process to guarantee the control and predictability of the entire development process (Alves, 2009). This necessity led to the creation of a novel area of knowledge "Requirements Engineering", responsible for the development of new methods and techniques to deal with requirements (Jirotka, 2000). The REng process is not a singular task rather it is a composition of the following core activities: “requirements elicitation; requirements analysis and negotiation; requirements specification; and requirements validation” (Lopes et al, 2006). As software development evolved, for example, from large batch systems to object-oriented programming, so too did these techniques.

Several techniques for requirements elicitation have been proposed and introduced over time. First, there are the so-called classical or traditional techniques. Traditional techniques include a group of common and widely-used data gathering techniques. Examples of these techniques are questionnaires, interviews, and documentation analysis. (Nuseibeh, Easterbrook, 2000). With the advent of the Internet, a demand for rapid creation of efficient systems was created, and consequently, the need of new approaches for development (Miller et al, 2000). Thus, Reverse Engineering came among the traditional methods to base the development of a new software product on the analysis of the existing system. This process includes analysing existing documents, data, code and interfaces as a starting point to develop a new system.

According to Jirotka and Goguen (1994), these traditional techniques focus on the notation of requirements description, rather than the concepts and contexts of the information. Traditional methods tend to concentrate on the common questions of computer science, as social issues such as collaboration and conflict between stakeholders in the project are seen as uncomplicated (Jirotka, Goguen, 1994). These methods take for granted that it is an easy and straightforward task to get the core elements, their relationships and the processes involved, from the data gathered.

Other more complex approaches were proposed to improve the gathering of information. For example, Model-driven techniques drive the information elicitation through the use of a given model. Such an approach can support the analysts performing the requirements elicitation process in a more technical way. Goal-based and scenario-based approaches are examples of these techniques (Nuseibeh, Easterbrook, 2000). However, even improving the process of elicitation, they still lack of addressing social issues, just like the traditional methods.

With time, focus in this area changed from technical features to the concern with the user and the user's expectations. Following Easterbrook's study, from the psychology came the basis for a new group of techniques. Cognitive psychology brings a comprehension of the troubles individuals may face when describing their necessities (Posner, 1993). For example, domain experts usually have a large expertise that is not easily described; hence their responses to questionnaires and interviews normally do not reflect their real behaviour. Furthermore, the analysts sometimes have to gather user's preferences for interfaces rather than just building it based on their own expertise (Nuseibeh, Easterbrook, 2000).

Currently, there are several techniques based on cognitive engineering. Some of these techniques were originally developed for knowledge acquisition. These techniques include protocol analysis, laddering, card sorting, repertory grids, among others (Nuseibeh, Easterbrook, 2000). Protocol analysis, for example, is a cognitive technique that allows observation of the reasoning process of an individual during the realization of a task (Nguyen, 2007). This technique is widely used to make the evaluation of the usability of a system, when a user must narrate aloud each step of the activity being performed (Olmsted-Hawala et al, 2010). The information gathered during the narration of the activity by the user can guide the design (Lottridge et al, 2009).

Other cognitive methods relate requirements elicitation directly to knowledge and how this knowledge can be gathered, transmitted and fully apprehended. 'Knowing' is more than just a collection of organisational, task and personal knowledge but also the ability to analyse, adapt, transform and make this knowledge useful in a given situation. "The essence of knowledge is the human ability to act in a knowledgeable way" (Hsieh, 2006). Thus, culture and personality directly interfere in the way individuals exchange knowledge.

Cognitive techniques assume that domain expert usually have a large expertise that is not easily described, so a technique was created to try to solve the problem a different way. This technique is named User Led Requirements Construction (ULRC). It is a social interaction trying to solve the cultural gap between client and developer. Instead of trying to increase the ability of comprehending user needs, this technique passes the responsibility of building the requirements specification to the client itself. The technique aims to provide the ability for users to model their requirements (Flynn and Jazi, 1998). As mentioned by

Coughlan and Macredie (2002), the approach is composed of three phases. In the first phase users are trained by the requirements team on techniques to write a specification. In phase two, client and requirements team together create the model of the existing situation. This model can be used to guide the client in phase three on how to design the new features for the future system, in a process of several iterations (Coughlan and Macredie, 2002).

Another group of techniques is based on the notion that when stakeholders are gathered in groups, it is easier for them to explain their ideas, often evolving from the ideas of others. Group elicitation techniques aim to encourage collaboration among stakeholders, exploring the dynamics of the group to better elaborate the needs of the system. Among these techniques, there are brainstorming, focus groups, workshops, JAD (Joint Application Development) and RAD (Rapid Application Development) (Maiden, Rugg, 1996).

According to Hender and Rodgers (2001), with the evolution of the world with numerous and even faster changes, creativity has become a very important asset. There are methods that aim to increase creativity (Hender and Rodgers, 2001). Linus Pauling once said: "If you want great ideas, you need to have lots of ideas". Brainstorming is the best-known technique that enables the generation of many ideas (Goman, 2005). Brainstorming is a group technique created in the late 1930s to increase group creativity in an environment free of criticism.

Another group technique 'Focus Groups' is used for well delimited problems. Focus groups are widely used in market research as a group technique to get opinions and perspectives about a product or a specific situation (Zarinah and Siti Salwah, 2009).

JAD is a group technique created in the 1970s to raise business requirements. JAD gathers in a single group business and IT professionals during a workshop that aims to reach a consensus on requirements for a new system (Mathews, 2007). According to Jones (2000), well–run JAD-like (facilitated) workshops "still rank as one of the best methods for dealing with user requirements" (Gottesdiener, 2000). Due to the difficulty of gathering executives and the high level requirements discussed, JAD is normally applied in the early phases of a project life cycle (Mathews, 2007).

Several times clients are spread across more than one department or even in more than one organisation, when analysts have to deal with different perspectives of the work process. Checkland's (1988) Soft Systems Methodology (SSM) recognizes issues with conflicts of interest and negotiation and suggests guided mediation in meetings where stakeholders with distinct goals and impressions debate and hence increment their comprehension of the system to be built. SSM, created in the 1960s, has its grounding in systems theory. "Systems theory attempts to take a holistic view of the interrelations of component parts – the wider picture" (Rose and Haynes, 1999). Such a technique models human activities, leading to a better understanding of critical issues, thus facilitating the discovery of possible solutions (Morcos and Henshaw, 2009). The main advantage of SSM is that it can support attempts to obtain a mutual general understanding of distinct objectives in a common system (William, 2005). Consequently, in the case of requirements engineering it must be used mainly when there is more than one department or sector involved in a system development, and it is important to establish the boundaries of participation of each one. Also it can be used when there are conflicts of opinions about a process or workflow. Following Checkland, Easterbrook utilises concepts coming from social and organisational

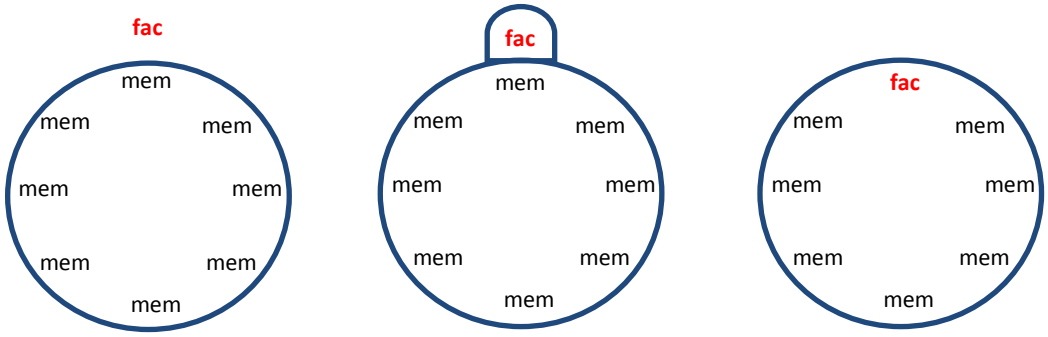
psychology to suggest that it is crucial to disclose and manage conflict in the requirements phase (Jirotka, Goguen, 1994).

All the group techniques mentioned have as a central point the fact that there must be a person responsible for guiding all members through the whole process of requirements elicitation (Viller, 1991). Furthermore, it is common sense that the success of group techniques relies strongly on the skills of that person. Redl (1942) invoked the concept of the facilitator as the core person who can organize the group emotionally so that there is a uniformity of intentions, and that the group organization occurs through this facilitator. According to the author, this facilitator is "someone who provides an object of identification, an object of drives, and ego support for the other members of the group" (Redl, 1942).

According to Macaulay (1998), it is well known that physical environment and technology normally do not interfere in the work of the facilitator. The issues that typically influence group work include "a combination of (a) lack of process or method, (b) interpersonal problems, and (c) political problems". Often times it is necessary to solve emotional problems or issues of political influence prior to organising the work process (Macaulay, 1998).

The facilitator can only concentrate on the control of the process with no influence on content, or be the work leader who controls the debates and organizes the necessary documentation (Crawford, 1994). Douglas (1970), notices three different forms of participation of the facilitator in the group (Figure 5). In the first the facilitator is a person external to the group that has control and power over it. In the second the facilitator works giving assistance to other members of the group without interference, so that they are able to

work effectively. In the third the role of facilitator is performed by a member of the group (Douglas, 1970).



1. Separate 2. Included but different 3. Full membership

Figure 5: Membership status of facilitator (Douglas, 1970)

To Heap (1968), there is an evolution of the role of the facilitator during the group work to adapt to the needs of each moment, due to the dynamics of any group work. To Douglas (1970), initially the facilitator plays the role of the core person. During the work, the group starts to create its own identity, and the members begin to see themselves as part of the group, when the facilitator becomes another member (Douglas, 1970). Close to the end of the work the centralization of control is once again needed to guide the group to meet deadlines and to terminate the work satisfactorily.

For some time systems were developed based mainly on the perceptions of the analysts with the notion that they, as software experts, would better understand what the needs of the systems were. With time and numerous failures came the idea that users should be part of the team that would elaborate the requirements for a new system. Thus, in the 1970s came the idea of participatory design, created in Scandinavia, with the aim of

increasing the participation of the employees on how a new technology should be developed and applied in the work setting, initially on factories (Kensing and Blomberg, 1998).

Prototyping is an approach based on participatory design, used to improve elicitation when there is a large amount of uncertainty in the information, or when it is important to have the opinion of the stakeholder in the beginning of the requirements process (Davis, 1992). Prototyping is a technique where the stakeholders can use a future version of a system to obtain some knowledge about using it in a setting close to the real one, and also to test new features (Preece et al, 2002). It uses the evaluation of the user to improve or test a product. It is easier for the client to understand the development idea or even to have new ideas when there is a starting point to stimulate it. Prototyping can also support several other techniques, for instance, as a starting point of a group technique or as a basis of a questionnaire (Nuseibeh, Easterbrook, 2000).

With the conclusion that organisations are composed of social and technical elements each one with its particular dynamics, but that these aspects are independent from each other came the socio-technical approaches (Jirotka, Goguen, 1994).

According to Jirotka and Goguen (1994) there are diverse ways that an approach designated 'Socio-technical' could address both social and technical issues. A first way to address both issues is to include a novel level of analysis to the current requirements approaches, incorporating social issues along with the existing technical ones. However, there is a difficulty to determine what those issues are and how to analyse them. A Participatory Design (PD) approach involves the user in the design phase in order to reduce these difficulties. However, analysts still struggle to make design choices that involve social issues. A third line of research states that it is not possible to separate social from technical

issues as they are intertwined, and thus suggests creating analytic classifications directly from the stakeholders (Jirotka, Goguen, 1994). These approaches are also known as contextual techniques due to their interest in obtaining a profound comprehension of the context in which systems are inserted.

In the 1990s these socio-technical techniques appeared as an alternative to traditional and cognitive approaches (Goguen, Linde, 1993). Numerous problems in software development made it clear that purely technical methods were not supporting properly the creation of software solutions. According to Dittrich et al (2009), traditional approaches were not taking into account all the complexity involved in software development as well as all the different points of view that take part in the solution. Furthermore, in his study Suchman (1987) states the importance of context in interactive systems, in order to adapt them to specific situations of use, which has been a problem according to social scientists. Thus, new approaches should be used, including qualitative research. Consequently, ethnographic observations and qualitative research could be useful to increase system development efficacy. In order to define conversation and interaction standards, these approaches also included ethnomethodology and conversation analysis (Viller, Sommerville, 1999). Ethnomethodology avoids understanding the work based purely on theoretical concepts, rather the work is observed in terms of its social organisation through stakeholder interactions. It focuses on the social structure of the work, as the activities are performed in the environment. To ethnomethodology tasks are performed according to the peculiarities of each situation, thus a given activity can occur in diverse ways, including having different outcomes for each situation (Martin, Sommerville, 2006). As long as Ethnography comes from the fields of anthropology and sociology, rather than any technical area of software

development, it is not as rigid as other techniques (Jirotka, Goguen, 1994). An ethnographic study focuses on the stakeholders and their interactions while performing the activity as well as the whole context where it is performed. To this approach understanding the activity and all the social interaction involved during the activity is essential for a clear comprehension of the environment. Such approaches can also address conflicts and their sources that happen during work activities in order to find possible solutions for these conflicts. Bartocci et al (2008) for example, observed ethnographic methods used in a hospital to gather data. The study developed a way of informing ethnographic results effectively to all stakeholders involved including users and the requirements team, in a way to increase client confidence and commitment in the new solutions, by making the client's staff directly report their findings and prioritisations (Bartocci et al, 2008).

Ethnography was incorporated as a technique to elicit requirements but was also improved with the introduction of technology to support it. Many researchers and practitioners perform ethnographies – that is, they observe the stakeholders while performing activities, many times with the support of video (Jirotka, Luff, 2006). The latter conducted a video-based field study in a dealing room. They noted that there is a "growing body of experience with video-based ethnographies on technological development in various domains, including air traffic and other control rooms, healthcare, public settings such as museums, and more experimental technologies, including media spaces and ubiquitous computing." According to them, video-based ethnography cannot be easily used with all requirements approaches, however to support the comprehension of a complex setting, it is unquestionably useful (Jirotka, Luff, 2006).

Corroborating with socio-technical approaches, Goguen and Linde (1993) comment that "the problems of requirements elicitation cannot be solved in a purely technological way, because social context is much more crucial than in programming, specification and design phases." Lately, sociological approaches have been increasingly used to develop interactive systems. Among these methods, ethnomethodology is the most popular (Dourish, Button, 1998). Since the acknowledgement that social issues are crucial to develop systems – mainly in the areas of computer supported cooperative work (CSCW) and human-computer interaction (HCI) – researchers have more understanding of social interactions in the environment that can improve the efficacy of the design, once the systems will be inserted and used in a social world. Currently, there is a common knowledge that any technical system is intertwined with the social interactions of the environment (Martin, Sommerville, 2006). CSCW has increasingly focused on the social interaction involved in using software systems, and created a framework in which sociology is directly used. However, after sometime, sociological views start to be more intertwined with HCI. The area of HCI, originally based on psychological and technical aspects, also began to incorporate sociological methods, which now became an integral part of this area of research (Dourish, Button, 1998). Dourish (1995) mentions the current HCI perspective, that comes from sociological studies, which observes the work as something that adapts to situations as they occur and thus depends on timely and local decisions. To Dourish, research based on anthropology or sociology can support studies grounded in other disciplines. This perspective has been raising debates about issues previously forgotten, such as the role of emotions in tasks previously considered cognitive (DePaula, Dourish, 2005) (Boehner et al, 2005).

Combining the advantages of participatory design and ethnography, Kensing et al (1996) developed a new approach: 'MUST'. This method gathers end users, managers and analysts, combining ethnographic methods and intervention. "Ethnography and intervention contrast in terms of their basic approaches and intended results: ethnographers originally strove not to change the phenomena they were studying, while interventionists deliberately set up activities to change the organisation in order to learn from the reactions to the change" (Kensing et al, 1996). Kensing et al (1996) argues that in practice, the combination of ethnography and intervention generates learning about the organisation while allowing the discussion of new proposals in situ.

A novel approach in relation to system development and consequently to requirements elicitation emerged in mid-90s with the so-called 'lightweight' software development methodology. These methods came as a reaction to the 'heavyweight' (Larman, Basili, 2003) or traditional methodologies, characterised by strong procedures and heavy documentation, and were labelled Agile Methodologies according to the 'Agile Manifesto' divulged in 2001 (Beck, 2001). According to the Manifesto, Agile Methodologies are a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through cooperation between multiple stakeholders, including client and development teams.

These methodologies are distinguished by practicality, PD and continuous iterations. According to Lindstrom and Jeffries (2004), software projects fail due to numerous motives: "not clearly communicating the requirements; the business problem is not solved by the requirements; the changing nature of requirements; incorrect requirements; and omitting

resources before fully understanding the requirements." A solution to these issues passes through increasing communication between users and the development team (Cockburn, Highsmith, 2001). These methods encourage less documentation and more face-to-face communication. In the agile method 'XP' (extreme programming), for example, requirements are described in Stories rather than UML Use Cases. Davies (2001) comments that the purpose of the XP stories is to allow developers to implement solutions iteratively rather than attending any formal process. These stories should be small to be implemented in a single iteration. Moreover, as requirements change constantly the ideal is that the descriptions of requirements are flexible and easy to update. These methods criticise Use Cases mentioning that they represent an abstraction of the system, bringing many technical terms and links to other UML diagrams, which are not clear to the user (Davies, 2001). In contrast, XP Stories do not have a formal structure. According to Davies (2001), another problem with Use Cases is that there is a lack of participation of the client in the phase of modelling these requirements, while in XP users are co-producers of the stories.

However, even advocates of Agile methodologies agree that this methodology does not meet every case. The key conception in Agile methods is that the team can be progressively productive when the time and cost of the interaction between people are reduced (Cockburn, Highsmith, 2001). For this purpose, the team should be co-located to enhance communication. They also note that agile methodologies can be used together with other existing system development practices (Paetsch et al, 2003). This is not a unique characteristic of Agile methods. Trying to preserve the best characteristics of several elicitation methods Christel and Kang (1992), propose to integrate techniques rather than just selecting one.

Just like system development itself, requirements elicitation approaches have been evolving over time towards the notion that it is not just a matter of gathering needs, rather it is a much more challenging task of understanding the environment where the system will be embedded including the relationships needed to perform each activity. Besides traditional techniques, there are techniques to understand the thought process of the client, methods that notice requirements elicitation as a group task, methods that include users in the requirements team, and methods that aim to understand the whole social process that takes place inside the client environment. Each approach was created to solve some kind of issue and thus is useful in some circumstance.

In conclusion, no method or approach will be applicable to all environments, teams or types of systems. Given the features and issues mentioned above and being cognisant of the large quantity of methods and techniques that are available, it is important that the team chooses the best method or technique in each situation in order to get the best understanding of the clients' needs. Aranda et al (2005), for example, mention that "since techniques employed during the requirement elicitation process influence the quality of the requirements, it is possible to improve the success of the products by improving the way techniques are selected." In each case there must be an analysis based on system, personnel and environment characteristics in order to select the best approach to elicit requirements. To support this selection many approaches have been proposed. The next section will discuss some of these approaches.

2.3 Selection of the Elicitation Method

Whilst various techniques exist for eliciting requirements, a critical issue for requirements engineers is how to select those techniques for a particular exercise. The aim of Requirements Elicitation is to collect the needs of the clients. However, to fulfil this purpose analysts face technical and social issues. To perform Requirements Elicitation, there are many methods available, each with its advantages and disadvantages. However, some techniques are often used incorrectly or in inappropriate situations, others simply are not used, and usually traditional techniques such as interviews, are frequently used. Analysts normally choose a specific elicitation method for any combination of the following motives: "(1) It is the only technique that the analyst knows, (2) It is the analyst's favourite technique for all situations, (3) The analyst is following some explicit methodology, and that methodology prescribes a particular technique at the current time, and (4) The analyst understands intuitively that the technique is effective in the current circumstance" (Hickey, Davis, 2003). As mentioned by Ayalew and Masizana-Katongo, (2008), the reason behind the difficulty of knowing which technique to use is the lack of information about existing techniques and situations where they would be appropriate.

For a long time, it was common knowledge that requirements engineering was performed for a particular user who was supposed to have clarity about what were his real needs. However, currently there is the notion that requirements engineering is performed in several domains, including products for the market or for customers who wish to expand some business. With all these distinct situations, diverse techniques were created: traditional techniques such as interviews and questionnaires, techniques focusing on the analysis of

conversation and interaction, and techniques that analyse the discourse structure as shown by Goguen and Linde (1993). Nuseibeh and Easterbrook (2000) comment that characteristics of the project will also influence the selection of the technique. Thus, one of the biggest challenges of system analysts is to choose the proper elicitation technique for a given system (Tsumaki, Tamai, 2005). With the vast number of existing survey techniques all requirements analysts should select and combine the appropriate techniques in each situation. However, most analysts always use their favourite techniques in any circumstance (Gotel, Cleland-Huang, 2009). In order to help this difficult choice Tsumaki and Tamai (2005) group the existing techniques into four areas: "Domain decomposition; Goal-oriented; Scenario-based; Brainstorming." After grouping they characterise these methods using two dimensions: 'static/dynamic' and 'open/closed'. The first dimension observes how the elicitation process itself should be performed, either focusing on static structures or dynamic behaviours; while the second focuses on the setting to be studied, either relatively closed or open. After the characterisation, they compare the features above with the project's prospective features to support the selection of the elicitation technique. Such an approach gives a broad indication of the method to be used; however, still relying severely on the analyst expertise.

Hickey and Davis (2003) present another method to support the selection of elicitation techniques. According to them, the selection of the technique should be guided by: "what requirements are known and, which are as yet unknown; the characteristics of the problem domain; the characteristics of the solution domain; and finally, the characteristics of the project." In their approach, they use the characteristics above in a formula to identify the best method for a given project. Similarly, Jiang et al (2008) suggest an approach to the

selection of REng techniques based on the attributes of the project under development and the specific domain. Their research uses attributes to classify elicitation methods to help in the selection process. The importance of these issues is unquestionable; however, their approaches lack any addressing of social issues that are equally important in this phase.

Ayalew and Masizana-Katongo (2008) propose a ' group decision support approach' for the selection of requirements elicitation methods. This approach is based on a decision-making process that involves a group of experts opting for a specific method or a group of methods that would be proper for a given situation. The 'group decision support process' is based on the 'Analytic Hierarchy Process' which stipulates a formal model for prioritising the selection aspects and the pertinence of the methods in relation to each aspect. The analytic hierarchy process will rely on the experience of the experts of the domain; however, without analysing the thought process of all other stakeholders.

Scogin (2006) related types of systems and factors that affect requirements elicitation. According to her study, the factors will influence more or less depending on the type of system. For example, domain knowledge is a factor that is more relevant in a web system than in a development based on commercial-off-the-shelf (COTS), while time and costs are extremely important in security critical systems but not in web systems. Interestingly, in all four types of systems analysed: COTS, Safety critical, Security critical and Web services, the relationship between the customer and the developer obtained the maximum value meaning a highly influencing factor to the effectiveness of the requirements elicitation task.

Aranda et al (2005) developed a cognitive-based approach to distributed requirements elicitation processes where they use personal characteristics to select the best

elicitation technique. They used some techniques from the field of psychology, known as 'Learning Style Models', to select groupware tools and elicitation techniques according to the cognitive styles of stakeholders. Aranda et al (2005) observed that knowledge flows in two directions: from clients to analysts and from analysts to clients, as shown in Figure 6:

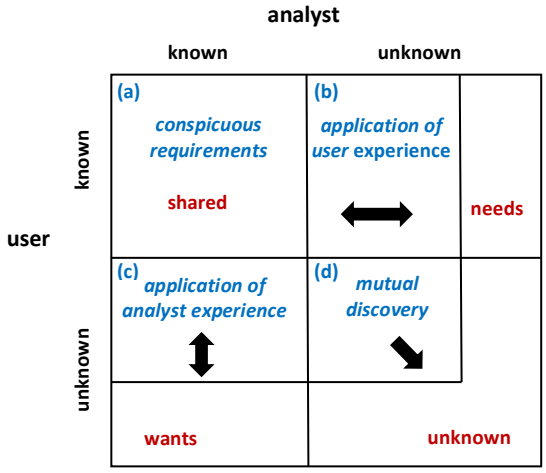


Figure 6 – Communication challenges (Davey and Cope, 2008)

To arrive at the proper technique to use, individuals first must be categorised, and this classification is used to help the elicitation technique selection (Figure 7). Their approach is focused on the relevance of communication during the requirements elicitation process in GSD (Global Software Development) projects. Communication difficulty is important to any kind of development rather than just distributed ones. However, this issue must be addressed together with other several equally relevant issues.

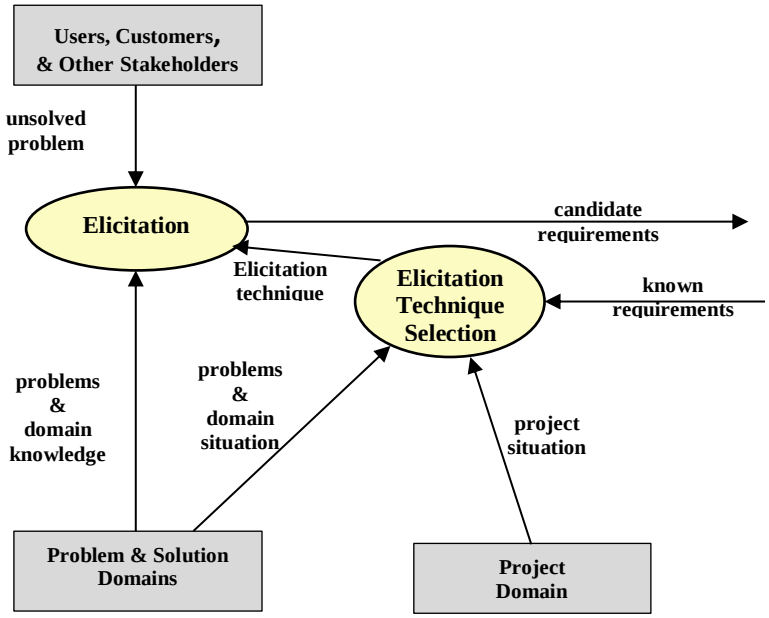


Figure 7- Details of elicitation activities (Hickey and Davis, 2003)

Liang et al (2009) studied several techniques according to what kind of support each one could provide for the requirements team. Once analysed according to their abilities these techniques were grouped. Finally, characteristics of the projects were correlated to project characteristics in order to select an adequate technique for that project.

Carrizo et al (2008) performed a very complete study on the existing literature searching for evidence that could prove the adequacy of several requirements elicitation techniques to a determined condition. In their study, they divide the factors in four areas: "elicitor, informant, problem domain, and elicitation process". Among the characteristics addressed by their study are: "number of informants, time allocated to elicit requirements, type of information involved, type of system, type of development approach, and experience of the analysts". Their research indicates when the studied techniques could be used, should be avoided, or there was no evidence to indicate either option. Although their study was very

broad and well developed, they discarded some aspects such as: personal factors, type of task and complexity of the domain. Besides, there was no specific guide for the technique, just a general indication of possible techniques, yet still leaving a great part of the task of technique selection to the analyst.

As noted by various researchers, because of the complexity of requirements elicitation and due to the direct influence of this phase on the project success, it is very important that it is performed in a very professional and conscious way. To guarantee a well performed requirements elicitation it is crucial that the appropriate method be used. It is very difficult to choose the right elicitation method or technique because this decision must be guided by several project and personnel characteristics. It is also important that the analyst notices that teams involved in system development are multidisciplinary groups of people with different backgrounds and, consequently there are social issues to manage. Consequently, to support this difficult and important choice, researchers developed numerous selection approaches based on several different influencing factors as seen in the table below. However, due to the several issues involved in this selection, the final choice is still alarmingly dependent on the analysts' skills and knowledge, and the results have been shown to be constantly inadequate.

Framework	Influencing factors
Maiden and Rugg (1996) "ACRE"	Number of participants Time allocated to elicit requirements Type of knowledge Size of requirements team
Davis et al (2006)	Types of systems Types of development approach Domain knowledge Project Characteristics
Scogin (2006)	Types of systems Domain knowledge Customer/developer relationship Developer skill Time Cost
Aranda et al (2005)	Stakeholders' preferences Cognitive features
Tsumaki et al (2005)	Project size Project structureness User task comprehension Developer task comprehension
Carrizo et al (2008)	Training in elicitation techniques Elicitation experience Familiarity with domain People per session Consensus among informants Informant interest Expertise of the informant Articulability of the informant Availability of time of the informant Location/Acessibility Type of information Availability of information Problem definedness Project time constraint Process time
Jiang et al (2008)	Project size Team size Requirements volatility Project category Degree of safety criticality Time constraint Product type

Table 1: Existing frameworks and the respective influencing factors

2.4 Summary

The first goal of this chapter was to give an idea of the motivation behind the thesis. Thus, the background of the researcher was presented as well as a generic view of the current development process of the Brazilian Navy. The Brazilian Navy was facing several issues regarding system development, and the consequences were delays and sometimes even project failure. As a lead assessor of the Brazilian Quality Model I was in the position to compare the system development in the Navy with the development in other Brazilian military and civilian institutions. Based on these comparisons it was possible to notice that some military features interfere directly or indirectly in the development success.

As authors agree that a large number of the problems faced by system development teams starts with requirements, this thesis focuses on the area of requirements. The chapter presented an overview of the requirements engineering area, highlighting the causes that make this one of the most challenging phases of the development process.

Once the motivation was understood the chapter focused on the evolution of the requirements elicitation approaches. Each new approach brought a group of techniques, and each technique has its goal and characteristics. Consequently, it is important to understand each one to know when they should be applied.

The next step was to present the existing frameworks to select elicitation techniques. With the large number of existing techniques, analysts struggle in the task of choosing the best one and consequently in eliciting requirements. To support the selection of requirements

elicitation technique to be used in each circumstance, several frameworks were developed. Each framework uses a group of features to support the selection of techniques that could or could not match the features existing in the environment under study. Consequently, there is the need of understanding better the military environment.

The next chapter will present a literature review of the military environment in order to understand it, focusing on similarities and differences from other domains. It will present the characterisation of the environment, some of its specific features, and the existing military frameworks.

Chapter 3

The Military Environment

This chapter will study the military environment, aiming to characterise it. After a general introduction the military domain is characterised according to Harrison (1972). This characterisation allows us to perceive general aspects of the environment. Once the domain is characterised we will study general aspects of the system development in this domain. From this, some general problems related to system development in the domain can be found. Finally, the existing frameworks for system development in the military domain will be presented with the analysis of their goals and characteristics.

3.1 Introduction

According to Goguen (1993), "a computer-based system is built for people and by people; it is intended to serve some human purpose, and it exists in some human context. Consequently, the most vexing difficulties are social, political and/or cultural" (Goguen,

1993). Currently, the majority of studies on requirements engineering focus on technical aspects while there are still many social issues to be addressed.

Furthermore, Button and Sharrock (1994) state the importance of the organisational culture on social and technical issues:

> *"To develop a system is not just a matter of following the method, but also to make this method work in the organisational environment. In these environments, many times technical issues are subordinated to organisational ones."*

Consequently, any selection of technique will need to take into account the environment where the system will be embedded and its characteristics, including social and political ones.

3.2 Characterisation of the Military Environment

According to Robbins (1998), "a group is more than the sum of its individual members, as well as an organisation is more than the sum of its member groups." Consequently, it is easier to comprehend an institutional comportment if we include some level of formality when studying individual and group conduct. There are several factors that influence organisational behaviour such as processes, rules and policies, organisational structure, and institutional culture. (Robbins, 1998).

Thus, to continue this research and find a satisfactory solution it is important to characterise the military environment according to its social structure and internal culture.

As mentioned by Hudlicka (1999), the military environment is unique and as such has several peculiarities. However, these peculiarities, although exaggerated in military institutions can be found in other institutions, and consequently, this environment can be categorised with other institutions. Harrison (1972) has studied organisational ideologies and categorised them according to the way the work should be organized; the way authority should be exercised; people rewarded; people controlled. According to this study there are four types of institutions in relation to their organisational culture (Handy, 1999):

- The power culture – it depends on a central power origin. Power and influence spread from the center into rays that lose strength as we head into the extremities (Handy, 1999)

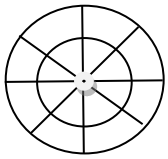


Figure 8 - The power culture (Handy, 1999)

- The task culture – this is a culture that focuses on the tasks and projects under development. The strength and power are usually in the knots of the net. This culture aims to allocate the person with the knowledge in the task to perform the task. Thus, the influence is based upon the expertise of the individual to perform activities. The focus is on a common goal rather than individual power or personal gains (Handy, 1999).

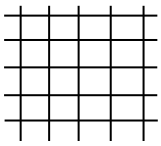


Figure 9 – The task culture (Handy, 1999)

- The person culture – this organisation focuses on individuals, and has a minimal structure. The members are together just to ease the way for their personal goals (Handy, 1999).

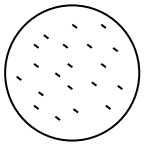


Figure 10 – The person culture (Handy, 1999)

- The role culture – that culture has as its main characteristic a strong bureaucracy. Its strength is in their bases and their predetermined functions. All the work and the relationships between the bases are arranged around formal rules and regulations with rigid definitions and procedures. There are also rules for the resolution of conflicts. The function to be performed is independent of the person that fills it, leaving the individual in a secondary position. Functional definitions are generic, so they can be performed by a variety of people. The power is based on positions and not on knowledge of the work (Handy, 1999).

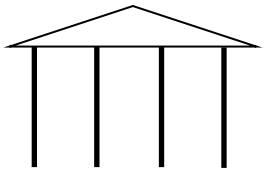


Figure 11 – The role culture (Handy, 1999)

Handy (1999) continues mentioning that a culture cannot be literally characterised as it is something that is noticed or felt by the workers. Thus, what can be characterised is the predominant culture in an organisation.

According to what was mentioned above a military environment can be strongly perceived as a role culture. In a military environment, there are written rules for every aspect of the job and even for the expected behaviour. The expected behaviour is related to the rank of the individual. In these institutions, there are books with all of the behavioural expectations of each role with the consequent punishment for any transgression.

> *"…an officer should always have so much in the way of pride that would never admit physical inadequacy until he dropped dead or unconscious"* (Goffman, 1961).

Another important aspect of a role culture observed in the military setting is that the power is in accordance with the position while the expert power comes in a second place.

From sociology comes a study from Goffman (1961) mentioning total institutions. This research considers as a rule in the society that people usually work, rest and have fun in distinct settings, with distinct authorities, dealing with distinct people and with no unique general plan. A total institution breaks this rule by having all these parts of an individual's life under a single structure (Goffman, 1961). Goffman (1961) mentions, among the total institutions: army barracks and ships. One important aspect of this kind of institution is that all situations of life are organized by the same singular authority. Furthermore, "authority in a total institution is directed to a multitude of conduct: dress, deportment, manners, that constantly occur and constantly come up for judgement" (Goffman, 1961). Although many

activities in armed forces take place inside offices, the principal activities happen in ships, barracks or airfields, and goals are all geared to these activities. Thus, all the behavioural aspects are brought from those places to all other tasks. Terms such as "on board", specifically related to ships are also used in Navy offices. Another study (Douglas, 1986) states that in any organisation particular problems are dealt with as if they were institutional ones, and if this institution relies on authority the solution will be based on increasing authority.

Hofstede (1991) studied power distance in several countries. Power distance is related to how people react to inequality in a society, and studies the perception of distance between superiors and subordinates. In his research, he concludes that this perception is strongly dependent on the national culture. However, after understanding the military culture as a role culture, with several distinct characteristics and with the hierarchy as one of its pillars, characteristics shared by these organisations all over the world, it is possible to infer that the military culture would be stronger than the national culture in the determination of the power distance in these institutions. Ratifying this statement, it is possible to observe a uniformity of behavior related to hierarchy and consequently, to all group tasks in these institutions independently of country or continent as will be shown in the qualitative research presented in the next chapter.

3.3 Software Development in the Military Domain

Certain domains have such specific characteristics that require the observation and sometimes adaptation of existing development system policies in order to achieve better

performance. These particular environments need a background study of their features to support the development of several systems. Members of these environments must determine the appropriate requirements elicitation technique or composition of techniques, according to their features, to gather the knowledge necessary in order to develop needed systems. Several projects studied the possibility of using concepts common to a specific environment to support the creation of specifications for novel systems in this environment (Jirotka, Goguen, 1994).

According to Jirotka and Goguen (1994), the efficacy of domain-based approaches within the area of requirements relies on a number of influencing aspects. There must be a group of experts in the domain with the knowledge to feasibly solve problems in that domain. This domain must have specific information complex enough to justify being reused in numerous different systems. Finally, this common information must be stored in a way to be easily shared with any analyst who needs it (Jirotka, Goguen, 1994).

Bolton et al (1994) present an approach in which a domain is defined to be a group of the so-called 'situated systems' already developed for a specific knowledge area. The term 'situated system' includes the system itself together with the environment where it is inserted. In their approach, the system is composed of elements that relate to one another to achieve common goals, while the environment is defined by other elements that provide information to or receive information from this system, however, being independent from the system (Bolton et al, 1994).

In the study of complex environments, the use of ontology (a concept's specification) as a means of addressing these issues has gained some devotees. "Ontologies in general, are

a shared understanding (semantics) of a certain domain, axiomatized and represented formally as a logical theory in a computer resource" (Jarrar, 2005). Ontologies can make the specific concepts of a certain environment available and understandable by any member of an information system development team. Furthermore, the domain concepts inside a given information system can be replaced by the concepts of another domain, making this system portable between domains, especially if the system is strongly parameterized. However, the ontology itself has its own lifecycle and demands a team to keep it updated and thus useful, making an ontology-driven development approach extremely costly and complex (Yildiz, Miksch, 2007). As critical challenges that face the endeavour of the ontology development lifecycle, Jarrar (2005) mentions: "the influence of a specific purpose (what it is made for) on the ontology developer; the difficulty of identifying and isolating the reusable components (allowing the reuse of the general-purpose parts of an ontology); different usability perspectives (different purposes of what an ontology is made for and how it will be used) lead to different or even to conflicting axiomatizations, although these axiomatizations might intuitively be in agreement at the domain level; and the continuous growth and intensive maintenance of emerging ontologies currently (and for the immediately foreseeable future)."

As mentioned above, when there are several systems in the same environment, it is strongly recommended that any previous knowledge of this environment, and also about the existing systems, be registered and organized in order to facilitate and standardize new systems to be created. This will also help new team members to understand faster the main issues to be addressed in a specific environment.

Among the existing specific environments, there is a significantly complex one: the military domain that is the object of this research. In this environment, one encounters many of the most challenging issues in the market, such as multidisciplinary teams and global software development, and there are other special characteristics that maximize the complexity of these issues, (for example, a large amount of confidential information) also including other new issues that significantly influence the development of systems, such as hierarchy and high personnel turnover. The military domain and the related term 'military software' refers to software produced for a uniformed military service such as an Army, Navy, Marine Corps, Air Force, Coast Guard of any nation, also including software produced for the Ministry of Defense or equivalent in each country (Jones, 2000). Comer et al (1990), for example, state that the military environment is very peculiar, and that there is the need for a careful analysis of all the previous consequences of introducing new systems or technologies, as well as adaptations needed to be in conformance to this specific reality. An environment with such specific features leads to the need for a 'translation' from its terminology to common language. As noted by Isaac (2010), people without any contact with this environment or its personnel will usually misunderstand the terms and acronyms used, which would lead to the necessity of some translation into 'civilianese'. Weber and Aha (2002) specify the need for a 'disambiguation method' in military systems, as well as other technical and specific domains (such as medical or financial), in which several terms used daily can have more than one meaning, making it difficult to use the system across similar subdomains such as the Navy and Marine Corps.

"The military domain includes almost all types of application, but it is still homogenous from user and regulation points of view" (Codur, Dogru, 2009). These systems

vary from very common applications virtually identical to their civilian counterparts such as payroll and personnel information to extremely specific ones, such as software associated with weapon systems; command, control and communication systems; and logistical applications.

This complex domain employs a large percentage of all the workers in the software area. In the United States (US) in 2000, for example, the military software domain employed around 16 percent of the total software employment force (Jones, 2000). Similar numbers can be found in many other countries all over the world. While most countries opt to buy their military packages from specific countries with a high expertise in the area such as the China, France, Germany, Japan, Russia, United Kingdom, and the US, some other countries such as Argentina, Brazil, India, Israel, Mexico, Taiwan, the Ukraine, and both South and North Korea still develop their own military software (Jones, 2000).

Codur and Dogru (2009), mention situations where a military system was developed over almost 30 years, which shows, as a peculiarity of these systems, that they tend to have a long time of development. Another strong characteristic of this domain, including system development, is that all activities are normally regulated by norms and standards; some of them are quite old such as MIL-STD-498 (Military Standard 498) created in 1978 (Codur, Dogru, 2009). However, there is a major problem in this domain related to these standards. They are so complex that the productivity of defense applications is lower than any other software subindustry (DoD, 1987) due to the enormous amount of documentation generated on military systems, which is up to three times that of comparable civilian projects. It is not uncommon for the management of paper documents to account for up to 50 percent of the

total costs of large defense projects (Jones, 2000). Heavy regulation inhibits the use of standard commercial practices in software development and also encourages the building of new software rather than purchasing items in the market (DoD, 1987).

Hierarchy is one of the pillars of this complex domain. Jones (2000) describes the hierarchical organisational structure as the usual practice for military software in contrast to the matrix organisational structure, normally found in civilian companies in the software area. In this type of structure, each manager reports directly to a higher-level manager, up to the level of the senior manager responsible for the entire system. However, according to Jones (2000), productivity decreases in the same rate that the quantity of management levels increases, which shows another reason for software problems in this domain (Jones, 2000).

The US DoD (1987) states that the main concerns in the military environment are not related to technical issues and are normally referred to norms, personnel, and skills among others. An example of an issue inherent in this environment is a strong political and hierarchical influence in technical decisions as evidenced by this statement regarding problems in the US Department of Defense: "If the military software problem is real, it is not perceived as urgent by most high military officers and DoD civilian officials" (DoD, 2000).

Searching for an improvement in software development in this domain, in 1985 the US Department of Defense funded the Software Engineering Institute (SEI) to study ways to add formality to software development practices in order to decrease the quantity of failures in software systems. Thus, SEI created the Capability Maturity Model (CMM) (later updated to the CMMI - Capability Maturity Model Integrated), a model of software development good practices together with an assessment method (Jones, 2000) that would allow the

verification of the software process maturity of the suppliers of the US Department of Defense. In 1994, the US Secretary of Defense, based on the low productivity of the domain, decided that the armed forces would adopt civilian best practices instead of heavy military standards. To the DoD (2000) following commercial practices is mandatory as "technology, particularly technology relating to software, is changing more rapidly than ever before, and there is no doubt that the commercial marketplace, not the DoD, is in the driver's seat". However, as Jones (2000) mentions, the military environment tends to be very conservative and resistant to change, especially adopting civilian practices: "we don't do it that way." A task force gathered by the US DoD reviewed in 2000 "six major DoD-wide studies that had been performed on software development and acquisition since 1987. This task force was created to address the problem of defense development and acquisition programs continuing to experience 'software problems'. These problems have resulted in significant cost overruns, schedule slips, and performance difficulties. These studies contained 134 recommendations, of which only a very few have been implemented. Most all of the recommendations remain valid today and many could significantly and positively impact DoD software development capability" (DoD, 2000). The task force concluded that a back-to-the-basics approach would be needed, and would provide "significant improvement in the performance of software intensive DOD programs" (DoD, 2000). However, military organisations are not-for-profit, thus, if essential mission requirements are attended, there is no motivation to improve the productivity, especially exchanging military practices for civilian ones (Jones, 2000).

In the same period, SEI recognized the importance of personnel skills and human relations and included these issues in CMM. Common characteristics in the military

environment are salaries under the market and no concern about living conditions in the offices like ergonomics. Other issues within the defense industry are teams composed of civilian and military (DoD, 1987); project managers with lack of knowledge in software engineering (Gotterbarn, 2002); and difficulty in retaining specialized personnel (Jones, 2000). "With limited ability to rapidly change worker compensation, it is increasingly difficult for the DoD to hire and retain software talent" (DoD, 2000). Consequently, a group of experts in software hired in the market in 1995 by the US Navy called the "Airlie Council" recommended to "treat people as the most important resource". The Airlie Council included the most famous specialists in the area such as Dr. Victor Basily, Dr. Barry Boehm, Dr. Tom McCabe, Ed Yourdon, Larry Putnam, Tim Lister, Tom DeMarco and Capers Jones. The fundamental practices defined by the Airlie Software Council of the US Software Program Managers Network were still not implemented in 2000 (DoD, 2000). Military software project managers in the area of Defense live within the budgets limited by government decisions. Moreover, military software project managers usually do not know how to deal with human issues that can affect the morale such as office and workstation conditions, as well as ergonomics. Although in both battle situations and software projects good morale increases productivity, this improvement continues to be constrained by the inability of leaders (Jones, 2000). There is also a lack of specialized courses for training in military software with its peculiarities and the project managers in the military environment are commonly known to lack software engineering knowledge. DoD (2000) mentions that "inexperience and/or unqualified personnel at all levels are a major contributor to DoD software problems. This is particularly disturbing since software frequently constitutes a majority portion of DoD program budgets and risks." Further complicating the situation, it is

usual to utilise personnel hired in the market for a percentage of military software technical work because contractors can be brought in fairly quickly, and have relevant experience suitable to the projects in need of support.

A more sensitive issue has been the fact that compensation and benefit levels within the defense contractor community have lagged behind the civilian sector for software personnel. This salary difference is one of the reasons that make several experts in the domain look for better gains in the civilian world. In the US in the year 2000, for example, while compensation for project managers in the defense sector was US$75,200 per year the average in the computer sector was US$108,500 per year (Jones, 2000).

Other issues that are increasing the complexity of the requirements process can be mitigated due to the specific features and yet the specific culture intrinsic to this environment. Recently, many companies are expanding their market to other countries, as part of the worldwide globalisation process. Thus, these companies are creating distributed structures to support this new business, including geographically distributed teams located in offices around the world, and, sometimes, teams from other organisations. Although there is evidence of global teams that work properly, there is a consensus that distance can increase organisational difficulties. Several aspects can be affected by geographical distance such as communication and management, which directly influences the way requirements are elicited and modelled, as well as the way development is controlled and validated. It seems natural that GSD be a significant concern for the armed forces, as they have many organisations all over the country and representatives all over the globe, as well as the need to perform joint exercises and activities with foreign armed forces with common objectives. Currently, there

are many international systems being developed that affect more than one country such as systems that work with the safety of life at sea. Crebolder et al (2009), for example, comment that "the nature of Canadian Forces operations is fast becoming one that involves joint operations, multi-national coalition forces, and inter-agency organisations working together within a networked environment." However, Hargreaves and Damian (2004) observe that there are military characteristics that minimize the negative effects of GSD since military organisations deliberately create a different subculture to make viable communication and reduce possible conflicts between individuals from distinct countries and cultures, such as the Military Phonetic Alphabet. Their research even suggests that civilian distributed teams should adopt certain postures inherent in the military environment.

In order to source solutions it is the best practice in the market to have performance appraisals for both technical staff members and project managers. Usually after these appraisals, there should be an agreement by employees and managers about goals and targets. Military organisations do perform these appraisals; however, consensual agreements are difficult to implement in a rigid hierarchical organisation (Jones, 2000).

With several issues and problems regarding the military domain, it is important that the sources of problems are identified. Perrow (2008) states that "in the military, specification errors are much more common than operator errors." According to him, the development team struggles to foresee all possibilities of use for the system, especially operational systems, as these systems deal with war situations (Perrow, 2008). In agreement Grudin (1994) states that due to the impossibility of predicting all possible use situations of a system, a solution will never resolve all problems in the setting. Dittrish (1998) for example,

proposed the ‘observation of unanticipated use’ arguing that computer systems depend also on the interpretation of the situation by the analyst. Defects in requirements or in specifications can lead to several types of failures, sometimes resulting in fratricidal engagements (Nitzberg, 1999). A real example of system requirement failure is presented by Jackson (2004): during the Afghanistan war a handheld GPS receiver communicator was being used to set a distant target for an air strike, but during its operation the equipment, which was already set for the target, showed a low battery warning message. Due to the battery change, the equipment reset to its present coordinates instead of the target coordinates, previously set by the operator. This change went unnoticed by the operator who gave the order to attack, which resulted in the deaths of himself and his entire platoon. The software should never allow the operator to set his own coordinates, but this situation was not foreseen during requirements elicitation (Jackson 2004).

According to DoD (1987), the work of specification is essential, and yet it represents a small percentage of the total cost of a system. The DoD (2000) states that the observation made by the 1987 DoD task force that “requirements-setting and management are the hardest part of the software task” is still valid (DoD, 2000). Thus, any increase in the effort allocated to the requirements phase will increase development efficacy without many increases in the cost. The most challenging task of the software development is the definitions of the requirements, including non-functional requirements, and the priorities of different requirements (DoD, 1987). Mistakes in requirements can strongly affect effectiveness, budget, and schedule. It is common in software development to over specify some functions which the influence on performance will be noticed just in posterior phases. Another common difficulty is to design the user interface. DoD states that clients are not able to

define the requirements without testing the system in an environment close to the real one, and without participating iteratively in the specification. “The systems built today are just too complex for the mind of man to foresee all the ramifications purely by the exercise of the analytic imagination” (DoD, 1987). As a possible solution to specification problems DoD (1987) suggests two separate contracts, a first one to perform the specification, properly validated by prototypes, and a second one to develop the system (DoD, 1987).

The peculiarities of the military environment are discussed by Weber and Aha (2002) who developed a tool, LET (Lesson Elicitation Tool), which uses domain and linguistic expertise to guide users to store and disseminate lessons learned in the military field. Their tool is categorised as Lessons Learned Systems (LLS), which are "knowledge management initiatives for supporting the capture, validation, storage, distribution, and reuse of organisational lessons." This kind of tool is especially useful in a military environment where there are lessons learned stored for several years in all its areas. Their solution is composed of numerous domain entities, which makes it a domain-specific solution that was built based upon the identification of patterns through the observation of several lessons collected in armed forces (Weber, Aha, 2002).

Hudlicka (2009) studied knowledge elicitation in complex military environments. The author mentioned that in knowledge elicitation, there are theoretical and practical challenges that affect this elicitation. The theoretical challenges include “selecting the best structure and level of specificity for the interview process; matching the elicitation method to the desired knowledge type, function, and formalism; accessing intuitive and compiled knowledge, which is difficult to articulate; accessing and representing uncertain and

idiosyncratic knowledge; and selecting the best representational formalism for the elicited knowledge." Among the practical issues, the author includes: "subject matter expert selection; conduct of the interview sometimes using multiple experts or multiple knowledge-engineers; preparation and 'homework' such as obtaining the necessary background material to maximize the productiveness of the expert interviews; recording and data analysis with decisions of what data to record and how, and selection of the best method for data analysis; and selecting the appropriate computer tools for administering the knowledge elicitation sessions, and for recording and analysing the data." In addition to the issues related to knowledge elicitation mentioned above by Hudlika (2009), there are also issues related to the specific environment. The military environment is mentioned due to its complexity and consequently, the theoretical and practical issues that must be confronted. From the theoretical perspective, this is due to "the large quantity of information and knowledge that must be processed; the complexity of this knowledge; the challenge of eliciting and representing uncertainty in knowledge and reasoning; and the considerable difficulty of eliciting the largely intuitive knowledge that underlies much of the more complex tactical battlefield decision-making." From the practical perspective, this is due to "the difficulty of finding and accessing the domain experts; issues with classified information; and the difficulty of conducting elicitation in the actual performance environments." As it can be seen from the work above, there are numerous issues that make this environment such a unique one. Thus, Hudlicka (2009) ends mentioning that no technique will be effective alone and recommends the use of a set of techniques to gather the necessary knowledge in this kind of environment.

The variety of systems existent in this domain and yet the urgency of developing systems and devices which are increasingly powerful makes the military domain a major exporter of technology and development standards. As Nitzberg (1999) noted, the Global Positioning System (GPS) was developed by the US military to support its missions and now is incorporated into many civilian devices; the Internet itself, was developed under the patronage of the United States Defense Advanced Research Projects Agency (DARPA) (Nitzberg, 1999); and in the Software Engineering field, one of the most widespread process improvement approaches in the market, CMM, was developed in response to a request from the US Department of Defense to SEI.

3.4 Existing Military Frameworks

The direct or indirect influence of computer systems in all areas of knowledge and even on business success grows continuously. More and more executives understand that the success of their business depends on software support. In the military domain, that holds an important part of the benefits involved in software, IT relevance has also been understood. This acknowledgement is perceived by the complex and expensive existing frameworks created for the armed forces of some developed countries.

In 2003 the United States of America developed the DODAF (Department of Defence Architecture Framework) based on the IBM architectural framework (Zachman, 1987) for organizing architecture models in an institution. The goal of the DODAF was "to provide a

foundational framework for developing and representing architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures across organisational, joint, and multinational boundaries" (DODAF, 2009). The DODAF is composed of eight viewpoints: All Viewpoint; Capability Viewpoint; Data and Information Viewpoint, Operational Viewpoint; Project Viewpoint; Services Viewpoint; Standards Viewpoint; and Systems Viewpoint. Each viewpoint provides a different aspect of the capability.

Based on the DODAF, the Ministry of Defense of the United Kingdom developed the MODAF (the Ministry of Defense Architecture Framework), an organisational architecture that provides a formal structure for the collection of data with several views to support the enhancement of a business area (MODAF, 2010). Each view provides a different standpoint of the business under examination to support understanding of how this business operates. The MODAF defines the following viewpoints: Strategic context; Operational context; Service context; System architecture; Supporting Standards; Acquisition context (MODAF, 2010).

Both approaches are similar and support the general vision of any IT project in the military domain. However, they do not specify techniques or methods of how to elicit the information required to build systems.

Different from the previous frameworks, the French Ministry of Defense established specific experimentation frameworks called “laboratoires technico-operationel” (LTO) in order to create novel theories and to investigate technologies in military contexts (Merindol, 2010). The main focus was to systematize the involvement of military users in the

determination of novel technological solutions. The approach was developed to support an improvement in the participation of the military users of the systems in solutions developed for armed forces by other organisations (Merindol, 2010). As with the American and British approaches, there was no focus on elicitation methods or techniques, rather it is a methodology to enhance end user participation, specifically on command and control systems.

In general, all three approaches mentioned above focus on the notion that outsourcing system development is the common approach. All these ideas focus on IT project development in a very broad sense with several guidelines for procurement and contract management. Although very complete and useful, none of the approaches provide guidance specifically on how to perform elicitation of the knowledge involved.

In conclusion, software is crucial to military services. Thus, software engineering also became critical to the military area as it is to any civilian area. However, this complex discipline still faces many difficulties due to all the factors involved. Consequently, it is essential that the military domain continues to support research in the area that can bring even more formality to the software development process than the existing frameworks already brought, in order to increase the rate of predictability of this process (Jones, 2000). Thus, the goal of this study is to propose a framework, to be discussed in subsequent chapters, whose purpose is to support requirements elicitation in the military environment and may facilitate studies in civilian organisations that have features in common with the military domain. This framework could be complementary to any of the existing frameworks mentioned above.

3.5 Summary

This chapter presented the military environment. First, it studied the domain according to an existing organisational characterisation. From this study, we could conclude that the military environment can be characterised mainly as a role culture. This domain is extremely hierarchical, with strong role definition. There are several regulations to each task and an expected behaviour to every role. Furthermore, the power is related to the position rather than technical expertise.

After this characterisation, we presented a general view of software development in the military domain according to the literature. It was mentioned that in a specific domain where several systems are developed, it is possible to establish some basic concepts and structures that can be used in several projects. However, it is a difficult task that will demand a large effort and some previous infrastructure. Specifically in the military domain, there are several issues to be dealt with. Issues such as heavy regulation, budget constraints, strong hierarchical structure, and uncertainty of the data are some of these issues.

Due to all these features some specific frameworks for the military domain were presented throughout this chapter. All three frameworks described, have a broad focus encompassing not just system development, but the whole process of contracting enterprises to develop systems, and all activities involved. However, as the focus is more commercial and contractual, they lack supporting and defining some specific tasks and phases of software development such as requirements elicitation.

The next chapter will present an exploratory qualitative research performed mainly in the Brazilian Navy with the aim of gathering the issues related to system development in the military environment in a more specific and detailed way. Due to the lack of research available in this domain for security reasons, especially presenting problems, this qualitative research was essential to be the basis of the proposed solution.

Chapter 4

Exploratory Qualitative Research

This chapter will present an exploratory qualitative research performed in the Brazilian Navy to gather specific problems regarding system development. To allow the generalisation of data, we will also present interviews performed in the Brazilian Air Force and Brazilian Army, as well as with personnel related to foreign armed forces. These interviews led to a selection of issues to be addressed. This is the first contribution of the research. These issues were organized and grouped. After the interviews several Brazilian Navy teams performing requirements elicitation were observed to confirm the issues obtained in the interviews. To allow the triangulation of data, documentation was also gathered and analysed to also confirm the issues. Finally, we present an analysis of the results and the conclusions of the whole Part I.

4.1 Introduction

In parallel to the literature research, there were a gathering and analysis of data from the field following 'Grounded Theory'. Grounded Theory is "the discovery of theory from data systematically obtained from social research" (Corbin and Strauss 1990). According to Corbin and Strauss (1990) Grounded Theory besides revealing relevant conditions also determines how the stakeholders react under those conditions, and the consequences of their actions. This approach was selected due to the ease access that I have, as a Navy officer, to the teams and information in the Brazilian Navy.

Corbin and Strauss also mentioned that as in other qualitative approaches, the data for a grounded theory can come from various sources. Data collection methods include interviews and observations, with other common sources, such as: documents, newspapers, books among others – anything that might help understanding the setting under investigation. After collecting the data, there is an analytic process, in this case coding. Another important characteristic of this approach is that the first collection results guide the following rounds of data collection (Corbin and Strauss 1990).

In this research, as the first source of data, a semi-structured interview was prepared to ascertain the opinions of Brazilian Navy personnel regarding problems encountered during systems development. Thirty interviews were conducted with executives, managers and analysts as well as some client representatives of the Brazilian Navy. These interviews were conducted in Brazilian Portuguese and then translated into the English language. These interviews were conducted in an inductive way. After translating the interviews, the relevant issues identified were assigned a color and number code. As a final step, the results were

organized according to the number of respondents as well as separated by the role of the informant.

As a second source of data, ethnographic observations of teams as they performed requirements elicitation activities in the Brazilian Navy were undertaken. These ethnographical studies were guided by the findings from the interviews in a deductive way. When it was possible, meetings were filmed subject to security clearance. Ethnographic fieldwork was conducted, especially during the requirements elicitation phase, within projects that were developing administrative and operational systems or, according to the nomenclature used by the US Department of Defense (DoD), non-mission-critical and mission-critical systems (DoD, 1987). Operational systems are systems where the information handled, usually classified, is directly related to the main activity of armed forces, including all tactical, operational and strategic levels of information. These systems cover areas such as command and control, logistics, decision support, operational intelligence, search and rescue, tactical control, weapon's control, military simulation, damage control and communications. In the Brazilian Navy, operational systems deal with information relating to ships, their crews, their armaments, and the operations carried out by them. Administrative systems deal with information regarding administrative tasks as with any other civilian institution, such as payment systems and medical care. However, these are tailored to the military environment. The Brazilian Navy also has hybrid systems that span both operational and administrative areas. This fieldwork permitted the observation of behavioral aspects that could be influencing requirements tasks as well as technical aspects related to the correct use of elicitation techniques. Issues such as standardization of procedures among the teams and the influence of hierarchy could also be better understood.

The observation outcomes could confirm issues raised during the interviews and lead to a more complete and accurate picture of the environment under study.

Finally, to guarantee the triangulation of the data (Yin, 1994) documentation was used as a third source of information (Figure 12). Rules and regulations of the Brazilian Navy were studied and analysed in a deductive way.

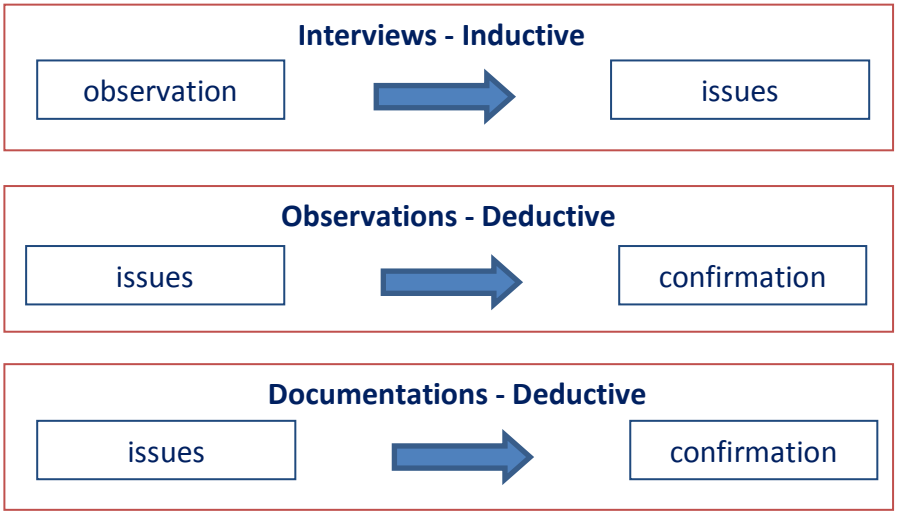


Figure 12 – Summary of the qualitative research methodology

At Oxford University any research that deals with human subjects must be submitted to the Central University Research Ethics Committee. The committee must observe measures taken by the researcher to guarantee the confidentiality of personal information and opinions. Thus, in the research, each interviewee and the researcher signed an 'Informed Consent Statement' (Appendix F). One copy of each document was retained by the respective interviewee and another copy was sent to the committee. To undertake the observations the researcher attended one lecture about ethnography and was trained by the supervisor (an

ethnographer) to remain a silent observer without interfering in the setting. Before each observation these rules were presented to the participants.

4.2 Interview Results

As mentioned previously, the idea of studying system development in the military with the aim of increasing its chance of success came from my own experience in the Brazilian Navy. In order to address correctly the research and to challenge my assumptions, thirty interviews were performed with managers and developers. The interviews were carried out with the presence of the researcher on trips to Brazil. Among the interviewees, there were four executives, nine managers, fourteen analysts and three client representatives. The goal of the interviews was to gather information from the various roles existing in the system development teams in order to obtain different perspectives on the current issues. I would then be able to compare my own experiences and conclusions with other members' points of view. Furthermore, there would be a solid grounding from which to begin my investigation of the related literature. Therefore, these semi-structured interviews were conducted according to the Grounded Theory approach (Corbin and Strauss 1990). These interviews were conducted in Brazilian Portuguese and then translated into English. Once translated the interviews were open coded and relevant issues identified were assigned a color and number code to permit the use of Grounded Theory. Several issues were elicited during the interviews. Some of these issues are interrelated. These interviews were ordered according to the number of respondents who mentioned each issue. Below is the list of issues in descending order of the number of respondents who mentioned the issue out of the thirty interviews.

1 **Lack of guidelines to perform Requirements Elicitation** and the consequent lack of standardization in the way teams perform this task. There is no guideline to guide specifically how to perform the task of requirements elicitation. Currently, each team is free to perform the task using any technique. As teams do not know any other technique, all teams use interviews to perform this task. (Mentioned by 26 interviewees)

2 **The hierarchy.** A characteristic which is intrinsic to military environments. Higher ranks often influence system development by translating their desires into orders and consequently, into requirements. Issues such as increasing scope without increasing time are very common. Additionally, technical decisions regarding scope and schedule of the system oftentimes are made by non-technical people from the upper tiers in the naval hierarchy. Military organisations have rigid hierarchies (command structures), and these structures are essential for the primary purpose of the military, but with respect to system development and, specifically, requirements engineering, at times, this feature has produced negative consequences. Unlike in civil organisations, there is no scope for 'flattening' the management structure in armed forces. (Mentioned by 24 interviewees)

3 **The third issue is related to the client**. Many issues with the client were mentioned such as: **lack of commitment, lack of knowledge about the business and unavailability to meet the team**. These issues are related to the high turnover of client staff. It is generally the case that a client representative who knows that he will not be on the project for its entire life is consequently, not committed to the system. By the same token, as a client representative will often be in a department for a relatively short period of time, he will not have sufficient expertise or familiarity with the environment to be of any real help to the development team. (Mentioned by 23 interviewees)

4 The **political influence** associated inevitably with the military hierarchy. There are some political decisions that postpone some systems, interrupt some developments and so on.

Other political decisions influence the infrastructure and even computational ambience. (Mentioned by 23 interviewees)

5 **Lack of continuity due to the turnover among the client staff**. As mentioned before, client staff is continually changing during system development, and each new clients' representative has his own expectations, which often lead to requirements changing. (Mentioned by 22 interviewees)

6 **Delay itself**. Many interviewees mentioned that most of the projects in the Navy are delayed or the scope is reduced to avoid delay, which clearly indicates that they were often behind schedule. (Mentioned by 21 interviewees)

7 **Only one technique to perform Requirements Elicitation: non-structured interviews**. There is just one technique used by the Navy teams to perform requirements elicitation: conduct interviews; not because it's the best one rather because nobody knows any other. Besides which interviews are conducted based solely on particular teams combined experience. (Mentioned by 20 interviewees)

8 **Uncertainty of who is responsible for performing Requirements Elicitation**. Usually, the manager performs this task because no team has been formed at the time of the performance of this step. Besides, no team leader is worried about hiring someone with experience in this task when the assumption is that there is no need for specific skills to perform it. (Mentioned by 19 interviewees)

9 **Lack of time to perform Requirements Elicitation**. There is pressure by the client to start developing as soon as possible. Usually, the client does not understand that this task is important and takes time to be done well in order to avoid future development problems. (Mentioned by 19 interviewees)

10 **Constant change of scope during the project**. Due to high client turnover with different expectations and thus a lack of time available to perform requirements

elicitation properly, the scope keeps changing during development. (Mentioned by 18 interviewees)

11 **Late evaluation of client / Requirements without the participation of the client or without the participation of the actual user**. Usually, the real user does not participate in the elicitation because the customer elects a representative to be in contact with the development team. (Mentioned by 13 interviewees)

12 **Lack of project management culture**. A large percentage of managers in the Brazilian Navy assume their roles without having gained any experience in the area. Usually, they are trained to work as Navy Officers and have only become managers because of the ranks that they hold. (Mentioned by 13 interviewees)

13 **Turnover of the development team.** As is the case with the client team, the development team also has a large turnover of personnel. The manager is usually an officer who can be transferred at anytime and non-military members often move to other jobs in the market. (Mentioned by 9 interviewees)

14 **Lack of communication within the team / Multicultural team**. Some members of the team do not know the expectations and desires of the client because they are removed from the requirements elicitation phase to avoid direct interference from the client in the development phase. Besides that, there are several communication difficulties within the team as teams are composed of members with a very different background: military, civilians with a long career in the Navy and civilians with temporary contracts. (Mentioned by 11 interviewees)

15 **Other non-technical tasks of the manager and the military**. Normally, in the Navy, there are numerous administrative and military tasks to be performed that are not related to the project such as: new reports, military ceremonies and many others that demand time away from the project. (Mentioned by 7 interviewees)

16 **Teams under-skilled**. Usually in Navy systems there are specific features that need particular skills in the team, and these skills are hard to find in the market. When professionals with the required skill sets can be found they will normally command high salaries, however, the Navy is regulated by governmental laws that restrict wages. Consequently, the Navy is forced to hire people who are lacking the skills needed with the aim of developing those skills internally. This leads to delays due to lack of competence and an increased risk of errors going unnoticed. Of course, after developing their skills there is always the possibility to lose these professionals to the market with no compensation for the Navy for the resources expended. (Mentioned by 6 interviewees)

17 **Bad estimates**. As the manager has to estimate the development program timescales and costs before he has a full understanding of what his client's needs are and because he is not trained to perform requirements elicitation correctly, systems schedules are almost invariably underestimated. (Mentioned by 6 interviewees)

18 **Lack of concern about non-functional requirements**. Usually, the development team focuses on functional requirements thereby neglecting non-functional requirements. Issues such as the client infrastructure which may be required to house the new system are often considered very late in a project's lifecycle, sometimes only when system deployment is imminent. (Mentioned by 6 interviewees)

19 **Concern about the financial restrictions of the annual budget and bids**. The Navy has to operate within a finite annual government budget. In operating under this regime the Navy has to comply with several rules such as the requirement to forfeit unused funds at the completion of the fiscal year. As a consequence, the client and the development manager are often more worried about the budget for the project than its requirements. (Mentioned by 5 interviewees)

20 **Systems developed based on bad processes or processes not described.** Many times clients are unaware of their department's own processes and usually these processes are

not formally described, which makes them difficult to understand. Another issue is that normally there is insufficient time to understand the problem so teams restrict themselves to developing a system suitable to the existing process rather than trying to improve this process. (Mentioned by 4 interviewees)

21 **Lack of infrastructure for testing / Lack of exercises to test operational systems**. The first issue is that there is a lack of infrastructure to test systems before using the client's computers, so many problems surface when the system is deployed. The second issue is related to operational systems, due to the large amount of money needed to prepare a test environment. (Mentioned by 4 interviewees)

22 **Information that not all the team can access.** As the team is a mixture of military people and civilians, there is some information that can't be shared with non-military personnel. (Mentioned by 4 interviewees)

23 **Changing computing environment during development because of delays**. When a system development experiences long delays it often means that changes are required in the computer environment. When this occurs there is an ironclad guarantee of further delays. (Mentioned by 3 interviewees)

24 **Lack of knowledge of the environment by the development team / Multicultural characteristic of the team.** Due to the multicultural characteristic of the development team with military personnel, civilians from the Navy and workers hired in the market place, there is a difficulty to share an understanding of the problem to be solved among the team. It is also difficult to share an understanding of the terms and acronyms used in the environment. (Mentioned by 3 interviewees)

25 **Lack of formalization of contract changes**. Due to high client turnover, there are many changes to the project scope which are often not formalized. (Mentioned by 2 interviewees)

26 **Acceptance is not focused on requirements**. Sometimes acceptance testing is not derived from the requirements. There is also not a formalization of the acceptance tests. (Mentioned by 2 interviewees)

27 **Systems that assist more than one organisation.** Some systems are developed to assist several organisations each of which may have vastly different expectations of it. (Mentioned by 2 interviewees)

28 **Sharing staff between projects**. Due to the lack of financial resources and the lack of physical space to accommodate teams properly some projects in the Brazilian Navy have to share professionals. (Mentioned by 1 interviewee).

ID	Issue	Quantity
1	Lack of a document to guide requirements elicitation / lack of standardization of development	26
2	The hierarchy	24
3	Client - Lack of commitment of the client / lack of knowledge about the business / Unavailability to meet the team	23
4	The political influence	23
5	Lack of continuity due to the turnover of client staff	22
6	Delay itself	21
7	Only one technique to elicit requirements: non-structured interviews	20
8	Uncertainty of who is responsible for performing requirements elicitation	19
9	Lack of time to perform requirements elicitation	19
10	Constant change of scope during the project	18
11	Late evaluation of client / requirements without the participation of the client or without the participation of real user	13
12	Lack of project management culture	13
13	Lack of communication within the team	11
14	Turnover of the development team	9
15	Other non-technical tasks of the manager and the military	7
16	Teams under-skilled	6
17	Bad estimates	6
18	Lack of concern about non-functional requirements	6
19	Concern about the financial restrictions of the annual budget and bids	5
20	Systems developed based on bad processes or processes not described	4
21	Lack of infrastructure for testing / Lack of exercises to test operational systems	4
22	Information that not all the team can access.	4
23	Changing computing environment in the middle of development because of delays	3
24	Lack of knowledge of the environment by the development team / Multicultural characteristic of the team	3
25	Lack of formalization of contract changes	2
26	Acceptance is not focused on the requirements	2
27	Systems that assist more than one organisation	2
28	Sharing staff between projects	1

Table 2 – Issues mentioned in the interviews / quantity of respondents that mentioned the issue

To gain a better understanding of the results, the issues were also observed according to the roles of the respondents: executives, managers, analysts and client representatives. Below we present the main issues mentioned by the informants in descending order of percentage. After each list of issues a table presents the quantity and percentage of respondents that mentioned that issue.

Executives (4 informants)

- Delay itself
- Lack of guidelines to perform requirements elicitation and the consequent lack of standardization in the way teams perform this task.
- The hierarchy.
- Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team.
- The political influence.
- Lack of continuity due to the turnover among the client staff
- Constant change of scope during the project.

ID	Characteristic	Quantity of respondents	Percentage of respondents (total 4)
6	*Delay itself*	4	100 %
1	*Lack of guidelines to perform requirements elicitation*	3	75 %
2	*The hierarchy.*	3	75 %
3	*Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team*	3	75 %
4	*The political influence.*	3	75 %
5	*Lack of continuity due to the turnover among the client staff*	3	75 %
10	*Constant change of scope during the project*	3	75 %

Table 3 – Main issues mentioned by the executives in descending order of percentage of executives mentioning each issue

Managers (9 informants)

- Lack of guidelines to perform requirements elicitation and the consequent lack of standardization in the way teams perform this task.
- The hierarchy.
- The political influence.
- Uncertainty of who is the responsible for performing requirements elicitation.
- Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team.
- Only one technique to perform requirements elicitation: non-structured interviews.
- Constant change of scope during the project.

ID	Characteristic	Quantity of respondents	Percentage of respondents (total 9)
1	*Lack of guidelines to perform requirements elicitation*	9	100.0 %
2	*The hierarchy.*	9	100.0 %
4	*The political influence.*	9	100.0 %
8	*Uncertainty of who is the responsible for performing requirements elicitation.*	9	100.0 %
3	*Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team*	8	88.9 %
7	*Only one technique to perform requirements elicitation: non-structured interviews*	8	88.9 %
10	*Constant change of scope during the project*	7	77.8 %

Table 4 – Main issues mentioned by the managers in descending order of percentage of managers mentioning each issue

Analysts (14 informants)

- Lack of guidelines to perform requirements elicitation and the consequent lack of standardization in the way teams perform this task.
- Lack of continuity due to the turnover among the client staff.
- The hierarchy.
- Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team.
- Delay itself.
- The political influence.
- Lack of time to perform requirements elicitation.
- Only one technique to perform requirements elicitation: non-structured interviews.

ID	Characteristic	Quantity of respondents	Percentage of respondents (total 14)
1	*Lack of guidelines to perform requirements elicitation*	14	100.0 %
5	*Lack of continuity due to the turnover among the client staff.*	13	92.9 %
2	*The hierarchy.*	12	85.7 %
3	*Client - lack of commitment, lack of knowledge about the business and unavailability to meet the team.*	12	85.7 %
6	*Delay itself.*	12	85.7 %
4	*The political influence.*	11	78.6 %
9	*Lack of time to perform requirements elicitation*	11	78.6 %
7	*Only one technique to perform requirements elicitation: non-structured interviews*	10	71.4 %

Table 5 – Main issues mentioned by the analysts in descending order of percentage of analysts mentioning each issue

Client Representatives (3 informants)

- The political influence.
- Delay itself.
- Lack of knowledge of the environment by the development team/ Multicultural characteristic of the team.
- Bad estimates.
- Concern about the financial restrictions of the annual budget and bids.
- Systems that assist more than one organisation.

ID	Characteristic	Quantity of respondents	Percentage of respondents (total 3)
4	*The political influence*	3	100.0 %
6	*Delay itself*	3	100.0 %
24	*Lack of knowledge of the environment by the development team/ Multicultural characteristic of the team*	3	100.0 %
17	*Bad estimates*	2	66.7 %
19	*Concern about the financial restrictions of the annual budget and bids*	2	66.7 %
27	*Systems that assist more than one organisation*	2	66.7 %

Table 6 – Main issues mentioned by the client representatives in descending order of percentage of representatives mentioning each issue

Some conclusions can be raised from the results above showing some differences between the roles. As it can be observed, executives have as their main concerns issues that are related to the result of the system. Usually executives are worried about the general vision of the systems and the issues that can jeopardize the overall success. At the same time, the main concerns of the managers are political questions such as political influence and hierarchy as they are normally lower ranks in comparison to client representatives and executives, thus having to submit their work to the higher ranks' will. The analysts, even focusing on almost the same issues, have as their main concerns technical aspects such as lack of guidelines to perform requirements elicitation and lack of time to perform requirements elicitation. They are responsible for directly performing the activities, consequently wanting guidelines to support their work. Finally, client representatives have as their main preoccupations, the multicultural background and the lack of knowledge of the environment by the requirements team, as representatives have to interact directly with the requirements team, and these issues increase communication problems leading to difficulties on having their needs understood.

The results from the interviews not only corroborated the initial assumptions that I made from my previous background in the Navy, but also brought to light some new issues and even possible solutions for the problems. Furthermore, the respondents did not restrict themselves to answer the questions; they additionally made some suggestions as follows:

Using some technique such as Agile Methodology or RUP (Rational Unified Process). With these methodologies, the iterations are smaller and faster, which could reduce the impact on the project of high client turnover.

Using observation. Oftentimes, the client is not well informed about his own area's processes and very often the development team has to speak to a representative rather than the real user. The presence of the development team inside the client's environment would reduce the potential negative impact of this lack of knowledge by the client's representative.

Using participatory design. The presence of the client in the developing team could increase the sense of commitment to the project by the client. The client would also understand and thus respect the complexity of the tasks performed by the development team.

Using prototypes. A prototype could help the client to assess the new system before deploying it.

Creating ontology. An ontology would help new members to understand the military environment and also produce a standardization of information in all systems. This would reduce in the medium term the time needed to model a system.

Web tools to support the client to request changes and observe their status. A tool to help the client to request formally any changes and to track their status would formalize the requesting procedure and consequently, reduce requirements changes.

Two separate contracts: one for Requirements Elicitation and one for Development. This suggestion would let teams spend sufficient time performing requirements elicitation, as the focus of the first contract would be just requirements.

4.2.1 Interview Analysis

My initial assumptions about requirements elicitation in the military domain were corroborated by the results from the interviews conducted. Additionally, some new issues were raised by the respondents such as lack of infrastructure for tests, and team members with lack of technical expertise. The respondents also proposed solutions to some of the issues raised. These suggestions will be considered in the proposed solution.

In order to facilitate the search for issues in observations and documentation as well as to build the solution, the issues mentioned in the interviews were grouped in Table 7 as follows. The "general" (G) group corresponds to issues that will not be directly addressed by the solution; however, their results tend to be better just from the existence of a guide for the requirements elicitation task. A second group named "hierarchy" (H) is related to hierarchical issues as well as political influences. The third group is "multicultural teams" (MC) related to the different background of the members of the requirements team. Another group is related to the "turnover rate" (TO) of the members of client and development teams. Finally, there is a group of issues that will neither be addressed nor be influenced by the solution. These issues are named "not addressed" (NA). These main categories were selected based on the possibility of having a direct influence on requirements elicitation tasks and the possibility of being directly influenced by the solution.

Issue	Main Issue Related G – General H – Hierarchy TO – Turnover rate MC – Multicultural team NA – Not addressed
Lack of a document to guide requirements elicitation	G
The hierarchy	H
Client - Lack of commitment of the client / lack of knowledge about the business / Unavailability to meet the team	TO
The political influence	H
Lack of continuity due to the turnover of client staff	TO
Delay itself	G
Only one technique to elicit requirements: non-structured interviews	G
Uncertainty of who is responsible for performing requirements elicitation	G
Lack of time to perform requirements elicitation	G
Constant change of scope during the project	G
Late evaluation of client / requirements without the participation of the client or without the participation of real user	G
Lack of project management culture	G
Lack of communication within the team	MC
Turnover of the development team	TO
Other non-technical tasks of the manager and the military	NA
Teams under-skilled	NA
Bad estimates	G
Lack of concern about non-functional requirements	NA
Concern about the financial restrictions of the annual budget and bids	NA
Systems developed based on bad processes or processes not described	G
Lack of infrastructure for testing / Lack of exercises to test operational systems	NA
Information that not all the team can access.	NA
Changing computing environment in the middle of development because of delays	G
Lack of knowledge of the environment by the development team / Multicultural characteristic of the team	MC
Lack of formalization of contract changes	G
Acceptance is not focused on the requirements	G
Systems that assist more than one organisation	G
Sharing staff between projects	NA

Table 7 – interview issues and the main related groups of issues

This initial case study and the literature review identified the main issues to be used as the basis for a new round of data gathering when observations and documentation analysis were conducted. The issues under investigation were: hierarchy, multicultural teams, and the turnover rate. Interestingly the principal issues raised were social issues rather than technical ones. This result corroborates the assumption that the requirements phase is one that depends on several human interactions thus leading to numerous sources of conflicts among the stakeholders. Consequently, for a solution to be successful in this phase it should also address these social issues.

4.3 Observation Results

In addition to the literature review and the interviews already conducted, as a further source of information, there was an observation of teams in the Brazilian Navy as they perform requirements elicitation. As a Brazilian Navy officer I have full access to all personnel and information needed for the thesis such as development teams, project documents and requirements meetings, which allowed me to gather data from projects and the elicitation process. It was also possible to participate as a silent observer on requirements meetings. To support the analysis of the meetings video was used (subject to the approval of the Navy). These observations involved administrative and operational systems, or according to the nomenclature used by the US Department of Defense (DoD), non-mission-critical system and mission-critical system (DoD, 1987). The observations were undertaken in Brazil with my presence as a silent observer. In frequent contact with the Department of System Development in the Brazilian Navy I was informed about the schedule of each project and

organized the trips to Brazil in order to match the period of requirements elicitation of the projects under study.

In this research, three administrative systems and four operational systems were observed in January and in June of 2011. The following systems were observed during 17 meetings: Health Navy System, Identification System, Financial System (administrative systems); Intelligence System, Command and Control System, Officers Evaluation System, Area Control System (operational systems).

During these meetings, several aspects were observed corroborating the issues elicited during the interviews.

4.3.1 Hierarchy

The first aspect observed with numerous examples was the hierarchical influence existing in the meetings. Usually, the manager of the development team is a lieutenant or a lieutenant commander while the client representative is normally a captain, which is a much higher rank. This hierarchical difference showed a high influence on the main decisions taken during the development.



Figure 13: Client in a prominent position to reinforce the hierarchical difference

Sometimes, even technical decisions are made based upon the hierarchy and not on expertise. Aspects such as how to elicit requirements are discussed by the client representative, who usually does not have any expertise in requirements elicitation.



Figure 14: Client discussing a questionnaire presented by the requirements team.

"You will have to allocate one person or two for each division!" *[Client representative telling the development manager how many people should participate in the requirements meetings – client interference in technical decisions]*

Another aspect of the hierarchy difference is the lack of participation shown by the lower ranks in the presence of higher ranks. During the requirements meetings it was observed that usually the lower ranks only participate when directly asked by the higher ranks as can be seen in Figure 15.

Lowest ranks
In the meeting



- Client Manager: Mary! *[Client manager asking the lower rank to answer the question]*
- Mary (Client member – lower rank): I think the booking module is really not good [*lower rank agrees with the Client manager speaking in an almost inaudible volume – showing discomfort in the presence of higher ranks]*

Figure 15 – Requirements workshop with multiple ranks

4.3.2 Multicultural teams

The second aspect observed during the meetings was the multicultural aspect of the development team. In the Navy, there are three different types of workers: military personnel, civilians with a long career in the Navy and civilians hired in the market place. With extremely distinct backgrounds, there is a high difficulty of communication within the team. Even among the military personnel there is a difference between technical personnel and career personnel. Due to the very specific environment and the distinct backgrounds, there is consequently a different level of understanding of the domain. During the meetings, these differences show a strong influence on the comprehension of the environment and its behavioral aspects as well as the terms used. These comprehension difficulties lead to a communication problem which directly affects requirements elicitation tasks.

Civilian team member: "… going back, you said that the Military Organisation has to make a system that is compatible with the Naval Health Department"

Client Representative: "I did not say that"

Team Manager: "He didn't say that"

[The civilian team member misunderstood the idea presented by the client representative]

Some terms and acronyms are specific and are common sense among the military personnel, but have no meaning to the civilian members.

Military Team Member: "This questionnaire is much broader than what we did in the NEPTUNE project." *[The NEPTUNE project is well-known to the military personnel, but unknown to the civilians]*

This background difference together with the hierarchical aspect leads the client representative to prefer to work on requirements tasks with military personnel rather than with civilians.

Client: "The questionnaire is very detailed, which suggests that bound to the questionnaire I have someone qualified, preferably a military" *[Client representative mentioning that he would rather work with a military member of the requirements team]*

4.3.3 Turnover rate

The third aspect observed on the meetings is the extremely high turnover of the military personnel and consequently, the client representative. As an example, there is an operational system (decision support) where there were eight different client representatives in less than two years. In another operational system (command and control), there were four different client representatives within a period of two years. Each new client representative comes with different expectations and ideas, which inevitably leads to changes of scope.

4.4 Documentation Review Results

To enable the triangulation of data (Yin, 1994) a third source of information was needed. Thus, there were gathering, selection and analysis of documents to confirm the issues mentioned during the interviews and corroborated by the observations.

4.4.1 Hierarchy

There are many documents that reinforce the hierarchical structure of any armed force. These concepts of hierarchy and discipline are part of the pillars of any armed force as shown in the following document:

"Military Hierarchy and Discipline"

> Art. 2nd – discipline is the strict observance and the full compliance of laws, regulations, rules and provisions justifying military organisation, and coordinate its functioning, smooth and harmonious, perfect for fulfillment of the obligation on the part of each and every one of the components of this organisation.
>
> Sole paragraph – the military discipline manifests itself primarily by: obedience to the orders of the superior; total use of energy to the service; righteousness attitudes; spontaneous cooperation for the benefit of collective discipline and efficiency of the institution.
>
> Art. 3rd – Military Hierarchy is the ordering of the authority at different levels, within the military structure. Sorting is done by ranks or graduations; within the same rank or graduation, is by seniority in rank or at graduation.
>
> Sole paragraph – respect for hierarchy is embodied in the spirit of compliance to the authority sequence."
>
> (RDM, 2009)

Furthermore, there are also documents that evidence duties and responsibilities of the military personnel to guarantee that the regulations and hierarchy are respected and followed under any circumstances.

> § 3 – the discipline and respect for hierarchy must be kept in all circumstances of life between active duty military, remunerated reserve and retired.
>
> (Military Statute, 2004)

Finally, armed forces also present documents that will determine when sanctions and penalties must be applied in the case of any military that infringe any of those rules or regulations.

Duties and responsibilities of officers:

a) To know, observe and enforcing his subordinates the provisions of this Ordinance and current legislation;

b) Comply, with commitment, loyalty, promptness and dedication the orders received;

c) Employ every effort for the good performance of the tasks and functions entrusted to him, keeping himself up to date regarding the operating, administrative and technical knowledge for this need.

d) To require that his subordinates perform, with promptness and correctness, all the tasks for which they have been established;

(OGSA, 2004)

Art. 6th – Disciplinary Infraction is any action or omission contrary to the obligations or to military duties laid down in laws, regulations, in standards and in the provisions in force on the basis of the Military Organization, since not focusing on what is capitulated by military Penal Code as a criminal offence.

(RDM, 2009)

Art. 7th – Disciplinary Infractions are:

1. address or refer to superior disrespectfully;
2. censor acts of superior;
3. respond to superior in a disrespectful way;
4. address the superior to handle service issues or personal issues without following hierarchical structure;
5. the subordinate does not, in uniform or not, to greet superior in uniform or in civilian clothes, since knowing him; or do not pay him homage and signs of consideration and respect under military regulations;
6. avoid deliberately answering the subordinate greetings;
7. failing to comply with the order received from the competent authority;
8. slow, without good cause, the fulfillment of the order received from the competent authority;
9. advise or compete for the failure to comply with any order of the competent authority or for delaying its implementation;
10. intentionally induce or compete for others in contravention;

(RDM, 2009)

4.4.2 Multicultural teams

In the Brazilian Navy, all teams are almost always composed of military, career civilians, and temporary civilians. Each of these groups has a different background and culture. Such compositions can be noticed in the following figure. Consequently, there are several misunderstandings between these members. In an attempt to reduce these misunderstandings, there are some introductory courses and presentations to all the civilians who join development teams in the organisation. The goal is to reduce the gap of understanding on military subjects between military and civilian personnel.

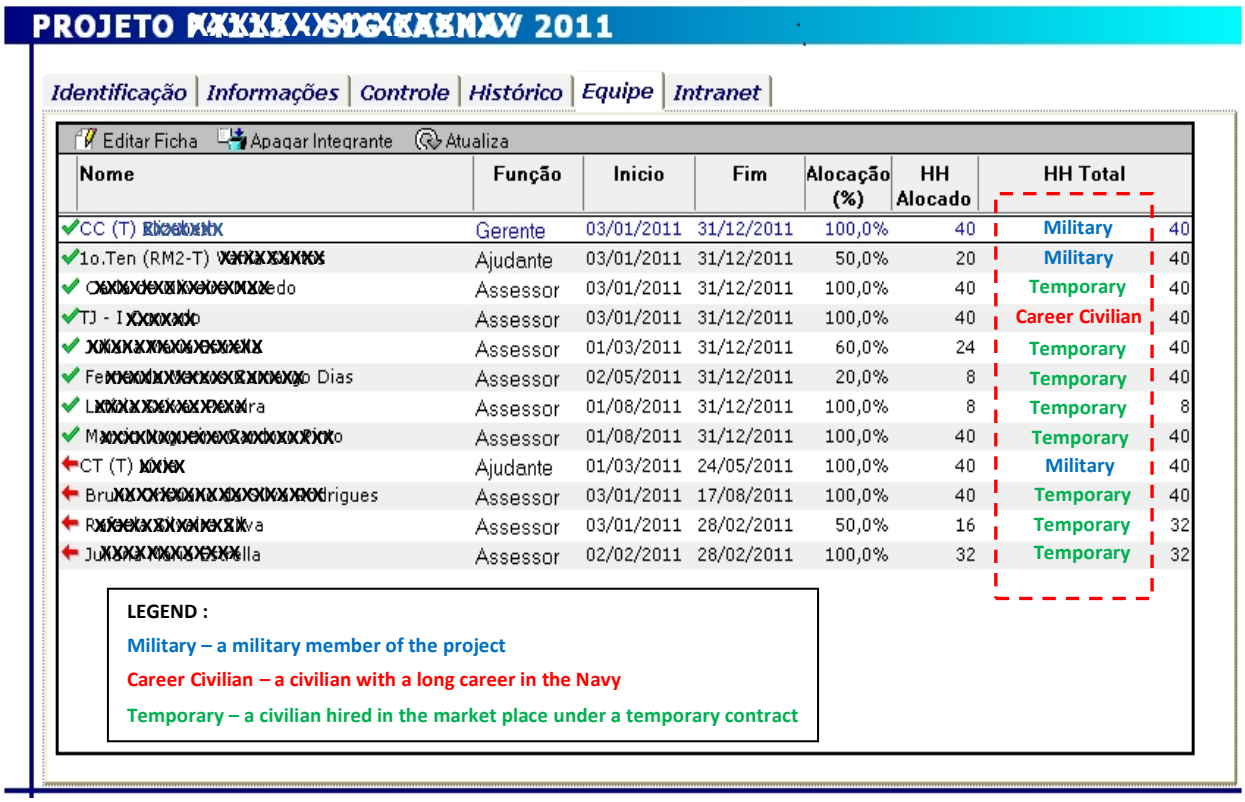


Figure 16 – Multicultural Team in a Project

4.4.3 Turnover Rate

In the Brazilian Navy, it was possible to find documents for career requisites that demand the officers be constantly transferred to other settings in order to keep their career updated. Consequently, there is a high turnover of both managers and client representatives during the development.

According to the PCOM (2007) (Career Plan for Navy Officers) the planning of the military career is a responsibility of each of the Brazilian Armed Forces Commands. In the case of the Brazilian Navy, there are several courses, and the official courses are classified into undergraduate courses, training, and career graduate courses, according to the following table:

<table>
<tr><th colspan="3">CLASSIFICATION OF COURSES</th></tr>
<tr><td colspan="3">Undergraduate Course Officer</td></tr>
<tr><td colspan="3">Formation Course for Officers</td></tr>
<tr><td rowspan="5">Career courses</td><td colspan="2">Specialization Courses</td></tr>
<tr><td colspan="2">Training Courses</td></tr>
<tr><td colspan="2">Staff Course to Intermediate Officers</td></tr>
<tr><td rowspan="3">High Military Studies Course</td><td>Staff Course for Senior Officers</td></tr>
<tr><td>Higher Education Course</td></tr>
<tr><td>Maritime Policy and Strategy Course</td></tr>
<tr><td rowspan="3">Post-graduate courses</td><td colspan="2">Special Technical Qualification Courses</td></tr>
<tr><td colspan="2">Extraordinary Courses</td></tr>
<tr><td colspan="2">Advanced Training Courses</td></tr>
</table>

Table 8 - Classification of courses in the Brazilian Navy (PCOM, 2007)

The table above presents courses that provide progressively the opportunity to obtain the qualifications required for the exercise of the positions set out in the table of personnel. The approval in career courses is one of the requirements for access to upper ranks in military career (PCOM, 2007).

As well as the client team, the development team, and consequently, the requirements team, suffers with high turnover rates. The consequences will depend on the experience and the role of the member being transferred. This high turnover rate can be observed in Figure 17, where in a specific project during the year 2011, 4 out of 12 members were transferred, which means a rate of around 33.3 % of turnover rate in a period of one year.

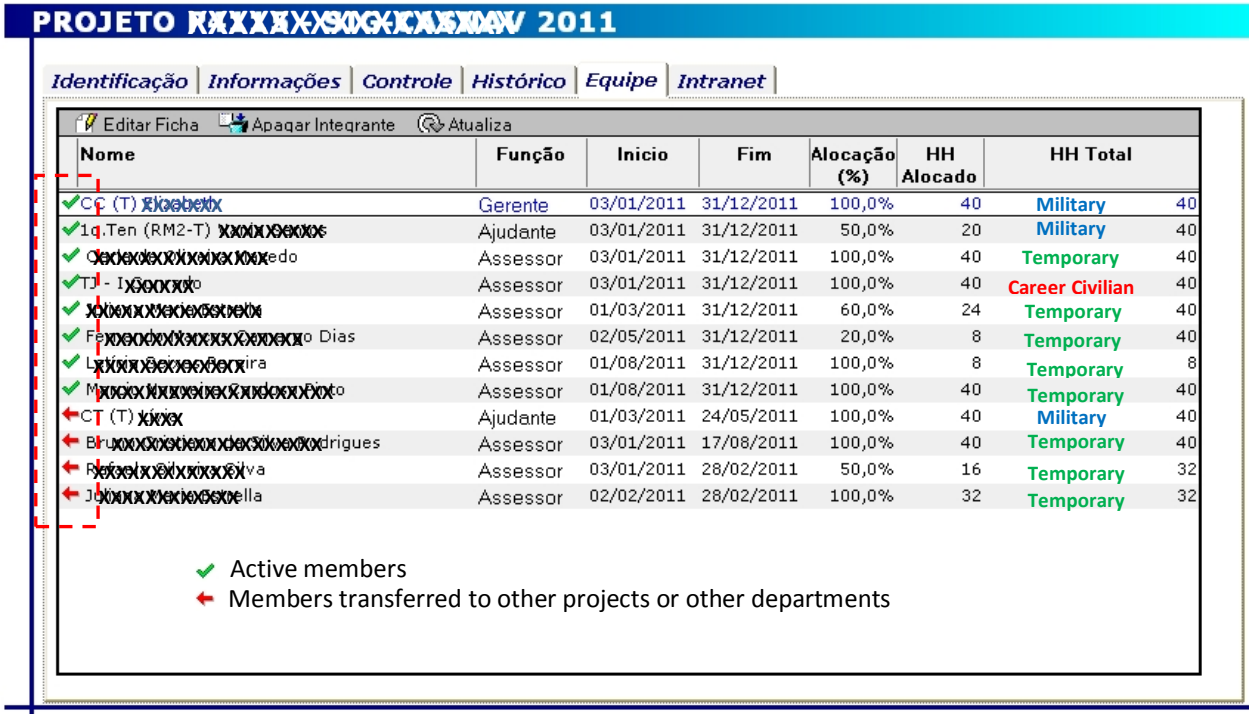


Figure 17 – Turnover in a Project

4.5 Other institutions

In order to guarantee that the problems mentioned above are not exclusive to the Brazilian Navy, another two military institutions were included in the research: the Brazilian Army and the Brazilian Air Force. Thus, to generalise the research another seven interviews were conducted in the Brazilian Air Force, and another four interviews were conducted in the Brazilian Army in June of 2011. In the Brazilian Air Force, three managers and four analysts were interviewed while in the Brazilian Army, two managers and two analysts were interviewed.

While in the Navy the main development is centralized in one department with two departments, in the Air Force and in the Army there are several departments responsible for system development. However, in terms of the research results the same issues were mentioned by the interviewees.

As was expected due to the foundations of any armed force, the hierarchical influence in technical decisions is common in the environments and is shown in the following declaration of an analyst:

> Manager: "The General wanted it immediately! So I didn't have enough time to talk to the stakeholders!" *[the analyst didn't have time to elicit requirements properly due to a superior order]*

Another aspect corroborated in the other military institutions observed was the multicultural communication difficulty. Just like the Navy, the Army and the Air Force use civilians hired in the market place to complete their staffing needs. Thus, it was also

expected that the same communication difficulties were faced by their teams. The following two statements corroborate this assumption.

> Analyst: "In the beginning he *[the analyst]* was just a listener ... after six months hearing those terms the guy started to understand the requirements…"

> Analyst: "He [*the requirements analyst*] doesn't know air defense and nobody else here has any knowledge on air defense. …in the specification he just looked at the format not the meaning…"

With the evidence presented above it was possible to conclude that problems in software development and specifically in requirements elicitation are common in all the Brazilian armed forces. Consequently, a solution for the Navy should also be useful for the Army and the Air Force. However, the similarity of problems could be due to the Brazilian national culture, and thus more general research was needed as can be seen in the next section.

4.6 Foreign Institutions

To discard the idea of research that would benefit just Brazilian armed forces it was necessary to search for problems in armed forces from other countries. Thus, to permit a generalisation of the solution, some people involved in military software development from countries other than Brazil were interviewed. Some of the interviewees are military personnel involved in system development and some work or worked for enterprises hired to develop systems for armed forces. In total seven people were interviewed: three military officers (one retired) and four civilians working for private companies that developed

systems for armed forces. Among the respondents, there are people related to several countries and continents: two related to countries on the American continent, four related to European countries, and one related to an Asian country.

First, a general comment showed the peculiarity of the environment and the difficulty related to the main purpose of the armed forces:

> "It is really hard to test military systems as they should be used in the battlefield."

As the whole area of system development faces problems all over the world independently from the domain, the same reality was expected in the military institutions investigated. Thus, all the respondents mentioned problems in software development directly related to requirements elicitation.

> "A very small percentage of systems is delivered on time"

Specifically to the issues highlighted it was important to confirm whether they are independent of national culture. Unsurprisingly all interviewees mentioned an extremely high political and hierarchical influence in technical decisions, emphasizing that this is crucial in the main activities of an armed force but oftentimes prejudicial in technical tasks, as shown in the flowing statements:

> "It is a highly political environment"

> "The environment is too rigid, which must be appropriate for war, but not appropriate for other activities as dynamic as technological activities."

Problems such as the extremely high turnover were also mentioned, and the consequences seem to be the same. Like in the Brazilian Armed Forces the high turnover rate

leads to a lack of commitment by the client as they will not be in the department by the time the development ends.

> "Due to the high turnover clients are not as committed to the systems as in the private sector"

Another issue mentioned is a lack of knowledge shown by some managers in several countries, as some assignments happen for hierarchical reasons. Due to the hierarchy, the highest rank in the development team has to be the manager, even if he/she does not have the expert knowledge to play this role.

> "Managers were assigned to the systems for political reasons rather than their expertise in the area."

With all the similarities found among the answers of the interviewees related to these different countries even in different continents involved in the research, there is an indication that the military culture is more relevant to requirements elicitation in a military environment than the national culture. This is due to the very specific nature of the core activities of this environment that demand a high commitment and also some unusual behavior, strongly guided by hierarchy, as well as very specific terms, acronyms and even language that goes beyond national borders. Consequently, it is likely that a solution that worked for the Brazilian Navy also would benefit the armed forces throughout the world.

4.7 Conclusions of Part I (Chapters 2, 3 and 4)

The literature review in Chapter Three showed that the military environment is a complex and unique one with several specific characteristics that influence any task performed in this domain. A deeper analysis was needed to define the specific issues to be addressed, thus a qualitative research was performed. From the interviews, numerous issues were highlighted. Many of these issues were confirmed during the observations and documentation analysis. Furthermore, these issues were also found in other Brazilian Armed Forces as well as in Foreign Armed Forces. Thus, it is possible to conclude that there is a similarity in the armed forces around the world in terms of problems that influence requirements elicitation.

Observing the existing frameworks to support the selection of elicitation techniques presented in Chapter Two it is possible to observe that none of them addressed the issues highlighted by the qualitative research. As a result, a novel framework would be needed to solve the problem addressing the issues gathered.

In order to guide the solution to be proposed, the issues were organized and grouped. Issues such as lack of guidelines, uncertainty of who is responsible for elicitation, the use of only a single technique to perform elicitation and the lack of project management culture will be minimized when guidelines to perform requirements elicitation exist. All other issues highlighted in the interviews are grouped into three key challenges: multicultural teams, hierarchy, and high staff turnover. From these issues, some relationships could be established.

- When the cultural difference increases there is a natural increase in the difficulty of communication due to misunderstandings provoked by the lack of knowledge of the social implications of the military environment.

- An increase in the hierarchical difference leads to a decrease in the collaboration during requirements tasks as usually the lower ranks tend to avoid giving their own impressions and ideas for the system even though being an important part of the stakeholders group. At the same time, the higher ranks sometimes use their authority to inhibit discordant opinions. This issue is usually even bigger in the development of operational systems.

- An increase in the rate of client representative turnover drives an increase in the number of change requirements and consequently scope. Every new client representative has a different idea of the system and with his/her hierarchical authority tends to induce strong changes in the requirements previously elicited.

Based on these correlations, presented in Figure 18, a solution will be proposed to reduce the impact of them on the requirements elicitation process.

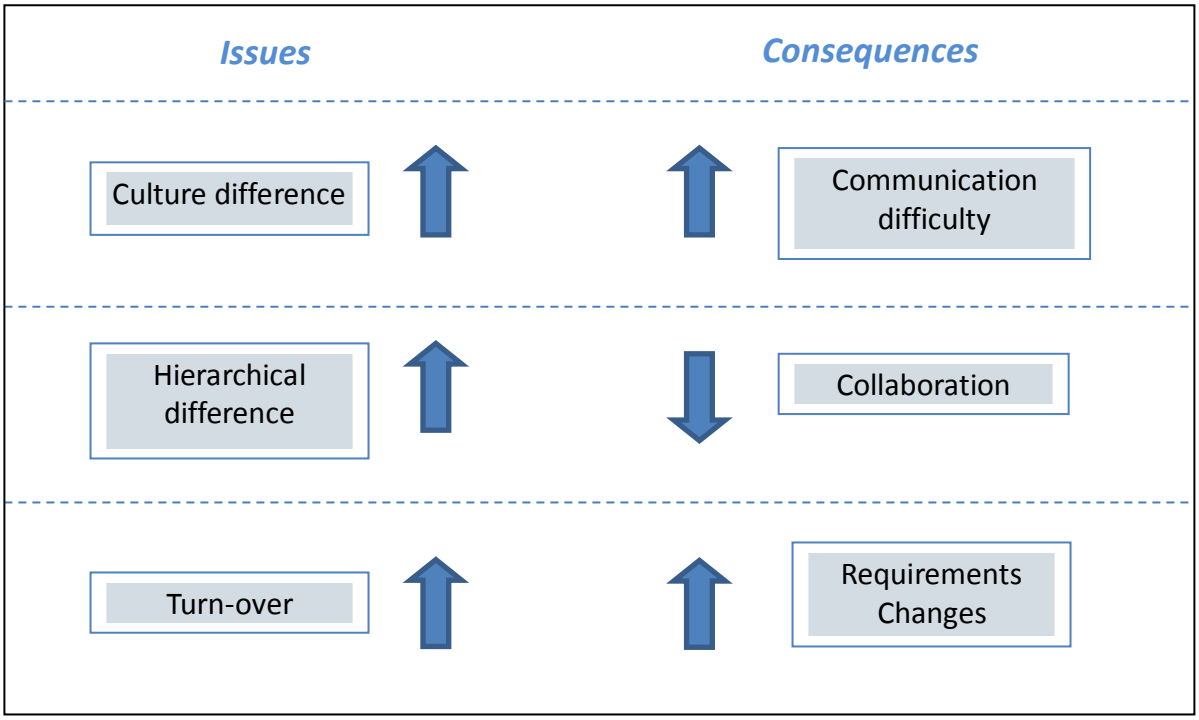


Figure 18 – Main issues to be addressed by the framework and their consequences

Part II

Building the Solution

Chapter 5

Conflicts in Requirements Engineering

As there is a consensus in the literature that requirements elicitation is a group task, the first aim of this chapter is to conceptualise and understand group tasks in general. In a group task there is the need for constant interaction between the personnel involved. This interaction inevitably leads to conflicts. Grudin (1994) mentions that system development problems usually have non-technical sources, but have their root in group interaction. Considering the issues raised in the first part of the research it is reasonable to conclude that each issue is a potential source of conflicts. Hierarchical differences, cultural differences or personnel changes may all lead to perspective differences or misunderstandings. Thus, the solution focuses on resolving conflicts as a way to reduce requirements problems. The solution will be based on the 'Conflict Theory' from Poundy (1967). The first step is to conceptualise conflicts and to understand the sources of conflict in requirements elicitation tasks. Then, we will correlate conflicts and requirements engineering, presenting the existing research on this area. Finally, the process of conflicts will be displayed with the focus on the existing research.

5.1 Introduction

"Requirements engineering is a multidisciplinary, human-centred process" (Nuseibeh and Easterbrook, 2000). If a REng process is being used this implies that there is the need for a system to support some activity. However, invariably, such a system will change the environment and the social order in which the activity takes place. Therefore, it is essential that the REng process be aware of all the implications of the changes generated by the new system. Thus, to gain a deeper understanding of the environment and the social interactions affected, REng relies on other disciplines such as cognitive and social sciences that can provide the theory needed to ground the process as well as presenting some techniques to support activities such as elicitation and modelling. In the REng process, conflicts of interest arise among the stakeholders and, consequently, there must be some kind of negotiation. These conflicts represent one of the biggest issues in REng. Research in REng in the last decade reveals a number of sources of conflict. They include conflict between stakeholders' perceptions about the problem; the many goals of a design; suggested solution components; stated constraints; conflicts in resource usage (Damian et al, 2003). Initially, in REng practice, conflicts were thought to be a problem and therefore, something to be eliminated. Over time research began to suggest that the existence of conflict was an opportunity to showcase new perspectives. In order to take advantage of these opportunities and the divergent ideas that spawned them, some methods (such as the "WinWin" system process model proposed by Boehm et al, 1994) began to include negotiation as a legitimate and worthwhile contributor to the requirements elicitation process.

Furthermore, the requirements elicitation activities must be seen as group tasks as they involve several stakeholders and deal with a considerable number of desires and expectations. Like any other group activity, it encompasses individual and group issues demanding a set of skills from managers and mediators to control the dynamics of such tasks.

Now that we understand that social issues are as relevant as technical issues, especially in a complex environment such as the military one, it is important to delve further into the complex task of requirements elicitation in relation to conflicts.

5.2 Groups and Conflicts

In many organisational situations group tasks are required. Sometimes to increase productivity or creativity, sometimes because there are various people involved. The requirements elicitation process is one that depends upon the involvement of several people, as long as there is more than one organisation and more than one team involved. Consequently, this complex activity will always rely on group interaction. Therefore, it is important to understand more about groups and their features. McGrath (1983) noted that groups and their memberships can be observed from varied angles. Furthermore, he mentioned that groups are such a complicated subject that it would not be possible to comprehend all aspects and concepts involved at the same time. McGrath developed a framework (Figure 19) in order that the whole might be understood so that the numerous factors involved could be studied individually. The framework encompasses several variables. The core variable is the “group interaction factor” and focuses on the way the

members interact and behave together. The second factor, "properties", relies on the fact that all individuals have their own beliefs, inner values, habits and characteristics that guide the way they see and act in the world. Another important aspect is the pattern of relations among group members, the existence of prior relationships among them, and the way they influence one another. These patterns are called "group structure". A fourth aspect equally relevant is that any group activity takes place in a certain "environment", and is influenced by particular environmental characteristics. Finally, any group is gathered to perform some "tasks". Each task has its own characteristics that can also influence the way the interactions happen. As can be noticed in the framework, all these factors and their interactions make a complex structure that shapes the way the group will interact.

However, these factors are not the only concern in group interaction. The process itself and its results are forces that lead to changes in these input factors. While group interactions are happening, they induce changes on members themselves, in the group structure, in relationship patterns, and in the way the group interacts with the task and the environment. In other words, the group changes its members and the members change the group. This makes the understanding of groups even more complex.

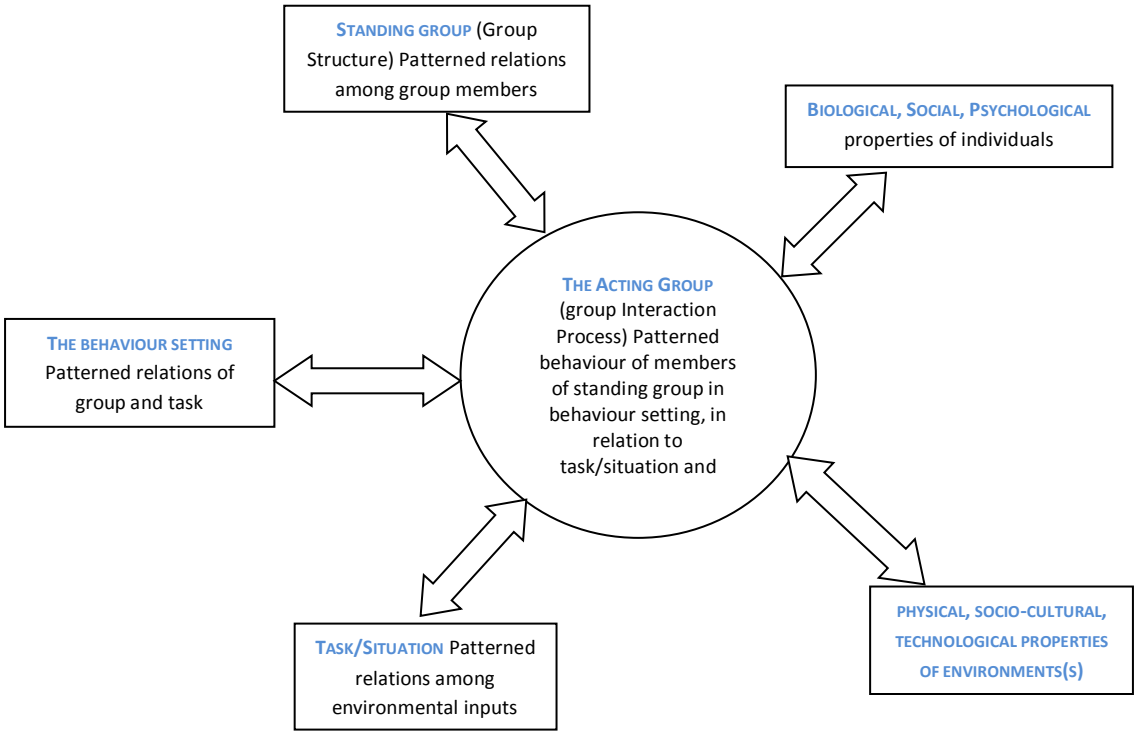


Figure 19 – Factors involved in group interaction (McGrath, 1983)

With all the factors that impact on group interaction, it is important to understand better the tasks performed by requirements elicitation groups. McGrath (1983) defines four general processes for group tasks: to Generate (alternatives), to Choose (alternatives), to Negotiate and to Execute (Figure 20). 'Generate tasks' encompasses generating plans and generating ideas. 'Choose tasks' includes intellective tasks and decision-making tasks. 'Negotiate tasks' involves by resolving conflicts of viewpoint or resolving conflicts of interest. Finally, 'Execute tasks' include resolving conflicts of power and psycho-motor tasks against an objective or standard excellence (McGrath, 1983). These tasks are divided in the horizontal axis of the figure in conflicts or cooperation. These definitions show that when dealing with group tasks, either we are trying to solve different types of conflicts, or we are relying on cooperation. In fact, to cooperate successfully in many situations will depend

upon first resolving conflicts. In the case of requirements elicitation, the activities performed permeate both sides. During Requirements Elicitation there are activities relying on cooperation, such as generating ideas, as well as there are others that rely on dealing with conflicts such as solving conflicts of viewpoints and interests.

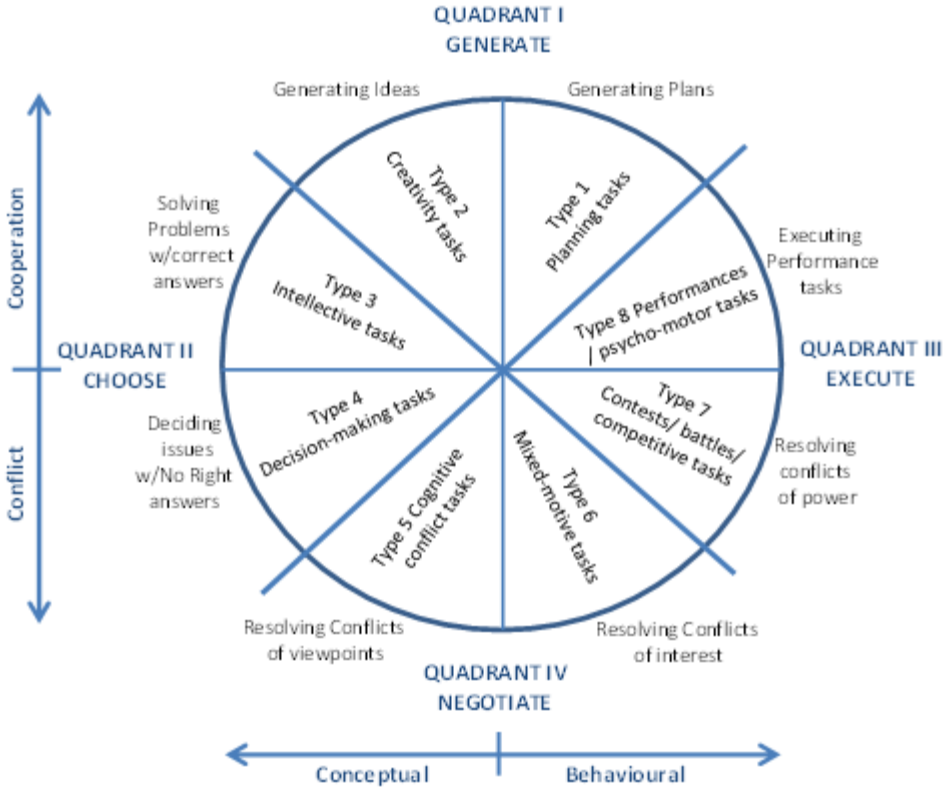


Figure 20 – Classification of Group tasks (McGrath, 1983)

Similarly, Easterbrook (1994) states that "conflict is a common phenomenon in interactions both between individuals and between groups of individuals and that to assume an absence of conflicts is naïve. Inherent differences between individuals' experiences, personalities and commitment make the potential for conflict innate to any group of individuals".

According to Easterbrook, systems are built to facilitate individuals to work together, therefore they must be developed focusing on collaboration. To make it possible it is essential to discover obstacles to collaboration and thus to know how to enhance collaborative work even when conflicts arise. Consequently, dealing with conflict satisfactorily is one of the key points to make group work viable and effective (Easterbrook, 1994). His study can be applied specifically to a requirements perspective. Requirements meetings are group meetings that involve various groups with different collective and individual goals and expectations, which generate conflicts. Thus, understanding conflicts can be used as a tool to support the choice of requirements elicitation techniques.

To Goguen (1993), the goal of requirements engineering is to determine what functionalities a system should have in order to be successful. Furthermore, he states that "the most vexing difficulties are social, political and/or cultural." Thus, an aspect to be studied is the presence of conflicts in an organisation and their influence in organisational tasks such as system development. According to Horowitz (1962), "there must be a social equilibrium within any organisation." The way to achieve this equilibrium is getting a consensus about the tasks to be performed and the way they should be performed. There are several definitions for consensus (Horowitz, 1962). In the role theory, "consensus is seen as an accord between role behaviour and role expectation" (Gross, 1959). From game theory comes a more collaborative perspective that says: "two parties or groups are playing to gain a maximum, but they are prepared to settle for less within the recognized limits" (Green, 1960). While the first definition describes the difficulty in trying to use the right role at the right moment, the second definition demonstrates the agreement needed on requirements.

Consensus Theory (Horowitz, 1962) states that "an increase in the amount of consensus yields an increase in functional efficacy; while inversely a decrease in the amount of consensus creates social disorganisation and dysfunction." However, there are conflicts that will make consensus more difficult to achieve. "A member of an organisation is said to have conflictful behaviour if he consciously, but not necessarily deliberately, blocks another member's goal achievement" (Poundy, 1967). Both theories from Horowitz and Poundy come from the 1960s, however, are still valid in any discussion about conflicts in a general form, and are still used by several theorists to define conflicts, their causes and other closed issues.

Conflict Theory (Poundy, 1967) defines three types of conflict between the subunits of formal organisations:

- Bargaining model - this model deals with conflicts among groups in dispute for insufficient resources;
- Bureaucratic model – this model analyses superior-subordinate conflicts, or in general, conflicts between different hierarchical levels;
- System model – this model is directed to a conflict on a common level or conflict between departments that have some related work but with different objectives and which therefore require some kind of coordination between groups.

These three forms of conflict will be taken into account in this study either for the type of tasks presented in requirements elicitation or due to the specificity of the military domain.

The bargaining model is present as any requirements elicitation involves at a certain point a bargaining conflict. As long as there are two groups of interest, client organisation and development organisation, deciding upon a system with limited human and material resources will necessitate renegotiation and perhaps compromise. The client will usually try to get as many functions as possible with the available resources, while the development organisation will try to keep this amount to a minimum to guarantee the completion of the contract. According to Poundy (1967), a significant aspect of interest-group conflicts is that negotiation is usually a responsibility of representatives who must deal with two conflicting problems: "(1) securing consensus for the negotiated solution among the respective group members, and (2) compromising between the demands for flexibility by his opposite number and the demands for rigidity by his own group." The level of perceived conflict will rise close to the time limit for the new system, and interest-group conflicts are characterised by schedule pressures (Poundy, 1967).

From the bureaucratic model perspective, as the military domain is based on hierarchy, any relationship within the environment will have, to certain extent, some kind of hierarchical interaction. According to Dahrendorf (1959), "although conflict is endemic in society, it arises primarily from the structure of authority." "Vertical conflicts within an organisation arise because superiors attempt to control the behaviour of subordinates, and the subordinates resist such control" (Poundy, 1967). A conflict relationship is made from a sequence of interlocking conflict episodes. During requirements elicitation in a military environment at least three hierarchical relationships are present: hierarchical difference between client and requirements team; hierarchical difference between stakeholders; and the hierarchical difference between requirement team members. Because hierarchy is present in

all levels of a military organisation, usually lower ranks will reduce the interaction with the higher ranks by obeying orders without giving any discordant opinion, which is dangerous for the requirements elicitation as just some of the stakeholders will actually be heard. These conflicts for control together with the dispute for resources are an important issue during requirements elicitation tasks.

From the system model point of view usually in an organisation there is more than one sector dealing with the same information that will be manipulated by the systems to be developed. This often leads to an interrelationship between the sectors. As noted by Poundy (1967), normally the sectors involved have different goals, and thus it will be a source of conflict. To resolve these conflicts, there are two ways: reducing the difference of goals or reducing the interrelationship (Poundy, 1967).

A fourth source of conflict mentioned in Conflict Theory is the role conflict model (Poundy, 1967). The role conflict model treats the organisation as a set of roles, each role inserted in a specific subset of roles that generate demands for the person to whom the task is assigned. The conflict is perceived when a role receives, from its subset of roles, requirements that are not compatible with what is expected of that role (Poundy, 1967). Easterbrook (1994) states that well-characterised roles diminish conflict. Furthermore, in many situations, there is the occurrence of multiple roles or 'selves'. Mintzberg (1973), for example, performed a study with five managers and concluded that managers in their jobs perform at least ten different roles: "Figurehead, Leader, Liaison, Monitor, Disseminator, Spokesman, Entrepreneur, Disturbance Handler, Resource Allocator and Negotiator". Marks and MacDermid (1996) mention the theory of role balance wherein a person needs to define

a balance between his roles. According to their study, research establishes that these roles are hierarchically ordered and one role will always be the prevailing one. On the same subject, Mead (1964) notes that if there is a conflict between two or more of our 'selves', we have a problem to solve. In the military domain, all military personnel have to deal with at least two different roles: the military or hierarchical role and the technical or expert one. Douglas (1986) states that a manner of sustaining social order is through the social features and standards of a society. According to Goffman (1961) "authority in a total institution is directed to a multitude of conduct: dress, deportment, manners, that constantly occur and constantly come up for judgement." Thus, the military environment with all its characteristics and patterns, leads its military personnel to use almost always the military role. When there is a technical task, such as Requirements Elicitation, and the expert role should be the primary one, there is a conflict of roles and usually the result of the task is negatively impacted by this conflict. Decisions that should be made based on technical expertise end by being made based on hierarchical authority.

All these conflicts can restrain the organisation's productivity, stability or adaptability (Poundy, 1967). However, when conflicts occur, a natural motivation for reducing them arises, and thus the parties have consciously endeavored to employ initiatives to eliminate these conflicts. The most natural reaction to dealing with conflict is an attempt at a decision by consensus and not a direct solution to the problem (Poundy, 1967). Another common occurrence is the institutionalisation of solutions to deal with recurring conflicts. As previously mentioned, an institution that has authority as one of its pillars will increase the level of authority to resolve conflicts (Douglas, 1986).

Similarly, but in a software perspective, Goguen (1993) analysed social issues in the REng discipline. To him, as this discipline relies on interactions of different teams, there are several sources of social issues and possible conflicts within and between these teams. According to Goguen (1993), three groups participate in the requirements process: the client organisation, the requirements team and the development team. "This three-fold division gives rise to six areas where social issues can arise: within the client organisation, within the requirements team, between the client and the requirements team, between the development team and the requirements team, within the development team and between the development team and the client" (Goguen, 1993).

Within the client organisation, Goguen (1993) mentions that in large institutions with many different and sometimes discordant areas it is difficult to determine who should be understood as 'the client'. Furthermore, many times the client representatives that interact with the analysts are not the real users. The people who will actually use the system must be found, otherwise they may feel excluded and could subvert the system. In our case study, the hierarchical issue arose as the lower ranks avoided participating or giving opinions due to hierarchical pressure, or the client representative, many times the highest rank among the client team, decides to provide all the ideas and goals for the system.

Regarding the Requirements Team, Goguen (1993) states that "the team itself is a social organism." Among the issues present, he mentions the organisation of work, what procedures and notations should be used and that each group participant has particular perceptions of the institution. From our observations, another issue arose: the different

backgrounds among the members (military and civilians) and the consequent communication difficulty.

Goguen (1993) mentions that professional and commercial relationships between the client organisation and the requirements team underlie any requirements task. Requirements are the clauses in the contract between these two teams that will guide the development. He mentions the differences that will take place once the system is ready such as some workers losing their jobs or being transferred to other departments. It is also common for stakeholders to hide some aspects of their work or to behave differently in the presence of strangers. He mentions in addition, the difficulty in managing the constant changes to requirements (Goguen, 1993). Again from our observations, the different backgrounds of client members and the requirements team create communication barriers. Another issue present from our case study was the hierarchical difference between client team and requirements team as long as both are from the same organisation and subject to the same hierarchical system. Usually, as the client representative has a much higher rank than the manager of the requirements team, he/she tries to drive the elicitation. Furthermore, this issue could be amplified by the cultural difference existing within the requirements team and between the requirements team and the client team. The extremely high turnover on the client team also appeared among the issues as each new client representative comes with different ideas and expectations. As mentioned by Wallington (2004), a new boss will almost always bring a new set of expectations and priorities and will want to put his/her own stamp on the organisation. The change of a member of the staff or the boss of a group represents a social change. A social change is a source of intergroup conflict (McGrath, 1983). Changes are constantly shifting the ground of possibly antagonistic ideas and positions of leadership and predominance over

the group. The potential value of changes in the social environment generates a new order of things with new demands, new difficulties and new misunderstandings (Mack, Snyder, 1957).

Goguen (1993) opines that usually the requirements team and the development team do not have the opportunity to exchange ideas because often times the requirements team is dismissed after completing its task. In the case of our study, normally the requirements team is part of the development team thereby reducing the impact of this issue.

Within the development team, Goguen (1993) mentions that social issues can influence the quality and quantity of work. Finally, between the client and the development team he mentions that problems can arise if this relationship is denied.

From all the above, it can be concluded that several kinds of conflicts can arise during requirements elicitation. There are intragroup conflicts between the members of each group involved, there are intergroup conflicts between those groups, and there are also intrapersonal conflicts that involve one person and his inner perceptions of himself. All these sources of conflicts make requirements elicitation an extremely complex process. Something must be done to deal with all these conflicts.

In the presence of any conflict, something must be done and managers may pursue one of at least two approaches: change attitudes or change behaviour (Neilsen, 1972). If they change behaviour, open conflict is often reduced, but usually groups still dislike each other; the conflict may simply become less visible as the groups are separated from each other. Changing attitudes, on the other hand, often leads to fundamental changes in the ways that

groups get along (Steers, 1988). Steers presents a set of conflict-resolution strategies to solve conflicts. Some of the resolutions focus on changing behaviour such as: physical separation; increasing specifications of rules and regulations; and limiting intergroup interaction. Among the resolutions to change attitudes the author mentions: confrontation and negotiation; third party consultants; intergroup training. As noted also by Poundy (1967), if conflicts are significantly simpler, participants will try to resolve them within the own group through conciliation (changing attitudes). On the other hand, when conflict is intense, usually the way to solve it is by eliminating any relation to the other party (changing behaviour).

However, the way to deal with conflict will also depend on each stakeholder involved, as each person has a particular way of dealing with conflicts. Thus, to choose between changing attitudes or behaviour you must understand the stakeholders individually and as a group. Thomas (1976) presents a model with five different approaches that a person can use in the presence of conflict based on two-dimensional space of possibilities (Figure 21). The two dimensions are the desire to attend your own expectations (assertiveness) and the desire to satisfy another's expectations (cooperation). Below there are the five possible personal approaches to a conflict:

- Competitive. This mode is characterised by a member trying to dominate the activities, without considering the opinions of others.

- Collaborative. In this way participants take into account the differences to seek a decision that benefits all.

- Avoidant. When the conflict is perceived, but avoided by one or more participants.

- Accommodative. When one side decides to sacrifice some own interests in favour of the interests of the other side.

- Sharing. When each party abdicates some interests to reach a common solution.

Thomas concludes that each of these five styles is convenient in a certain situation; consequently, parties should choose the approach based on circumstantial factors.

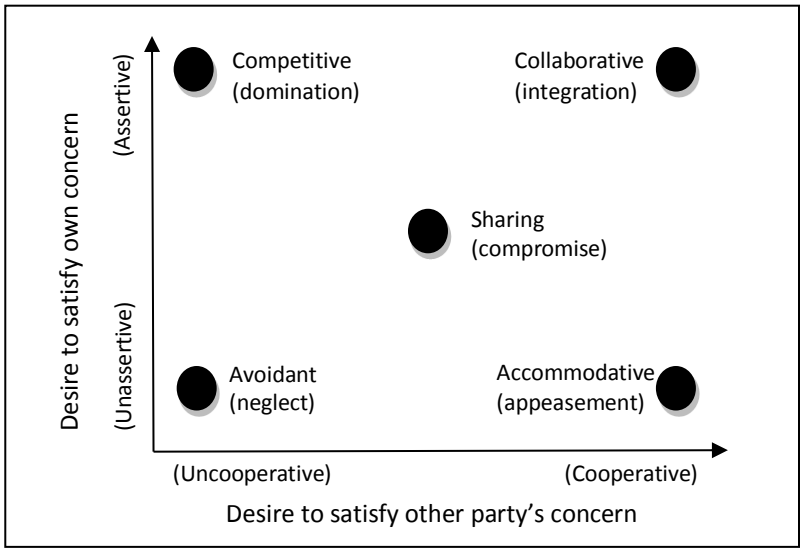


Figure 21– Behavioural modes of tackling conflict with (in brackets) the outcome sought in each mode. Thomas (1976)

These differences in the way of dealing with conflicts increase the difficulty facing managers in controlling the conflicts in order to improve the efficacy of group tasks. Furthermore, such characteristics can be exaggerated in a military environment. Due to its

characteristics, this environment contributes to the increase in the incidence of some behaviours such as the competitive or accommodative modes in the presence of hierarchical difference. Dahrendorf (1959), for example, mentions that "conflict is endemic in society, but it arises primarily from the structure of authority", one of the pillars of any military organisation. Thus, and mainly in institutions such as a military one, it is important that the manager or the person responsible for the group can foresee the approach used by each member of the group in order to control the conflicts and the conflictful behaviour to guide the energy of the group towards creativity and consequently, better solutions.

According to Bailey (1982) predicting human performance in any situation requires an understanding of the human, the activity being performed, and the context in which it is performed. The analyst has different degrees of control over each of these three elements. The interaction between the elements is also critical. Although to Bailey, the human is the most complex of the three elements mentioned. The performance of an individual when performing a task can be affected by several inner issues that exist for that individual, besides the conditions of the task or the setting. Thus, it is important that the analyst verify which conditions could influence positively or negatively each individual and use it as a source of information to guarantee the effectiveness of the elicitation tasks. (Bailey, 1982). As mentioned earlier in this chapter, it is also important that the analyst identifies which modes of tackling conflict each stakeholder will normally use. Thus, a stakeholder analysis must be performed to identify the possible sources of conflict as well as how each stakeholder handles existing conflicts that could disturb the activity of requirements elicitation.

> *Behaviour generally is predictable if we know how the person perceived the situation and what is important to him or her. While people's behaviour may not appear to be rational to an outsider, there is a reason to believe it usually is intended to be rational, and it is seen as rational by them. An observer often sees behaviour as non-rational because the observer does not have access to the same information or does not perceive the environment in the same way*. (Lawler and Rhode, 1976)

According to Freeman (1984), any group or individual in an institution that can directly or indirectly influence or be influenced by the accomplishments of the institutional objectives must be considered a stakeholder. MacCaulay (1994) identifies four categories of stakeholders in any computer system: those responsible for design and development; those with a financial interest; those accountable for introduction and maintenance; and those who have an interest in its use.

Sharp et al. (1999) developed an approach for stakeholder analysis that presents as a starting point what they call baseline stakeholders. From these, they select supplier and client stakeholders where the former provides information or supporting tasks to the baseline, and the latter provides processes or inspects the products of the baseline stakeholders. Besides the baseline stakeholders, they identify the so-called satellite stakeholders which interact with the baseline stakeholder in a variety of ways such as: communicating, reading a set of rules or guidelines, searching for information among others. Their approach is based on five steps as follows:

1. Identify all specific roles within the baseline stakeholder group;

2. Identify supplier stakeholders for each baseline role;

3. Identify client stakeholders for each baseline role;

4. Identify satellite stakeholders for each baseline role;

5. Repeat steps 1 to 4 for each of the stakeholder groups identified in steps 2 to 4.

Once the stakeholders are identified, it is important to decide which is the best way to obtain requirements from each stakeholder. Authors agree that each stakeholder has its own characteristics and consequently, will deal with situations in a different manner, as shown in the following statement.

> *A fundamental aspect of collaborative work is that individuals are not identical, and will approach the same task with differences in their expectations, goals and preferred styles of working. They will have different amounts of time to commit and to the resolution of a problem, and even different notions of what the problem is. These differences will at times lead to conflict.* (Easterbrook, 1994)

According to Robbins (1998), the conditions under which conflicts arise are the following:

- Communicational, including insufficient exchange of information, noise, and semantic differences that arise from selective perception and difference of background.

- Structural, which includes the goal compatibility of members of the group, jurisdictional clarity, and leadership style.

- Personal factors, including individual value systems and personality characteristics.

Consequently, stakeholder analysis should be the starting point for the proposed solution to identify the conditions aforesaid.

Everything presented above leads to the impression that conflicts are necessarily disruptive. However, conflicts are not necessarily destructive; they can be functional or dysfunctional depending on whether it facilitates or inhibits the organisation's productivity, stability or adaptability (Poundy, 1967). Functional conflicts happen when two people have different opinions or desires about a subject but both want the subject to be resolved, so having a common goal. Dysfunctional conflicts are related to emotions, fears and aversions, not necessarily being related to a subject. Conflicts caused by emotions are more difficult to solve in a cooperative way than functional conflicts (Deutsch, 1973). Destructive or dysfunctional conflicts tend to expand and escalate independently of their initial causes. Furthermore, it tends to continue even after becoming irrelevant. According to Deutsch (1973), this expansion happens on several levels: the size and numbers of issues involved; the number of motives and participants implicated in each side of the issue; the size and number of the principles and precedents that are perceived to be at stake; the costs that participants are willing to bear in relation to the conflict; the number of norms of moral conduct from which behaviour toward the other side is exempted; the intensity of negative attitudes toward the other side. "The processes in which these conflicts escalate produce

some effects such as impoverishment of communication; hostile attitudes; oversensivity to differences" (Deutsch, 1973). There is the notion that the solution has to be imposed by one side by means like superior force. Such a way of solving the conflict can be easily used in an environment where hierarchy is one of the pillars. With all these effects and several consequences it becomes clear that this a type of conflict must be avoided.

On the other hand, several authors note that conflict can also be productive (Poundy, 1967). Deutsch (1969), for example, suggests that conflict can lead to an increase in the motivation to resolve issues and move beyond a standard solution. Thomas (1976) found that satisfaction of group participants can be increased when they notice that their divergent opinions will not disrupt the group. Finally, Hall (1971) mentions that "conflict effectively managed is a necessary precondition for creativity." These authors are all referring to functional conflicts, where the main issues are divergent opinions or desires. According to Deutsch (1973), "there are three psychological elements in this process: the existence of motivation to solve the problem; the development of the conditions that permit the reformulation of the problem once an impasse has been reached; and the concurrent availability of diverse ideas that can be flexibly combined into novel and varied patterns." In addition, to Deutsch (1973), a variety of reasons generate a fertile environment for the creative process, but all have in common the fact of providing a setting free from threats and personal criticism. Threats lead to a reduction of tolerance to ideas outside of the standard; excess pressure leads to stereotype and to blunt the way of thinking (Deutsch, 1973). In the case of the requirements process, creativity is essential to discover the most effective and efficient solution available to the problem. However, in many cases, the group process will create an inhospitable environment for creative ideas. Thus, it is important that the

requirements team attempt to develop an adequate environment inside the group of stakeholders in order to encourage functional conflicts, but keeping control at all times.

5.3 Issues related to conflicts

5.3.1 Communication

"Lack of good communication can easily turn a System Information Project into a modern-day Tower of Babel" (Frese, 2003). According to Steers (1988), the organisation structure has an important influence on communication patterns and behaviour in the organisation. To explain such an influence the author uses the concept of communication networks that show all possible communications among a group of people. One of the communication patterns mentioned by Steers is named chain. The chain represents a pattern where most communication flows up or down a formally defined chain of command, like any military organisation. The chain pattern exhibits a considerable centralization of power and authority, and a moderate speed of communication. Furthermore, it does not lead to positive attitudes by employees, largely because of a lack of participation in decisions. Another disturbing characteristic mentioned about the chain pattern is the lack of commitment to the decisions of the members at the bottom of the chain since they have little input in the decision-making process (Steers, 1988).

As the requirements elicitation process relies on many interactions of different groups with different goals and expectations, communication is of crucial importance, and any communication difficulty must be addressed to increase the effectiveness of the requirements

elicitation process. All issues elicited during the data analysis are related to communication each in a different way. Hierarchy was mentioned as a factor that contributes to decrease collaboration and thus can be seen as inhibiting the possibility of communication of lower ranks. The turnover brings new people to the requirements elicitation task during the development with different ideas but also with a lack of knowledge of the sector or activity to be helped by the system. However, the issue that is most related to communication is "multicultural teams". Gathering team members from different backgrounds can lead to several misunderstandings during the requirements phase. Furthermore, when requirements are being elicited in an extremely peculiar environment such as the military one, this aspect will have an even bigger impact. Any environment has its social characteristics and patterns, but as mentioned by Goffman (1961) in a total institution these characteristics are exaggerated, and thus it is not just a matter of understanding jargons and acronyms, rather it is a matter of comprehending a whole conduct aspect that guides all behaviour in the domain.

To have a complete understanding of any interaction it is usually necessary to have at least some previous knowledge on the subject or the environment where it takes place. Wittgenstein (2009) in his work "Language Games" said: "some foreknowledge is necessary for understanding; we need to know the role that 'table' will play in our language game". In the same direction, Goffman (1974) uses the term frame in the following sense: " I assume that definitions of a situation are built up in accordance with principals of organisation, which govern events […] and our subjective involvement in them; 'frame' is the word I use to refer to such of these basic elements as I am able to identify."

Thus, in any interaction, there are frames applied by the participants which can lead toward an understanding or some misunderstandings (Goffman, 1974): “When an individual recognizes a particular event, he tends to imply in this response one or more frameworks or schemata of interpretation of a kind that can be called primary… the individual decides what is going on, choose the primary framework, given his particular interests."

However, Goffman (1974) also mentions that these frames are influenced by the culture of a group or organisation: “The primary frameworks of a particular social group constitute a central element of its culture, especially insofar as understandings emerge concerning principal classes of schemata." Consequently, an individual who is a stranger to a group has a bigger chance of misunderstanding the utterances, as long as he might apply the wrong frame: “The various kinds of ambiguity, including vagueness and uncertainty have their counterpart in error that is beliefs, uninduced and erroneous, as to how events at hand are to be framed. …the individual act on the basis of wrong premises. He misframes events " (Goffman, 1974).

Another important concept related to frames is “Footing” (how the mode and frame of conversation are determined and how that is controlled or not by participants) (Goffman, 1988). Thus, the participants can also try to control an utterance by using a specific frame.

In the area of system development, Lin and Silva (2005) argue that “management of information system adoption is a social and political process in which stakeholders frame and reframe their perceptions of an information system.” Orlikowski and Gash (2000) developed the technological frame analysis as a way to explain the Requirement process. According to their definition technological frames are “that subset of members’ organisational frames that

concern the assumptions, expectations, and knowledge they use to understand technology in organisations. This includes not only the nature and role of the technology itself, but the specific conditions, applications, and consequences of that technology in particular contexts".

With all these concepts and statements, it can be seen that a multicultural group eliciting requirements or just interacting will face bigger problems in gaining a complete understanding of one another as many times they will not share the same frames, especially if the client team is also considered.

5.3.2 Collaboration

With regard to considerations about the military environment and the existing sources of conflict during requirements elicitation, a more collaborative approach is needed. Solutions such as participatory design have been used in many environments. In the military environment, efforts to reduce the hierarchical interaction and increase collaboration have been tried. The hierarchical relationship, although indispensable in the battlefield, can reduce effectiveness on technical tasks. The French Ministry of Defense, for example, is using an approach named LTO (Laboratoire Technico-Operationnel) with some principals to support cohesion and consensus as well as anonymity of ideas (DGA, 2009).

All these issues can be understood when we observe the participation framework (Goffman, 1988). According to this framework, there are two groups of roles involved in any social interaction: production roles and reception roles. In one group, there are three production roles: animator, author and principal depending on the relation to what will be said. The

reception roles are: addressed recipient, unaddressed recipient, over-hearers and eavesdroppers related to the intention of the utterer to address or not a hearer, and the right to be present at the meeting. Hall (1959), points out that "the area of non-verbal language (or language of behaviour) is equally important as the use of the spoken word." According to Steers (1988), this silent language can be seen in a multitude of ways:

- Time. An indication of the importance of an individual may be perceived by the time that this person should expect to be received by another.

- Space. The working space of an individual provides several tips on his/her importance to the organisation such as: the size of the table, if this person has an individual office or advisors.

- Dress. The simple observation of the way someone dresses, gives us an idea of the person. Do the clothes indicate a manager or a worker? A military officer or enlisted person?

- Physical appearance. Such factors as general look, posture, and sympathy tell us signals (rightly or wrong) about the individual's temperament.

- Titles. Our perceptions of others can be affected by the titles or pre-names they possess.

- Interpersonal interaction. The way people treats us shows their hierarchical superiority or inferiority as well as his/her intentions.

Specifically, in a total institution, Goffman (1961) stated that authority is directed to a multitude of conduct rather than just spoken language. Thus, when using group techniques that gather several stakeholders in a meeting such as Workshops, Brainstorming, JAD or Focus Groups, there are numerous ways of being addressed rather than just by speech. The military uniform conveys authority; a gesture can carry an implicit order; a tone in the voice can be directed to a different recipient than the speech. Consequently, any type of group technique must be carefully used to keep the focus on the speech that can be more easily controlled by the facilitator rather than subliminal messages that can be unnoticed and inhibit participation.

According to Downs (1967) horizontal or lateral communication is more frequent than vertical communication in organisational settings since people feel more open and free communicating with peers than with superiors or subordinates. Furthermore, because peers are most likely to share a common frame of reference, message distortion is less likely. This will be exacerbated inside a strong hierarchical structure. In the presence of an authoritarian management approach, it will be difficult for a subordinate to express opinions and ideas in front of his/her superiors, especially in an environment such as the military that has hierarchy as one of its pillars.

Consequently, to make a collaborative task possible in such an environment, there must be a careful selection of techniques. However, it is not enough to select the right technique, but also to guarantee the perfect foundation to make it efficient. Things such as the wrong group, the wrong moment to perform a technique or even the wrong posture to guide a meeting can make a useful technique fail, even in perfect conditions. Thus, there

must be a careful selection of the technique followed by an equally careful preparation of the groups, tasks and places to perform each technique in a proper way.

5.3.3 Power and Politics in Requirements Engineering

Power and politics have been discussed over many years in several areas of human knowledge. However, little attention has been given to these subjects within the area of system development or specifically requirements engineering. As a group process that deals with desires and perspectives, naturally stakeholders will use the tools at hand to make their desires prevail. Thus, it is natural that the requirements process will be highly influenced by power and politics. Rost (2004) stated that there are political reasons for project failure such as: a project can threaten someone's job; fear of worsened work conditions; disagreement on the lead assignment to the project; fear of losing power over the work or resources; fear of more control over himself/herself. Thus, power and politics must be a concern of requirements engineers.

To Bergman (2002), politics relevant to Requirements Engineering can be divided into functional politics and resources politics. The first one deals with the functional goal of the system, while the other is related to the control of the resources needed to build the system.

In the sixties, several studies were undertaken to study power and politics in various areas. French and Raven (1959) defined five sources of power:

- legitimate – the right to give orders, and expect compliance and obedience;
- reward – ability to offer rewards to another for compliance;
- expert – person's superior skill or knowledge;
- referent – from personal characteristics
- coercive – power to punish others for noncompliance;

Later, Hersey et al (2000) included other two sources of power: the connection power and the informational power. Connection power or networking is related to using interpersonal relationships to influence decisions or attitudes. This source of power is based on the organisation's powerful personnel that a person knows. Informational power is based on the ownership of information that is important to others or to the organisational in general. In the so-called information era, this power is extremely relevant.

Based on the different sources of power, Milne and Maiden (2011) have suggested an approach to map the flow of power in an organisation. Their method is a simple diagram that shows the power that each stakeholder has over other stakeholders (Figure 22). However, such a diagram is not easy to create for any institution as most types of power are not explicit. In the military domain legitimate, coercive and reward powers are regulated, making the diagram easier to be built. Independently of the diagram, this research shows that attention must be given to the way power and politics are handled in an institution in order to guide requirements elicitation. For example, the selection of personnel that will attend a requirements meeting can be strongly affected by the power relation between the stakeholders. However, the study does not consider the political relationship between client and requirements teams, mainly when they belong to the same organisation. When the

development is in-house, requirements and client teams are under the same structure of power leading to political issues between these teams.

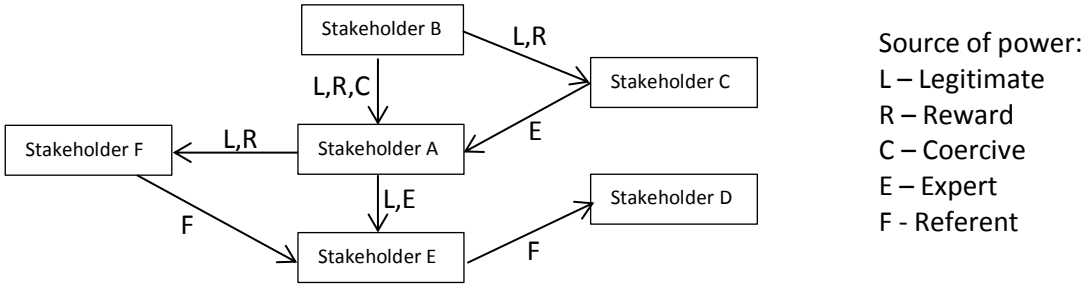


Figure 22 – Social network with sources of power (Milne, Maiden, 2011)

Bergman et al (2002) state that rather than the normal assumption of REng being that the need for a system is a problem searching for a solution, many times specialists invert the order to solutions searching for problems in order to make their own competencies useful, using their organisational power to force this situation. As a result, the requirements process becomes a solution (based on the specialist skills) searching for a problem and after that a problem searching for a solution, which can lead to a system that is not in the best interest of the organisation.

Furthermore, in agreement with the system model from the conflict theory presented earlier in this chapter, Bergman et al (2002) states that, especially when the system involves several departments with several authorities, rather than the expected organisational goal congruence, what is usually found is the existence of several different goals based on personal or department perspectives that will need some type of decision-making process in order to be resolved. This decision-making process can be based on negotiation, but usually is strongly guided by organisational power and politics. This perception drove Aurum and

Wohlin (2003) to consider the requirements process as a decision-making process that must consider political, social, cultural and organisational issues. In agreement, Bergman et al argue that "large-scale systems are constructed through a political decision-making process."

Consequently, it is essential that requirements analysts consider organisational politics when eliciting and developing requirements, especially regarding the consequences of the novel system to the power of the authorities involved. Tasks such as the selection of the elicitation technique must take into account possible political influences that can jeopardize the efficacy of the elicitation process.

5.4 Existing Literature on Requirements Engineering based on Conflicts

As mentioned in the previous section, requirements elicitation is a group task that should be performed in a collaborative way. Being a group task it is expected that several conflicts arise that could jeopardize the efficacy of this complex task. Thus, numerous authors studied the requirements phase through the point of view of conflicts.

According to Cooper and Bendiab (1998), "to support negotiation, there are seven fundamental areas, which require consideration: conflict detection – to limit and detect the conflict occurrence; conflict identification strategy selection – to identify a given conflict type; negotiation team formation – to form the team of agents who will select the resolution strategy of the identified conflicts; negotiation management – to control a collaborative/competitive negotiation session; negotiation monitor – to collect and store

information that is needed to support and guide the decisions; solution preparation – to use strategies for a group of agents to apply their own domain knowledge to provide an optimal solution to a conflict; and solution evaluation – to facilitate the rating of potential solutions." However, no research is devoted to discovering a means of limiting the occurrence of a conflict when it arises. The main research found on conflict focuses on conflict resolution, after the conflict is already occurred.

Easterbrook (1994b) presents an approach to solve requirements conflicts with computer-supported negotiation. His approach is based on the idea of separating people from the problem, in order to avoid the polarising nature of arguing from entrenched positions. This approach uses two or more viewpoints to be combined as a new viewpoint without discarding the primary ones (Figure 23). This new viewpoint is used to help in understanding the original viewpoints and understand the existing conflicts. Once the conflicts are understood, an exploration of possible solutions begins, and the resolution criterion is agreed and applied. Finally, a resolution is selected.

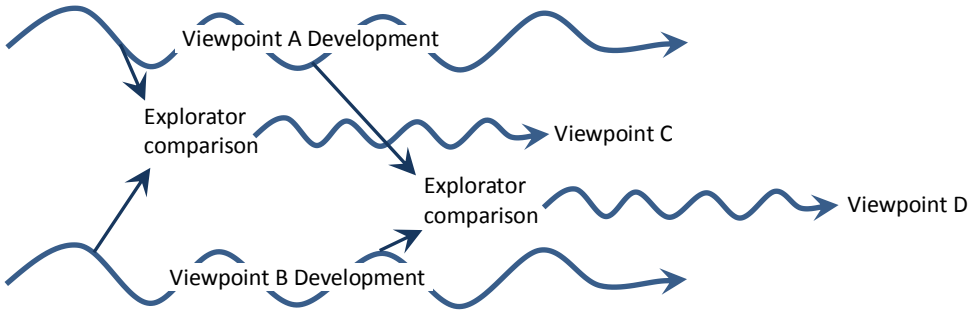


Figure 23 – Parallel Viewpoints of Easterbrook model (Easterbrook, 1994)

Robinson and Fickas (1994) developed an approach to assist requirements conflict resolution through a combination of multi-agent multi-criteria optimisation and heuristic resolution generation. Their approach is supported by a tool that has a conflict detector to compare specifications and a conflict resolver that provides analytic and heuristic alternatives for each conflict. Their tool provides a graphical interface where a skilled person can make an informed choice of alternatives and negotiation methods. The individual solutions are then merged into a non-conflicting specification that can be verified by the original requirements writers. Similarly, Spaccapietra (1994) presented a solution based on structural conflicts once they are already modelled in a database.

To Kontoya and Sommerville (1996) REng involves dealing with divergent ideas and perspectives on different levels of abstraction and developed a solution based on viewpoints. Their approach is based on services and states two types of viewpoints: 'direct viewpoints' and 'indirect viewpoints'. Direct viewpoints are related to users who obtain services directly from the system while indirect viewpoints are related to people who are influenced by the services provided by the system without having a direct interaction with it. However, they note that several existent methods depend on the analyst's skills to determine the core entities of the system to be developed without much or any support from the method. Finally, the paper mentions that discordant opinions about the requirements may create conflicts, although admitting that their approach does not permit an analysis of possible merging of viewpoints. In the same direction, Nuseibeh et al (1994) define a strategy to integrate multiple viewpoints. Their approach claims a definition of inter-viewpoint rules. After defining such rules, there is a tool to handle inconsistencies and structural consequences.

Sommerville et al (1998) mention that "REng is carried out to a tight schedule so it is not always possible to collect all desirable information, and that political and organisational factors influence technical requirements". They present an approach for requirements elicitation that follow the idea of establishing stakeholders' "win" conditions to support requirements prioritisation and negotiation. According to them these conditions are essential to guarantee that all important conditions (political, economic and technical) were taken into account properly when solving requirements conflicts. Their method follows the concepts of viewpoints, where a viewpoint is a representation of a stakeholder's perspective and modelling format that naturally differs from one individual to another. According to them attaching information gathered to the viewpoints avoids missing critical information and facilitates the linking of information to its source. Their method 'PREview' introduce the notion of 'concerns'. Concerns are high-level criteria related to business goals to guide the requirements process. The PREview process has as its first step the definition and elaboration of concerns that will guide the following steps of the requirements process. (Sommerville et al (1998). However, their method does not mention how to obtain those concerns.

Lamsweerde et al (1998) presented an approach to manage conflicts in Goal-Driven requirements engineering. According to them, several issues arise during the requirements phase as there are several stakeholders involved, each one with its own requirements and goals in mind. Thus, these issues must be solved at some point of the development process to guarantee a successful system. Their approach states that a constant refinement of goals up to the point where they become assignable requirements is necessary. Usually this process has several steps that can occur in a top-down or bottom-up way. Their process suggests using

interviews to gather the goals and analysis of existing documentation. They also mention the possibility of using other techniques, although without specifying when to use each one.

Klein (1991) states that there are two types of conflict situations: Competitive and Cooperative. In Competitive situations, parties focus on their own gain without any concern about a global solution if this solution will not bring specific benefits for themselves. In Cooperative situations, each party is ready to make concessions in order to find a solution that will benefit all groups involved. To Klein, each of these conflict situations has a specific strategy to deal with. His approach focuses on the cooperative situations, which also means that he does not address dysfunctional conflicts. The approach is based on the assumption that several conflicts happen in a standard way, which means that there are experts capable of indicating the best way to solve the conflict as long as the type of conflict is identified.

From an analysis of the literature presented above several common aspects can be distilled. The first aspect to be mentioned is related to the phase of the conflict's lifecycle that these approaches focus on. According to Easterbrook et al (1993) conflicts follow a set pattern. For Poundy (1967), a conflict episode has five steps (Figure 24): latent conflict (conditions), perceived conflict (cognition), felt conflict (affect), manifest conflict (behaviour), and conflict aftermatch (conditions).

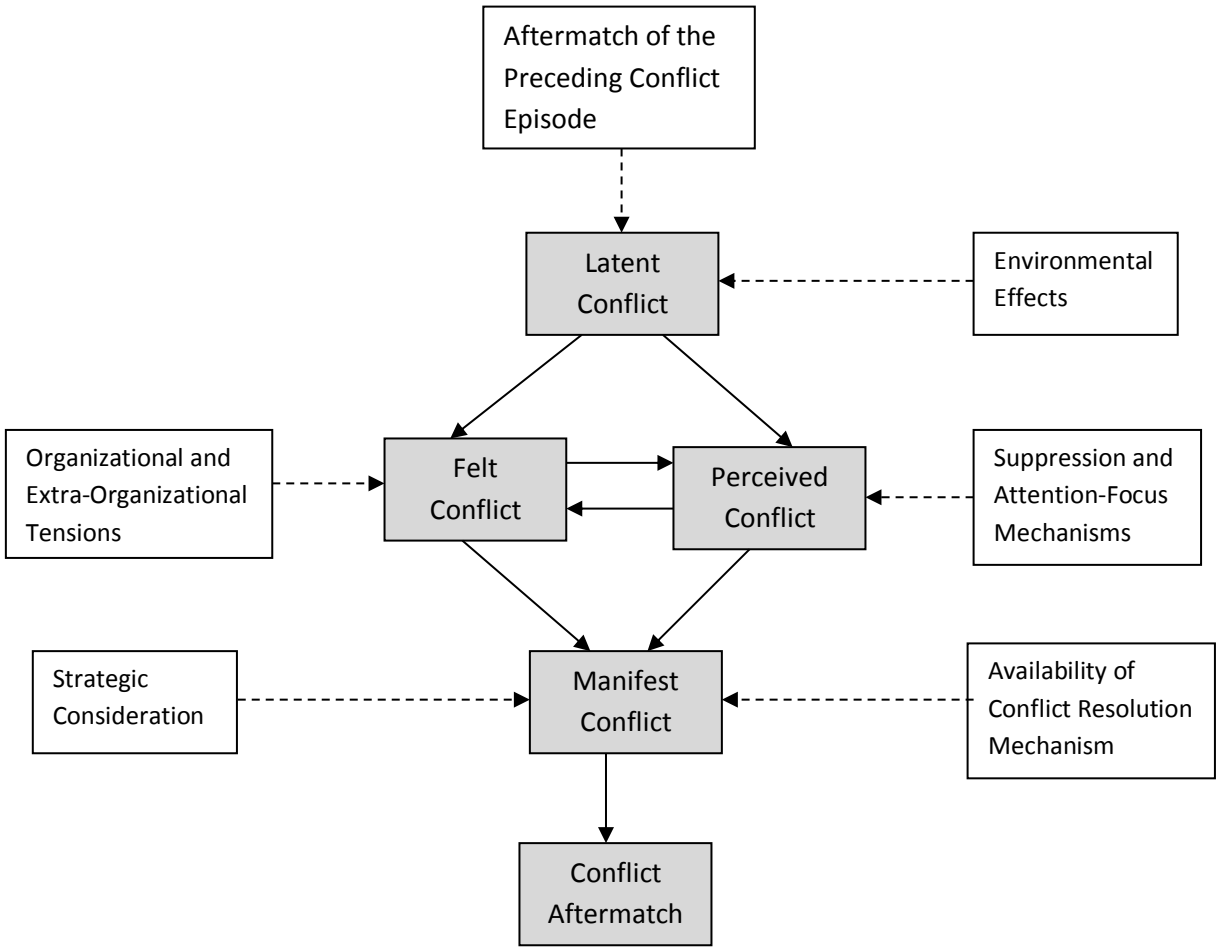


Figure 24 – The Dynamics of a conflict episode (Poundy, 1967)

After a careful research, all the literature found about conflicts and requirements deal with the conflicts when they are already manifest, without trying to prevent them from becoming manifest. Thus, the approaches present proposals to solve the conflicts that already exist (Figure 25).

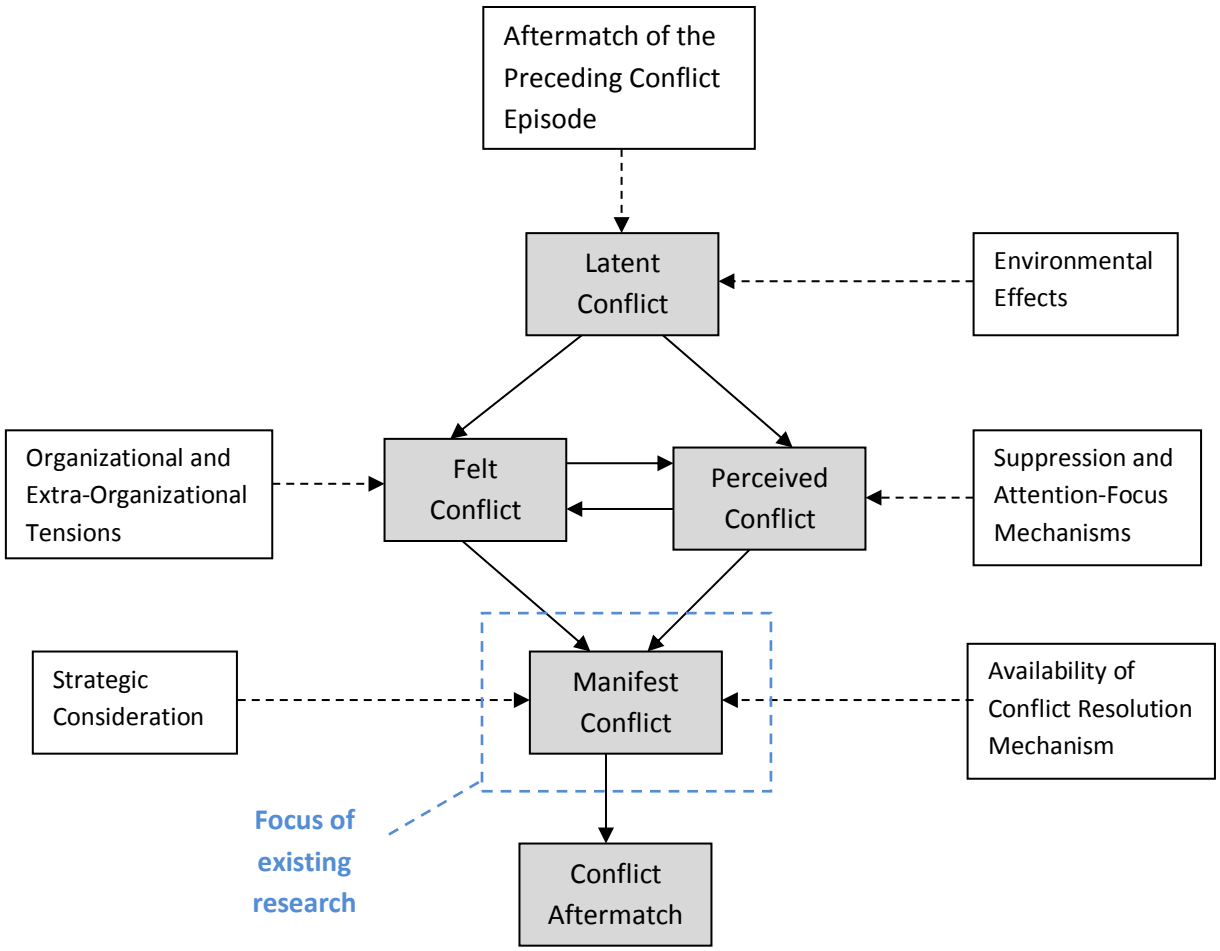


Figure 25 – Focus of the existing research on conflicts and requirements.

A second relevant aspect to be mentioned is that all the literature found focus on dealing with conflicts between client's perspectives of the system. Certainly, such conflicts are extremely important. However, the requirements phase has many other stakeholders involved. These approaches suffer from a lack of observation of conflicts that can arise amongst these other stakeholders. Requirements tasks depend on the interaction of three distinct groups (Goguen, 1993): Requirements Group, Development Group and Client Organisation, consequently, several social issues can arise from the interactions among and within all these groups.

Finally, as these approaches usually focus just on conflicts between client's perspectives of the system, consequently, they focus only on functional conflicts as they are conflicts of client's points of view towards the perfect solution. As previously mentioned, functional conflicts are positive and arise between stakeholders that aim to get to a common solution. Consequently, main goal of these approaches is to address a way to merge or select these different points of view in order to build a better solution. However, no approach was mentioned to deal with negative conflicts, the dysfunctional ones. Such conflicts can be extremely prejudicial for any group task and are not necessarily based on finding a better solution. In many cases, they permeate personal issues.

5.5 Summary

After understanding the existing gaps and issues to be addressed in the military environment and the specific task of requirements elicitation, and to gain the understanding that the main issues to be addressed by any solution should discuss social problems, a solution started to be constructed.

All the issues found in the literature as well as in the exploratory qualitative research can be translated as conflicts between stakeholders, including the requirements, development and client teams. Thus, to ground the solution this chapter expatiated on conflicts and their implications to requirements tasks.

As there is a consensus that requirements elicitation is a group task, the chapter started by discussing groups and group tasks. To manage group tasks is either to resolve conflicts or to search for collaboration between individuals.

Once groups are understood, it is important to understand possible conflicts in organisations. The conflict theory presented three conflict models: bargaining, bureaucratic and system. Each one is shown to be present in requirements tasks and in the military environment. Further a fourth model is presented: the role model. This model discusses the existence of multiple roles played by each individual and the consequent conflicts of these roles. The perspective of requirements groups as sources of conflicts were also discussed.

After that, communication and power were presented as a concern of the requirements elicitation activity as well as the perception of the task as extremely dependent on collaboration.

Finally, the existing research on conflicts related to requirements was presented. It was possible to conclude that all current work in the area focuses on conflicts that already exist. Furthermore, dysfunctional conflicts were not addressed by any of the studied research.

The next chapter will present the proposed solution that aims to fill the gap of the existing research, such as focusing on conflicts before they become manifest and the involvement of stakeholders other than the client team.

Chapter 6

The Framework

After understanding the problem in the three first chapters, conflicts were discussed in the previous chapter which will be the grounding for the solution.

As mentioned in the previous chapter, there are two possible consequences for a conflict. “If conflicts are relatively small, then the participants are likely to try to resolve the conflict within the context of the existing relationship. When the conflict is intense and there is no prospect for the re-establishment of equilibrium, the conflict will be probably reduced by dissolving the relationship” (Poundy, 1967).

Based on these two possible approaches to solve a conflict, a solution was developed, trying first to increase collaboration and thus search for a consensus. For this purpose, there are already many possible techniques such as JAD, Workshops, Focus Groups and even Participatory Design. Secondly, when the collaboration becomes hard to achieve other solutions will be used in an attempt to reduce direct interaction, consequently, reducing sources of conflict. As Easterbrook (1994) mentions “the more communication there is between people, the more opportunities there are for conflict”. Techniques such as Protocol Analysis, Ethnography and Observation, although not intentionally created for this purpose,

will reduce interrogative interaction, switching to a more informative interaction between the client representatives and the requirements team, while techniques such as User-led Requirements Construction (ULRC) will reduce interrogative interaction between client representatives and the requirements team by passing the responsibility of the requirements specification to the client.

The second contribution of the thesis, the framework, will be presented in this chapter. First, the idea of the solution will be discussed, with its component parts and reasons as well as the techniques selected. Later in the chapter we present the solution as it was provided to the Navy teams as a guideline.

6.1 Framework structure

Based on all the research and the initial findings a solution is being proposed. The solution followed the conflicts theory and the possible approaches in the presence of a conflict.

As mentioned in the previous section when a conflict is faced each person has a particular way of dealing with it. As a group task, requirements engineering involves several people, consequently providing several sources of conflict. As in the presence of the conflict there may be different behaviours among the stakeholders, it is important that the requirements analyst controls these behaviours in order to guarantee the efficacy of the requirements elicitation and to support an environment of general collaboration. This control can be even more relevant in a military environment because its characteristics can

contribute to enhancing some non-collaborative behaviours. Thus, the solution will try to give the control of the situation, and consequently the stakeholders' behaviours, to the analyst in this peculiar environment.

This control is difficult to obtain once the conflict is already manifest, especially if it is a dysfunctional conflict. Thus, while the usual approaches to deal with conflicts focus on "Mechanisms of Conflict Resolution" when the conflict is already in progress, our approach will focus on the early stages of the conflict life cycle trying to prevent dysfunctional conflicts from happening. The solution as a result, will focus on the latent conflict and the environmental effects in an attempt to prevent potential sources of dysfunctional conflict appearing in the initial stages of conflict: conflict felt, and conflict perceived; when the source of the conflict already exists and tensions are starting to grow (Figure 26). In this case, a change in the technique is still possible as the conflict is not a manifest conflict yet. However, the solution is not trying to eliminate all kinds of conflict. As mentioned in previous chapters conflicts can be functional or dysfunctional depending on whether it facilitates or inhibits the organisation's productivity, stability or adaptability.

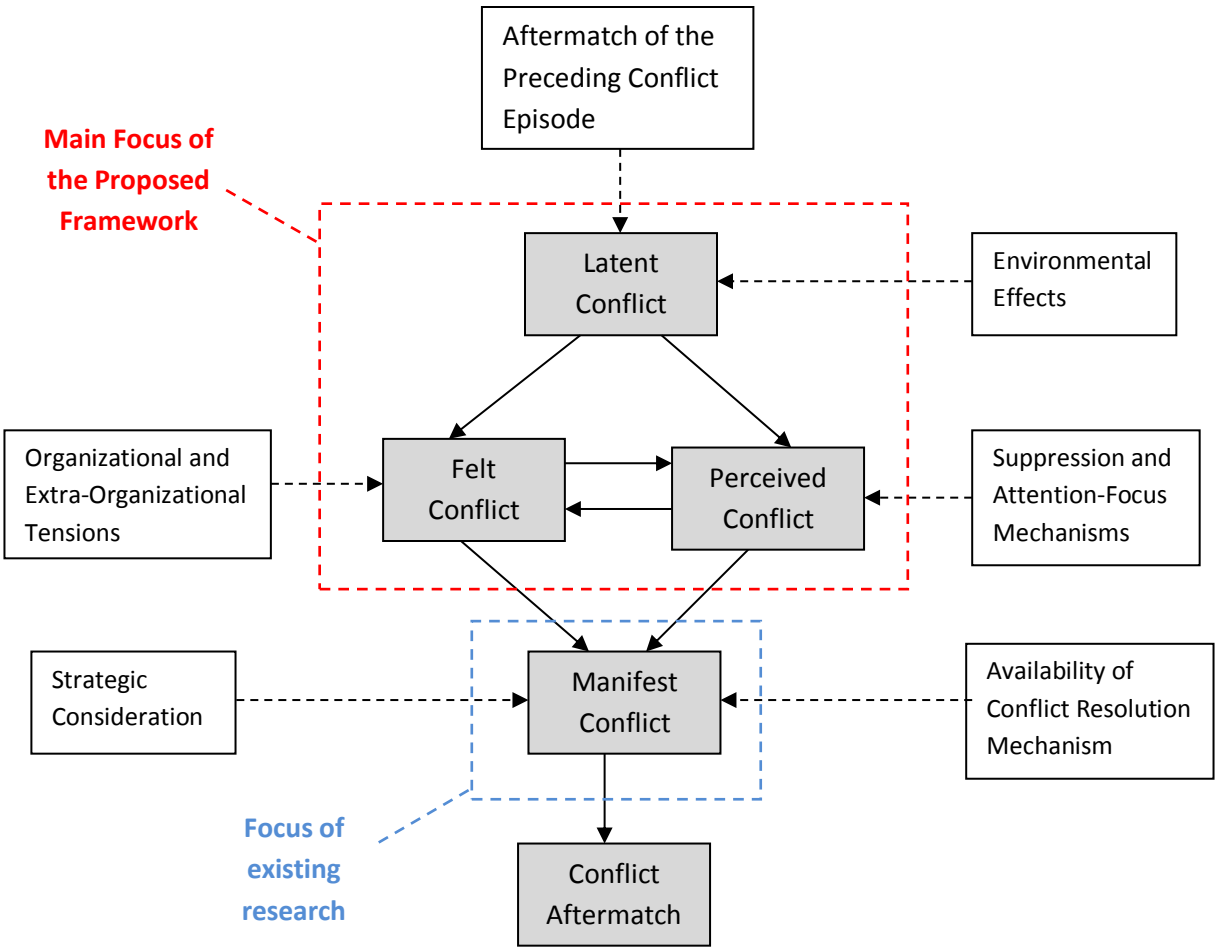


Figure 26 – Comparison between the focus of the existing research and the focus of this work according to the phases of conflict

As regards the dysfunctional conflicts, the solution focuses on reducing the sources of such conflicts as they can inhibit creativity or even drastically reduce collaboration. On the other hand, when dealing with functional conflicts, as they are a potential force for increasing creativity, the proposed framework will encourage them, however, giving the control of them to the analyst to keep the requirements elicitation working in a collaborative way.

The proposed framework will usually try to increase collaboration before and in the presence of the conflict. However, sometimes, increasing collaboration will just increase the intensity of the conflict. In such cases, the solution will use alternatives to perform the elicitation reducing direct interaction among stakeholders. Based on these assumptions the research will try to prove the following hypotheses:

Hypothesis 1

- When increasing collaboration, there will be an increase in the consensus and thus an increase in the efficacy of requirements elicitation.

Hypothesis 2

- In the presence of an intense conflict, by reducing the direct interaction between the stakeholders involved there will also be the elimination of the source of such conflict and consequently, there will be an increase in individual collaboration.

In order to prove these hypotheses and consequently, evaluate the framework, new rounds of data collection were performed and analysed after using the framework. The Evaluation Case Study, described in the following chapter, provided such an analysis.

To follow the idea of reducing dysfunctional conflicts, it is mandatory that the sources of conflict are identified in order to guide the choice of the right approach to perform requirements elicitation. As mentioned in the previous chapter, it is also important that the analyst identifies which modes of tackling conflict each stakeholder will normally use. Thus, as a first step of the framework, a stakeholder analysis must be performed to identify the possible sources of conflict as well as how each stakeholder handles existing conflicts that could disturb the activity of requirements elicitation.

A. Stakeholder Analysis

An extremely important aspect to be studied before choosing the best elicitation technique to be used in a specific situation is first to understand with whom the analysts will be dealing. In fact, who the stakeholders involved in the elicitation process are in terms of personality, expectations and work style. The first step thus, is to identify who are all the stakeholders involved.

The solution will not propose a specific way of identifying the stakeholders, as long as they are properly identified. As a suggestion to support the activity, we suggest using Sharp et al.'s (1999) approach that presents as a starting point what they call baseline stakeholders. From these, they select supplier and client stakeholders where the former provides information or supporting tasks to the baseline, and the latter provides processes or inspects the products of the baseline ones. In addition to these, they identify the so-called satellite stakeholders which interact with the baseline stakeholder in a variety of ways such as: communicating, reading a set of rules or guidelines, searching for information among others. Their approach is based on five steps as follows:

1. Identify all specific roles within the baseline stakeholder group;
2. Identify supplier stakeholders for each baseline role;
3. Identify client stakeholders for each baseline role;
4. Identify satellite stakeholders for each baseline role;
5. Repeat steps 1 to 4 for each of the stakeholder groups identified in steps 2 to 4.

Once the stakeholders are identified, it is important to decide which is the best way to obtain requirements from each stakeholder. Consequently, stakeholder analysis should be the starting point to the proposed solution to identify the conditions above. Many studies have

addressed stakeholder analysis. Each analyst must define which characteristics should be taken into account when choosing the appropriate elicitation technique. Thus, we present the following table as a guide for this analysis based on some characteristics of the literature and some of the study findings, including military features:

Characteristic	Description
Stakeholder Name	*Identification of the stakeholder*
Rank	*Military rank or equivalence*
Group	*Group to which the stakeholder belongs*
Organisation Role	*The role played generally by the stakeholder in the organisation*
Activities / Tasks	*Activities and tasks related to the system developed undertaken by the stakeholder*
Project Role	*The role of the stakeholder in the project*
Stakeholder's superior	*The superior to whom this stakeholder must report*
Project Impact (high/ medium/ low)	*What level of impact to the system that stakeholder could have*
Project Expectations	*What the project expects that stakeholder to do in order to support the development of the project*
Perceived Attitudes / Risks	*Stakeholder's attitudes that could influence positively or negatively the development*
Management Strategy (authoritarian / participative)	*What kind of management the stakeholder usually uses*
IT Knowledge (high / medium / low)	*What level of IT knowledge the stakeholder has*

Table 9 – Stakeholder Analysis

As shown in the table above, there are some significant rows that are strongly recommended by the solution as they reflect the issues gathered during the Motivational Case Study. Some rows are important in identifying the stakeholder and the stakeholder's relevance for the project: Stakeholder Name; Organisation Role; Activities and Tasks; Project Role; Project Expectations; Group. The other rows help in determining the risk of that stakeholder raising conflicts:

- Stakeholder's Superior – this row will help defining the participants of a meeting in order to avoid any kind of intimidation that could influence negatively the elicitation.

- Project Impact – this item will show how the attitudes or the expectations of that stakeholder could influence the success of the project.

- Perceived Attitudes or Risks – this row will describe the usual behaviour of that stakeholder and outline the possibility of participating in group techniques or to show the best approach to use with this stakeholder.

- Management Strategy – this issue shows if the stakeholder has an authoritarian or participative way of making decisions or dealing with daily tasks. Such a strategy can determine the possible approaches to be used with him/her. Mainly, if the stakeholder is in the highest hierarchical position within the client organisation, it will be determinant to define the best approach. This could mean changing the approach or at least excluding this stakeholder from group tasks.

- IT Knowledge – the level of IT knowledge could make some possible techniques difficult to implement such as ULRC or even participatory design. For certain techniques some level of IT knowledge is desirable.

- Rank – in a military environment this characteristic can determine, together with others, the approach that could be used and in which groups this stakeholder could participate.

Furthermore, when individuals are in groups, they act differently than they do when they are alone. Thus, the behaviour of people in groups is different than the sum total of all the individuals acting in their own way. Consequently, we suggest a second level of analysis based on group characteristics for each group or department involved in using the system to be developed:

Characteristic	Description
Group	*Group identification*
Activities / Tasks	*Activities related to the project developed by the group*
Related Groups (and what kind of relationship)	*Other related groups that are involved in the project*
Manager	*Group manager identification*
Management Strategy (participative /centralized)	*What the management strategy in the group is*
Size	*Quantity of members*
Meetings	*What kind of meeting usually takes place in the group and how often.*
Turnover Rate (high / medium / low)	*What the usual turnover rate in the group is*
Systems Types (operational / administrative)	*What kind of systems that group usually deals with*

Table 10 – Group Analysis

Again, some rows are important to identify the group such as: group name, activities / tasks, related groups. The other characteristics will support the decision of the approach based on the issues mentioned in the research:

- Manager – knowing who the manager is based on the stakeholder analysis, especially regarding his/her managerial strategy, will guide which possible

approaches could be used in that group. This could also determine that the manager should be addressed separately from the rest of the group.

- Managerial Strategy – this issue shows the usual approach used by that group, showing the way that group is used to working.

- Meetings – the way they are used to doing their meetings can give guidance to how they will perform elicitation (ie. normally people tend to be in conformance with their culture).

- Turnover Rate – the turnover rate has a high influence on the efficacy of the requirements elicitation, and so it should also be used to guide the selection of the elicitation technique. An elevated turnover rate should indicate the use of more indirect techniques to enable a separation between the information and the informant.

- System Type – this row will show if the system is operational or administrative giving an idea of the existing level of security and military formality. Operational systems tend to have strong security issues, and the information tends to be extremely specific and full of military terms, acronyms and jargon.

After understanding each stakeholder and each group of stakeholders and their characteristics it will be easier for the analyst to choose which elicitation technique is best suited to each situation and project. To support this choice the framework provides a set of techniques, its characteristics and when to use them, in order to obtain good performance in

the requirements elicitation task and to achieve project success in the eyes of all stakeholders. These techniques are divided into: increasing collaboration, increasing participation and reducing interaction (Figure 27).

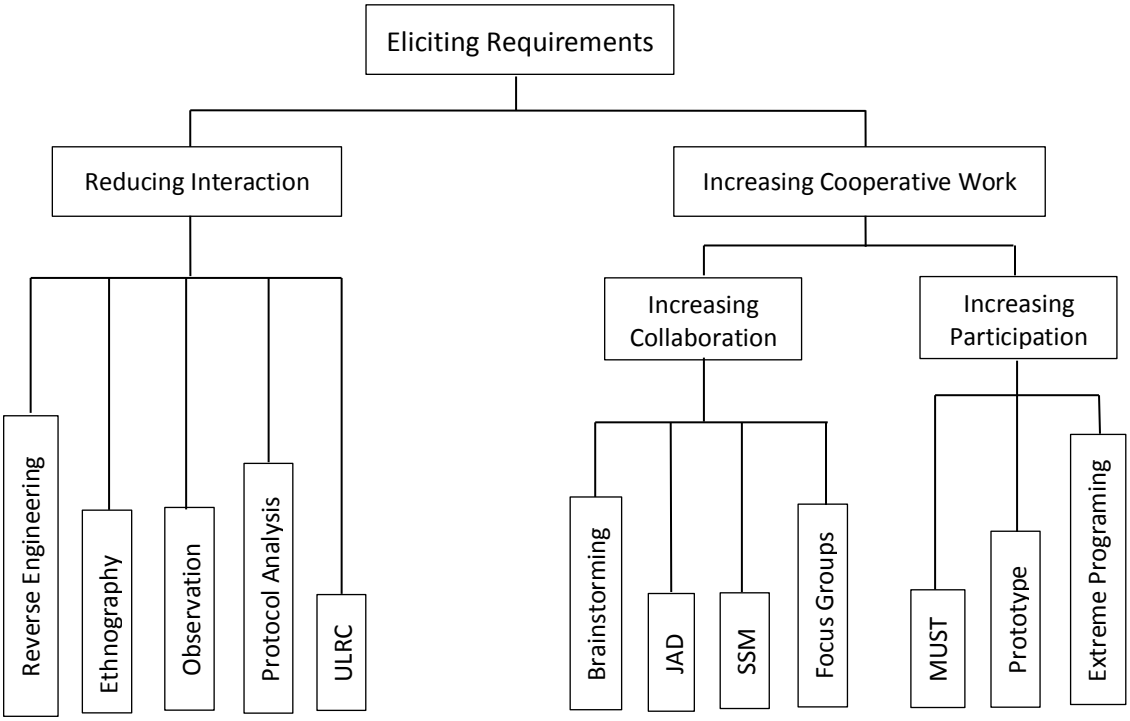


Figure 27 – The Framework - Classification of requirements elicitation techniques

B. Increasing Collaboration

As mentioned in the previous chapter, there is an understanding of requirements engineering as a group activity. Consequently, the first and most natural step in attempting to improve requirements elicitation is to increase collaboration between the stakeholders. Following this idea, there are several group elicitation techniques such as workshops, JAD

(Joint Application Development), brainstorming, focus groups and SSM (Soft System Methodology) that might be applied.

All group techniques independent of purpose or format rely on the skills of a person to guide these tasks and guarantee their efficacy. This person is known as a facilitator or mediator. The facilitator can be integrated into the group as a member or can be in a remote position to guide without interfering.

The mode of the facilitator will depend on the technique used as well as in the phase of the group's lifecycle. The process of requirements is composed of several stages (Figure 28). Some of them take place with clients and requirements team working together and some of them within the requirements team.

During the first stage, eliciting requirements, there are two key moments. In the initial moment there must be a decision of what kind of system will be delivered and a draft of the scope. Such decisions are usually made by the highest level stakeholders and sometimes with some political influence. In a military setting the hierarchy can also influence this stage. The facilitator will need some authority to maintain the focus of the meetings and suppress any attempts to use political influence. Thus, the facilitator should be of a high rank but preferably with no participation in the project. Usually, military institutions hire retired officers with relevant knowledge. These retired officers could be used as facilitators as they still hold their high rank, however, with no strong relationship with the project and no career aspirations that could restrain their actions. The second pivotal moment is when the requirements are effectively elicited. In this second moment there must be an analysis of the stakeholders involved. If there are some tensions between client team and requirements team

the facilitator could remain this high ranking officer or a civilian without relation to the project, however, usually these officers or civilians will have a high cost, and the ideal would be to switch to a more participative approach where the requirements team leader would work as a facilitator.

During the second stage, as it is performed within the requirements team, the facilitator will naturally be the requirements team leader.

In the following stages again there must be an analysis of the stakeholders involved. Preferably, the facilitator would be the requirements leader, although, if the tensions between the two teams still remain the retired officer or the civilian could be used again, always subject to an analysis of the costs involved.

At the end of the project, there is the acceptance of the system which will usually involve the higher ranks again. During this stage, a facilitator with some power should once more be used as it is a phase that will often have elevated tensions in any software project.

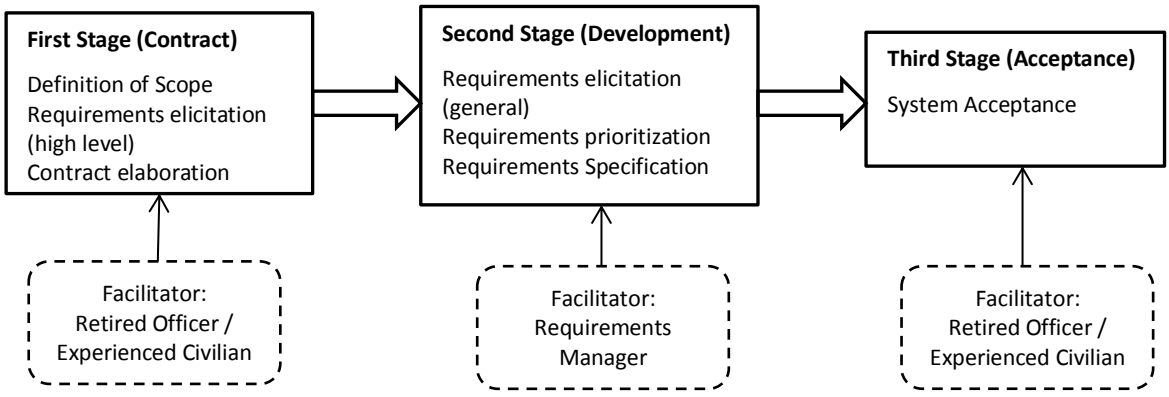


Figure 28 - Requirements stages and activities, and recommended facilitator

With the right facilitator at the right moment, any of the group techniques could be used. Each group technique will also indicate some other necessary skills and the knowledge of the process by the facilitator.

Once it is decided what group technique will be used, an important issue is which stakeholders you can put together in discussion. As mentioned in the previous chapter, horizontal or lateral communication is more frequent than vertical communication in organisational settings since people feel more open and free communicating with peers than with superiors or subordinates. Furthermore, because peers are most likely to share a common frame of reference, message distortion is less likely. This fact will be exacerbated inside a strong hierarchical structure. In the presence of an authoritarian management approach, it will be difficult for a subordinate to express opinions and ideas in front of his/her superiors, especially in an environment such as the military that has hierarchy as one of its pillars. If the management approach is based on participation, and the leader of the client's team is capable of accepting someone else's opinions, including those hierarchically inferior to him/her, then it is possible to group all stakeholders together in the same meeting, always subject to the restrictions of physical space. If there could be any kind of embarrassment among the stakeholders, there must be some separation according to hierarchical levels. For example, there could be a group of officers and a second group of non-commissioned officers. From what was observed, even among commissioned officers there could be some level of embarrassment. In these cases, there could be even more groups.

Furthermore, if there is more than one group of clients during the requirements elicitation, it is likely that divergent ideas between the groups arise. As previously stated,

when there are tensions, it is crucial to separate the people from the problem. Consequently, it is important to choose carefully the way to introduce the ideas from one group to the other. Preferably, the ideas from one group should be presented to the other group by the requirements team as if they were general ideas originated by the requirements team itself. Therefore, there will not be a previous rejection of the ideas of another group, because they are considered hierarchically inferior. In other words, the ideas will be judged on merit and not according to who received and proposed them.

i) Brainstorming

Brainstorming is a group technique created to increase group creativity in an environment free of criticism. In the case of requirements, due to its characteristics it must be used in a group with a participative management approach, to keep the environment free of criticism, and usually in the early stages of the requirements elicitation process in order to raise general ideas for the system.

The framework suggests this technique as a possible tool to increase creativity and collaboration among stakeholders when the environment allows collaboration among different hierarchical levels. Therefore, it is important that the requirements team perceives any possible threat to participation or the use of hierarchy on the process. In the presence of a mild presence of authoritarian behaviour the division of the group on hierarchical levels is a possible solution. However, if authoritarian behaviour is impossible to keep under control this technique should not be used.

There are other creative techniques, such as Assumption Reversals or Analogies. They can also be used in such situations, although they need some external stimulus and thus some previous preparation. They could generate a better result, if the brainstorming is generating basic and uniform ideas, due to their manipulation of the initial information.

ii) JAD / Workshop

JAD is a technique usually used to develop business requirements, when executives and IT personnel are grouped together in a workshop well-structured to gather requirements with a consensus.

The framework proposes such a technique to generate only the high level goals of the project as long as it is expensive to gather all the stakeholders in an environment outside the work environment. To establish such a meeting the requirements team will need support from higher authorities to convene higher ranks in such a time-consuming task. As the meeting will be discussing high level goals for the system with high authorities involved, there should not be a high hierarchical difference between the participants in the meeting. In the military setting the use of uniform could be abolished during the meetings in order to keep a more collaborative environment. Furthermore, the facilitator must have the authority to control such high hierarchical level participants, which would indicate a retired high ranking officer or a civilian with a high level of known expertise.

iii) Focus Groups

'Focus Groups' is a group technique that groups stakeholders to gather information about a specific subject. Due to the characteristic of fast meetings focused on one specific subject, in the case of requirements elicitation, this technique must be used to clarify a particular requirement or discuss a specific issue raised during the requirements process. Its characteristics indicate its usage in the framework to solve some specific functional conflicts that can affect the development. The facilitator in this case could be the manager of the requirements group, as the discussion will be extremely specific. It is important, though, that the facilitator keep the focus on the issue at hand.

iv) SSM (Soft System Methodology)

SSM is generally applied to situations where there are conflicts among stakeholders. Consequently, in the case of requirements engineering it must be used mainly when there is more than one department or sector involved in a system development, and it is important to establish the boundaries of participation from each one. It also can be used when there are conflicts of opinions about a process or workflow.

In the military domain, there are numerous systems that permeate more than one department. These systems typically struggle to fulfil the expectations of all the departments involved. Furthermore, usually there is redundancy of information sometimes with conflicting definitions or structures. In this environment, hierarchical pressure is often applied to the requirements team as each department prioritises its own expectations and

consequently, uses hierarchical and political influence. SSM would allow the requirements team to confront the different expectations in a collaborative meeting. To ensure the collaborative environment, however, a high ranking facilitator should be used, such as a retired officer or a civilian with known expertise.

C. Increasing Participation

When there is space for an even closer collaboration of the client in the development, other techniques can be used. These alternatives will work with the participation of the client in the requirements team.

i) MUST

An even bigger level of collaboration would be to encourage the participation of the client in the design. MUST has its principals based on Participatory Design. MUST is a technique that uses ethnographic techniques and intervention together and relies on a committed participation of managers and workers.

In the case of the environment under study, this technique could be used to create the participation of the user during the design of the new system, especially with clients with high levels of criticism, although with a participative management approach. A high ranking client would have the opportunity to discuss ideas as the tasks are being performed, and the lower ranks would have the opportunity to prove empirically their opinions based on their practical expertise. For this technique to be successfully used it is important that the client

team has the ability to suffer interventions and also to accept criticisms on the way they perform their tasks. It will be useful as well to try new possible solutions.

ii) Prototyping

It is easier for the client to understand the development idea or even to have new ideas when there is a starting point to stimulate them. The prototype is an incomplete version of the system, developed with the only purpose to enhance the understanding of the system by the user and to gather novel ideas, being discarded after that.

In the framework proposed it can be used to support any of the other techniques in order to evaluate the system, but is especially useful to explore vague or ambiguous requirements. However, sometimes there exists hierarchical pressure during the presentation of a prototype. If there is a tension between client team and requirements team, the presence of a high hierarchical level facilitator is indicated.

iii) Agile Methods

Agile methods are a group of system development methodologies with continuous iterations and incremental development where the development groups are composed of clients and developers. Effective user participation is common in Agile approaches.

However, even advocates of the Agile methods agree that this methodology does not meet every case. The main idea is to reduce time and cost by reducing the need for information exchange and control. All developers will be responsible for some requirements in all its stages. Thus, to make better use of the Agile Methods, it is necessary not only that

client characteristics such as participatory approach and availability of time are present, but also the team needs a set of skills such as experience, organisation, speed and cohesion.

In our solution, we indicate one Agile method, the 'Extreme Programming' (XP), when the development team has the skills and expertise required as well as the client having the commitment and availability of time to the project. Furthermore, it is important that there are no tensions between client and development teams.

D. Reducing Direct Interaction

Sometimes conflicts within the group are so intense that even a group facilitator can have difficulties in guiding the group to fulfil its goals. When conflicts reach a high level of intensity with a very improbable chance of being solved through discussion, the best way to solve it is by separating the litigious parts. In these cases increasing communication will just exacerbate the conflict. Following this idea we propose, when conflicts are too intense, that analysts should use techniques that would reduce the interrelation between the stakeholders or even between client team and requirements team. Some techniques are based on observation of tasks and the social interaction during those tasks. Consequently, there will be no intense direct interaction between client and requirement teams. These techniques also will not confront directly antagonistic opinions. With the use of these techniques, the development will be based on the requirements team perception rather than the client team perception, which makes the solution independent of the client that is present in the moment, avoiding criticism by other current or future client organisation members. Techniques such

as ethnography, observation, protocol analysis or ULRC will support the reduction of the interaction in different ways and levels.

i) Reverse Engineering

Reverse engineering is a process by which one bases the development of a new software product in the analysis of the existing system. This process includes analysing existing documents, data, code and interfaces as a starting point to develop a new system.

Such a method can be used only if the documentation is at least useful and reasonably updated, and mainly if there are some characteristics of the existing system with the approval of the client team. It is an especially useful method if the goal of the development is merely to update the platform of the existing system. This method will provide a considerable amount of information for the new system with a reduced need for interaction between the client team and the requirements team. However, it could also lead to the development of what may be merely a copy of the existing system; consequently not fulfilling expectations. This would indicate that this technique shall be used in conjunction with other techniques to elicit the new functionalities.

ii) Ethnography

Ethnography, an approach that comes originally from anthropology, allows us to understand the culture of a given society or organisation. As long as ethnography can provide a deep understanding of the environment under study it could be used in any situation to gain an understanding of the client organisational culture. However, due to its costs and the time

needed it will be recommended at least for systems that will attend organisations with a complex and very specific nature. In such organisations, it is important to know the terms, jargon and acronyms used. Although, in the military setting, the whole environment is full of specificities, a deep understanding is mainly important in operational systems as they deal with tasks just performed on this domain. In the case of possible conflicts, it will provide a deep understanding without the necessity of interviewing the customers individually or even in a group, consequently reducing the amount of interactions needed and therefore, the sources of conflict.

An ethnographic approach will also support developments where there is a high turnover, since the requirements are being gathered based on the observation of the analysts and not on the opinions of the client stakeholders. This will insure against the idea of a new boss changing the procedures from previous incumbent's approach, in order to put his "stamp" on the system being developed as long as the system is not being developed based on the old boss ideas, but on the requirements teams ethnographic data. A problem that could occur is the distortion of the viewpoint of the requirements engineer leading to a wrong modelling of the system. Thus, during the validation of the specification, when stakeholders have to agree with it, there could be tensions between requirement and client teams.

Ethnography is often time consuming. An alternative was proposed by Millen (2000) is: Rapid Ethnography. In this approach, analysts perform rapid observations to obtain a general view of the environment in a short period of time. Instead of a wide view of the organisation, the research question to the ethnographer must be sufficiently well defined so that the research team has a clear idea of what to observe (Millen, 2000).

iii) Participant Observation

Participant observation is a qualitative method based on ethnographic studies. The analysts observe workers in order to understand the possible situations they face in their jobs as well as their reactions to these situations.

In the case under study, this technique can be used to understand some activities as well as the interactions needed to perform these activities. Like ethnography, it will provide the perspective of the analyst without interviewing or relating the observation to a client's perspective. The interactions between the teams will be reduced to clarifying some aspects.

iv) Protocol Analysis

Protocol analysis is a method based on cognitive psychology to analyse behavioural aspects. This method relies on the out loud description of a task while the task is performed.

This method should be used to clarify specific tasks such as the usage of an electronic sheet, thus avoiding several meetings or interviews. Although this technique is criticized for not being natural, as the observed person could bias the task due to being under study, it will be useful in an individual task and better than various interviews with the client. The usage of filming is encouraged to further reduce the need for other meetings. Consequently, possible conflicts between clients and analysts will be minimized. Another gain is related to multicultural teams. New analysts or analysts without any knowledge in the domain can have

an understanding by watching the recording instead of soliciting explanations from the client or other analysts.

v) ULRC (User Led Requirements Construction)

URLC is a social process created to reduce the gap between user and developer. In this technique, the client team is responsible for writing the requirements. To make it possible the requirements team performs a hands-on training for the client team.

In our solution, this technique can be used to reduce tensions and criticisms between the client team and the requirements team, especially if these criticisms are reinforced by hierarchical issues. As long as the specification is developed by the client team, there is no space for criticism or hierarchical pressure and consequently, conflicts. However, to make this approach effective there is the need of some IT knowledge and high domain knowledge by the client team.

6.2 The framework provided for the teams

This framework was developed to support the selection of requirements elicitation techniques according to some influencing factors.

The first step is to define who are the stakeholders involved in the project following Sharp et al's (1999) process. A stakeholder in an organisation is any group or individual who

can affect or is affected by the achievement of the organisation's objectives. Follow the sequence below:

1. Identify all specific roles within the baseline stakeholder group (users, developers, legislators and decision-makers);

2. Identify supplier stakeholders for each baseline role (provides information or supporting tasks to the baseline stakeholders);

3. Identify client stakeholders for each baseline role (provides processes or inspects the products of the baseline stakeholders);

4. Identify satellite stakeholders for each baseline role (interact with the baseline stakeholder in a variety of ways such as: communicating, reading a set of rules or guidelines, searching for information among others);

5. Repeat steps 1 to 4 for each of the stakeholder groups identified in steps 2 to 4.

Stakeholder 1 (Name / Role)	Supplier1 (Name / Role), …Supplier n (Name, Role)
	Client1 (Name / Role), … Client n (Name / Role)
	Satellite 1 (Name / Role), … Satellite n (Name / Role)
… Stakeholder n (Name / Role)	Supplier1 (Name / Role), …Supplier n (Name, Role)
	Client1 (Name / Role), … Client n (Name / Role)
	Satellite 1 (Name / Role), … Satellite n (Name / Role)

Table 11 - List of Stakeholders

Once the stakeholders are defined the next step is to perform a stakeholder analysis. Fill the table below with the information according to each project stakeholder's characteristics.

Characteristic	Description
Stakeholder Name	*Identification of the stakeholder*
Rank	*Military rank or equivalence*
Group	*Group to which the stakeholder belongs*
Organisation Role	*The role played generally by the stakeholder in the organisation*
Activities / Tasks	*Activities and tasks related to the system developed undertaken by the stakeholder*
Project Role	*The role of the stakeholder in the project*
Stakeholder's superior	*The superior to whom this stakeholder must report*
Project Impact (high/ medium/ low)	*What level of impact to the system that the stakeholder could have*
Project Expectations	*What the project expects that stakeholder to do in order to support the development of the project*
Perceived Attitudes / Risks	*Stakeholder's attitudes that could influence positively or negatively the development*
Management Strategy (authoritarian / participative)	*What kind of management that stakeholder usually uses*
IT Knowledge (high / medium / low)	*What level of IT knowledge the stakeholder has*

Table 12 – Stakeholder characteristics

For each group present in the table above, there must be a definition according to the table below:

Characteristic	Description
Group	*Group identification*
Activities / Tasks	*Activities related to the project developed by the group*
Related Groups (and what kind of relationship)	*Other groups related that are involved in the project*
Manager	*Group manager identification*
Management Strategy (participative /centralized)	*What the management strategy in the group is*
Size	*Quantity of members*
Meetings	*What kind of meeting usually takes place in the group and how often.*
Turnover Rate (high / medium / low)	*What the usual turnover rate in the group is*
Systems Types (operational / administrative)	*What kind of systems that group usually deals with*

Table 13 – Group characteristics

Once the Stakeholders and Groups are known and characterised the next step is to determine which technique to use in each situation:

1) Guidelines for the selection of techniques

1.1 If there is a high probability of turnover, preferably use the following techniques:

- Reverse Engineering – as long as there is an existing system documented or at least the database structure to study;
- Participant Observation - to observe some tasks that involve more than one stakeholder;
- Ethnography - to observe the whole culture of the client organisation;

- Protocol Analysis – to understand specific tasks such as the use of an electronic sheet, for example.

If it is necessary, some group techniques could be used such as:

Brainstorming, JAD or SSM.

In these cases, the following techniques should be avoided:

Interviews, questionnaires, prototyping and agile.

1.2 When the client's manager's management approach is authoritarian, the following techniques should be used:

- Reverse Engineering – as long as there is an existing system documented or at least the database structure to study;
- Participant Observation - to observe some tasks that involve more than one stakeholder;
- Ethnography - to observe the whole culture of the client organisation;
- Protocol Analysis – to understand specific tasks such as the use of an electronic sheet, for example.

In these cases, the following techniques should be avoided:

Interviews, questionnaires, group techniques, MUST, prototyping and agile.

1.3 When there are members on the requirements team with no familiarization with the domain the following techniques should be used:

- Reverse Engineering – as long as there is an existing system documented or at least the database structure to study.

- Participant Observation - to observe some tasks that involve more than one stakeholder;

- Ethnography - to observe the whole culture of the client organisation;

- Protocol Analysis – to understand specific tasks such as the use of an electronic sheet, for example. (preferably filming it)

In these cases avoid:

Avoid involving members with lack of domain knowledge in the meetings of group techniques.

Do not allow these members to perform interviews.

1.4 If the management approach of client team is participative use the following techniques:

- Brainstorming – to gather ideas with a significant group of stakeholders
- JAD – preferably in the beginning of the elicitation to discuss high level requirements
- Focus Groups – to discuss some specific issues that are unclear.
- Prototypes and Storyboarding to validate and discuss ideas.
- MUST – to discuss some ideas while the team is performing the tasks.

As complementary techniques the following techniques could be used: interviews, ethnography, participant observation and protocol analysis.

1.5 If the client has high level of expertise in the domain and some IT knowledge, the ULRC technique can be used. This technique is highly indicated when there are tensions between the requirements team and the client representative.

To make the ULRC technique effective there must be commitment as well as availability of time from the client.

In the ULRC technique, requirements team members with lack of knowledge in the domain should not participate.

1.6 If there is more than one organisation, department or sector involved there must be an understanding of the flow of information and hierarchy between these areas.

1.6.1 If there are no tensions between the areas the following method should be used:

- SSM.

1.6.2 If there are tensions between the areas these relations should be defined by the following techniques:

- Interviews – Interview each area responsible separately and confront the answers.
- Reverse Engineering – observe documents that define the responsibilities of each area.
- Ethnography – observe the environment and focus on the need of information from other areas.
- Participant Observation – observe tasks and focus on the need of information from other areas.

1.7 In the specific case of a very experienced team and a client with time availability, high commitment and a participative managerial approach the following technique could be used

- Extreme programming

1.8 When there is a high hierarchical difference between users:

- Avoid gathering users with a high hierarchical distance from one another in the same group when using group techniques
- When presenting ideas from a user to another in a high hierarchical distance, avoid mentioning the source of information.

2) Summary of the Guidelines

Situation	What should be used	What should be avoided
High turnover	• Reverse Engineering • Participant Observation • Ethnography • Protocol Analysis *Could also be used:* Brainstorming, JAD or SSM	• Interviews • Questionnaires • Prototyping • Agile • MUST
Authoritarian management approach	• Reverse Engineering • Participant Observation • Ethnography • Protocol Analysis	• Interviews • Group techniques • MUST • Prototyping • Agile
Requirements team members with no familiarization with the domain	• Reverse Engineering • Participant Observation • Ethnography • Protocol Analysis	• Avoid involving members with lack of domain knowledge in the meetings of group techniques. • Do not allow these members to perform interviews.
Client team with participative managerial approach	• Brainstorming • JAD • Focus Groups • Prototypes and Storyboarding • MUST	
Client expert in the domain / Client with IT knowledge / Tensions between client and requirements teams	• ULRC Could also be used: • Questionnaires	• Group techniques • Prototypes and Storyboarding • MUST • Agile
Various areas involved with tensions between the areas	• Interviews – Interview each area responsible separately and confront the answers. • Reverse Engineering – observe documents that define the responsibilities of each area. • Ethnography – focus on the need of information from	• SSM.

	other areas. • Participant Observation – focus on the need of information from other areas.	
Various areas involved with **NO** tensions	• SSM.	
Extremely experienced development team Client team with time availability, high commitment and participative managerial approach	• Agile	
High hierarchical difference between users		• Avoid gathering users with a high hierarchical distance from one another in the same group when using group techniques • When presenting ideas from a user to another in a high hierarchical distance, avoid mentioning the source of information.

Table 14 – Summary of guidelines

To complement these guidelines a step by step guide explaining each technique was also provided for the teams so they could know how to perform the activities. This guide is presented in Appendix C.

6.3 Summary

This chapter presented the framework that is the main goal of this research. Such a framework was based on the Conflict Theory with the understanding of the requirements elicitation as a group task. Any group task due to the numerous interactions needed, provide

a large amount of conflict sources. Therefore, while this task is performed it is likely that several conflicts arise. Consequently, the conflict theory seemed to be the perfect theory to ground the solution.

The framework was based specifically on the possible ways to resolve a conflict. According to the theory, the first attempt to solve the conflict should be by increasing the communication and thus the collaboration. However, in the presence of an intense conflict increasing communication usually increases the intensity of conflict. Thus, in these cases, the solution is to reduce the relationships.

The framework analysed several techniques and grouped them according to these possible conflict resolutions in order to reduce the sources of conflict. There are three groups of techniques in the framework. The first group aims to increase collaboration by using group techniques. The second group is based on the participation of the client in the development team. This is possible only if the client has the proper commitment and time availability. Finally, the third group aims to reduce the interaction between stakeholders when there is a source for an intense conflict.

After presenting the framework with the explanations for each technique the chapter presented a guideline for the teams on how to perform the selection. Such a guideline does not explain the theory behind the selection as the teams need a more straightforward support to the selection.

Chapter 7

Evaluation of the Framework

Once the framework was developed, it was necessary to assess its efficacy in a military environment. The goal of the framework was to improve the efficacy of the requirements elicitation in a military environment and as a consequence the possibility of success in terms of fulfilling client's needs and expectations. Consequently, the framework was offered to some teams in the Brazilian Navy, and those teams were supposed to provide a qualitative evaluation based on interviews.

At the same time, metrics were being collected by the teams to provide a quantitative evaluation of the framework. The metrics were defined based on the Goal Question Metrics paradigm, an approach to support the definition of metrics based on business goals that will be discussed further. While the qualitative evaluation provided an indication of the efficacy of the framework, the quantitative evaluation provided an indication of its effectiveness. This chapter presents these evaluations and the results with the analysis.

7.1 Qualitative Evaluation

After the development of the framework, three rounds of fieldwork were undertaken. Seven teams, three administrative and four operational, were using the framework with the suggested techniques. Each team could decide which techniques should be used based upon the characteristics of the projects and the stakeholders guided by the framework. The teams received a copy of the framework composed of a table for stakeholder and group analysis and a list of the techniques with the procedures to implement it, pros and cons, and the situations in which the technique should be used in a military setting based on the issues gathered during the research. I trained the teams to perform properly the techniques, previously to the actual requirements elicitation tasks, without interfering during the tasks. Each system is discussed below with its characteristics and the techniques used before and after the framework as well as teams' comments about them.

SYSTEM 1

This system is an administrative system being developed to support communication among higher authorities. It was started already using the framework. The manager of the development team is an officer of a medium rank, and among the clients are the highest authorities in the Navy.

After the Framework

The high hierarchical difference could lead to tensions and difficulties among the client team and requirements team. Furthermore, the client organisation is based in a city two hours

distant by air from the development team. Another important feature is that the client team has among them some people with high IT knowledge. Consequently, the manager opted to use the technique URLC. Such a technique would avoid tensions between the teams and also reduce the necessity of travelling several times during the elicitation, which would increase the system cost considerably.

After the elicitation, the manager decided to use fieldwork observation for a week to be sure that the specification was presenting a real and updated picture of the system without questioning directly the client team. The techniques were used, and they worked satisfactorily. Due to the client's high IT knowledge, the specification showed a high level of standardization and objectivity. There were no tensions between requirement and client teams and the client, up to this moment, is satisfied with the approach selected.

SYSTEM 2

This is an operational system that deals with classified information. The client is just one Military Organisation. The level of security of the system is extremely high. The requirements team is formed by two military and three civilians. There was practically no consensus among the stakeholders and the client representative has a high IT knowledge. Furthermore, there was high level of uncertainty in the requirements as it was a totally new system and there was a difficult access to the information. All these characteristics created an arduous environment in which to perform elicitation.

Before the Framework

The first technique used was unstructured interviews. However, the requirements team was struggling to elicit requirements with the high ranking client members as they did not have the availability of time to discuss the requirements and when there was a meeting, they did not like to discuss some basic issues with the civilians with no previous military knowledge in such an important operational system. These problems together with the big hierarchical difference and the authoritarian management approach presented by the client led to high tensions between the teams.

After the Framework

As a second attempt the requirements team used URLC. The technique succeeded partially as just one client member was involved and also as the requirements team openly criticized the specification. So even with an initial specification that could be used as a starting point the tensions became even higher. At the same time, the requirements team decided to use a prototype to discuss the requirements with the other team members. The prototype was also ineffective as there were too many characteristics to discuss including low level requirements. The client team members with a high rank didn't want to discuss such a low level of requirements. Finally, they decided to use a storyboard with just the main features of the system. Such a technique reduced the time needed to discuss the requirements and also increased the client's perception of progress. With the client member who built the specification, the team started to use participatory design. This technique proved to be effective, significantly reducing the tensions between the teams.

SYSTEM 3

This is an operational system that deals with classified information about military operations. The system encompasses all Armed Forces in Brazil. This system has been in development for more than seven years, and it encompasses several organisations. The development team, that is also responsible for requirements elicitation, is composed of three military and three civilians. As the civilians have been working in the system for many years, they already have a good knowledge about the terms and regulations of the environment. There is a turnover of the sponsor among the three Brazilian Armed Forces which has led to changes in the main ideas about the system. There is not much availability of the client, and client team and development team are located in different cities. There are two distinct levels of clients: the operators and the decision makers. Though there is a high hierarchical difference between the two groups, there is also a high respect for each role responsibility, which practically eliminates any possible tensions within the client team.

Before the Framework

In the beginning of the development there were some tensions between client and development teams due to the hierarchical difference and the physical distance. The first technique used was unstructured interviews. This technique was ineffective because of the uncertainty of the requirements. After struggling for two years the development team noticed that most of the requirements were discovered during military practice operations. Consequently, they decided to be present during these operations while the system was being used, using participant observation. This technique brought a much better understanding of the environment and increased the confidence from the client. After changing the

requirements elicitation approach to observation, those tensions initially observed between teams were practically eliminated. However, there was still some inefficacy in the elicitation as after the observations, new development ideas were tried, and then the client came to the development team city to evaluate them. It led to a long wait for the client to see some suggestions of functionalities that could not be approved.

After the Framework

To reduce this time of trying possible ideas the development team changed to a participatory approach (MUST) using ethnography with intervention. The requirements team was present during the operations also suggesting modifications to the system while the client was using it. This approach together with technical meetings in the development environment was extremely effective. Besides the stability of the requirements, another consequence was the elimination of any possible tensions, as the requirements team could discuss new ideas on-site with the participation of the client. Although the number of meetings was not reduced they became more effective.

"Nowadays we meet to discuss high level requirements; before we just discussed bugs…increased synergy between the teams" *Development team manager.*

SYSTEM 4

System 4 is an administrative system that deals with financial information. The development team is composed of two military and two civilians and includes the requirements team formed by one military and one civilian. Though there is high interest in the system, there is a low level of consensus among the stakeholders. Additionally, there is a large hierarchical difference between stakeholders, which led to tensions within the client team. The system will support several departments of one Military Organisation. There is also a high turnover in the client team. This system is upgrading an existing one and several Excel sheets. There are various routine tasks that will be included in the system.

Before the Framework

Initially, the requirements team used unstructured interviews and group meetings. The result was far from satisfactory. There was the need for several meetings to discuss each requirement, and in each meeting time was spent to explaining routine tasks and also terms and jargon to the civilian requirements analyst. To the clients, it was a waste of time, culminating in a lack of trust in the requirements team.

After the Framework

After studying the framework the requirements team decided to use mainly protocol analysis with filming during the routine tasks. Thus, each stakeholder performed his/her tasks with a camera pointed to the screen, so the electronic sheets could be clearly seen by the requirements team as the stakeholder performed each action. To gain further knowledge about the tasks a second camera was prepared to film the whole room where the task was

taking place (Figure 29). Consequently, it was also possible to observe the interaction between the stakeholders during such tasks. At first, the client was intimidated by the camera due to the fear of who would watch the movie. After the intent has been well explained the fear vanished. The combination of protocol analysis and observation was effective and reduced the tensions between teams. The civilian requirement team member could also understand better the tasks and features of the domain by watching the film and clarifying any issues about the requirements or the environment with other team members, without asking the client. Thus, the need for meetings was reduced considerably while the trust in the requirements team grew again. While prior to the framework, meetings were used to understand the tasks but also to understand the jargon and even to discuss basic knowledge, after the framework the meetings were used to discuss details and new ideas for the system. As an example, while with interviews there was the need for ten meetings to discuss one sheet, with the protocol analysis to understand a similar sheet there was the need for just four meetings. Furthermore, currently, to validate the requirements, a storyboard is presented to the client showing, in a simple and graphic way, what is being done and how it is being done.

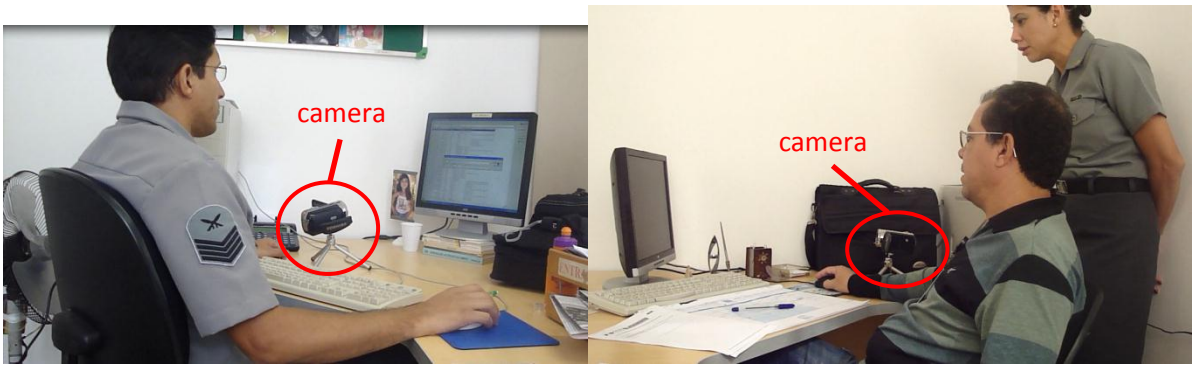


Figure 29 – protocol analysis and observation performed in System 4

"Now [after the Framework] they perceive what already exists and what is missing … there is a sense of what is good and bad and what can be thrown away and what can just be adapted" *Civilian team member*

"The department repeats a routine process, and people have difficulty in reporting what they do. They even did not know what was difficult, and now (after the Framework) they understand." *Requirements Team Manager*

SYSTEM 5

This system is an administrative system with all information classified to support the work of one department. The development team is composed of one military and three civilians with one military and one civilian in the requirements team. Among the clients, there are high ranking officers who access the system to get information and sergeants who operate the system. Although there is a high hierarchical difference, the management approach is participative.

Before the Framework

At first, the requirements team was performing elicitation using unstructured interviews. The effectiveness was low and there was the need for several meetings due to the low level of consensus. According to the requirements manager, "It was very confusing and there were a lot of disagreements". Increasing difficulty, there were several requirements that involved more than one sector.

After the Framework

After using the framework, the requirements team decided to use a composition of techniques. For the more generic issues that involved more than one sector, they decided to use brainstorming to take advantage of the participative management approach. As there is a small number of stakeholders without hierarchical pressures among them, all stakeholders were gathered in a single group. According to the requirements team manager, the improvement was easily noticed: "They became more involved and left the room smiling"… "The stakeholders reached consensus and there was the sense of collaboration". Another gain was that a single meeting substituted interviews with eight different stakeholders and solved the problem. For the tasks that involved one sector or one person they decided to use the participatory approach with observation and intervention together to validate the requirements. According to the requirements manager, this approach did not reduce the quantity of meetings, although it increased significantly the productivity of each meeting, with much more meaningful issues being discussed.

SYSTEM 6

This system is an administrative system with all the information classified. The system focuses on just one department, but this department has several sectors all over the country. There will be training all over the country but requirements are usually defined by the users of the main office. There are high interest and elevated consensus among the stakeholders with a low turnover rate. There is a significant hierarchical difference between the clients, but the management approach is participative. The hierarchical difference between the teams is insignificant. There is a system to be updated. However, some parts of the existing version

already work well and should be included in the new version. The development team is composed of two military and two civilians, and the requirements team is composed of two military members. Despite being formed by two military, one of them has practically no experience in the domain. This system was started after the framework.

After the Framework

As there was an existing system with several good features, the requirements team used reverse engineering with a good result. To complete the elicitation, they used workshops (Figure 30) also with a good result as the management approach is participative. However, as the requirements team decided to gather all stakeholders, independently of rank, in a single group, the lower ranks did not give opinions during the workshops. Thus, it was necessary to perform some separate interviews with the lower ranks to complete the information gathering exercise.



Figure 30 - Workshop with multiple ranks

SYSTEM 7

This system is an operational system that involves several operators. This is an extremely critical real time system. It encompasses difficult activities that need extreme attention and concentration and involve the control of a specific area. The operators work in shifts, and due to the tensions involved they are not allowed to do overtime. Usually no one other than the operators is allowed in the working area for security reasons. The system runs 24 hours per day and seven days per week.

Before the Framework

The first attempt was based on unstructured interviews. The access to the operators was difficult as they performed critical tasks in turns. Thus, the interviews had to be performed after the work period. This approach was not effective as there never seemed to be enough meetings.

After the framework

After the framework, there was a second attempt based on participant observation. While before the framework the interviews had to be performed after the work period, the observations could take place during their jobs, however, with serious caution. The operators preferred the second approach (observation) as it did not demand extra time after work. Furthermore, as it is an extremely detailed and peculiar job, the observation led to a more complete understanding, and consequently, to fewer meetings.

7.2 Quantitative Evaluation

Together with the qualitative evaluation a quantitative evaluation was conducted. Teams collected metrics before and after using the framework. Thus, the comparison between these metrics before and after the use of the framework would indicate the evolution of the requirements elicitation tasks due to the support provided by the proposed solution. These metrics cannot guarantee that any improvement demonstrated was due to the use of the framework, as there is no way to control all the factors involved, such as, for example, the increasing experience of the requirements team. However, if the improvement shows significant numbers, there is a strong indication of the influence of the framework on the cases. Teams were guided on how to collect the metrics, and the data collected during two years was sent to the researcher periodically. The quantitative evaluation could first corroborate the qualitative evaluation of the efficacy of the framework, besides that it could evaluate the efficiency of the framework in terms of how much improvement was possible with the use of the framework.

To define the metrics to support the evaluation of the framework effectiveness we used the "Goal Question Metrics" (GQM) paradigm. According to Solingen and Berghout (1999), "the principle behind the method is that measurement should be goal-oriented." GQM defines a certain goal, refines this goal into questions and defines metrics that should provide the information to answer these questions. Once the questions are answered, the results support the definition of operational goals, allowing us to assess whether the objectives were achieved or not. Thus, "GQM defines metrics from a top-down perspective and analysis, and interprets the measurement data bottom-up" (Solingen, Berghout, 1999).

In the case of this work, the goal of the metrics will be to "Evaluate the efficacy and efficiency of the requirements elicitation framework proposed" as it can be seen in Figure 31.

In order to make such an evaluation possible the following questions should be answered:

Q1. Was there an improvement in the requirements elicitation process with the utilisation of the new framework?

Q2. Are the clients' needs better elicited with the new framework?

Finally, to answer these questions the following metrics were collected:

M1. Number of requirements meetings needed before and after the framework.

M2. Quantity of requirements meetings needed in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework.

M3. Time needed to elicit in a certain project two requirements of similar complexities, one elicited before and one elicited after the framework.

M4. Number of requirements changes before and after the framework.

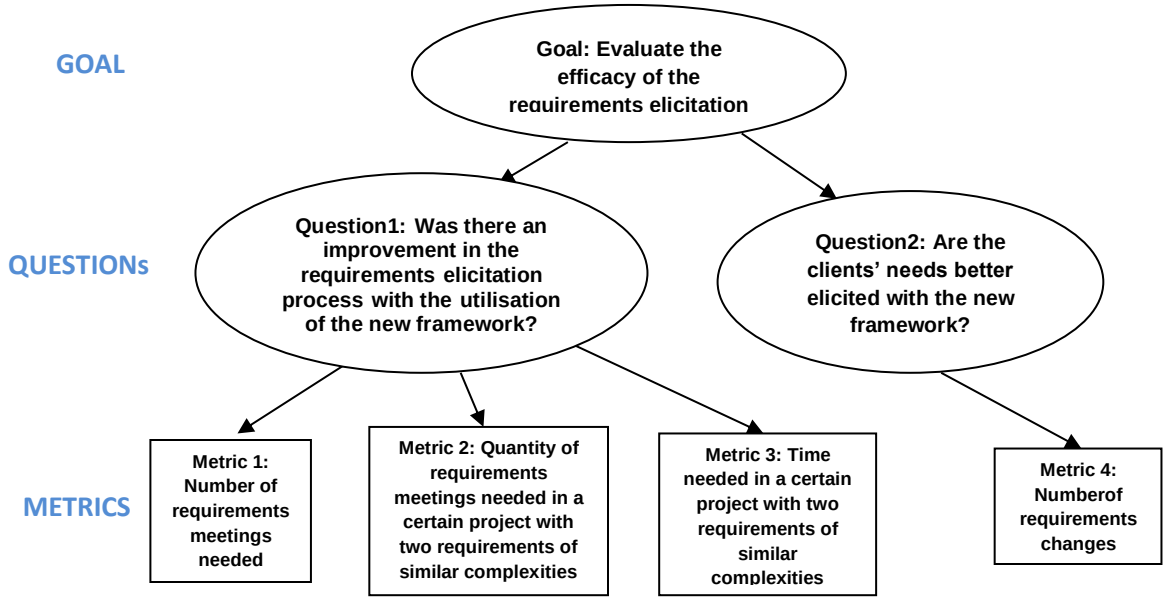


Figure 31 – GQM applied to the current research

The next section will present a brief explanation of each metric. A deeper explanation of the collection and analysis process as well as the partial numbers of the metric are presented in Appendix D.

Measure 1 - Number of requirements meetings needed before and after the framework.

This measure will quantify the number of requirements meetings (any meeting with one or more client stakeholders to perform requirements elicitation using any technique) needed to elicit requirements in a group of projects in each use case before and after the usage of the framework proposed. To calculate this metric it is necessary that the projects have initiated the requirements elicitation process before the framework and continued to perform this process with the framework. Thus, it will be possible to compare the number of

requirements meetings needed before and after the usage of the framework. The hypothesis is that if the need for requirements meetings is reduced it would mean that the requirements meetings are now more effective indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole.

Measure 2 - Difference between quantity of requirements meetings needed in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework.

This measure will quantify the number of requirements meetings needed to elicit requirements in a certain project in a certain complexity of requirements before and after the usage of the framework proposed. The complexity will be calculated in Use Case Points (an approach to determine the complexity of a system before building it (UFPE, 2012)) that will be described in the Appendix E. The comparison will be between the number of requirements meetings needed to elicit a requirement of a certain complexity before the framework, and the number of requirements meetings needed to elicit a requirement of similar complexity after the framework. To calculate this metric it is necessary that the project has initiated the requirements elicitation process before the framework and is still performing this process. The hypothesis is that if the need for requirements meetings is reduced it would mean that the requirements meetings are now more efficient indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole. To improve the analysis of this metric, teams and clients will be the same, to maintain a fixed set of other factors involved,

consequently reducing the factors that could be influencing the result. Thus, it is easier to determine the influence of the framework on the result of the metric.

Measure 3 - Difference between time needed to elicit in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework.

This measure will quantify the amount of time needed to elicit requirements in a certain project with a certain complexity of requirements before and after the usage of the framework proposed. The complexity will be calculated in Use Case Points. The comparison will be between the amount of time needed to elicit a requirement of a certain complexity before the framework, and the amount of time needed to elicit a requirement of a similar complexity after the framework. To calculate this metric it is necessary that the project has initiated the requirements elicitation process before the framework and is still performing this process. The hypothesis is that if the amount of time needed is reduced it would mean that the elicitation is now more efficient indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole. The results should support measure two because the number of meetings could be reduced but if the total time to elicit the requirements increases, the efficiency will not have been increased and just the format of the elicitation process would have been modified. Like measure two teams and clients will be the same, for the same reason of improving the analysis.

Measure 4 - Number of quantity of requirements changes before and after the framework.

This measure will quantify the number of requirements changes across several projects before and after the usage of the framework proposed. To calculate this metric it is necessary that there are at least two projects finished after the framework is used. The hypothesis is that if the quantity of requirements changes is reduced it would mean that the requirements elicitation is now more effective, indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole.

7.2.1 Quantitative Evaluation – Results

Measure 1 - Number of requirements meetings needed before and after the framework

The metric was collected in the same systems observed in the qualitative evaluation.

System	Quantity of meetings needed before the Framework	Quantity of meetings needed after the Framework
2	**10**	**12**
3	**12**	**6**
4	**18**	**9**
5	**14**	**4**
7	**15**	**8**

Table 15 – Results of measure 1

Systems 1 and 6 were not considered as there was no elicitation before the framework. Metric 1 indicates that in general the number of requirements meetings needed was reduced with the Framework which indicates an increase in the efficacy of these meetings as well as the efficiency of the requirements process. In the case of system 2, there was a small increase in the number of meetings as there were mistakes in the application of the ULRC technique leading to the need of another technique to complement it, with the consequent need of new meetings.

Measure 2 - Difference between quantity of requirements meetings needed in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework

The systems selected were systems 4 and 5. System 4 was responsible for implementing a group of Excel sheets. Consequently, it was easy to isolate parts of the system to compare elicitation before and after the framework. Two sheets were selected to be compared based on complexity similarities and also that both sheets were used by the same person. The complexities of the requirements were calculated using Use Case Points. For each sheet, we counted the quantity of meetings required to elicit the respective requirements.

System 4		
Metric 2	**Requirement 1**	**Requirement 2**
Framework	Before	After
Use Case Points (complexity)	31.1	35.8
Techniques used	Interview	Observation / Protocol Analysis
Meetings Needed	10	4

Table 16 – Results of measure 2- System 4

System 5 was responsible for implementing a new version of an existing system. For the same requirements, there were two attempts. First, the requirements team tried to use unstructured interviews. In the second attempt, for the same requirements with the same team they used Brainstorming and MUST.

System 5		
Metric 2	**Requirement 1**	**Requirement 2**
Framework	Before	After
Complexity (Use Case Points)	69.0	69.0
Techniques used	Unstructured Interviews	Brainstorming / MUST
Meetings Needed	13	6

Table 17 – Results of measure 2- System 5

Metric 2 indicates that the quantity of requirements meetings needed for requirements elicitation was significantly reduced in both systems 4 and 5 for selected requirements of similar complexities with the use of the Framework, which indicates an increase in the efficacy of these meetings as well as the efficiency of the requirements process.

Measure 3 - Difference between time needed to elicit in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework

The same systems and requirements as Measure 2, systems 4 and 5, were selected to be compared based on complexity similarities. For each situation, we counted the amount of time required to elicit the respective requirements.

System 4		
Metric 3	**Requirement 1**	**Requirement 2**
Framework	Before	After
Complexity (Use Case Points)	31.1	35.8
Techniques used	Interview	Observation / Protocol Analysis
Time Needed with the Client to Elicit the Requirement	14 hours and 15 minutes (855 minutes)	3 hours and 25 minutes (205 minutes)

Table 18 – Results of measure 3- System 4

System 5		
Metric 3	**Requirement 1**	**Requirement 2**
Framework	Before	After
Use Case Points (complexity)	69.0	69.0
Techniques used	Unstructured Interview	Brainstorming / MUST
Time Needed with the Client to Elicit the Requirement	22 hours and 30 minutes (1350 minutes)	9 hours and 25 minutes (565 minutes)

Table 19 – Results of measure 3- System 5

Metric 3 indicates that the amount of time needed for requirements elicitation was significantly reduced in both systems 4 and 5 for selected requirements of similar complexities with the use of the Framework. As long as both requirements before and after have similar complexities, the results indicate an increase in the efficiency of the requirements process.

Measure 4 - Number of quantity of requirements changes before and after the framework

This metric should consider changes on high level requirements (requirements with a high level of abstraction used to define the scope) before and after the framework. All seven systems mentioned in the qualitative research were considered to be part of the metric. However, just systems 3 and 4 were finished before the ending of this research, thus being capable of counting the number of requirements changes before and after the use of the framework to permit a comparison.

Metric 4				
System	**Number of Requirements before the framework**	**Number of changes before the framework**	**Number of Requirements after the framework**	**Number of changes after the framework**
System 3	39	12	13	3
System 4	43	7	16	3
Total	82	19	29	6

Table 20 – Results of measure 4

Rate before the framework = 19/82 = 0.23 changes / requirement

Rate after the framework = 6/29 = 0.21 changes / requirement

Even with the rate showing a decrease in the number of changes in requirements on these two systems after using the framework, it is not possible to know if such a difference is related to the use of the framework because there are several other uncontrollable variables involved. The difference cannot be considered an expressive rate. Furthermore, there is no indication of the severity of the changes or the complexity of the requirements. To make this metric effective there should be a rigid control on the severity of the changes and the complexity of the requirements as well as the involvement of other systems. It would also be necessary to increase significantly the period of collection.

7.3 Revision and Extension of the Framework

In general, the framework attended its purpose by increasing collaboration in many cases, and reducing interaction when tensions exist. Both purposes showed to be effective to improve the requirements elicitation process. However, even improving the elicitation in the Brazilian Navy, after the usage of the framework, some issues that could affect its efficacy were raised.

The first observation is that it was not enough to explain the techniques to the teams, especially the inexperienced teams. Sometimes even using the technique properly some social behavior can lead to the failure of a technique. As mentioned by Jia et al (2005), in CSCW systems many of the conflicts derive from social human behavior. As an example it was mentioned that in system 2 the technique ULRC was used properly and for the correct reasons. Tensions between the requirements and client teams were high; there were a hierarchical pressure and lack of military knowledge by the requirements team civilian members. Besides all these issues, the client representative has a high IT knowledge. However, even with all these indicators to use the technique the open criticism of the specification built by the client ended up increasing the tensions between the teams that were already high. Consequently, there is the need for the analysts to undertake training in elicitation to improve their skills, or at least be given some advice on appropriate behavior. At the same time in system 5, the brainstorming was effective as it was performed by a highly experienced and skilled analyst. Such observations led us to the conclusion that especially in group techniques where a facilitator is needed, the skills of the analyst are as

important as the technique used. As long as we cannot guarantee this training, some advice for recommended behavior will be included in the framework.

Another issue raised was that teams were lacking in some techniques that were group techniques, although still reducing interaction. The need was pointing to techniques where opinions can be discussed without direct interaction. Especially for complex decisions where there could be bad consequences if the wrong decision is made, involving stakeholders with authoritarian behaviour and in several levels of the hierarchical structure. Consequently, two other techniques were included in the framework. These techniques are not specifically for the area of requirements elicitation, but for managerial decision making. Delphi and Nominal Group Techniques were included. Although created for decision making, the Delphi method has been widely used in information system research (Okoli and Pawlowski, 2004).

Finally, some teams tried to use reverse engineering and struggled with the lack of documentation available. The use of reverse engineering, although highly recommended in many situations, is usually hampered due to the lack of documentation available or at least the lack of updated documentation available to support this technique. Thus, to support the gathering of data for the novel system based on existing functionalities a new technique will be included named "Analogical Reuse of Requirements Frameworks" (Massonet, Lamsweerde, 1997). This technique is based on constructing requirements based on other existing systems and their documentation in the same domain that can provide information about the structure of common knowledge in this domain. This technique will take advantage of the numerous common and unique features existing in a domain as specific as the military one.

i) Nominal Group Technique

As mentioned by CDC (2006) "Nominal (meaning in name only) Group Technique (NGT) is a structured variation of a small-group discussion to reach consensus". In this technique a facilitator sends issues to be discussed to several participants that will send their opinions back and later prioritize the opinions or ideas of the other participants. (CDC, 2006). According to MacGraw and Seale (1988) if characteristics of multiple expert teams such as status and rank appear to threaten the effectiveness of the consensus decision making, the requirements engineer may choose to work individually with each expert or use the Nominal Group Technique. The NGT is a problem solving procedure that reduces negative effects (i.e. non-participation, conflicts) that may be caused by face-to-face interaction among members of the multiple expert team (MacGraw and Seale, 1988).

To reduce political or hierarchical influence even further, instead of allowing the participants to discuss ideas, these ideas can be given prior to the meeting, presented by the moderator; and then prioritised individually, without any discussion. This variation will preserve the anonymity of the ideas; reducing the possibility that the ideas presented by the higher authorities be given the biggest votes as usually the owner of the idea tends to defend it with more emphasis.

ii) Delphi Technique

The Delphi method, belonging to the subjective-intuitive methods of foresight, was developed in the 1950s in operations research (Cuhls, 2003) and was funded by the US Air Force. The goal was to create a method that could reach the most consistent consensus among a group of experts (Dalkey and Helmer, 1963). According to Linstone and Turoff (1975), Delphi can be defined as a method that allows a group of experts to make a decision within a structured process of communication. "To accomplish this 'structured communication' there is provided: some feedback of individual contributions of information and knowledge; some assessment of the group judgement or view; some opportunity for individuals to revise views and some degree of anonymity for the individual responses" (Linstone, Turoff, 1975).

In contrast to Nominal Group Technique, the Delphi technique never allows decision participants to meet in person (MacGraw and Seale, 1988). According to Cuhls (2003) this approach consists of a survey conducted in two or more rounds and provides the participants in the second and later rounds with the results of the previous round so that the experts' answers from the second round on can be supported by the other participant's opinions. It provides qualitative as well as quantitative results and has beneath it explorative, predictive even normative elements (Cuhls, 2003).

According to Okoli and Pawlowski (2004), a core positive aspect of this method is that it does not allow experts to have a direct confrontation. As Dalkey and Helmer (1963) observed, by controlling the interaction between the experts the method leads to more technical opinions and less bias from emotions. Furthermore, the opinions are gradually

constructed and consequently better structured. Direct confrontation, often leads to standard opinions, criticism of novel ideas, defence of a stand once taken, and agreement with partners (Dalkey, Helmer, 1963). It is useful when there are divergences or political dominance so strong that the anonymity of ideas and opinions is essential to guarantee an effective result

Yousuf (2007) mentions that this technique is particularly useful when there are divergences or political dominance so strong that the anonymity of ideas and opinions is essential to guarantee a valid result.

Consequently, in our framework, such a method will permit some degree of collaboration; however, without personalizing the information and also eliminating sources of dysfunctional conflict, as long as there is no face-to-face interaction.

iii) Analogical Reuse of Requirements Frameworks

Massonet and Lamsweerde (1997) present the idea of specification reuse. Their approach suggests reusing fragments of requirements specification to reduce elaboration time and increase requirements quality. Requirements analysts tend to use their experience to solve gaps in the knowledge of the requirements of a system under development, sometimes reusing ideas of previous systems developed (Massonet, Lamsweerde, 1997). Analogical reasoning techniques are useful in these cases. Their approach suggests a library of cases that have been specified in the past.

In a domain as specific and complex as the military, specifications tend to be big and detailed; however, such a domain also presents several common aspects among the systems.

Several data structures and classes are used similarly in a considerable number of systems. Although the technique suggests the creation of a library of existing fragments of specifications, as a starting point to the approach the specifications themselves can be stored in a common repository to be consulted by the analysts of all systems. This technique would reduce the interaction as many common domain concepts were already elicited for previous systems and were also already modelled. Thus, the analysts would not have to interact with the client to elicit them, rather just confirming the similarity. The approach would also reduce time to model this information. Another meaningful gain is that the analysts with lack of knowledge in the domain could also use these previous specifications to understand it without the need of interacting with the client.

With the inclusion of these techniques the framework was updated (Figure 32).

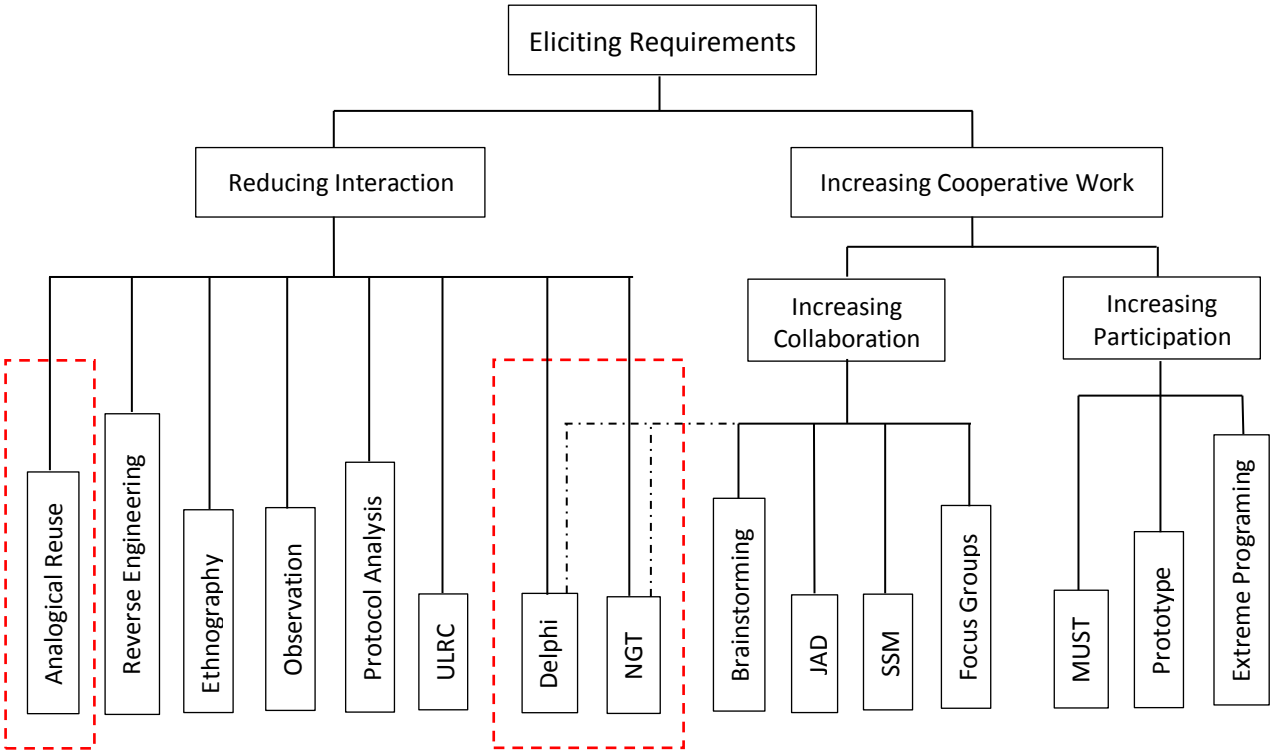


Figure 32 – Updated Framework

Just like the framework, the guidelines were also updated to include the new techniques as shown in the following table:

Situation	What should be used	What should be avoided
High turnover	• Reverse Engineering • **Analogical Reuse** • Participant Observation • Ethnography • Protocol Analysis *Could also be used:* Brainstorming, JAD or SSM	• Interviews • Questionnaires • Prototyping • Agile • MUST
Authoritarian management approach	• Reverse Engineering • **Analogical Reuse** • Participant Observation • Ethnography • Protocol Analysis • **Delphi** • **NGT**	• Interviews • Group techniques • MUST • Prototyping • Agile
Requirements team members with no familiarization with the domain	• Reverse Engineering • Participant Observation • Ethnography • Protocol Analysis • **Analogical reuse**	• Avoid involving members with lack of domain knowledge in the meetings of group techniques. • Do not allow these members to perform interviews.
Client team with participative managerial approach	• Brainstorming • JAD • Focus Groups • Prototypes and Storyboarding • MUST	
Client expert in the domain / Client with IT knowledge / Tensions between client and requirements teams	• ULRC Could also be used: • Questionnaires • **Delphi** • **NGT**	• Group techniques • Prototypes and Storyboarding • MUST • Agile
Various areas involved with tensions between the areas	• Interviews – Interview each area responsible separately and confront the answers. • Reverse Engineering – observe documents that	• SSM.

	define the responsibilities of each area. • Ethnography – focus on the need of information from other areas. • Participant Observation – focus on the need of information from other areas. • **Delphi** • **NGT**	
Various areas involved with **NO** tensions	• SSM	
Extremely experienced development team Client team with time availability, high commitment and participative managerial approach	• Agile	

Table 21 – Updated Guidelines for the teams

7.4 Summary

Once a proposal is presented it needs to be tested to evaluate if it is useful and the degree of improvement that it brings. The framework was evaluated first with a qualitative approach. This evaluation was based on interviews with the Brazilian Navy team personnel after using the framework. Such an approach allowed us to conclude that the framework was bringing a significant improvement to the task of requirements elicitation, thus giving an idea of the framework efficacy.

After this result, quantitative research was performed to evaluate the improvements provided by the framework. To allow this evaluation, four metrics were defined, collected, and the results analysed comparing the task of requirements elicitation before and after the framework. The results showed a significant improvement, after the use of the framework leading to the conclusion that the framework increased the efficiency of the task. However, some interviews showed some possible improvements to the framework. Consequently, the framework was updated based upon the research results, and some additional techniques were included.

Chapter 8

Conclusions and Future Work

The aim of this research was to improve system development in military environments. However, many authors working in this field agree that more than 70 percent of system failures are related to requirements problems (Davis et al, 2006) (Lindquist, 2005). Thus, the first step towards improving system development would logically be addressing shortcomings in the requirements elicitation process. The first goal of the literature review was to better understand the requirements area and the complexity of the requirements process. After that, the existing techniques were studied with their main characteristics. Finally, the existing frameworks to support the selection of elicitation technique were expatiated focusing on the influencing factors used to guide the technique selection. Once the existing techniques and frameworks were well known, it was time to analyse the military domain. The domain was studied focusing on its characteristics and then characterised as an environment guided by well-defined roles and strong regulations. The existing frameworks for the military domain were also studied. These frameworks have a higher-level goal of organizing the whole area of system development, especially regarding contractual aspects, however without detailing some specific aspects such as elicitation techniques selection.

After the literature review a further study was necessary to determine exactly the issues to be addressed by the solution. A qualitative research was performed to discover what were the principal issues influencing requirements, according to the opinions among the people involved in system development in the Brazilian Navy. The results of the qualitative research showed that the main concerns of the workers were social issues rather than technical ones. The existence of social issues among the concerns was expected, however not in such a high percentage, as almost all the main concerns were social. This response corroborates the literature that mentions the requirements phase as the most influenced by human aspects. However, even being common sense that several social issues are present during this phase, the existing frameworks usually address just technical issues such as: size of the team, type of project, type of information, availability of information, number of COTS, criticality, domain knowledge, experience from the analysts, time allocated to elicit requirements, among others. Consequently, a framework addressing the social issues would be a contribution to the requirements community.

The list of issues, the analysis of these issues and the perception that the main influencing aspects on the requirements phase in a military environment are social rather than technical are the first contributions of this research. The issues mentioned exist due to all the human interaction required during this phase. Political influence in technical decisions, the need of changing rules and procedures from a new boss due to the high turnover, and the misunderstandings raised in multicultural teams, are all human aspects that influence the efficacy of the requirements phase.

Once the issues to be addressed were determined a solution was developed. The first step was to understand the nature of the requirements elicitation tasks. When studying this activity in the literature it can be noticed that it is essentially a collaborative task. Furthermore, the efficacy of this task is strongly related to the degree of collaboration among the people involved. Now that the task is understood, the nature of the military environment must be revealed. Studying types of organisation we noticed that the military environment is essentially a role organisation with a strong hierarchical structure. Such a characteristic can be explained by the core activity of any armed force. In the battlefield, any decision is based on hierarchy. However, this organisational characteristic contrasts with the collaborative nature of the requirements elicitation task. Such contrasting characteristics are one of the reasons for the large number of social issues raised during this phase in this environment.

Finally, to develop a solution for these issues, there should be a theoretical background to explain all the organisational problems mentioned. The explanation came from the Conflict Theory. According to this theory, there are several conflicts in any organisation. There are three main sources of organisational conflicts: fight over scarce resources; fight for control (hierarchy); and sectors with different goals. The theory also mentions a fourth source of conflict: the conflict between different personal roles. Both hierarchy and role conflicts are intrinsic to the military environment. As long as the explanation for the problems came from the conflict theory the solution was also based upon the solutions presented by the same theory. The theory mentions that there are essentially two ways of solving conflicts. In the case of small conflicts, the solution is to increase collaboration. In the case of more intense conflicts, the parties involved should be separated from one another. In the same direction, the solution was based on the same two possibilities.

Even with the current notion that social issues are as important as technical ones the existing frameworks are mainly based on technical influencing factors such as type of system, size of the project and size of the team. Our approach is based on the social issues raised in the qualitative research, although also taking advantage of the existing frameworks. Consequently, the solution proposed separates the techniques in three groups. The first group gathers several group techniques that aim to increase the collaboration between the individuals involved. Such techniques strongly depend on the skills of the facilitator. Thus, the solution presents some recommendations to guide the way of organizing the stakeholders into groups to guarantee the efficacy of group techniques in the military environment, avoiding, for example, hierarchical intimidation. The second group of techniques is based on the participation of the client in the development team. To be effective, these techniques depend on the availability and characteristics of the client to be effective. The third group of techniques will aim to reduce the direct interaction needed to elicit requirements. These techniques are indirect techniques that will keep client organisation and requirements teams to a minimum of face-to-face interaction and even when some interaction is needed it will be in a way to depersonalize the information. Although the techniques in the third group were not created with the purpose of reducing interaction, this research found this new useful perspective of these techniques and applied it in the solution.

It is also important to note that not all conflicts are negative. The conflict theory mentions two types of conflicts: dysfunctional and functional. Functional conflicts happen when the individuals involved in the conflict have in common the goal to solve the problem, however, in different ways. As both people involved want to solve the problem, they will easily try to reach a consensual solution. This form of conflict is positive and even increases

creativity. On the other hand, dysfunctional conflicts happen when there are more than just discordant opinions, rather there are emotions involved and personal factors guiding the conflict. This last type of conflict contributes to reduce the efficacy of any group task and thus should be eliminated. All the groups of techniques present in the solution proposed try to reduce the sources of dysfunctional conflicts while trying to encourage functional conflicts as long as the control remains with the facilitator.

Once the framework was defined, it was used by several teams in the Brazilian Navy to evaluate its efficacy. A qualitative research was performed with the teams involved and the results showed that in general team members were satisfied with the framework. According to their opinion, there was a significant improvement in the elicitation activities specifically and in the requirements phase as a whole. To gain an even more detailed assessment, including the framework effectiveness a quantitative evaluation was conducted. Some metrics were defined using GQM and collected. The quantitative results corroborated the qualitative response and indicated a significant improvement in the elicitation effectiveness. Consequently, there could be a conclusion that the framework was attending its purpose of improving the efficacy of the requirements phase.

However, the evaluation also showed areas to be improved. Regarding the quantitative evaluation, some metrics were not effective due to the small period available to collect them. The metric "number of requirements changes before and after the framework" did not show any significant result. Analyzing the data it showed that, to be effective, this metric would need the minimum of another two or more years of data collection as it depends on the completion of the development of some systems and the systems available

would take more than the period of the thesis to be completed. The metrics "number of requirements meetings needed" showed a significant decrease in the number. Such a decrease in this metric indicates an improvement in the requirements process. However, to be more reliable there should be a measurement of systems where the complexity could be similar, which was not possible during the period under study.

Regarding the qualitative evaluation, some teams asked for a hybrid technique where there could be an increase in the collaboration yet reducing interaction. As there was no specific requirements elicitation technique that could solve this problem the solution came from the management area. Two techniques were then incorporated: Delphi and Nominal Group Technique. These techniques increase collaboration by showing the opinions from one participant to the other with no or few direct interactions and thus with depersonalization of ideas. However, these techniques will demand from the analysts the time to organise the ideas and to wait for the opinions, which could jeopardize the schedule. Consequently, there should be the stakeholder analysis to evaluate the possibility of using these techniques. Finally, some teams showed the intention of using previous ideas from the domain, however, without any existing ontology or sometimes with no previous experience in the domain. The perfect solution in this case would be to create ontology for the domain with meta-concepts and the interactions among them. However, such a solution would demand a large amount of time and would be so complex that should be the subject of a whole new research. To solve the problem within the time of this research, we included a technique to support the exchange of knowledge between the teams with less effort. The technique “Analogical Reuse” encourages this exchange of requirements concepts between projects and could be implemented as a start with a repository of specifications.

The framework itself is the second contribution of this research together with the notion that social issues are as important as technical ones on the requirements engineering discipline. Both the construction and the evolution of the framework followed the "action research" (Reason, Bradbury, 2007) approach taking into account the opinions of the requirements analysts involved. The framework proposed showed its efficacy through the qualitative evaluation and also showed an improvement in the requirements elicitation process with the quantitative evaluation. According to all requirements team members the framework brought some direction to the selection of elicitation techniques, especially regarding the complex problem of personal factors rather than just technical ones. In this complicated environment, many times the selection must be based on these social factors, as long as they are the factors that influence the most the efficacy of requirements elicitation. For this reason, the teams involved in the research approved the solution as a guide to help them to increase the elicitation efficacy by supporting them on controlling the organisational conflicts that constantly arise during this task.

By interviewing the Brazilian Army and the Brazilian Air Force it was possible to determine that the problems found in the Brazilian Navy are similar to the problems found in all Brazilian Armed Forces. During these interviews, we also presented the solution to these Armed Forces as they showed a big interest as they face the same issues. However, it was not possible to try the solution on these armed forces due to the research schedule. In the same direction, people related to several foreign armed forces involved in software development were interviewed. The results of these interviews showed that requirements teams from other countries face the same issues when developing systems in the military environment. Some of them requested the solution, once completed. From this, it could be possible to believe that

the solution is generalizable to any Armed Force around the world, as long as they share similar goals, and the regulations and procedures are all based on the same principles: "Hierarchy and Discipline". Indeed, this study could also serve as a blueprint for research into other stages in the system development lifecycle of the military environment, in particular, the design and construction phases.

Presenting the research to members of some civilian organisations, again they mentioned that many of the issues studied are similar in their institutions and that the solution proposed could support their requirements process as well. To prove the efficacy of the solution proposed in civilian organisations will be here proposed as future work. However, the idea of using social issues to guide the requirements process has already proved effective in the literature and corroborated by this research. The novel idea would be to understand what would be those issues in a civilian organisation.

Another important gain would be added to the research if a tool could be built based on the issues gathered by the research, indicating the best technique with the probability of efficacy in each case.

Finally, the research indicated the need for an ontology to this complex environment to support and standardize the elicitation and specification of systems. There could be core ontology for any Armed Force with derivations for each speciality such as Navy, Army and Air Force depending on the structure of each country. Such an ontology would support the elicitation of new systems, as long as they could be based on core concepts previously elicited for other systems.

References

References

Alves,P., (2009) “Requirements Engineering for Web Applications” Post-graduation monograph in Computing Science - Universidade Federal de Pernambuco

Aranda G., Vizcaíno A, Cechich A., Piattini M., (2005) “Towards a Cognitive-Based Approach to Distributed Requirement Elicitation Processes” Proceedings of the Fourth IEEE International Conference on Cognitive Informatics Pages 322-330 / 2005

Aurum, A., Wohlin, C., (2003) “The fundamental nature of requirements engineering activities as a decision-making process,” Information Software Technology, 45, 2003, pp. 945–954

Ayalew, Y., Masizana-Katongo, A., (2008) “A Multi-Criteria Group Decision Support Approach for Requirements Elicitation Techniques Selection” Asian Journal of Information Technology 7 (2): 40-52. ISSN: 1682-3915 – Medwell Journals

Bailey, R., (1982) “Human Performance Engineering: a Guide for System Designers” Prentice-Hall

Bartocci, G., Potts, L., Cotugno C. (2008) “Experience Report: Communicating Ethnographic Findings Effectively within Multidisciplinary Teams and to your Clients” Conference’08, September 22-24, Lisbon, Portugal

Beck, K., et al, (2001). "Manifesto for Agile Software Development". Agile Alliance. http://agilemanifesto.org/

Bergman, M., King, J., Lyytinen, K., (2002) “Large-scale requirements analysis revisited: the need for understanding the political ecology of requirements engineering,” Requirements Engineering, Volume 7, Number 3, 2002, pp. 152-171

Boehm, B., Rose, P., Horowitz, E., Lee, M.J., (1994) “Software requirements negotiation and renegotiation aids: a theory-W based spiral approach” Proceedings of ICSE 95

Boehner, K., DePaula, R., Dourish, P., Sengers, P., (2005) “Affect: From Information to Interaction” Proceedings of the 4th decennial conference on Critical Computing: between sense and sensibility - Pages: 59 - 68 - ISBN:1-59593-203-8

Bolton, D., Jones, S., Till, D., Furber, D., Green, S., (1994) “Using domain knowledge in requirements capture and formal specification construction” Requirements Engineering Social and Technical Issues, Chapter 6, Academic Press

British Council (2012)“The Delphi technique”
http://www.britishcouncil.org/eltons-delphi_technique.pdf (accessed in 18/01/2012)

Button, G., Sharrock, W., (1994) “Occasioned practices in the work of software engineers” Requirements Engineering Social and Technical Issues, Chapter 9, Academic Press

Carrizo, D., Dieste, O., Juristo, N., (2008) “Study of Elicitation Techniques Adequacy” 11th Workshop on Requirements Engineering

CDC (2006) “Gaining consensus among stakeholders through the Nominal Group Technique” Evaluation Briefs, Department of Health and Human Services, no.07 november 2006

Checkland, P., (1988), “Soft System Methodology: An Overview” Journal of Applied System Analysis, 15

Christel, M., Kang, K., (1992) “Issues in Requirements Elicitation” Technical Report - CMU/SEI-92-TR-012 - ESC-TR-92-012 - September 1992

Cockburn, A., Highsmith, J., (2001) “Agile Software Development: The People Factor” Software Management, pag 131-133, November 2001

Codur, K., Dogru, A., (2009) “Evolution of Software Development Standards in the Military Domain and Effects on Software Applications” Proceedings of the joint international and annual ERCIM workshops on principles of software evolution (IWPSE) and software evolution (Evol) workshops / Amsterdam, The Netherlands - SESSION: Empirical studies and experience reports Pages: 41-46

Comer, T.N., Commer, D.J., Rodjak, (1990) “Developing reusable software for military systems, why it is needed, why it isn't working” ACM SIGSOFT Software Engineering Notes -Volume 15, Issue 3 Pages: 33 - 38

Cooper, J.R., Seok-Won Lee , Gandhi, R.A. , Gotel, O., (2009) “Requirements Engineering Visualization: A Survey on the State-of-the-Art” Fourth International Workshop on Requirements Engineering Visualization (REV), 2009

Cooper, S., Bendiab, T., (1998) “Consensus: multi-party negotiation support for conflict resolution in concurrent engineering design” Journal of Intelligent Manufacturing, 9, 155-159

Corbin J., Strauss A., (1990) “Grounded Theory Research: Procedures, Canons and Evaluative Criteria” Qualitative Sociology, Vol. 13, no.01

Coughlan J., Macredie R., (2002) “Effective communication in requirements elicitation: A comparison of methodologies”. Journal of Requirements Engineering

Crawford A., (1994) “Advancing Business Concepts in a JAD Workshop Setting” Ed. Prentice Hall

Crebolder, J., Pronovost, S., Lai, G. (2009) “Investigating Virtual Social Networking in the Military Domain” 14th ICCRTS - “C2 and Agility”

Cuhls, K., (2003) “Delphi Method” Fraunhofer Institute for Systems and Innovation Research, Germany

Dahrendorf R.,(1959) “Class and Class Conflict in Industrial Society” Ed. Routledge and Kegan Paul

Dalkey, N., Helmer, O., (1963) “An experimental application of the Delphi method to the use of experts” Management Science 9, pp. 458-467

Damian D., Eberlein A., Shaw M., Gaines B., (2003) “An exploratory study of facilitation in distributed requirements engineering” Journal Requirements engineering 2003, vol. 8, no1, pp. 23-41 - Springer, London

Davey B., Cope C., (2008) “Requirements Elicitation – What’s Missing?” Issues in Informing Science and Information Technology Volume 5, 2008

Davies, R., (2001) “The Power of Stories” Fourth International Conference on eXtreme Programming and Agile Processes in Software Engineering, pp. 46-49

Davis, A., (1992) “Operational Prototyping: A New Development Approach” Software, 9(5): 70-78

Davis, A.M., (1990) “Software Requirements: Analysis and Specification” Prentice Hall - ISBN:0-13-824673-4

Davis, C. J., Fuller, R. M., Tremblay, M. C., Berndt, D. J., (2006). “Communication challenges in requirements elicitation and the use of the repertory grid technique” Journal of Computer Information Systems, 78

Delbecq, A.L., Van de Ven, A.H. and Gustafson, D.H. (1975) “Group Techniques for Program Planning: A Guide to Nominal Group and Delphi Processes” . Glenview, IL: Scott, Foresman & Company, 1975. “Nominal Group Technique”

DePaula, R., Dourish, P. (2005)“Cognitive and Cultural Views of Emotions” Proceedings HCIC, (2005)

Deutsch, M., (1973) “The Resolution of Conflict, constructive and destructive processes” Yale University Press

Deutsch, M., (1969) "Conflicts: Productive and Destructive" Journal of Social Issues25 (1):7-41, in Easterbrook (1993) "CSCW:Cooperation or Conflict" p. 26.

DGA, (2009), "Strategic Plan for Research & Technology in Defense and Security" Direction Générale de l'Armement – Ministère de la Défense

Dittrich, Y., (1998), "How to make Sense of Software.Interpretability as an Issue for Design" Research Report 19/98, University of Karlskrona Ronneby 1998.

Dittrich, Y., Randall, D., Singer, J., (2009) "Software Engineering as Cooperative Work" CSCW 2009 Volume 18, Numbers 5-6, 393-399

DoD, (1987), "Report of the Defense Science Board Task Force on Military Software" Office of the Under Secretary of Defense for Acquisition- Washington, D.C. 20301

DoD, (2000), "Report of the Defense Science Board Task Force on Defense Software" Office of the Under Secretary of Defense For Acquisition and Technology- Department of Defense – United States of America.

DODAF, (2009) "DoD Architecture Framework Version 2.02" DoD Deputy Chief Information Officer (accessed in 22/10/2012)
http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf

Douglas, M., (1986) "How institutions think" Syracuse

Douglas, T., (1970) "A Decade of Small Group Theory" Bookstall Publications

Dourish, P., (1995), "Developing a reflective model of collaborative systems" ACM Transactions on Computer-Human Interaction (TOCHI), Volume 2 , Issue 1 Pages: 40 - 63 ISSN:1073-0516

Dourish, P., Button, G., (1998) "On 'Technomethodology': Foundational Relationships between Ethnomethodology and System Design" Human Computer Interaction, 13(4), 395–432, 1998

Downs, A., (1967) "Inside Bureaucracy" in Steers, R., Introduction to Organisational Behavior, (1988)Third edition, The Scott, Foresman Series in Management and Organisations, p. 391

Easterbrook, S., (1994) "CSCW: Cooperation or Conflict?" Computer Supported Cooperative Work, Springer-Verlag

Easterbrook, S., (1994b) "Resolving requirements conflicts with computer-supported negotiation" Requirements Engineering, Academic Press ltd.

Easterbrook, S., Nuseibeh, B., (2004) "Fundamentals of Requirements Engineering?" Publisher Addison Wesley - ISBN 0201619199

Ebert, C., Wieringa, R., (2006) "Requirements Engineering: Solutions and Trends" Engineering and Managing Software Requirements, Chapter 20 – Springer Berlin Heidelberg Part 3 453-476 – April 2006

Fife, W., (2005) "Doing Fieldwork: Ethnographic Methods for Research in Developing Countries and Beyond" New York: Palgrave Macmillan

Flynn D., Jazi M. (1998) "Constructing user requirements: a social process for a social context" Info Systems J (1998) 8, 53-83

Fontana, A., Frey, J.H., (2005) "The interview: From neutral stance to political involvement. In Zhang and Wildemuth (2009) "Unstructured Interviews"

Freeman R., (1984) "Strategic Management: A Stakeholder Approach" Pitman, Boston

French J., Raven, B., (1959) "The bases of social power," In D. Cartwright (ed), Studies in Social Power, 1959, pp. 150-167

Frese, R., (2003) "Project success and failure: what is success, what is failure, and how can you improve your odds for success?" Retrieved February 13, 2012 from http://www.umsl.edu, 2003

Goffman E., (1961) "Asylums" Pelican Books

Goffman E., (1974) "Frame analysis" Northeastern University Press

Goffman E., (1988) "Exploring the interaction order" Polity Press

Goguen J., (1993) "Social issues in requirements engineering" Proceedings of IEEE International Symposium on Requirements Engineering, 1993.

Goguen J., Linde C., (1993) "Techniques for Requirements Elicitation" 1st IEEE International Symposium on Requirements Engineering (RE'93), San Diego, USA, 4-6th January 1993, pp. 152-164

Goman C., (2005) "The linkage leader - Brainstorming techniques". Linkage Inc. Burlington, MA

Gotel, O., Cleland-Huang, J., (2009) "Next Top Model: A Requirements Engineering Reality Panel" 17th IEEE International Requirements Engineering Conference

Gotterbarn, D., (2002), "Reducing Software Failures: Addressing the Ethical Risks of the Software Development Lifecycle" Australasian Journal of Information Systems, Vol 9, No 2 (2002)

Gottesdiener E., (2001) "Facilitated Workshops in Software Development Projects". Software Management 2001

Green A., (1960) "Sociology: An Analysis of Life in Modern Society" third edition, MacGraw-Hill p. 65

Gross N., (1959) "The sociology of Education" Sociology today: problems and prospects, New York Basic Books, 1959. p. 140

Grudin, J., (1994) "Computer Supported Co-operative Work: History and Focus" IEEE Computer, 2, (1994) 5,19-26

Hall, E. (1959) "Silent Language" Anchor Books, A Division of Random House Inc.

Hall, J., (1971) "Decisions, decisions, decisions" Psychology Today5; 51-54, 86-87, in Easterbrook (1993) "CSCW:Cooperation or Conflict" p. 26

Handy C., (1999) "Understanding organisations" Fourth Edition, Penguin Books

Hargreaves, E. J. and Damian, D. E., (2004) "Can Global Software Teams Learn from Military Teamwork Models?" In Proc. of the Int'l Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland, May 24, 2004, IEE, pp. 21--23

Harrison R., (1972) "How to Describe Your Organisation" cited in "Understanding Organisations" Handy C., p. 181

Heap K. (1968) "The social group worker as central person" Case conference, cited in Douglas (1970)

Hender J., Rodgers T., (2001) "Improving Group Creativity: Brainstorming Versus non-Brainstorming Techniques in a GSS Environment". Proceedings of the 34th Hawaii Conference on System Sciences. IEEE

Hersey, P., Blanchard, K. H., Johnson, D. E. (2000) "Management of organisational behavior: Leading human resources" (8th ed.). Englewood Cliffs, NJ: Prentice Hall

Hickey, A., Davis, A., (2003) "Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes" 36th Annual Hawaii International Conference on System Sciences (HICSS'03)

Hofstede, G., (1991) “Cultures and Organisations – Intercultural Cooperation and Its Importance for Survival” – Profile Books Ltd

Horowitz I. (1962) “Consensus, Conflict and Cooperation: A Sociological Inventory” Social Forces, University of North Carolina Press

Hsieh, Y., (2006) “Culture and Shared Understanding in Distributed Requirements Engineering” IEEE International Conference on Global Software Engineering (ICGSE'06)

Hudlicka, E., (1999) “Knowledge Elicitation in Complex Military Environments” IEEE Conference and Workshop on Engineering of Computer-Based Systems, 1999. Proceedings. ECBS '99, pp. 328-334. - Location: Nashville, TN, USA

IEEE, (2004) “Guide to the Software Engineering Body of Knowledge” IEEE Computer Society. pp. 1–1. ISBN 0-7695-2330-7 http://www.swebok.org

Isaac, K., (2010) “Out of Uniform: Resume Tips for Military Professionals” www.baseops.net/transition/resumetips.html (accessed on 02/10/2010)

Jackson, M. (2004) "Seeing more of the world" IEEE Software 21 - November/December 2004 ,pag. 83-85

Jarrar, M., (2005) “Towards Methodological Principles for Ontology Engineering” Doctoral dissertation, Vrije Universiteit Brussel, Faculty of Science

Jia, H., Tang, W., He, T., Kong, F., (2005) “A Conflict Resolution Methodology Based on Technology and Sociology – A Complex Approach” The 9th International Conference on Computer Supported Cooperative Work in Design

Jiang L., Eberlein A., Far, B., Mousavi, M., (2008) “A methodology for the selection of requirements engineering techniques” Journal Software and Systems Modeling - Publisher Springer Berlin / Heidelberg - Volume 7, Number 3 / July, 2008

Jirotka, M., (2000), “An Investigation into Contextual Approaches to Requirements Capture” Doctoral dissertation, Computing Laboratory, University of Oxford

Jirotka, M., Goguen, J., (1994) “Requirements Engineering – Social and Technical Issues” Academic Press Limited. ISBN 0-1238-5335-4

Jirotka, M., Luff, P., (2006) “Supporting requirements with video-based analysis” IEEE Software Magazine, 23(3):42-44, 2006

Jones, C., (2000), “Software Assessments, Benchmarks, and Best Practices” Addison-Wesley ISBN 0-201-48542-7

Kensing, F., Blomberg, J., (1998) “Participatory design: Issues and concerns” Computer-Supported Cooperative Work 7 pp. 167-185 Kluwer Academic Publishers

Kensing, F., Simonsen, J., Bodler, K., (1996), “MUST – a Method for Participatory Design.” Proceedings for the fourth biennal conference of participatory design. Computer professionals for Social Responsibility pp. 129-140

Klein, M., (1991) "Supporting Conflict Resolution in Cooperative Design Systems" Transactions on Systems, Man and Cybernetics. Special Issue on Distributed Artificial Intelligence. Volume 21, Number 6, December 1991

Lamsweerde, A., Darimont, R., Letier, E., (1998) “Managing Conflicts in Goal-Driven Requirements Engineering”, IEEE Transactions on Software Engineering, Vol. 24 no. 11

Larman, C; Basili, V., (2003). "Iterative and Incremental Development: A Brief History" IEEE Computer Society, Computer , June 2003, 36 (6): 47–56

Lawler E., Rhode J., (1976) “Information and Control in Organisations” Pacific Palisades, CA: Goodyear, p.22

Leite, J.C., (2009) “Engenharia de Software”.
http://www.dimap.ufrn.br/~jair/ES/ (Accessed in: 27/09/2010).

Liang, P., Avgeriou, P., Clerc, V., (2009), “Requirements Reasoning for Distributed Requirements Analysis Using Semantic Wiki” 2009 Fourth IEEE International Conference on Global Software Engineering - Limerick, Ireland July 13-July 16

Lin, A., Silva, L., (2005) “The social and political construction of technological frames,” European Journal of Information Systems, 14, 2005, pp. 49-59

Lindquist, C. (2005) “Fixing the Software Requirements Mess” CXO Media Inc. November 15, 2005

Lindstrom, L., Jeffries, R., (2004) “Extreme Programming and Agile Software Development Methodologies” Vol. 21, Iss. 3, pp: 41-52, Information Systems Management

Linstone, H., Turoff, M., (1975) “The Delphi Method: Techniques and Applications” Addison-Wesley, London

Lopes, L., Prikladnicki, R., Audy, J., (2006) “Requirements Specification in Distributed Software Development – A Process Proposal” IEEE Software - Volume 23 , Issue 5 (September 2006) Pages: 38 - 44 - 2006

Lottridge D., Chignell M., Kastner M., Zhang Q., Alexandar A., Straus S., (2009) "Model-driven coding with VPAT: The verbal protocol analysis tool" TIC-STH 2009. IEEE pp. 808-813

Maasonet, P., Lamsweerde, A., (1997) "Analogical Reuse of Requirements Frameworks" Proceedings of the Third IEEE International Symposium on Requirements Engineering

Macaulay L.,(1998) "The Role of the Facilitator in Requirements Engineering" ICRE '98 Proceedings of the 3rd International Conference on Requirements – IEEE

MacCaulay L. (1994) "Cooperative Requirements Capture: Control Room 2000" in Requirements Engineering: social and technical issues, Jirotka M. and Goguen J., ed. Academic Press Ltd.

MacGraw, K., Seale, M., (1988) "Knowledge elicitation with multiple experts: considerations and techniques" Artificial Intelligence Review 2, pp.31-44

Mack, R. W., and R. C. Snyder. (1957) "The. analysis of social conflict toward an overview and synthesis." Journal of Conflict Resolution (June) 1:212-48

Maiden, N., Rugg, G., (1996). "ACRE: Selecting Methods For Requirements Acquisition" Software Engineering Journal, 11(3):183-192

Marks S., MacDermid S., (1996) "Multiple roles and the self: a theory of role balance" Journal of Marriage and family, Vol. 58, No.2 pp.417-432

Martin, D., Sommerville, I., (2006) "Ethnography and the social structure of work" Chapter 9 - Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective, Publisher: Springer London, ISBN978-1-84628-110-5

Mathews J., (2007) "JAD/Iterative Approach to Requirements Management". Pierson Requirements Group, Inc.

McGrath, J., (1983), "Groups: Interaction and Performance" Prentice Hall

McGraw, K., Seale, M., (1988) "Knowledge elicitation with multiple experts: considerations and techniques" Artificial Intelligence Review pp. 31-44

Mead G., (1964) "Selected writings". Indianapolis, IN: Bobbs-Merr, cited in "multiple Roles and the seilf: a Tehory of Role balance", Marks S., MacDermid S. (1996)

Mei C., (1999). "Joint Application Design/Development". University of Missouri-St. Louis. Yatco Systems

Melnik, G. (2004). "Introducing Agile Methods: Three Years of Experience". Proceedings of the 30th Euromicro Conference. IEEE. pp. 334–341

Merindol, V., (2010) "New approaches for military innovation in knowledge based economics: An inquiry into the new role of defense in innovation Processes" Institut pour le Management de la Recherché e de l'Innovation

Military Statute, (2004) "Estatuto dos Militares" Marinha do Brasil (Brazilian Navy), Servico de Documentacao da Marinha, Rio de Janeiro

Millen D., (2000) "Rapid Ethnography: Time Deepening Strategies for HCI Field Research". DIS 2000 Brooklyn, New York

Miller H., Jahnke J., Smith D., Storey M., Tilley S., Wong K., (2000) "Reverse Engineering: A Roadmap" Published by ACM 2000

Milne, A., Maiden, N., (2011) "Power and Politics in Requirements Engineering: A Proposed Research Agenda" 2011 IEEE 19th International Requirements Engineering Conference

Minichiello, V., Aroni, R., Timewell, E., Alexander, L. (1990). "In-depth Interviewing: Researching people." Hong Kong: Longman Cheshire Pty Limited

Mintzberg H., (1973) "The nature of managerial work" New York, Harper and Row, ISBN 0060445556

MODAF, (2010) "MODAF Meta Model" MODAF V1.2.004 Released May 2010 http://www.mod.uk/DefenseInternet/AboutDefense/CorporatePublications/InformationManagement/MODAF/ModafMetaModel.htm - (accessed in March 19th, 2012)

Morcos M., Henshaw M., (2009) "A Soft Systems Methodology for Transforming Organisations to Product-Service Systems (Application in Defense and Construction Industry)". 7th Annual Conference on Software Engineering Research (CSER 2009)

Neilsen, E.H., (1972) "Understanding and Managing Conflict" in Steers, R., Introduction to Organisational Behavior, (1988) Third edition, The Scott, Foresman Series in Management and Organisations, p. 371

Nguyen L., (2007) "Using Protocol Analysis to Explore the Creative Requirements Engineering Process" Information Systems Foundation

NIST, (2002) "The Economic Impacts of Inadequate Infrastructure for Software Testing" Gaithersburg, MD: National Institute of Standards and Technology, 2002

Nitzberg, S., (1999) "Ethics in Military and Civilian Software Development" Proceedings from ETHICOMP99

Nuseibeh, B., Easterbrook, S., (2000) "Requirements Engineering: A Roadmap" International Conference on Software Engineering - Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, Pages: 35 – 46

Nuseibeh, B., Kramer, J., Finkelstein A., (1994) "A framework for expressing the relationships between multiple views inrequirements specification" IEEE Transactions in Software Engineering, Vol 20, no. 10

OGSA, (2004)"Ordenanca Geral para o Servico da Armada" (General Ordinance for Navy Service) Marinha do Brasil (Brazilian Navy) – Servico de Documentacao da Marinha, Rio de Janeiro

Okoli, C., Pawlowski, S., (2004) "The Delphi method as a research tool: an example, design considerations and applications" Information and Management 42, pp.15-29

Olmsted-Hawala E., Murphy E., Hawala S., Ashenfelter K., (2010) "Think-aloud protocols: A comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability" CHI 2010, April 10-15

Orlikowski, W., Gash, D., (2000) "Using technology and constituting structures: a practice lens for studying technology in organisations" Organisation Science, Vol 11 No 4, Nov 2000, pp. 404-28

Paetsch, F., Eberlein, A., Maurer, F., (2003) "Requirements Engineering and Agile Software Development" Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE, 03)

Patton, M.Q., (2002), "Qualitative Research and Evaluation Methods" Thousand Oaks, CA Sage

PCOM, (2007) "Plano de carreira para Oficiais" (Career Plan for Navy Officers), 2007 Marinha do Brasil (Brazilian Navy), 8th revision

Perrow,C. (2008) "Software Failures, Security, and Cyberattacks" 3/31/08 after JB chngs

Posner, M. I., (1993) "Foundations of Cognitive Science" MIT Press

Poundy L., (1967) "Organisational Conflicts: Concepts and Models". Administrative Science Quarterly, 1967

Preece, J., Sharp, H., Rogers, Y., (2002) "Interaction Design". John Wiley & Sons

RDM, (2009), "Regulamento Disciplinar para a Marinha" (Disciplinary Regulation for the Navy) Marinha do Brasil (Brazilian Navy), Diretoria do Patrimonio Historico e Documentacao da Marinha, Rio de Janeiro

Reason P., Bradbury H., (2007) "Handbook of Action Research", 2nd Edition. London: Sage, 2007

Redl F., (1942) "Types of Group Formation, Group Emotion and Leadership" Psychiatry, vol V no.4

Robbins S., (1998) "Organisational Behaviour" 8th edition. Prentice Hall

Robinson,W., Fickas, S., (1994) "Supporting Multi-Perspective Requirements Engineering" Proceedings of the First International Conference on Requirements Engineering, IEEE, pp. 206-215

Rose J., Haynes M., (1999) "A Soft Systems Approach to the Evaluation of Complex Interventions in the Public Sector". Journal of Applied Management Studies, 8, 2, 199-216

Rost, J., (2004) "Political reasons for failed software projects," IEEE Software, vol 21(6), Sept/Oct 2004, pp. 102-104

Royce, W., (1970) "Managing the development of large software systems: Concepts and techniques" In Proceedings of IEEE WESTCON. (Los Angeles, CA), 1-9.

Ruhe, G., Eberlein, A., Pfahl, D., (2002) "Quantitative WinWin – A New Method for Decision Support in Requirements Negotiation" SEKE '02, July 15-19, 2002, Ischia, Italy. ACM 1-58113-556-4/02/0700

Scappapietra, S., (1994) "View Integration: A step in Solving Structural Conflicts" IEEE Transactions on knowledge and data engineering" Vol 6, no. 2

Scogin, A., (2006) "The Factors of Software Systems that Contribute to Requirements Elicitation. Software" Engineering Research and Practice: 296-302

Sharp H., Finkelstein A., Galal G., (1999) "Stakeholder Identification in the Requirements Engineering Process" Proceedings Tenth International Workshop on Database and Expert Systems Applications, 1999, pp. 387 – 391

Slocum, M., (2012) "The Simple Steps of Classic Brainstorming" The Triz Journal http://www.realinnovation.com/content/c070604a.asp (Accessed in 12/01/2012)

Šmite, D., (2006) "Requirements Management in Distributed Projects" Journal of Universal Knowledge Management, vol. 1, no. 2 (2006), 69-76

Solingen R., Berghout E., (1999) "The Goal/Question/Metric project: a practical guide for quality improvement of software development", The McGraw-Hill Companies

Sommerville, I., Sawyer, P., Viller, S., (1998) "Viewpoints for requirements elicitation: a practical approach" Proceedings of ICRE'98

Steers, R., (1988) "Organisational Behavior" Third Edition, The Scott, Foresman Series in Management and Organisations

Suchman, L., (1987) "Plans and situated actions: the problem of human-machine communication" Cambridge University Press New York, NY, USA - ISBN:0-521-33137-4

Thomas, K., (1976) "Conflict and Conflict Management". In M.D. Dunnette, Ed. Handbook of Industrial and Organisational Pstchology, pp. 889-935, in Easterbrook (1993) "CSCW:Cooperation or Conflict" p. 26

Tsumaki, T., Tamai, T., (2005) "A Framework for Matching Requirements Engineering Techniques to Project Characteristics and Situation Changes" Proceedings of SREP'05, Paris, France, August 29–30, 2005

UFPE (2012) "Estimativa de esforço baseado em pontos de casos de uso" http://www.cin.ufpe.br/~if717/slides/pontos-de-caso-de-uso.xls (accessed in 04/03/2012)

Viller S., (1991) "The group facilitator: A CSCW perspective" Proceedings of the second European Conference on Computer-Supported Cooperative Work

Viller, S., Sommerville, I., (1999). "Social Analysis in the Requirements Engineering Process: from ethnography to method" 4th International Symposium on Requirements Engineering (RE'99), Limerick, Ireland, 7-11th June 1999

Wallington, P., (2004) "Secrets of Success with a New Boss" CIO http://www.cio.com/article/32062/Secrets_of_Success_With_a_New_Boss?page=1&taxonomyId=3123 - (accessed in 23/02/2012)

Weber , R., Aha, D., (2002) "Intelligent Elicitation of Military Lessons"
Copyright 2002 Association for Computing Machinery. ACM - IUI'02, January 13-16, 2002, San Francisco, California, USA

Westfall, L. (2005) "Software Requirements Engineering: What, Why, Who, When, and How" World Conference on Quality and Improvement 2005

Wiegers,K., (1996), "Creating a Software Engineering Culture" Dorset house

William B., (2005) "A Theory of Enterprise Transformation", Systems Engineering, Wiley Periodicals, Inc.

Williams (2005) “Soft Systems Methodology” The Kellogg Foundation

Wittgenstein L., (2009) “Philosophical Investigations” 4th edition, Ed. Wiley-Blackwell

Yildiz, B., Miksch, S., (2007) “Ontology-Driven Information Systems: Challenges and Requirements” International Conference on Semantic Web and Digital Libraries (ICSD), Bangalore, India

Yin, R. K. (1994). Case study research: Design and methods. (2nd ed.) Thousand Oaks, CA: Sage

Yousuf, M., (2007) “Using Experts’ Opinions Through Delphi Technique” Practical Assessment, Research & Evaluation, Volume 12, Number 4, May 2007

Zachman, J., (1987) “A framework for information systems architecture” IBM Systems Journal, vol 26. No.3, 1987

Zarinah M.K., Siti Salwah S.,(2009) “Supporting Collaborative Requirements Elicitation Using Focus Group Discussion Technique”. International Journal of Software Engineering and its Applications. Vol.3, No.3, July 2009

Zave, P. (1997) “Classification of Research Efforts in Requirements Engineering” ACM Computing Surveys, 29(4): 315-321

Zhang, Y. , & Wildemuth, B. M. (2009). “Unstructured interviews”. In B. Wildemuth (Ed.), Applications of Social Research Methods to Questions in Information and Library Science

Figures

- *Figure 1 – Comparison between CMMI and MPS.Br maturity levels (Campos, 2012)*

- *Figure 2 - Waterfall Model, Spiral Model, Agile Model*

Waterfall
http://www.csse.monash.edu.au/~jonmc/CSE2305/Topics/07.13.SWEng1/html/text.html

Spiral
http://www.stylusinc.com/VirtualBusiness/CustomWebsites/Development.php

Agile
http://www.magpiebrain.com/2005/02/14/the-agile-release-process/

- *Figure 3 - Increasing costs of repair (NIST, 2002)*

http://gwaredd.blogspot.com/2010/02/game-development-in-post-agile-world.html

- *Figure 4 - A spiral model of the requirements engineering process*

Lopes, L., Prikladnicki, R., Audy, J., (2006) "Requirements Specification in Distributed Software Development – A Process Proposal" IEEE Software - Volume 23 , Issue 5 (September 2006) Pages: 38 - 44 - 2006

- *Figure 5: Membership status of facilitator*

Douglas T., (1970) "A Decade of Small Group Theory" Bookstall Publications

- *Figure 6 - Communication challenges*

Davey B., Cope C., (2008) "Requirements Elicitation – What's Missing?"
Issues in Informing Science and Information Technology Volume 5, 2008

- *Figure 7 - Details of elicitation activities*

Hickey, A., Davis, A., (2003) "Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes"
36th Annual Hawaii International Conference on System Sciences (HICSS'03)

- *Figure 8 - The power culture*

Handy C., (1999) "Understanding organisations" Fourth Edition, Penguin Books

- *Figure 9 – The task culture*

Handy C., (1999) "Understanding organisations" Fourth Edition, Penguin Books

- *Figure 10 – The person culture*

Handy C., (1999) "Understanding organisations" Fourth Edition, Penguin Books

- *Figure 11 – The role culture*

Handy C., (1999) "Understanding organisations" Fourth Edition, Penguin Books

- *Figure 12 – Summary of the qualitative research methodology – This research*
- *Figure 13 – Client in a prominent position to reinforce the hierarchical difference - This research*
- *Figure 14 - Client discussing a questionnaire presented by the requirements team - This research*
- *Figure 15 – Requirements workshop with multiple ranks - This research*
- *Figure 16 – Multidisciplinary Team in a Project – This research*

- *Figure 17 – Turnover – Rate in a Project- This research*
- *Figure 18 – Main issues to be addressed and their consequences – This research*
- *Figure 19 – Factors involved in group interaction (McGrath, 1983)*

McGrath, J., (1983) "Groups – Interaction and Performance" Prentice Hall

- *Figure 20 – Classification of Group tasks*

McGrath, J., (1983) "Groups – Interaction and Performance" Prentice Hall

- *Figure 21 – Behavioural modes of tackling conflict with (in brackets) the outcome sought in each mode.*

Thomas, K., (1976) "Conflict and Conflict Management". In M.D. Dunnette, Ed. Handbook of Industrial and Organisational Pstchology, pp. 889-935, in Easterbrook (1993) "CSCW:Cooperation or Conflict" p 26.

- *Figure 22 – Social network with sources of power (Milne, Maiden, 2011)*
- *Figure 23 – Parallel Viewpoints of Easterbrook model (Easterbrook, 1994)*

Easterbrook S., (1994) "CSCW: Cooperation or Conflict?" Computer Supported Cooperative Work, Springer-Verlag.

- *Figure 24 – The Dynamics of a conflict episode*

Poundy L., (1967) "Organisational Conflicts: Concepts and Models". Administrative Science Quarterly, 1967

- *Figure 25 – Focus of the existing research on conflicts and requirements – This research*
- *Figure 26 – Comparison between the focus of the existing research and the focus of this work according to the phases of conflict – This research*
- *Figure 27 – classification of requirements elicitation techniques – This research*
- *Figure 28 – Requirements stages and activities, and the recommended facilitator - This research*
- *Figure 29 – protocol analysis and observation performed in System 4- This research*
- *Figure 30 - Workshop with multiple ranks – This research*
- *Figure 31 – GQM applied to the current research – This research*
- *Figure 32 – Updated Framework – This research*

Appendices

Appendix A

Interview Questions (Exploratory Qualitative Research)

This research is designed to gain an understanding of the development of systems in the Brazilian Navy.

1. What are your name and current position?

2. What is your professional relationship with the Navy?

3. What is or was your participation in Navy software projects?

4. For how long have you participated in the development of software?

5. Do you think that in the Navy systems are usually delivered on time? If not, why?

6. Do you believe that currently in the Navy systems satisfy their customers?

7. In your opinion, do teams have the necessary infrastructure to develop systems?

8. Do you think that project activities are well explained to the members of the development team?

9. Do you think the expectations of customers are known by the development team during system development?

10. What is your opinion of the way clients and team members interact?

11. Do you believe that military systems are different from other systems? If so, why? If not, why not?

12. Do political decisions affect the development of systems in the Navy?

13. Do you think that, as a rule, the client commitment in the Navy is appropriate?

14. In your point of view, what features of the military environment influence the development of systems?

15. Do Navy teams have something to guide them during development?

16. What kind of information do team members have to do their jobs?

17. Is there something to guide teams on how to accomplish specific activities such as requirements elicitation?

18. How does the Navy undertake a review of requirements nowadays? What are the techniques used that you are aware of?

19. Is there a tool to support this activity?

20. Do you believe that teams have the time they need to elicit requirements?

21. How often do requirements change during development? Do they change more than they should?

22. Are development teams trained to elicit requirements?

23. In the projects in which you participated, who was responsible for eliciting requirements?

24. How did the teams you have participated in elicit requirements?

25. Do you know any elicitation techniques other than the ones you use?

26. Do you believe that the way requirements are elicited in the Brazilian Navy nowadays is the best way?

27. What problems do you notice during requirements elicitation?

28. Do you think the different working relationships between team members, the fact that there are military, civilians and consultants hired in the marketplace, affects requirements elicitation?

29. In general, how do people validate requirements in the Navy?

30. Do you believe that all team members have the same commitment to the project?

31. Do you notice any behaviour contrast related to the different work relationships?

32. Do you believe that other techniques could help requirements elicitation in the Navy?

33. To conclude, what else do you think causes difficulties for system development in the Brazilian Navy?

Appendix B

Interview Questions (Evaluation Case Study)

This research is designed to evaluate the effectiveness of the requirements elicitation framework presented.

1. How is your requirements team composed?

2. How much experience does your team have on requirements elicitation?

3. What was the elicitation technique used previously to the presentation of the framework?

4. Which technique or techniques did you use after the presentation of the framework?

5. Did you notice any changes related to the client's team behavior when using the new techniques? What changes did you notice?

6. Did you notice changes in the requirements team behavior? What changes did you notice?

7. Did you notice changes in the quantity of requirements meetings needed with the new technique? Was this quantity reduced or increased?

8. Did you notice any changing in the goals of the meetings? Were they more general or more specific with the new technique?

9. Did you perceive changes in the collaboration between the client team and the requirements team? What changes did you notice?

10. Did you notice changes in the collaboration between the client team members? What changes did you notice?

11. Did you notice changes related to the understanding of the system by the requirements team? Was it better or worse?

12. Did you notice changes related to the understanding of the system by the client's team? Was it better or worse?

13. Were there any changes in the number of requirements changes? Did it decrease or increase?

14. Did you notice any changes in the requirements phase as a whole? Which changes did you notice?

15. Is there any other comment about the framework that you would like to mention?

Appendix C

Guidelines for Elicitation Techniques

According to the literature, all the techniques have their pros and cons and specific situations when they should be used. Below we describe each technique, its basic steps, its advantages and disadvantages, and finally, the situations in which it should be used. The list includes some traditional techniques: structured and unstructured interviews and questionnaires, in order to guide the analysts when performing such common techniques.

1) Requirements Elicitation Techniques

1.1 Structured Interviews

Interviews are a widely used tool to obtain individuals' perceptions, opinions and ideas. Depending on the level of predefined organization there are three types of interviews: structured interviews; semi-structured interviews; and unstructured interviews (Fontana & Frey, 2005). In a structured interview all questions are predefined and there is not much space to adaption to a certain situation. All questions are repeated in the same way and order to all respondents. This format aims to reduce the influence of the interviewer in the result. Structured interviews and surveys are similar in terms of structure, however in the first questions are posed orally (Zhang, Wildemuth, 2009)

Steps (Fontana, Frey, 2005):

1. Getting in: accessing the setting;
2. Understanding the language and culture of the interviewees;
3. Preparing the questions to be asked;
4. Locating an informant. Not every person in the research setting will make a good informant;
5. Gaining trust and establishing rapport;
6. Capturing the data. Note-taking is a traditional method for capturing interview data. In a structured interview, note-taking usually works, however, when possible, it is good to record the interview.

When it should be used:

- It can be performed by analysts with little experience, but some experience in the field is necessary;
- It should be used when the problem is well defined;
- It should not be used at the beginning of the requirements elicitation phase;
- It is predominantly qualitative, although quantitative information can also be found;
- It should be used when there is the need of quick feedback;
- It should be used when there are relatively few specialists, or several but with the same basic understanding;
- It is good to raise basic and tactical information;
- The analyst should know how to lead the technique in order to keep the interview impartial;
- Respondents must have an interest in the system;
- It should not be used if there is a probable turnover of client representatives.

Advantages:

- Simple, direct and customizable to several scenarios;
- Easy to schedule.

Disadvantages:

- Difficult to separate facts from opinions;
- Requires time to perform with all stakeholders and also some time to consolidate the results.

1.2 Non-structured interviews

The unstructured interview technique derives from sociology and anthropology where it was used to understand realities and cultures. Minichiello et al. (1990) defined them as "interviews in which neither the question nor the answer categories are predetermined. Instead, they rely on social interaction between the researcher and the informant".

The analyst must perform the interview without a predefined structure and no preconceived ideas of the responses. Questions will be elaborated during the interview based on the responses obtained. Consequently, each interview can generate very distinct ideas and responses. These interviews can be used together with participant observation.

However, even a non-structured interview needs some previous preparation to guide the questions. "Non-structured interviews cannot be started without detailed knowledge and preparation, if you hope to achieve deep insights into people's lives" (Patton, 2002). "The researcher will keep in mind the study's purpose and the general scope of the issues that he or she would like to discuss in the interview" (Fife, 2005).

Steps (Fontana, Frey, 2005**):**

1. Getting in: accessing the setting;
2. Understanding the language and culture of the interviewees;
3. Deciding on how to present oneself. An unstructured interview is a two-way conversation. The quality of the conversation is influenced, to a great extent, by how the interviewer represents him- or herself;
4. Locating an informant. Not every person in the research setting will make a good informant;
5. Gaining trust and establishing rapport;
6. Capturing the data. Note-taking is a traditional method for capturing interview data. But in an unstructured interview, note-taking is likely to disrupt the natural flow of the conversation. Thus, when possible, it is preferable to record the interviews.

When it should be used:

- It must be performed by analysts with good experience to extract as much information as possible during the questions, but there is no need for experience in the domain;
- It should be used when the problem is not well defined;
- It can be used at any stage of the requirements phase;
- It is predominantly qualitative;
- It should be used when there is the need of quick feedback;
- It should be used when there are relatively few specialists, or several but with the same basic understanding;
- It is good to raise tactical and strategic information;
- Respondents must have an interest in the system;
- It should not be used when there is a large hierarchical difference between the customer representative and the analyst, especially if the management model used by the client is authoritarian;
- It should not be used if there is a probable turnover of client representatives.

Advantages:

- Simple, direct and customizable to several scenarios;
- Easy to schedule.

Disadvantages:

- Difficult to separate facts from opinions;
- Requires time to perform with all stakeholders and also much time to consolidate the results.

1.3 Questionnaire

A questionnaire is similar to a structured interview but the questions are sent to the respondent that must answer and return them to the analyst.

Steps:

1. Accessing the setting;
2. Studying the culture and language used in the domain;
3. Decide who will participate in the research;
4. Define the questions using the jargons of the domain;
5. Send the questionnaire;
6. Gather the information and analyse it;

7. Present the results.

When it should be used:

- Dispersed groups;
- Large number of stakeholders;
- Feedback needed fast;
- It is predominantly quantitative;
- There is the need of some experience in surveying;
- There is the need of some familiarity in the field;
- It should not be used if there is a possible turnover of client representatives.

Advantages:

- In comparison to other techniques does not use much time from clients;
- Low cost;
- It could be managed by only one analyst.

Disadvantages:

- Low return rate;
- Answers could not correspond to the reality.

1.4 Brainstorming

Brainstorming is a group technique to encourage creativity by gathering a list of ideas spontaneously without criticism.

The brainstorming must address a specific issue. Sessions with multiple issues tend to be inefficient. In addition, the problem should require the generation of multiple ideas. Groups must not exceed 12 participants. There must be beginners and experts in the group. Stakeholders must generate as many ideas as they can without any space for criticism. The group simply provides ideas that can lead to a solution not applying any analytic judgement about feasibility. Decisions are reserved for a later date.

This technique should be used when the client management model is participative. It is prudent to separate customers into different hierarchical groups. Then, the analysts should

present one groups' ideas to the other group without mentioning who was the author. The analyst-facilitator should not allow criticism of the ideas raised during the session.

Steps (Slocum, 2012)**:**

1. Form team consisting of subject matter experts as well as non subject matter experts;
2. Define the problem the team will brainstorm solutions for (create a concise problem statement);
3. Ensure the team understands the problem (baseline problem understanding);
4. Explain the brainstorming rules (No criticism – all ideas are valid/Quantity is important, not quality);
5. Initiate brainstorming;
6. Record each idea;
7. Affinitize the ideas;
8. Create an idea matrix and utilise a selection technique or voting to prioritise ideas for further analysis.

When it should be used:

- New systems and new ideas needed;
- Vague requirements;
- Good starting point;
- Exploratory requirements;
- Customer management model must be participatory.

Advantages:

- Generates several ideas in a relatively short period of time;
- With the appropriate facilitation, encourages innovative thoughts;
- Encourages team formation;
- Ideas can be constructed based on other ideas.

Disadvantages:

- Requires an experienced facilitator:
    - Avoid the trial of ideas;
    - To keep the focus of the meeting.

1.5 JAD (Joint Application Design) / Workshop

It is a workshop that gathers executives and IT personnel in several meetings to establish general goals for the system. Executives must be present to guarantee that the system provides the necessary information and reports. These two distinct groups of stakeholders can resolve their differences of opinion. There must be a strong preparation to guarantee the coverage of all important subjects.

According to Mei (1999), the following are the main steps: identify project objectives and limitations; identify critical factors of success; define project deliverables; define schedule of activities during the workshops; select participants; prepare the workshop material (graphics, spread sheets, etc.); organize the workshop; prepare to educate and inform the participants about the goals and details of the project; coordinate the logistics of the workshop (to be held outside of the workplace to avoid interruptions).

Steps (Mei, 1999):

1. JAD project definition;
2. Research on user requirement;
3. Preparation for the JAD session;
4. Conducting and facilitating the JAD session itself;
5. Predicting and obtaining approval of the final document that incorporates all decisions made.

When it should be used:

- A small group of experienced representatives without much time;
- Requirements may be raised immediately;
- System knowledge dispersed among various stakeholders;
- Consensus required between the stakeholders;
- It should be used when the management model is participative or with all stakeholders in the same hierarchical level.

Advantages:

- Powerful to reach consensus;
- Supports team formation;
- All stakeholders will have the same vision of the system;
- It can expose and solve political problems;

- Forge concordance between client and requirements teams;
- Results are available immediately.

Disadvantages:

- Difficulty to schedule a date with the availability of all stakeholders;
- Requires some preparation before the meetings;
- Difficulty to reach consensus;
- Require skills from the facilitator;
- Hierarchy could interfere within the meeting.

1.6 Delphi

Delphi is a panel for experts in the area. It is composed of a series of rounds. In the first round the panelists give their opinions on the subject. The opinions of each panelist are sent to all other panelists. In the subsequent rounds, the panelists will have the opportunity of reviewing their opinions based on other's opinions. All opinions are anonymous in order to guarantee depersonalization of ideas.

Steps (British Council, 2012):

1. The moderator establishes the goal and sends it to the panellists;
2. Panellists send their responses to the moderator;
3. The moderator collates the responses and circulates them anonymously after each round, as the basis for the next round;
4. (The panellists have at each stage a full record of what comments and nominations other panellists have made, but they do not know who made which comment or voted for which entry. Nor do they know the final result);
5. The moderator organizes a meeting to present the results.

When it should be used (British Council, 2012)**:**

- When you want to generate a lot of ideas and want to ensure that all members participate freely without influence from other participants;
- When you want to identify priorities or select a few alternatives for further examination;
- The problem does not lend itself to precise analytical techniques but can benefit from subjective judgments on a collective basis;

- The individuals needed to contribute to the examination of a broad or complex problem have no history of adequate communication and may represent diverse backgrounds with respect to experience or expertise;
- More individuals are needed than can effectively interact in a face-to-face exchange.
- Time and cost make frequent group meetings infeasible;
- The efficiency of face-to-face meetings can be increased by a supplemental group communication process;
- Disagreements among individuals are so severe or politically unpalatable that the communication process must be refereed and/or anonymity assured;
- The heterogeneity of the participants must be preserved to assure the validity of the results, i.e., avoidance of domination by quantity or by strength of personality ("bandwagon effect").

Advantages (British Council, 2012)**:**

- Shift a significant portion of the effort needed for individuals to communicate from the larger respondent group to the smaller monitor team;
- Powerful to reach a consensus without criticism from one another;
- Avoid conflicts between members.

Disadvantages (British Council, 2012):

- Time consuming;
- Possibility of imposing monitor view's and preconceptions of a problem upon the respondent group by overspecifying the structure of the Delphi and not allowing for the contribution of other perspectives related to the problem;
- Poor technique of summarizing and presenting the group response and ensuring common interpretations of the evaluation scales utilised in the exercise;
- It ignores and not explores disagreements, so that discouraged dissenters drop out and an artificial consensus is generated.

1.7 Nominal Group Technique

The nominal group technique (NGT) is a decision making process that can be used by groups of any size to make a decision in a short period of time taking into account all opinions. Each member gives a solution and a short explanation. Based on the solutions proposed all participants rank the ideas to find a unique solution.

Steps (Delbecq et al, 1975)**:**

1. Provide tables for participants;
2. Develop the focus question. Write the question as an issue statement which defines the issue;
3. Silently, brainstorm ideas. Participants are allowed 5 – 10 minutes to write down all their responses;
4. Round robin sharing of ideas. Each person shares one idea at a time in a round-robin format;
5. Discuss and clarify all ideas on the flip chart;
6. Ranking of alternatives;
7. Ranked items are discussed;
8. Second Ranking. From among the ranked items on the previous list, each participant ranks the ten items he now considers most important. This is done silently and independently as in step 3 (This ranking might be limited to five items per person.) The rankings are again averaged;
9. Final rankings are discussed;
10. Format for sharing results. Create a table listing all ranked items, beginning with the item ranked the highest.

When it should be used:

- When you want to generate a lot of ideas and want to ensure that all members participate freely without influence from other participants;
- When you want to identify priorities or select a few alternatives for further examination.

Advantages:

- Enhance one or more dimensions of effectiveness of decision-making groups;
- Powerful to reach a consensus without criticism from one another;
- Avoid conflicts between members.

Disadvantages:

- It lacks flexibility by being able to deal with only one problem at a time;
- There must be a certain amount of conformity on the part of the members involved in NGT;
- A large amount of time needed to prepare for the activity;
- Produces a large number of ideas.

1.8 SSM (Soft Systems Methodology)

This is a systematic approach to deal with real world problems. SSM requires participants to become aware of their personal perspectives and values, and these in turn determine what aspects of a situation in which everyone understands as being part of the system under study. This technique gathers members from different departments to create a common understanding of the interactions between these departments. The facilitator asks questions of each participant creating a diagram of the flow of information among the departments involved to reach a common understanding. Each participant can pose questions about the diagram to the facilitator. The facilitator can show possible improvements to the area under study.

Steps (Williams, 2005)**:**

1. Placing the situation of the problem;
2. Expressing the situation of the problem;
3. Formulating basic definitions of relevant systems;
4. Building conceptual models of systems of human activities;
5. Comparing the models with the real world;
6. Defining desirable and doable changes;
7. Taking actions to improve the situation of the real world.

In stage 4 of construction drawings are made with the participation of the members. The drawings represent views of the situation that are discussed by everyone.

Criteria that must be specified to ensure that a basic definition supplied is accurate and comprehensive (Williams, 2005):

Customers – who are the beneficiaries of the particular system? (Who will benefit or participate in its operations?)

Actors – who will be responsible for implementing and inserting data in the system?

Transformation – which transformations the system will perform?

World view – which world view that makes the transformation meaningful.

Owner – who has the authority to stop the system or modify the parameters of the system?

Environmental – elements outside the system whose constraints are taken as given.

When it should be used:

- Small group of specialists from several different interrelated areas;
- Overview of the System and its constraints;
- Knowledge is dispersed among stakeholders in several areas;
- The system will benefit various departments.

Advantages:

- It saves time;
- Provides a good feedback about behaviour, experiences and desires;
- Allows the participants to know constraints imposed by other departments involved.

Disadvantages:

- It could generate unending discussions;
- It could not reach consensus;
- It could be affected by personal or political issues.

1.9 Focus Groups

Focus groups are a group technique that gathers a group of people to obtain their opinions about a specific subject. This method can be used as a requirements elicitation technique to solve ambiguous requirements or other specific issues. This method gathers various levels of stakeholders. It generates interaction within the group to raise ideas on a theme. The moderator leads a discussion on a topic or specific service. Sessions usually last about two hours.

Steps:

1. Establish the goal of the meeting;
2. Establish the members to be involved;
3. Set a meeting;
4. Moderate the meeting avoiding other discussing then the subject.

When it should be used:

- Small group of experts who can spend time frequently;
- Complex requirements developed iteratively;
- Knowledge is dispersed among stakeholders;
- There should be a consensus among these stakeholders.

Advantages:

- It saves time;
- Provides a good feedback on attitudes, experiences and desires;
- Allows participants to know other opinions and to evolve from them.

Disadvantages:

- Issues of trust can lead to personal criticisms;
- In a homogeneous group tends to lose information;
- It requires an experienced facilitator to lead the discussion, keeping the focus and the time limit;
- Difficulty in gathering those involved.

1.10 MUST

MUST is a technique based on Participatory Design. According to Kensing et al (1996) bringing users to participate in the design is a way of reducing the possibility of a system not fulfilling their needs. MUST combines ethnography and intervention, and relies on the committed participation of workers and managers. "Ethnography and intervention contrast in terms of their basic approaches and intended results: ethnographers originally strove not to change the phenomena they were studying, while interventionists deliberately set up activities to change the organisation in order to learn from the reactions to the change" (Kensing et al, 1996). Kensing et al (1996) argue that this combination of techniques can provide a way of understanding the client organization while trying some possible solutions.

In the case of the environment under study, this technique could be used to create the participation of the user in the design of the new system, especially with clients with a high

level of criticism, although with a participative management approach. A high ranking client would have the opportunity to discuss ideas as the tasks are being performed, and the lower ranking members will have the opportunity to prove empirically their opinions based on their practical expertise. For this technique to be successfully used it is important that the client team has the availability to suffer interventions and also to accept criticisms on the way they perform their tasks. It will be useful as well to try new ideas.

Steps (Kensing, Simonsen, Bodker, 1996):

1. Project establishment – activity to select tools and techniques and to establish the team as a social unit;
2. Strategic Analysis – activity to clarify and delimit which work domains should be in the focus in the design project;
3. In-depth analysis of selected work domain – the purpose is to reveal and develop an understanding of the rationale behind current work practices (ethnography with intervention);
4. Developing visions of the overall change – to develop one or more visions of the overall change (Future workshops);
5. Anchoring the visions – in order for a vision to materialize, it needs to be deeply rooted in the organisation iterations, especially between activity 1 and 2 and between activity 3 and 4.

When it should be used:

- When there is the availability of the client to be in the development team;
- When tasks can be interrupted to gather new ideas of procedure;
- When the characteristics of the client allow criticism of the current procedures.

Advantages:

- Creates new ideas during the work;
- Allows the perception of problems while they occur;
- Brings the requirements team to the client environment;
- Develops the knowledge of the requirements team about the domain.

Disadvantages:

- It interrupts the work of the client;
- It can generate tensions due to constant criticism.

1.11 Prototyping / Storyboarding

This method is based on creating incomplete versions of the software being developed. These incomplete versions must be discarded after being used. These prototypes should be created in a short period of time without any concerns about good practices as they will be discarded. Prototypes support the gathering of information on a system and help the validation of parts of this system. This technique can be used together with any other technique. Storyboards and diagrams are also considered low quality prototypes. You can also create a schematic diagram of the system to be evaluated by the user and thus raise requirements.

Steps:

1. Identify the basic requirements, including the desired inputs and outputs. Details can be ignored;
2. Develop an initial prototype. Only interfaces;
3. Review prototype: users use the prototype and provide feedback;
4. Review and improve the prototype until the necessary requirements for the construction of the system are known;
5. Discard the prototype.

When it should be used:

- Predominantly, graphics systems;
- Vague requirements or requirements that require exploration;
- A small group of experts;
- Process with a few obscure points.

Advantages:

- A picture is worth a thousand words;
- With various involved it is easier to reach consensus;
- It allows feedback early in the project on the part of the user.

Disadvantages:

- It can be confused with the real system;
- It can be criticized for the lack of details;
- More complex functionalities can be difficult to implement;
- The Notation to represent the stream must be simple.

1.12 Extreme Programming

Extreme programming (XP) is a software development methodology that aims to put together the user and the developer in a single team. It is an agile methodology that constantly delivers versions of the software to increase productivity. Its strength relies on the client participation, constant interaction and constant deliveries.

Steps (Melnik, 2004):

- Release Planning: This is focused on determining what requirements are included in which near-term releases, and when they should be delivered. Customers and developers are both part of this. Release Planning consists of three phases:
    - Exploration Phase: In this phase, the customer will provide a shortlist of high-value requirements for the system. These will be written down on user story cards;
    - Commitment Phase: Within the commitment phase, business and developers will commit themselves to the functionality that will be included and the date of the next release;
    - Steering Phase: In the steering phase, the plan can be adjusted, new requirements can be added and/or existing requirements can be changed or removed.

- Iteration Planning: This plans the activities and tasks of the developers. In this process, the customer is not involved. Iteration Planning also consists of three phases:
    - Exploration Phase: Within this phase, the requirement will be translated to different tasks. The tasks are recorded on task cards;
    - Commitment Phase: The tasks will be assigned to the programmers and the time it takes to complete will be estimated;
    - Steering Phase: The tasks are performed and the end result is matched with the original user story.

When it should be used (Melnik, 2004):

- A very experienced development team that conjugates development and elicitation skills;
- Client with availability of time;
- Client with a participative approach;
- Knowledge of the domain is concentrated in few experts.

Advantages (Melnik, 2004):

- It reduces the cost of changes in requirements by having multiple short development cycles;
- Extreme programming encourages starting with the simplest solution;
- Fast feedback;
- It does not propagate requirements errors.

Disadvantages (Melnik, 2004):

- Sometimes increases the effort to change the software;
- The client could interfere in development techniques;
- It can lead to rework.

1.13 Reverse engineering

Reverse Engineering is the process to base the development of a new software product in the analysis of the existing system. For this purpose several sources of information can be used such as documentation, interfaces, specification and database amongst others.

Steps:

1. Inventory analysis - collection of documentation and other information relating to the software;
2. Restructuring documents - organize and prepare the documentation to be understood;
3. Reverse Engineering - extract information from the executable program source code or the original design and specification;
4. Restructuring of code and data - correctness of modules to make the system more easily understood and modified;
5. Forward Engineering - reconstruction of the software to improve their performance and, fundamentally, to meet new requirements.

When it should be used:

- When the existing system has characteristics that are appreciated by the clients;
- When the documentation is updated or at least usable;
- When the existing system is a highly graphical system and the interfaces can provide information;
- When the system will be updated mainly to upgrade the technology used.

Advantages:

- It reduces the need to talk to the clients to understand the system;
- It provides an overall vision of the system.

Disadvantages:

- Time consuming;
- It can lead to inserting defects or unwanted characteristics from the old system.

1.14 Analogical Reuse of Requirements Frameworks

This technique is based on specification reuse. It suggests reusing fragments of requirements specification to reduce elaboration time and increase requirements quality. Requirements analysts tend to use their experience to solve gaps in the knowledge of the requirements of a system under development, sometimes reusing ideas of previous systems developed (Massonet, Lamsweerde, 1997). Analogical reasoning techniques are useful in these cases. Their approach suggests a library of cases that have been specified in the past.

In a domain as specific and complex as the military, specifications tend to be big and detailed, however, such a domain also presents several common aspects among the systems. Several data structures and classes are used similarly in a considerable number of systems. Although the technique suggests the creation of a library of existing fragments of specifications, as a starting point to the approach the specifications themselves can be stored in a common repository to be consulted by the analysts of all systems. This technique would reduce the interaction as many common concepts of the domain were already elicited for previous systems and were also already modelled. Thus, the analysts would not have to interact with the client to elicit them, rather just confirming the similarity. The approach would also reduce time to model this information. Another meaningful gain is that the analysts with lack of knowledge in the domain could also use these previous specifications to understand it without the need of interacting with the client.

Steps:

1. Analyse the goal and the concepts involved in the system;
2. Observe other systems already developed in the organisation;
3. Find similar concepts in other systems;
4. Create a sketch of the model to your system based on the other existing systems;
5. Validate the model with the client;
6. Evolve the model.

When it should be used:

- When there are similarities with other systems;
- When the access to the client is difficult;

- When there are many analysts without domain knowledge;
- When the concepts of the system are complex.

Advantages:

- It reduces modelling time;
- It reduces interaction with the client;
- It enhances confidence in the requirements team by the client;
- It produces a model that was already partially validated.

Disadvantages:

- It can lead to a model being out of the scope;
- It can be time consuming;
- It can be hard to achieve;
- It depends on some level of standardization by the organisation.

1.15 Ethnography

Ethnography is a technique that came from anthropology where researchers sought to understand the behavior of societies. This technique when brought to the development of systems aims to generate an understanding of an entire sector or department through observation of interactions of the members of these departments or sectors. The observation of these interactions can provide a general vision of types of existing relationships as well as of jargons and acronyms used.

The procedure includes a requirements analyst observing daily the client department, recording everything relevant to the system and to the understanding of the work. Doubts could be resolved without interrupting too much the processes, not interfering in the natural flow of information.

This technique can be used as an initial phase of any system development. However, it is costly because the analyst is inserted into the customer's environment for some time. This analyst should have some experience to know what to watch.

It is a technique that should be used when dealing with customers of very complex and specific areas with many specific terms.

Steps:

1. Establish the environment to study;
2. Collect the data. (The analyst must be in the environment without interfering);
3. Analyse the data collected;
4. Validate the data.

When it should be used:

- When the domain is very specific and it has many specific characteristics;
- Analysts do not have any knowledge of the domain;
- Analysts have experience in the technique.

Advantages:

- Study in the natural environment;
- It enables the discovery of new needs;
- It enables knowledge of practices as well as organisational culture.

Disadvantages:

- Time consuming;
- It depends on the observations of the analyst;
- The analyst should know what to observe.

1.16 Participant Observation

Participant Observation comes from anthropology and aims to have a deep comprehension of a group of individuals, the tasks performed by them as well as the environment where these tasks take place. The participant stays in the setting for a considerable period of time and creates an involvement with the group under study.

Steps:

1. Establish the activities to study;
2. Explain the participation to the clients involved;
3. Observe the activity without interfering;
4. Analyse the data collected;
5. Validate the data with the client.

When it should be used:

- System already exists;
- System has a routine operation;
- Process without many obscure points.

Advantages:

- Provides a notion of the system during the usage;
- It captures aspects of the process that could not be written.

Disadvantages

- It takes too much time;
- It can influence the work generating distortions in the observation;
- It could show only the common cases, avoiding exceptions;
- It reviews the work as it is and could not generate new requirements.

1.17 Protocol Analysis

Protocol analysis is a method of psychological research that provokes verbal reports of the survey participants. Protocol analysis is used to study thinking in cognitive psychology. The user describes aloud every step of his activities while performing a specific task. Meanwhile, an analyst observes the task to understand it. If possible filming should be used.

Steps:

1. Establish the task to be observed;
2. Explain the technique to the person to be observed;
3. Perform the activity (filming if possible);
4. Analyse the data;
5. Validate the data with the client.

When it should be used:

- Experienced analyst;
- There is a consensus about the application;
- The specialists have high domain knowledge;
- There is high customer interest;
- There is an easy access to the information;
- Individual routine tasks.

Advantages:

- Provides qualitative data;
- It documents thoughts of the process related to performance.

Disadvantages:

- Time consuming;
- Difficult to analyse.

1.18 ULRC (User led requirements construction)

This technique aims to reduce the gap between client organization and requirements team. The requirements team describes the current situation together with the client in order to teach elicitation techniques to the customer personnel. Once understand the technique the client team does the elicitation of the new system requirements.

Steps:

1. Establish the clients to be involved in the process;
2. Elicit the existing procedure together with the client to teach him/her how to perform elicitation;
3. The client elicits the requirements for the new system;
4. Analysts analyse the specification and update it;
5. Validate the new specification with the client.

When it should be used:

- Experienced analyst;
- Client with some technical knowledge;
- There is time to train the client;
- There is client interest in raising the requirements;
- When the client management model is the authoritarian;
- When there are tensions between requirements team and client staff.

Advantages:

- Desires are written directly by the client;
- Generates elicitation knowledge in the client team.

Disadvantages:

- Time consuming;
- It can generate rework to write the specification.

Table of techniques and guiding characteristics

Technique	Definition	When to use it
Non-structured interview	Interview without predefined questions	• It must be performed by analysts with good experience to extract as much information as possible during the questions, but there is no need for experience in the field; • It should be used when the problem is not well defined; • It can be used at any stage of the requirements phase; • It is predominantly qualitative; • It should be used when there is the need of quick feedback; • It should be used when there are relatively few specialists, or several but with the same basic understanding; • It is good to raise tactical and strategic information; • Respondents must have an interest in the system; • It should not be used when there is a large hierarchical difference between the customer representative and the analyst, especially if the management model used by the client is authoritarian; • It should not be used if there is a probable turnover of client representatives
Structured Inteview	Interview with predefined questions	• It can be performed by analysts with little experience, but some experience in the field is necessary; • It should be used when the problem is well defined; • It should not be used at the beginning of the requirements elicitation phase; • It is predominantly qualitative, although quantitative

		information can also be found; • It should be used when there is the need of quick feedback; • It should be used when there are relatively few specialists, or several but with the same basic understanding; • It is good to raise basic and tactical information; • The analyst should know how to lead the technique in order to keep the interview impartial; • Respondents must have an interest in the system; • It should not be used when there is a large hierarchical difference between the customer representative and the analyst, especially if the management model used by the client is authoritarian; • It should not be used if there is a probable turnover of client representatives.
Questionnaire	Organized list of questions	• Dispersed groups; • Large number of stakeholders; • Feedback needed fast; • It is predominantly quantitative; • There is the need of some experience in surveying; • There is the need of some familiarity with the field; • It should not be used if there is a probable turnover of client representatives.
Brainstorming	Meeting to search for ideas without censorship of these ideas	• New systems and new ideas needed; • Vague requirements; • Good starting point; • Exploratory requirements; • Customer Management model must be participatory;

		• If necessary it should be separate in different hierarchical groups.
JAD / Workshop	It is a workshop that gathers executives and IT personnel in several meetings to establish general goals for the system.	• Small group of experienced representatives without much time; • Requirements may be raised immediately; • System knowledge dispersed among various stakeholders; • Consensus required between the stakeholders; • It should be used when the management model is participative or with all stakeholders in the same hierarchical level.
SSM (Soft System Methodology)	Meeting of various experts from different sectors to reach consensus about the system overview. At the meeting, a drawing is made with the participation of the stakeholders.	• Small group of specialists from several different interrelated areas; • Overview of the System and its constraints; • Knowledge is dispersed among stakeholders in several areas; • The system will benefit various departments.
Focus Groups	The focus group uses various participants: users or prospective users of the application. The moderator leads a discussion on a topic or specific service. Sections typically take about two hours.	• Small group of experts who can spend time frequently; • Complex requirements developed iteratively; • knowledge is dispersed among stakeholders; • There should be a consensus among these stakeholders.
Delphi	Delphi is a panel for experts in the area. It is composed of a series of rounds. In the first round the panelists give their opinions on the subject. The opinions of each panelist are sent to all other panelists. In the subsequent rounds the panelists will have the opportunity of reviewing their opinions based on other's opinions. Panelists will not meet in person.	• When you want to generate a lot of ideas to and want to ensure all members to participate freely without influence from other participants; • When you want to identify priorities or select a few alternatives for further examination; • The problem does not lend itself to precise analytical techniques but cart benefit from subjective judgments

		on a collective basis; • The individuals needed to contribute to the examination of a broad or complex problem have no history of adequate communication and may represent diverse backgrounds with respect to experience or expertise; • More individuals are needed than can effectively interact in a face-to-face exchange; • Time and cost make frequent group meetings infeasible; • The efficiency of face-to-face meetings can be increased by a supplemental group communication process; • Disagreements among individuals are so severe or politically unpalatable that the communication process must be refereed and/or anonymity assured; • The heterogeneity of the participants must be preserved to assure the validity of the results, i.e., avoidance of domination by quantity or by strength of personality ("bandwagon effect).
Nominal Group Technique	The nominal group technique (NGT) is a decision making process that can be used by groups of any size to make a decision in a short period of time taking into account all opinions. Each member gives a solution and a short explanation. Based on the solutions proposed all participants rank the ideas to find a unique solution.	• When you want to generate a lot of ideas to and want to ensure all members participate freely without influence from other participants. • When you want to identify priorities or select a few alternatives for further examination.
Prototyping / storyboarding	The method is based on creating incomplete versions of the software being developed. These incomplete versions must be discarded after	• Predominantly, graphics systems; • Vague requirements or requirements that require exploration;

	being used. Prototypes support the gathering of information to a system and help the validation of parts of a system. This technique can be used together with any other technique.	• Small group of experts; • Process with a few obscure points.
MUST	MUST is a technique based on Participatory Design. MUST combines ethnography and intervention and relies on the committed participation of workers and managers.	• When there is the availability of the client to be in the development team; • When tasks can be interrupted to gather new ideas of procedure; • When the characteristics of the client allow criticism of the current procedures.
Extremme Programming	This is an agile method based on strong participation of the client in the development team and small iterations with constant deliveries.	• A very experienced development team that conjugates development and elicitation skills; • Client with availability of time; • Client with a participative approach; • Knowledge of domain is concentrated in few experts.
Reverse engineering	Study of the existing system as the basis for the development	• The system already exists and is well documented; • There is time to perform this survey; • This technique can support any other technique; • Difficulty in achieving requirements by traditional methods; • The current system is well-documented and, in general is considered satisfactory by the users.
Analogical Reuse of Requirements Frameworks	This technique is based on specification reuse. It suggests reusing fragments of requirements specification to reduce elaboration time and increase requirements quality.	• When there are similarities with other systems; • When the access to the client is difficult; • When there are many analysts without domain knowledge; • When the concepts of the system are complex.

Participant observation	Its goal is to win a close and intimate familiarity with a given group of individuals and their practices through a strong involvement with people in their natural environment, usually over a long period of time. The observer is inserted into the work environment for a period of time.	• System already exists; • System has a routine operation; • Process without many obscure points.
Ethnography	This technique aims to generate an understanding of an entire area through observation of the interactions of the members of these departments or sectors. With the observation of these interactions, it is possible to obtain an overview of the existing relationships as well as jargons and acronyms used.	• When the domain is very specific, and it has many specific characteristics; • Analysts do not have any knowledge of the domain; • Analysts have experience in the technique; • Can be the starting point of any system, subject to cost analysis.
Protocol Analysis	The user describes aloud every step of his activities while performing a specific task. Meanwhile, an analyst observes the task to understand it. Filming can be used if allowed.	• Experienced analyst; • There is a consensus about the application; • The specialists have high domain knowledge; • There is high customer interest; • There is an easy access to information; • Individual routine tasks.
ULRC (User Led Requirements Construction)	The team of requirements describes the current situation together with the client in order to teach elicitation techniques to the customer personnel. Once understood the technique the client team performs the elicitation of the new system.	• Experienced analyst; • Client with some technical knowledge; • There is time to train the customer; • There is customer interest in raising the requirements; • It should be used when the client management model is the authoritarian; • It should be used when there is a possible tension between requirements team and client staff.

Appendix D

Metrics Specification and Procedures

Metrics Specification and Procedures

Measure 1 - Number of requirements meetings needed before and after the framework.

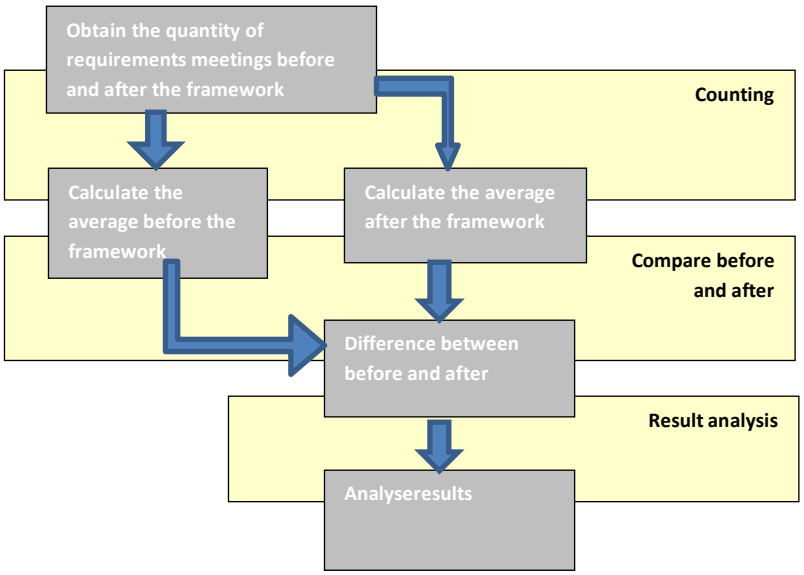
This measure will quantify the number of requirements meetings (any meeting with one or more client stakeholders to perform requirements elicitation using any technique) needed to elicit requirements in a group of projects in each use case before and after the usage of the framework proposed. To calculate this metric it is necessary that the projects have initiated the requirements elicitation process before the framework and continued to perform this process with the framework. Thus, it will be possible to compare the number of requirements meetings needed before and after the usage of the framework. The hypothesis is that if the need for requirements meetings is reduced it would mean that the requirements meetings are now more effective indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole.

Moment to Collect – this measure will be collected immediately before the usage of the framework and immediately after the ending of the requirements elicitation process.

Goal Definition	
Analyse	Framework
For the purpose of	evaluating
With respect to	Comparison between before and after its usage
From the viewpoint of	Project stakeholders
In the context of	Several projects in the Brazilian Navy

Abstraction Sheet for Measure 1	
Quality focus	Improve requirements elicitation
Current understanding	The efficacy of requirements elicitation is low
Variation factors	Difference between the average of requirements elicitation needed before and after the framework
Impact of variation factors	Higher the difference higher the improvement

Question definition for measure 1



SPECIFICATION MEASUREMENT	
General Information	
Identifier	**NBR_MEET**
Strategic Objectives	**1) Improve the elicitation process efficacy to improve client satisfaction**
Measurement Name	**Number of requirements meetings needed before and after the framework**
Measurement Objectives	Evaluate the efficacy of the requirements meetings before and after the framework
Type of Measurement	Derived Measure
Measurement Components	NBRBF = number before the framework NBRAF = number after the framework
Goal	(NBRBF – NBRAF) > 0
Calculation Formula	(NBRBF – NBRAF)

Unit of Measure	Unit
Information Collection	
Source (s) of data	Project Reports of the Requirements Meetings.
Procedure for collection	**How to collect:** Obtain the quantity of requirements meetings performed before and after the framework for each project. This measures will be compared to establish if the difference of quantities of meetings was reduced after the framework
Collection schedule	Twice to each project. Before the use of the framework and after the requirements elicitation phase.
Responsible	Requirements Analyst
Verification Procedure	The analyst should verify if the difference between the numbers was reduced after the framework
Information for Data Analysis	
Parameters	Goals
Procedure	There will be an analysis from the values and criteria for evaluation below: Range: <=0 Below Expected Action: framework considered ineffective. Range: >0 Expected Action: framework considered effective.
Frequency analysis	After all data collection
Responsible	Researcher

Measure 2 - Difference between quantity of requirements meetings needed in a certain project in two requirements with similar complexities, one elicited before and one elicited after the framework.

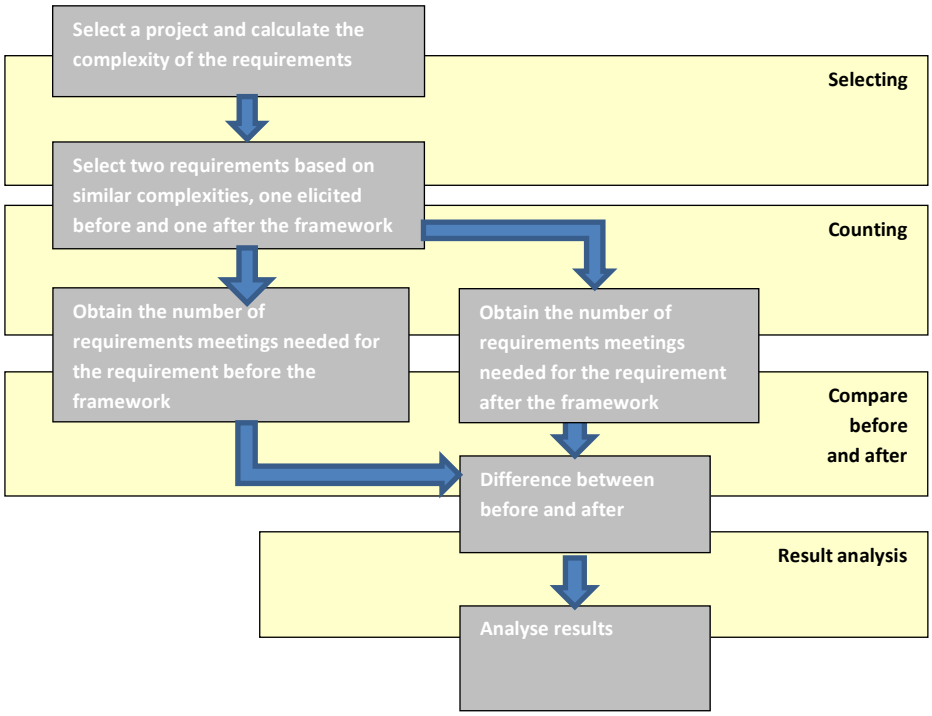
This measure will quantify the number of requirements meetings needed to elicit requirements in a certain project in a certain complexity of requirements before and after the usage of the framework proposed. The complexity will be calculated in Use Case Points (an approach to determine the complexity of a system before building it (UFPE, 2012)) that will be described in the Appendix E. The comparison will be between the number of requirements meetings needed to elicit a requirement of a certain complexity before the framework, and the number of requirements meetings needed to elicit a requirement of similar complexity after the framework. To calculate this metric it is necessary that the project has initiated the requirements elicitation process before the framework and is still performing this process. The hypothesis is that if the need for requirements meetings is reduced it would mean that the requirements meetings are now more efficient indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole. To improve the analysis of this metric, teams and clients will be the same, to maintain a fixed set of other factors involved, consequently reducing the factors that could be influencing the result. Thus, it is easier to determine the influence of the framework on the result of the metric.

Moment to Collect – this measure will be collected immediately after the elicitation of the requirement before the usage of the framework and immediately after the elicitation of the requirement after the framework.

Goal Definition	
Analyse	Framework
For the purpose of	evaluating
With respect to	Comparison between before and after its usage
From the viewpoint of	Project stakeholders
In the context of	A project in the Brazilian Navy

Abstraction Sheet for Measure 2	
Quality focus	Improve requirements elicitation
Current understanding	The efficacy of requirements elicitation is low
Variation factors	Difference between the quantity of requirements elicitation needed in a certain project in a certain complexity of requirements before and after the framework.
Impact of variation factors	Higher the difference higher the improvement

Question definition for measure 2



SPECIFICATION MEASUREMENT	
General Information	
Identifier	QTY_MEET
Strategic Objectives	1) Improve the elicitation process efficacy to improve client satisfaction
Measurement Name	Difference between quantity of requirements meetings needed in a certain project in two requirements with similar complexities one elicited before and one elicited after the framework.
Measurement Objectives	Evaluate the efficacy of the requirements meetings before and after the framework
Type of Measurement	Derived Measure

Measurement Components	QTYBF = quantity of requirements meetings before the framework QTYAF = quantity of requirements meetings after the framework
Goal	(QTYBF - QTYAF) > 0
Calculation Formula	(QTYBF - QTYAF) Where: QTYBF = quantity of requirements meetings before the framework QTYAF = quantity of requirements meetings after the framework
Unit of Measure	unit
Information Collection	
Source (s) of data	Project Reports of the Requirements Meetings.
Procedure for collection	**How to collect:** Select a project. Calculate requirements complexity. Obtain the quantity of requirements meetings performed to elicit a requirement before the framework and quantity of requirements meetings performed to elicit a requirement with similar complexity after the framework. This measures will be compared to establish if the quantities of meetings was reduced after the framework
Collection schedule	Twice to the selected project. Once for each selected requirement.
Responsible	Requirements Analyst
Verification Procedure	The analyst should verify if the quantities of meetings was reduced after the framework
Information for Data Analysis	
Parameters	Goals
Procedure	There will be an analysis from the values and criteria for evaluation below: Range: <=0 Below Expected Action: framework considered ineffective. Range: > 0 Expected Action: framework considered effective.
Frequency analysis	After all data collection
Responsible	Researcher

Measure 3 - Difference between time needed to elicit in a certain project with two requirements of similar complexities, one elicited before and one elicited after the framework.

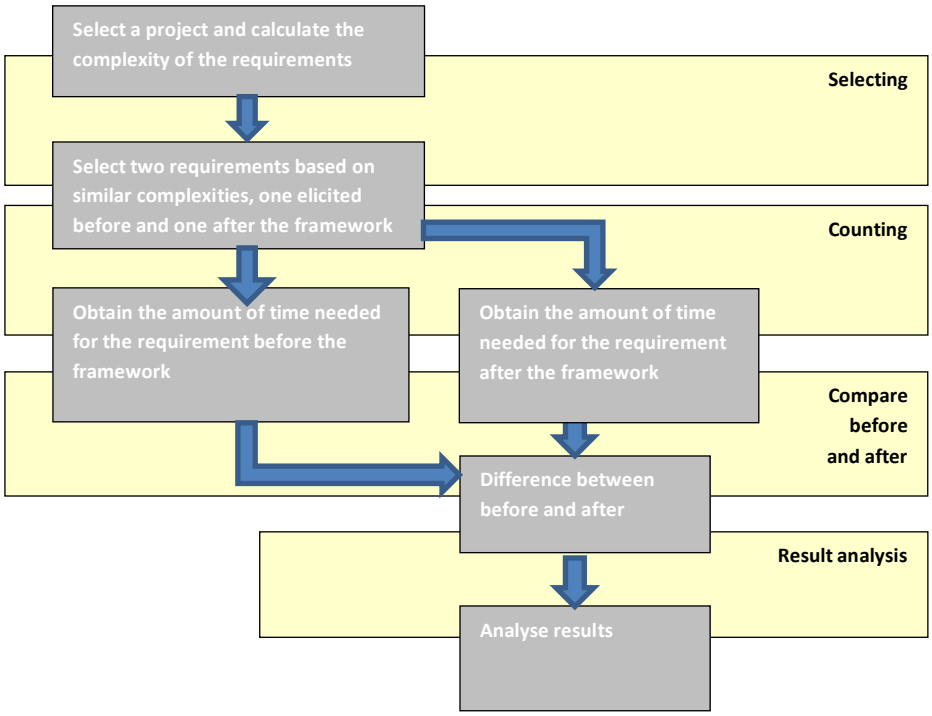
This measure will quantify the amount of time needed to elicit requirements in a certain project with a certain complexity of requirements before and after the usage of the framework proposed. The complexity will be calculated in Use Case Points. The comparison will be between the amount of time needed to elicit a requirement of a certain complexity before the framework, and the amount of time needed to elicit a requirement of a similar complexity after the framework. To calculate this metric it is necessary that the project has initiated the requirements elicitation process before the framework and is still performing this process. The hypothesis is that if the amount of time needed is reduced it would mean that the elicitation is now more efficient indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole. The results should support measure two because the number of meetings could be reduced but if the total time to elicit the requirements increases, the efficiency will not have been increased and just the format of the elicitation process would have been modified. Like measure two teams and clients will be the same, for the same reason of improving the analysis.

Moment to Collect – this measure will be collected immediately after the elicitation of the requirement before the usage of the framework and immediately after the elicitation of the requirement after the framework.

Goal Definition	
Analyse	Framework
For the purpose of	evaluating
With respect to	Comparison between before and after its usage
From the viewpoint of	Project stakeholders
In the context of	A project in the Brazilian Navy

Abstraction Sheet for Measure 3	
Quality focus	Improve requirements elicitation
Current understanding	The efficacy of requirements elicitation is low
Variation factors	Difference between the time needed in a certain project in a certain complexity of requirements before and after the framework
Impact of variation factors	Higher the difference higher the improvement

Question definition for measure 3



SPECIFICATION MEASUREMENT	
General Information	
Identifier	**AMT_TIME**
Strategic Objectives	**1) Improve the elicitation process efficacy to improve client satisfaction**
Measurement Name	**Difference between time needed with client in a certain project in two requirements with similar complexities one elicited before and one elicited after the framework.**
Measurement Objectives	Evaluate the efficiency of the requirements elicitation before and after the framework
Type of Measurement	Derived Measure
Measurement Components	TMBF = amount of time before the framework TMAF = amount of time after the framework
Goal	(TMBF - TMAF) > 0
Calculation Formula	((TMBF – TMAF) Where: TMBF = amount of time before the framework TMAF = amount of time after the framework
Unit of Measure	minutes
Information Collection	
Source (s) of data	Project Reports of the Requirements Meetings.
Procedure for collection	**How to collect:** Select a project. Calculate requirements complexity. Obtain the quantity of time needed to elicit a requirement before the framework and time needed to elicit a requirement with similar complexity after the framework. This measures will be compared to establish if the time needed was reduced after the framework
Collection schedule	Twice to the selected project. Once for each selected requirement.
Responsible	Requirements Analyst
Verification Procedure	The analyst should verify if the time needed to elicit a requirement was reduced after the framework
Information for Data Analysis	
Parameters	Goals

Procedure	There will be an analysis from the values and criteria for evaluation below: Range: <= 0 Below Expected Action: framework considered inefficient. Range: > 0 Expected Action: framework considered efficient.
Frequency analysis	After all data collection
Responsible	Researcher

Measure 4 - Number of requirements changes before and after the framework

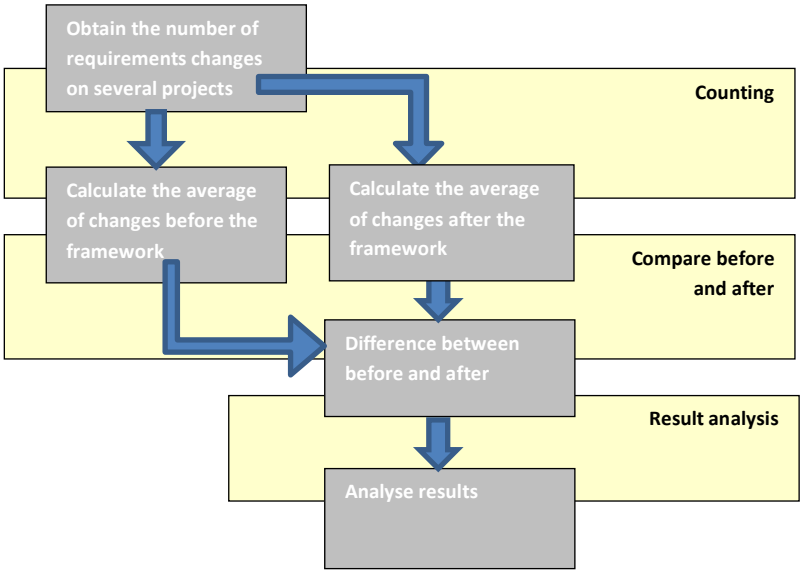
This measure will quantify the number of requirements changes across several projects before and after the usage of the framework proposed. To calculate this metric it is necessary that there are at least two projects finished after the framework is used. The hypothesis is that if the quantity of requirements changes is reduced it would mean that the requirements elicitation is now more effective, indicating that the framework influenced positively the efficacy of the requirements meetings and consequently, the requirements elicitation process as a whole.

Moment to Collect – this measure will be collected immediately before the usage of the framework and sometime after the framework is in use.

Goal Definition	
Analyse	Framework
For the purpose of	evaluating
With respect to	Comparison between before and after its usage
From the viewpoint of	Project stakeholders
In the context of	Several projects in the Brazilian Navy

Abstraction Sheet for Measure 4	
Quality focus	Improve requirements elicitation
Current understanding	The efficacy of requirements elicitation is low
Variation factors	Difference between the number of requirements changes before and after the framework
Impact of variation factors	Higher the difference higher the improvement

Question definition for measure 4



SPECIFICATION MEASUREMENT	
General Information	
Identifier	**NBR_CHG**
Strategic Objectives	**1) Improve the elicitation process efficacy to improve client satisfaction**
Measurement Name	**Difference between the quantity of requirements changes before and after the framework**
Measurement Objectives	Evaluate the efficacy of the requirements elicitation process before and after the framework
Type of Measurement	Derived Measure
Measurement Components	CHGBF = number of requirements changes before the framework CHGAF = number of requirements changes after the framework
Goal	(CHGBF – CHGAF) > 0
Calculation Formula	((CHGBF - CHGAF) Where: CHGBF = number of requirements changes before the framework CHGAF = number of requirements changes after the framework
Unit of Measure	unit
Information Collection	
Source (s) of data	Project Reports.
Procedure for collection	**How to collect:** Count the quantity of requirements changes before and after the framework. This measures will be compared to establish if the number of requirements changes was reduced after the framework
Collection schedule	Twice to the all projects.
Responsible	Requirements Analyst
Verification Procedure	The analyst should verify if the if the number of requirements changes was reduced after the framework
Information for Data Analysis	
Parameters	Goals

Procedure	There will be an analysis from the values and criteria for evaluation below: Range: <= 0 Below Expected Action: framework considered ineffective. Range: > 0 Expected Action: framework considered effective.
Frequency analysis	After all data collection
Responsible	Researcher

Appendix E

Complexity Calculation (Use Case Points)

Excel Sheet to Calculate Complexity

Effort estimation based on use case points

References

Resource Estimation for Objectory Projects de Gustav Karner
Applying Use Cases - A Pratical Guide de Geri Schneider e Jason P. Winters

1.0. Measuring Actors Complexity

Actor	Interface	Weight
Simple	Program Interface (API)	1
Medium	Protocol (Ex.:TCP/IP) or text interface	2
Complex	Graphical interface	3

	Quantity of Actors	Value
		0
		0
		0
Total	0	0

2.0. Measuring Use case Complexities

Use Case	Description	Weight
Simple	< 3 transactions or < 5 classes of analysis	5
Medium	4-7 transactions or 5 to 10 classes of analysis	10
Complex	> 7 transactions or > 10 classes of analysis	15

	Qty. Use Cases	Value
		0
		0
		0
Total	0	0

NAUCP	0

NAUCP = Non Adjusted Use Case Points

3.0. Considering Project Technical Factors

Factor	Description	Weight		Atributed	Value
T1	Distributed System	2			0
T2	Performance Goals	1			0
T3	on-line eficiency	1	*		0
T4	Processing complexity	1			0
T5	Reusable code in other applications	1			0
T6	Ease of installation	0.5			0
T7	Ease of use	0.5			0
T8	Portability	2			0
T9	Changeability	1			0
T10	Concurrency	1			0
T11	Security	1			0
T12	Direct access to third parties	1			0
T13	Need for special facilities for training users	1			0
				FatorT	0

TCF	0.6

TCF = Technical Complexity Factor

4.0. Considering Environmental Factors

Factor	Description	Weight	Atributed	Value
F1	Team's familiarity with RUP	1.5		0
F2	Team experience	0.5		0
F3	Team experience in Object Oriented Development	1		0
F4	Team analysts ' ability	0.5		0
F5	Motivation	1		0
F6	Stability of the requirements	2		0
F7	Interns or part-time employees	-1		0
F8	Technology knowledge and environment setup	-1.5		0
			EFactor	0

EF	1.4

EF = Environmental Factor

5.0. Use Case Points

PCU	PCUNA *FCT*FA	0.00

Calculation of Use Case Points of one Excel sheet to be implemented on System 4, elicited before the framework

Number of meetings needed to perform elicitation – 10

Number of use Case points – 31,1

Effort estimation based on use case points

References http://www.cin.ufpe.br/~if717/slides/pontos-de-caso-de-uso.xls
(accessed in 15/02/2012)

1.0. Measuring Actors Complexity

Actor	Interface	Weight
Simple	Program Interface (API)	1
Medium	Protocol (Ex.:TCP/IP) or text interface	2
Complex	Graphical interface	3

	Quantity of Actors	Value
		0
		0
	1	3
Total	1	3

2.0. Measuring Use case Complexities

Use Case	Description	Weight
Simple	< 3 transactions or < 5 classes of analysis	5
Medium	4-7 transactions or 5 to 10 classes of analysis	10
Complex	> 7 transactions or > 10 classes of analysis	15

	Qty. Use Cases	Value
	3	15
		0
	1	15
Total	4	30

NAUCP	33

NAUCP = Non Adjusted Use Case Points

3.0. Considering Project Technical Factors

Factor	Description	Weight		Atributed	Value
T1	Distributed System	2		0	0
T2	Performance Goals	1		5	5
T3	on-line eficiency	1	*	2	2
T4	Processing complexity	1		3	3
T5	Reusable code in other applications	1		5	5
T6	Ease of installation	0.5		1	0.5
T7	Ease of use	0.5		5	2.5
T8	Portability	2		4	8
T9	Changeability	1		2	2
T10	Concurrency	1		1	1
T11	Security	1		5	5
T12	Direct access to third parties	1		1	1
T13	Need for special facilities for training users	1		1	1
				FatorT	36

TCF	0.96

TCF = Technical Complexity Factor

4.0. Considering Environmental Factors

Factor	Description	Weight	Atributed	Value
F1	Team's familiarity with RUP	1.5	1	1.5
F2	Team experience	0.5	3	1.5
F3	Team experience in Object Oriented Development	1	3	3
F4	Team analysts ' ability	0.5	1	0.5
F5	Motivation	1	2	2
F6	Stability of the requirements	2	5	10
F7	Interns or part-time employees	-1	0	0
F8	Technology knowledge and environment setup	-1.5	3	-4.5
			EFactor	14

EF	0.98

EF = Environmental Factor

5.0. Use Case Points

PCU	PCUNA *FCT*FA	31.05

Calculation of Use Case Points of one Excel sheet to be implemented on System 4, elicited after the framework

Number of meetings needed to perform elicitation – 4

Number of use Case points – 35,8

Effort estimation based on use case points

References http://www.cin.ufpe.br/~if717/slides/pontos-de-caso-de-uso.xls
(accessed in 15/02/2012)

1.0. Measuring Actors Complexity

Actor	Interface	Weight		Quantity of Actors	Value
Simple	Program Interface (API)	1			0
Medium	Protocol (Ex.:TCP/IP) or text interface	2			0
Complex	Graphical interface	3		1	3
			Total	1	3

2.0. Measuring Use case Complexities

Use Case	Description	Weight		Qty. Use Cases	Value
Simple	< 3 transactions or < 5 classes of analysis	5		4	20
Medium	4-7 transactions or 5 to 10 classes of analysis	10			0
Complex	> 7 transactions or > 10 classes of analysis	15		1	15
			Total	5	35

NAUCP	38

NAUCP = Non Adjusted Use Case Points

3.0. Considering Project Technical Factors

Factor	Description	Weight		Atributed	Value
T1	Distributed System	2		0	0
T2	Performance Goals	1		5	5
T3	on-line eficiency	1	*	2	2
T4	Processing complexity	1		3	3
T5	Reusable code in other applications	1		5	5
T6	Ease of installation	0.5		1	0.5
T7	Ease of use	0.5		5	2.5
T8	Portability	2		4	8
T9	Changeability	1		2	2
T10	Concurrency	1		1	1
T11	Security	1		5	5
T12	Direct access to third parties	1		1	1
T13	Need for special facilities for training users	1		1	1
			FatorT		36

TCF	0.96

TCF = Technical Complexity Factor

4.0. Considering Environmental Factors

Factor	Description	Weight		Atributed	Value
F1	Team's familiarity with RUP	1.5		1	1.5
F2	Team experience	0.5		3	1.5
F3	Team experience in Object Oriented Development	1		3	3
F4	Team analysts ' ability	0.5		1	0.5
F5	Motivation	1		2	2
F6	Stability of the requirements	2		5	10
F7	Interns or part-time employees	-1		0	0
F8	Technology knowledge and environment setup	-1.5		3	-4.5
			EFactor		14

EF	0.98

EF = Environmental Factor

5.0. Use Case Points

PCU	PCUNA *FCT*FA	35.75

Calculation of Use Case Points of one Excel sheet to be implemented on System 5, elicited before the framework

Number of meetings needed to perform elicitation – 13

Number of use Case points – 31,1

Effort estimation based on use case points

References http://www.cin.ufpe.br/~if717/slides/pontos-de-caso-de-uso.xls
(accessed in 15/02/2012)

1.0. Measuring Actors Complexity

Actor	Interface	Weight		Quantity of Actors	Value
Simple	Program Interface (API)	1			0
Medium	Protocol (Ex.:TCP/IP) or text interface	2			0
Complex	Graphical interface	3		3	9
			Total	3	9

2.0. Measuring Use case Complexities

Use Case	Description	Weight		Qty. Use Cases	Value
Simple	< 3 transactions or < 5 classes of analysis	5		5	25
Medium	4-7 transactions or 5 to 10 classes of analysis	10		2	20
Complex	> 7 transactions or > 10 classes of analysis	15		3	45
			Total	10	90

NAUCP	99

NAUCP = Non Adjusted Use Case Points

3.0. Considering Project Technical Factors

Factor	Description	Weight		Atributed	Value
T1	Distributed System	2		1	2
T2	Performance Goals	1		3	3
T3	on-line eficiency	1	*	2	2
T4	Processing complexity	1		2	2
T5	Reusable code in other applications	1		1	1
T6	Ease of installation	0.5		2	1
T7	Ease of use	0.5		3	1.5
T8	Portability	2		1	2
T9	Changeability	1		2	2
T10	Concurrency	1		2	2
T11	Security	1		5	5
T12	Direct access to third parties	1		0	0
T13	Need for special facilities for training users	1		2	2
			FatorT		25.5

TCF	0.855

TCF = Technical Complexity Factor

4.0. Considering Environmental Factors

Factor	Description	Weight		Atributed	Value
F1	Team's familiarity with RUP	1.5		3	4.5
F2	Team experience	0.5		4	2
F3	Team experience in Object Oriented Development	1		4	4
F4	Team analysts ' ability	0.5		4	2
F5	Motivation	1		5	5
F6	Stability of the requirements	2		4	8
F7	Interns or part-time employees	-1		3	-3
F8	Technology knowledge and environment setup	-1.5		2	-3
			EFactor		19.5

EF	0.815

EF = Environmental Factor

5.0. Use Case Points

PCU	PCUNA *FCT*FA	68.99

Calculation of Use Case Points of one Excel sheet to be implemented on System 5, elicited after the framework

Number of meetings needed to perform elicitation – 6

Number of use Case points – 31,1

Effort estimation based on use case points

References http://www.cin.ufpe.br/~if717/slides/pontos-de-caso-de-uso.xls
(accessed in 15/02/2012)

1.0. Measuring Actors Complexity

Actor	Interface	Weight		Quantity of Actors	Value
Simple	Program Interface (API)	1			0
Medium	Protocol (Ex.:TCP/IP) or text interface	2			0
Complex	Graphical interface	3		3	9
			Total	3	9

2.0. Measuring Use case Complexities

Use Case	Description	Weight		Qty. Use Cases	Value
Simple	< 3 transactions or < 5 classes of analysis	5		5	25
Medium	4-7 transactions or 5 to 10 classes of analysis	10		2	20
Complex	> 7 transactions or > 10 classes of analysis	15		3	45
			Total	10	90

NAUCP	99

NAUCP = Non Adjusted Use Case Points

3.0. Considering Project Technical Factors

Factor	Description	Weight		Atributed	Value
T1	Distributed System	2		1	2
T2	Performance Goals	1		3	3
T3	on-line eficiency	1	*	2	2
T4	Processing complexity	1		2	2
T5	Reusable code in other applications	1		1	1
T6	Ease of installation	0.5		2	1
T7	Ease of use	0.5		3	1.5
T8	Portability	2		1	2
T9	Changeability	1		2	2
T10	Concurrency	1		2	2
T11	Security	1		5	5
T12	Direct access to third parties	1		0	0
T13	Need for special facilities for training users	1		2	2
			FatorT		25.5

TCF	0.855

TCF = Technical Complexity Factor

4.0. Considering Environmental Factors

Factor	Description	Weight		Atributed	Value
F1	Team's familiarity with RUP	1.5		3	4.5
F2	Team experience	0.5		4	2
F3	Team experience in Object Oriented Development	1		4	4
F4	Team analysts ' ability	0.5		4	2
F5	Motivation	1		5	5
F6	Stability of the requirements	2		4	8
F7	Interns or part-time employees	-1		3	-3
F8	Technology knowledge and environment setup	-1.5		2	-3
			EFactor		19.5

EF	0.815

EF = Environmental Factor

5.0. Use Case Points

PCU	PCUNA *FCT*FA	68.99

Appendix F

Informed Consent Statement

Embedding e-Science Applications – Designing and Managing for Usability

INFORMED CONSENT STATEMENT

(Declaração de Consentimento Informado)

You are invited to participate in a research study. The purpose of this study is to understand system development in the Brazilian Navy.

(Voce esta convidado a participar em um projeto de pesquisa. O proposito deste estudo e o de entender o desenvolvimento de sistemas na Marinha do Brasil.)

INFORMATION (Informações)

This study involves the following procedures (Este estudo envolve os seguintes procedimentos):

1. Personal interviews, either in person or by skype. (Entrevistas realizadas pessoalmente ou atraves do Skype)

2. The personal interview will take approximately 30-60 minutes per participant on average. Most participants will have one interview session. Some participants may be asked to take part in a follow-up interview to clarify information learned in the course of this research. (As entrevistas durarao aproximadamente de 30 a 60 minutos por participante em media. A maioria dos participantes participara de soomente uma reuniao. Alguns participantes podem ser convidados a participar de outras entrevistas para clarear alguma informacao obtida no curso desta pesquisa)

Interviews will be audio recorded and transcribed; photos and video recordings would also be made in the course of this research. (As entrevistas serao gravadas e transcritas; fotografias e gravacao de video tambem poderao ocorrer no curso desta pesquisa)

RISKS (Riscos)

No risks are foreseen once no information will be related to the person that provided it. Information damaging to your personal or professional reputation will not be reported.

(Nenhum risco esta previsto uma vez que nenhuma informacao estara associada as pessoas que as proveram. Nenhuma informacao que afete as reputacoes pessoais ou profissionais serao publicadas.)

BENEFITS (Beneficios)

The benefits of this research are that it will contribute to a scholarly understanding of the ways in which new approaches for requirements engineering may be advanced. Other benefits are that this research may enable those who are interested to become familiar with novel requirements engineering approaches.

(Os beneficios desta pesquisa são o de contribuir para o entendimento academico das formas como novas abordagens para engenharia de requisitos possam ser antecipadas. Outros beneficios são o desta pesquisa possibilitar os interessados em se familiarizar com novas abordagens de engenharia de requisitos.)

CONFIDENTIALITY (Confidencialidade)

We would like to reassure you that the data collected during this study will be treated according to current ethical guidelines for capturing audio and video data. Thus, the data captured shall not be passed onto any third party. This information will be used primarily for research purposes, and only researchers working on this project will have access to the original files storing this information. Any sensitive material can be destroyed at your request, and in addition, the recording device can be switched off at any point during the interview. We will send you the results of our analysis for acceptance. References to the data in publications will be anonymised.

(Gostaríamos de assegurar-lhe que os dados capturados durante este estudo serão tratados de acordo com diretrizes éticas atuais para capturar dados de áudio e vídeo. Assim, os dados capturados não serão passados para qualquer terceiro. Esta informação será usada principalmente para fins de investigação, e apenas pesquisadores trabalhando neste projeto terão acesso aos arquivos originais armazenando essas informações. Qualquer material sensível pode ser destruído a seu pedido, e além disso, o dispositivo de gravação pode ser desligado a qualquer momento, durante a entrevista. Nós lhe enviaremos os resultados de nossa análise para apreciação. Referências aos dados em publicações vão ser anonimizadas)

CONTACT (Contato)

If you have questions at any time about the study or the procedures, you may contact the researcher: Fernando Muradas, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK, +44 (0) 07861542751, or by e-mail at Fernando.muradas@comlab.ox.ac.uk. The principal investigator of the study, Marina Jirotka, can also be contacted by e-mail marina.jirotka@comlab.ox.ac.uk or telephone (+44 (0) 1865 610613)

(Se você tiver perguntas a qualquer momento sobre o estudo ou os procedimentos, pode contactar o investigador: Fernando Muradas, Departamento de Computacao da Universidade de Oxford, edifício Wolfson, Parks Road, Oxford, OX1 3QD, UK, + 44 (0) 07861542751, ou por e-mail Fernando.muradas@cs.ox.ac.uk. A investigadora principal do estudo, Dr. Marina Jirotka, também pode ser contatada pelo email marina.jirotka@oerc.ox.ac.uk ou telefone (+ 44 (0) 1865 610613)).

PARTICIPATION (Participação)

Your participation in this study is voluntary; you may refuse to participate. If you decide to participate, you may withdraw from the study at any time. If you withdraw from the study before data collection is completed your data will be returned to you or destroyed.

(Sua participação neste estudo é voluntária; Você pode recusar-se a participar. Se você decidir participar, você pode se retirar do estudo a qualquer momento. Caso você se retire do estudo antes de coleta de dados ser concluída seus dados serão destruídos.)

AUDIO, VIDEO AND IMAGES (Audio, Video e imagens)

Audio and/or video recordings and photographs may be collected during your participation in this research. This information will be used for research purposes, and only researchers working on this project will have access to the original files storing this information. If you withdraw from this study, the files containing your data will be destroyed. Edited versions of audio, video and photographic information from this research may be used in instruction, public talks, and publications of this research if you consent to your data being used in this way below. The images and audio will not be used for any additional purposes without your additional permission.

(Fotografias e gravações de áudio e/ou vídeo podem ser recolhidas durante a sua participação nesta pesquisa. Estas informações serão utilizadas para fins de investigação, e apenas

pesquisadores trabalhando neste projeto terão acesso aos arquivos originais armazenando essas informações. Se você se retirar do presente estudo, os arquivos contendo os dados serão destruídos.Versões editadas de áudio, vídeo e informações fotográficas de pesquisa podem ser utilizadas na instrução, palestras públicas e publicações desta pesquisa se você concordar com os dados sendo usados desta forma abaixo.As imagens e o áudio não servirão para quaisquer finalidades adicionais sem a sua permissão adicional.)

CONSENT (Consentimento)

I have read this form and received a copy of it. I have had all my questions answered to my satisfaction. I agree to take part in this study.

(Eu li este formulario e recebi uma copia do mesmo. Eu tive todas minhas questoes respondidas satisfatoriamente. Eu concordo em tomar parte deste estudo).

Printed name (Nome em letra de forma)_____ _____
e-mail: _____

Signature (Assinatura): _____ Date (Data): _____

Initial below (Iniciais abaixo)

_____ I agree to allow still images of me collected during this research to be used in presentations and publication of this research. (Eu concordo em permitir imagens estaticas de minha participação nesta pesquisa para serem utilizadas em apresentacoes e publicacoes desta pesquisa).

Consent form date: __ January 2010

(Data do Formulario de Consentimento)