UNIVERSITI TEKNIKAL MALAYSIA MELAKA

PROJECT COMPLETION REPORT
FOR
(SHORT TERM) RESEARCH GRANT

A FRAMEWORK FOR EMPIRICAL INVESTIGATION TO ATTAIN COST EFFECTIVE SOFTWARE REQUIREMENTS THROUGH NEGOTIATION

Principal Researcher : DR. SABRINA BINTI AHMAD

Co-Researchers : 1. DR. AZAH KAMILAH BINTI MUDA (FTMK)
2. NOOR AZILAH BINTI MUDA (FTMK)
3. ZAHRIAH BINTI OTHMAN (FTMK)

Project Code No. : PJP/2011/FTMK12(C)/S00894

Report Submission Date : 30 OKTOBER 2012

Department of Software Engineering

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

PJP/2011/FTMK12(C)/S00894

A FRAMEWORK FOR EMPIRICAL INVESTIGATION TO ATTAIN COST EFFECTIVE SOFTWARE REQUIREMENTS THROUGH NEGOTIATION

PENYELIDIK:
DR. SABRINA BINTI AHMAD
DR. AZAH KAMILAH BINTI MUDA @ DRAMAN
NOOR AZILAH BITNTI MUDA @ DRAMAN
ZAHRIAH BINTI OTHMAN

PEMBANTU PENYELIDIK:
AMIRAH FATEHANA BINTI AHMAD
MUHAMAD SUFFIAN BIN SIKANDAR GANI

Fakulti Teknologi Maklumat dan Komunikasi
Universiti Teknikal Malaysia Melaka
2012

This report is submitted to Universiti Teknikal Malaysia Melaka
for Final Research report

ACKNOWLEDGEMENT

Praise to Allah s.w.t for giving the researchers strength, patience and motivation to complete this research work.

We wish to gratefully acknowledge the Faculty of Information and Communication Technology (FTMK), Centre for Research and Innovation Management (CRIM), and Universiti Teknikal Malaysia Melaka (UTeM) for providing the grant for this research and subsequent support throughout the duration of the project. Without them, this project would not have been possible.

Dr. Sabrina Binti Ahmad
Dr. Azah Kamilah Binti Muda @ Draman
Noor Azilah Binti Muda @ Draman
Zahriah Binti Othman

ABSTRACT

Software requirements are not simply collected or obtained. The process is called 'elicit' and eliciting software requirements is challenging due to the existence of multiple resources with multiple perspectives. Requirements elicitation acquires appropriate techniques and skills in order to get the right requirements. In general, the set of requirements must describe the stakeholders' needs to ease their business processes and must be feasible in order to be realized within time, cost and technology constraints. The requirements statement is the basis for every project, defining what the stakeholders need and also what the system must do in order to satisfy that need. Further, defects in requirements are the most numerous in the software lifecycle and also the most expensive and time-consuming to correct. Therefore, it is crucial to minimize defects in requirements to save later effort to correct them. One promising approach to attain cost effective software requirements is to introduce formal negotiation. This research focuses on providing a framework for empirical study to estimate the benefit of negotiation. Benefits come in savings of rework when a defect has to be detected and removed at a later stage of development or operation. The benefit from savings depends on the severity of the defect and the impact it would have had on the development project; this may vary with the development phase in which it would have surfaced. The framework attempts to provide a guideline to empirically assess the effectiveness of requirements elicitation practice in order to reduce the number of defects and therefore estimates the savings resulting from negotiation. The reduction of defects here is presented in the terms of economic benefit obtained through negotiation.

(Keywords: software requirements negotiation, cost-benefit analysis, empirical investigation)

Key Researchers:

Dr. Sabrina Ahmad
Email: sabrinaahmad@utem.edu.my
Tel no: 06-3316637

Dr. Azah Kamilah Muda @ Draman
Email : azah@utem.edu.my
Tel no : 06-331 6597

Noor Azilah Muda @ Draman
Email: azilah@utem.edu.my
Tel no: 06-3316638

Zahriah Othman
Email: zahriah@utem.edu.my
Tel no: 06-3316692

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 Background

In 1996, it was stated (Jones, 1996) that deficient requirements are the single biggest cause of software project failure. Several hundred organizations were studied and discovered that requirements engineering (RE) is deficient in more than 75 percent of all enterprise. In other words, getting requirements right might be the single most important and difficult part of a software project. Consequently, much research has been done to improve the RE process to increase the chance of the project success. In 2001, an empirical study (Hofmann and Lehner, 2001) identified that RE practices clearly contribute to project success, particularly in terms of knowledge, resource allocation and process. In 2005, an industrial survey (Verner et al., 2005) was carried out in Australia and the U.S. to depict the relationship between RE and software project success. The survey showed that the most important project success prediction factors are that the requirements were good and that the requirements were managed effectively. Good requirements were defined as complete and consistent from the perspective of the respondent. These two factors alone correctly predicted 93% of successful projects. In relation with that, having good requirements is highly correlated with a high level stakeholders' involvement. Stakeholders are involved during elicitation process; which identifies the needs and bridging the difference among multiple stakeholders for the purpose of defining and distilling requirements to meet the needs of an organization or project while staying within imposed constraints (Berenbach et al., 2009).

1.2 Statement of Problem

The context in which requirements are elicited is usually a human activity, and the problem owners are people. It is seldom technical problems which inhibit productivity and quality (Zowghi and Coulin, 2005, Juristo et al., 2002). Instead the vast majority of requirements problems are related to human interactions, process and communications. One of the main problems during requirements elicitation is communication and understanding among the stakeholders. This involves conflicts, scope boundary and erroneous interpretation. The argument is supported (Zowghi and Coulin, 2005) and believed that requirements elicitation is inherently imprecise as a result of multiple variable factors, a vast array of options and decisions, and communication.

One promising area worthy of investigation is negotiation between the various system stakeholders. The advantages of implementing negotiation are the requirements represents all the stakeholders' perspective and perception, underlies a sound basis for resource estimation, improves system quality and minimizes the resources involved (Boehm and Egyed, 1998, Damian, 2000, Grunbacher and Syeff, 2005, Juristo et al., 2002, Kotonya and Sommerville, 1997, Pressman, 2005, Price and Cybulski, 2006, Sommerville, 2004, Zowghi and Coulin, 2005, Dorfman, 1997). Negotiation is seen as a prevention action to avoid or at least minimize the amount of defects that would otherwise be established in the requirements at a very early stage of RE process. This leads to the economic benefit of negotiation, which is the reduction in future effort of development and to the higher quality inputs on which development and project planning are based.

The benefit of implementing negotiation in which shall improve the quality of software requirements are mere speculation if it is not possible to provide empirical evidence.

The purpose of this research is to set up a framework to estimate the benefit of negotiation. Benefits come in savings of rework when a defect has to be detected and removed at a later stage of development or operation (Halling et al., 2003). The benefit from savings depends on the severity of the defect and the impact it would have had on the development project; this may vary with the development phase in which it would have surfaced (Biffl et al., 2001a, Biffl and Halling, 2003a). The framework attempts to provide a guideline to empirically assess the effectiveness of requirements elicitation practice in order to reduce the number of defects and therefore estimates the savings resulting from negotiation. The reduction of defects here is presented in the terms of economic benefit obtained through negotiation. Thus, the purpose of this research is to identify empirical processes and develop a framework for exercising empirical study which allows practitioners to assess negotiation benefit.

1.3 Objective

1. To investigate current requirements elicitation practice in industry
2. To propose and develop a model of requirements elicitation with negotiation
3. To model a framework for empirical study to assess software requirements cost benefit

2.0 METHODOLOGY

2.1 Cost-Benefit Analysis

Cost-benefit analysis is used in this research to estimate the benefits of deploying negotiation in the requirements elicitation process. The idea of cost-benefit analysis is to make different dimensions of a problem comparable to each other by pushing everything into an economic framework. Once everything is represented in economic terms, one can then calculate net gains and base decisions on these economic values. A cost benefit analysis finds, quantifies, and adds all the positive factors. These are the benefits. Then it identifies, quantifies, and subtracts all the negatives, the costs. The difference between the two indicates whether the planned action is advisable.

Cost-benefit analysis is useful because it allows business decisions to be analysed in advance (Boardman et al., 2006). While benefits are often harder to quantify, this effort provides analysis of the cost effectiveness of different alternatives in order to see whether the benefits outweigh the costs (Robinson, 1993). The aim is to gauge the efficiency of the interventions relative to each other and the status quo. The analysis is being implemented as it evaluates all of the potential costs and revenues that may be generated if the decision (in this case; using a negotiation process) is applied. The outcome of the analysis will determine whether the decision made is financially beneficial (Layard and Glaister, 1994).

When deploying IT for competitive advantage-type systems there are a need to create systems that cause shorter-term cost disadvantages to provide longer term gains which have acceptable risk (Boardman et al., 2006). Hence, cost-benefit can be a reductionist approach (Prest and Turvey, 1965) to project management in terms of determining whether to implement a certain requirement as it may fail to take into account longer term goals and visions of the organization.

In line with the cost-benefit analysis used in both Biffl's and Halling's research; the benefit of negotiation effort is the saved future effort for development which is a result of the higher quality of inputs for development and project planning. In this thesis, the benefit of estimated savings of rework comes from the defects not being introduced into requirements; the resulting cost of defects which are allowed to slip into the development process would be greater. Project managers, for example, can use the results for guidance in future development. The negotiation activities are an investment that saves money by preventing defects that would cause rework (Biffl et al., 2001b). In relation to that, cost-benefit analysis helps to determine in what context negotiation is likely to be worthwhile. Such an analysis balances the invested effort with likely saved staff hours from early defects reduction.

Even though negotiation reduces the occurrence of defects, the effectiveness of a negotiation in this thesis is defined as the ratio of defects found to the total number of defects. In order to allow the measurement of effectiveness, defects are seeded into the candidate list of requirements. Then, negotiation takes place to achieve an agreement. During the process, the requirements list is refined and would be expected to exclude the requirements containing defects in the agreement. This effort shows that defects are detected and resolved during negotiation process. The difference in inspection is that it only detects the defects from a completed requirements document whereas negotiation is able to identify potential defects during the creation of requirements, negotiate to resolve the defects and exclude them from the agreement of requirements.

Here, defect severity is considered based on the likely impact of a defect on further development (Halling et al., 2003):

- Low-severity defects (L) do not considerably increase development effort
- Major defects (M) potentially incur a considerable amount of rework and may increase project risk

- Critical defects (C) will most likely cause considerable rework and/or put the overall project success at risk.

In practice, the amount of rework to fix a defect often depends on the project stage in which a defect is found and removed. For example, an incorrect requirement may be easy to fix during requirements definition. However, the same defect may become a major problem during implementation since the foundation of architecture and design is based an incorrect requirement. Subsequently, much effort would have been needed to fix the defects. Therefore, each defect is distinguished by three cases based on risk expectations for development: the best case (B), a nominal case (N) and a worst case (W), with more or less increasing defect severity depending on the nature of the defect. In practice, the quality manager can track defects in a set of comparable projects to fine-tune the rating of the likely impact of a defect (Basili, 1993).

This benefit of savings depends on the severity of the defect and the impact that it would have had on the development project; this impact may vary with the development phase in which the defect would have surfaced. As indicated above, defects may slip into later development stages and thereby increase the risk/cost to the project.

The expected benefit per defect avoided in this thesis is based on the assumption of much other similar research. The assumption is based on the similar work done to estimate the benefit of detecting defects at an early stage of project development (Gilb and Graham, 1993, Halling et al., 2003, Biffl et al., 2001b, Biffl and Gutjahr, 2001, Biffl and Halling, 2003b). The benefit of finding a given defect depends on the difficulties this defect would have caused in a hypothetical project context. The greater the effort needed to fix the defects, the more benefit the project will have gained by reducing the defect through negotiation. This is not a fixed value but can be modelled with a probability distribution of expected savings. However, the probability distribution has to be validated. There are several approaches to determine the benefit for detecting and fixing a defect of a given severity class. Note that there are different severity classes introduced by different researchers. Gilb et al. (Gilb and Graham, 1993) and Biffl et al. (Biffl and Gutjahr, 2001) introduced two severity classes; major and minor. On the other hand, Biffl et al. (Biffl et al., 2001b) distinguished four severity classes which were numbered 0 to 3 to represent the magnitude of impact they have on development. Halling (Halling et al., 2003) used three severity class; low, major or critical and this is adopted in this thesis.

There are three probability approaches:

I. The simplest approach is to assign a single benefit value to each defect class. Gilb et al. (Gilb and Graham, 1993) and Biffl et al. (Biffl and Gutjahr, 2001) used this approach in which they assumed only two severity classes. They assumed an average savings of 8 hours for a major defect and savings of 1 hour for a minor defect.

II. Another approach would be to assume a probability distribution of benefits for each class. This can be for example, the most likely value of a triangle distribution based on the best, most likely and worst cases. The expected benefit for a given defect is determined from this benefit distribution (Biffl et al., 2001b). All defects in the document together would add 60% to the estimated project effort in the worst case, 5% in the most optimistic case, and 20% in the most likely case. Note that Biffl used four severity classes. For level 0 (trivial) defects it does not really matter when they are found. Level 1 (minor) defects that represent a local problem yield up to 1.5 hours, if found during inspection. Level 2 (major) defects may be hard to find during development and require multiple changes, thus they are value at 2 to 20 staff hours with an expected benefit of 8 hours. Level 3 (critical) defects may require reimplementation of the system, so they value generally about an order of magnitude higher than level 2 defects, which is rather cautious from an economic point of view. Table 2.1 shows general assumptions for benefits used by Biffl (Biffl et al., 2001b).

	Benefit for severity level x			
Severity level	0	1	2	3
Minimum	0.01	0.5	2	20
Most likely	0.1	1	8	80
Maximum	0.2	1.5	20	150

Table 2.1: General assumptions for benefits (Biffl et al., 2001b)

III. A more sophisticated approach includes benefit estimates for each development phase in which the defect could be detected. In an early phase the impact of a defect is rather low (e.g., in-house design), while in a later phase the impact is much higher, since a defect potentially spreads in related software products and various people are involved in removing the defect (e.g., operation at the customer site) (Halling et al., 2003). Table 2.2 shows benefit distribution used by Halling (Halling et al., 2003).

	Low	Moderate	Critical
Best case	0.5	1	2
Normal	1	4	16
Worst	2	16	128

Table 2.2: Benefit distribution (Halling et al., 2003)

Note that Tables 2.1 and 2.2 are for illustration to present other researchers' benefit distribution and NOT used in the calculation of this thesis. The cost associated with defects of different cases used in this thesis can be expressed in a 3 by 3 table (see Table 2.3). This is however motivated by benefit distribution in literature (Halling et al., 2003).

	Low	Moderate	Critical
Best case	1	2	4
Normal	2	8	32
Worst	4	32	256

Table 2.3: Expected benefit in hours per defect avoided during negotiation

In this work, three severity levels of defects and three phases are distinguished depending on the additional effort to fix a defect in a given class, if it is not prevented during negotiation. This is not a fixed value but can be modelled with a probability distribution of expected savings. As for the assumption of negotiation benefits, conservative (low) benefit values are used to stay on the conservative side in the economic evaluation. The total negotiation gain is then calculated as the difference between total negotiation benefit (i.e. summing up negotiation benefits for all detected defects) and negotiation cost (i.e. total effort invested in negotiation).

Therefore, the benefit distribution applied in this thesis is based on the understanding of three different defined severity classes of defects; even though other research made different benefit assumptions the arguments are still identical and allow expressing the magnitude of impact the defects have on development. For example, the more severe the defect impact is on the project then the higher is the risk that the development team may endure; more benefit is gained by omitting such defects.

2.2 Evaluation Criteria for Negotiation Performance

Negotiation Benefit

The economic ***benefit*** of negotiation is the future effort saved for development due to better quality input for development and project planning. From the set of defects found and from assumptions on the benefit of finding a defect during negotiation, the benefit of the negotiation can be determined. There are three approaches to determine the benefit, as was demonstrated in Section 2.1.

All three approaches were used and published in different papers (Biffl and Gutjahr, 2001, Biffl et al., 2001b, Halling et al., 2003). Here, the negotiation benefit is adapted from approach (**III**) and defined as the number of defects avoided multiplied by the benefit based on Table 2.1. The benefit is always in a low severity level and in a best case scenario to stay on the conservative side to assume low benefit for all the defects avoided (Halling et al., 2003).

$$NegotiationBenefit = defects \times benefitPerDefect$$

Negotiation Cost

The time invested by a nominal negotiating team (in staff hours) is used as ***direct negotiation costs.*** In a real project context further indirect costs would accrue such as negotiation planning and the delay of the project. However, indirect cost is not included in this research.

$$NegotiationCost = staffHours$$

Negotiation Effectiveness and Efficiency

The ***effectiveness*** of a negotiation process is defined as the ratio of defects found to the total number of defects present at the start of negotiation. The 'defects found' here means the defects excluded from the agreement at the end of a negotiation. Negotiation effectiveness is also an indicator of the product's quality, defined by the number of agreed requirements with a decrease or zero number of defects(Halling et al., 2003, Biffl and Halling, 2003b).

$$NegotiationEffectiveness = \frac{defectsFound}{TotalDefects}$$

Negotiation ***efficiency*** is defined as the number of defects found per person-hour.

$$NegotiationEfficiency = \frac{defectsFound}{personHour}$$

Net-gain

The ***net gain*** (Halling et al., 2003) is an economic indicator which shows the difference between negotiation benefits and negotiation costs. An activity that does not yield a net gain is not advisable from an economic point of view.

$$NetGain = NegotiationCost - NegotiationBenefit$$

Return on Investment

The ***Return*** on ***Investment*** (ROI) is defined as the net gain per invested cost unit or the interest earned on this investment. Usually an investor would choose an investment plan that maximizes the interest returned per invested unit (see also (Waters, 2008)). ROI analysis is one of several commonly used financial metrics for evaluating the financial consequences of business investments, decisions, or actions. The financial metric reveals some characteristic of the whole body of data that might not be obvious from simply reviewing the financial figures (Farris et al., 2006). The payback period cash flow metric, for instance, takes a series of cash inflows and outflows and measures the time it takes for investment returns to cover investment costs. The estimated payback periods of different potential investments can be compared, to help decide which alternative is the better investment. The wise investor, however, will also want to see other metrics for the same investment choices, as well, such as Net Present Value (NPV) and ROI. NPV is defined (Farris et al., 2010) as the sum of the Present Values (PVs) of the individual cash flows of the same entity. It is used for capital budgeting, and widely throughout economics, finance, and accounting, it measures the excess or shortfall of cash flows, in present value terms, once financing charges are met.

Comparing NPV to ROI, ROI analysis compares the magnitude and timing of investment gains directly with the magnitude and timing of investment costs (Farris et al., 2010). A high ROI means that investment gains compare favorably to investment costs. Therefore, ROI is used in this thesis because it provides appropriate information to the readers of the investment value of implementing negotiation in RE to the overall software project.

$$ROI = \frac{NetGain}{InvestedCostUnit}$$

Therefore, the evaluation criteria presented here are ***negotiation effectiveness, negotiation efficiency, net gain*** and ***return on investment.***

3.0 EXPERIMENTS

An empirical evaluation study has been done through a role play experiment to evaluate the benefit of exercising negotiation. The participants of the empirical study played roles as system stakeholders who need to deploy negotiation among them in order to identify the right requirements to be developed. The stakeholders for the system were the representative of students, lecturers, administrators and the university finance staff. The experiment was designed to allow negotiation during requirements elicitation phase and to evaluate the benefit of exercising it. Random observation done during the experiments showed that all the participating groups exercised more or less the same format as illustrated in Figure 3.1.

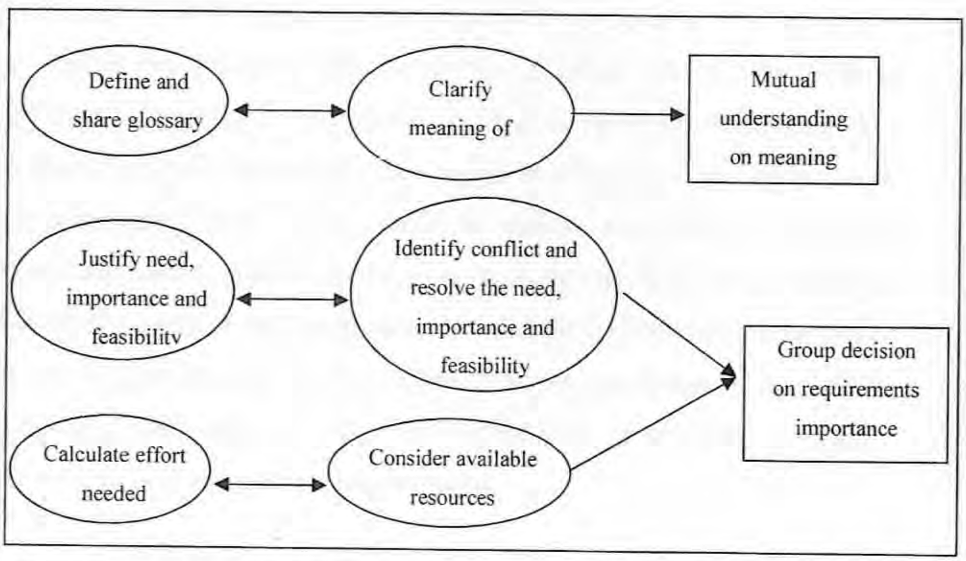

Figure 3.1: Parallel Activities in the Negotiation Process Guideline

3.1 The Device

The device for the experiment was a descriptive scenario, a list of forty requirements elicited from the descriptive scenario and groups of computer science students. A system which was familiar to the participants who played the role of the stakeholders was important. It reduced the pressure on understanding the system environment, the functionalities and the constraints. Thus, the system used in the experiment was Unit Registration System for students at Universiti Teknikal Malaysia Melaka. This was a system to enable students to register their choice of courses units. The students were third year, computer science students with software engineering knowledge background. Particularly, they were equipped with the negotiation theory and concept through formal lecture before the exercise.

3.2 The Protocol

The experiment consists of two stages to observe the achievement through negotiation and to distinguish the progress whenever additional time is given to negotiate. In order to ensure the existence of negotiation, a project constraint was inserted into the exercise. As an assumption, each group has 60 points which represents $60,000 and 60 days. The total effort needed to fulfil all the requirements are 120 points. Therefore, the students' groups have to drop some of the requirements and identify the most desired requirements worth 60 points. Furthermore, requirements difficulty level is introduced here to show that in real situation, different amount of effort is needed for different requirements. Complicated requirements need more effort compared to a simple one. The forty requirements are tagged as difficult, moderate and easy. Easy requirements need 2 points, moderate requirements need 4 points and difficult requirements need 6 points. Every stakeholder in a group owned 10 requirements (exception for the team leader) and the requirements were tagged clearly as 'S' for students, 'L' for lecturer, 'A' for administrator and 'F' for finance staff. Time was given for the participants to read the descriptive scenario and to understand their requirements.

Then, time was given to perform the negotiation in order to achieve an agreement on which requirements to have and which requirements not to have. During the negotiation process and whenever agreement was achieved, the team leader recorded 1(agree-to-have) or 0 (agree-not-to-have).

3.3 Threats to Validity

Whenever students are used as the subject for an experiment, a typical question will be asked if the experiment results are valid or not if compared to the real environment. Students are one of the most accessible sources of small scale project data. It has been shown that data gathered from students is generally applicable to the software industry. It was observed (Höst et al., 2000) that no significant differences between students and professionals for small tasks of judgment. According to (Tichy, 2001), using students as subjects is acceptable if students are appropriately trained and the data is used to establish a trend. These requirements are both fulfilled in this case.

A role play experiment always come with dilemma if the participants are really playing their role or incorporates their personal judgment. In order to minimize that possibility, prior to the experiment, the participants were given a formal lecture on negotiation with knowledge on the nature of a role play experiment and given ample time to explore their roles and their dedicated requirements. Observation done by the researcher throughout the experiment discovered that most of the time, the participants were playing the role given to them.

4.0 RESULTS

This section presents the analysis of negotiation performance in a controlled role play experiment exercising negotiation.

4.1 Negotiation Effectiveness

Negotiation effectiveness is based on the ratio of defects found to the total number of defects in the candidate requirements. The total number of defects is the same for all groups as 18 defects were seeded in the 30 candidate requirements prior to the negotiation. The total number of defects found during the negotiation by the six groups is given in Table 4.1.

Table 4.1 shows negotiation effectiveness for the six groups exercising negotiation and this indicated not really satisfying negotiation performance for all the groups. The lowest effectiveness in this case is 27.78% achieved by G4 while the highest is 83.33% effectiveness achieved by G1.

	G1	G2	G3	G4	G5	G6
Total of Defect	18	18	18	18	18	18
Total of Defect Detected	15	11	14	5	12	9
Percentage	83.33%	61.11%	77.78%	27.78%	66.67%	50%

Table 4.1: Negotiation Effectiveness

4.2 Negotiation Efficiency

Negotiation efficiency is defined as the number of defects found per group. The total effort time is unlimited. Table 4.2 shows negotiation efficiency achieved by 6 exercising negotiation. In average, 11 defects were found in 2.44 hours effort. Naturally, groups spend more effort negotiating should detect more defects but this is not the case with G4. This is due to the lack of cooperativeness among team members to strive for consensus. Based on the observation, the team let the time fly while they stayed idle and the team leader faced difficulties trying to get things moving. On the other hand, G3 detected 14 defects in 1 hour and 15 minutes only. This due to the commitment among team members and the ability of the team leader to control the situation and made negotiation happened.

	G1	G2	G3	G4	G5	G6	Average
No. Of Defect Found	15	11	14	5	12	9	11
Total Effort	2.45h	2.3h	1.15h	5.35h	2.25h	1.15h	2.44h
Negotiation Efficiency	6.1	4.8	12.2	0.9	5.3	7.8	6.18

Table 4.2: Negotiation Efficiency

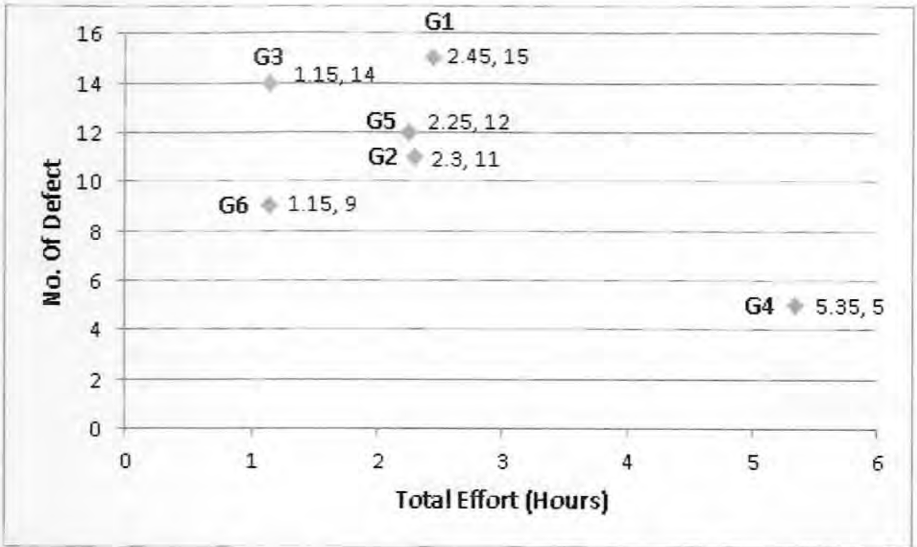

Figure 4.1: The relationship between the number of defects detected and effort spent negotiating

4.3 Net-gain

Net gain is the difference between negotiation benefits and negotiation costs. For the assumption on negotiation benefits, conservative (low) benefit value is used to stay on the conservative side in the economic evaluation. Table 4.3 shows the net-gain value which is calculated by the difference between negotiation benefits and negotiation cost. Negotiation cost is direct negotiation cost invested to negotiate in group by five participants. Overall, the average net gain achieved is 2.44 hours.

	G1	G2	G3	G4	G5	G6	Average
Negotiation Cost	2.45h	2.3h	1.15h	5.35h	2.25h	1.15h	2.44h
Negotiation Benefit	15	11	14	5	12	9	11
Net Gain	12.55	8.7	12.85	0.35	9.75	7.85	8.66

Table 4.3: Net Gain

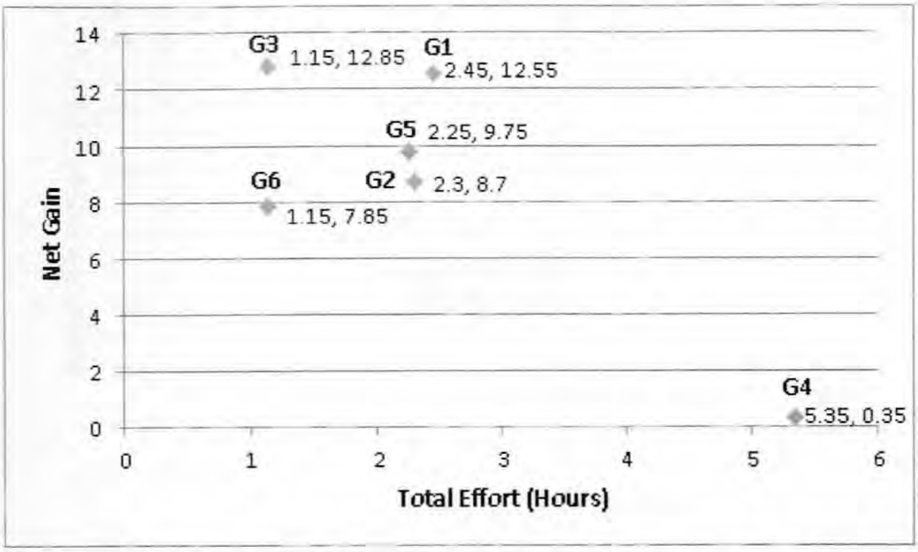

Figure 4.2: The relationship between the net gain achieved and effort spent negotiating

4.4 Return on Investment

Return on investment (ROI) is the net-gain per invested cost unit. Table 4.4 shows the return on investment negotiation for all the groups. The ROI is calculated based on a very optimistic assumption in which all the defects are assumed easy to fix. This means, the benefit value use here is very low. Still, the net-gain and the ROI shows positive value which suggest that negotiation activities worth an investment.

	G1	G2	G3	G4	G5	G6	Average
Negotiation Cost	2.45h	2.3h	1.15h	5.35h	2.25h	1.15h	2.442
Net Gain	12.55	8.7	12.85	0.35	9.75	7.85	8.675
Return On Investment	512%	378%	1117%	6.5%	433%	682%	521.42%

Table 4.4: ROI

can be saved. Therefore, negotiation saves time and money through preventive activity and agreement by the stakeholders; it then follows that development and quality control teams spend less time on rework.

6.0 REFERENCES

References

Basili, V. R. 1993, 'The Experience Factory and its Relationship to Other Improvement Paradigms', *4th European Software Engineering Conference,* London, Springer-Verlag, vol. pp.65-82.

Berenbach, B., Paulish, D., Kazmeier, J. & Rudorfer, A. 2009 *Software & Systems Requirements Engineering: In Practice,* McGraw-Hill, New York.

Biffl, S., Freimut, B. & Laitenberger, O. 2001a Investigating the cost-effectiveness of reinspections in software development. *Proceedings of the 23rd International Conference on Software Engineering.* Toronto, Ontario, Canada, IEEE Computer Society.

Biffl, S., Freimut, B. & Laitenberger, O. 2001b, 'Investigating the cost-effectiveness of reinspections in software development', *the 23rd International Conference on Software Engineering,* Toronto, Ontario, Canada, IEEE Computer Society, vol. pp.155-164.

Biffl, S. & Gutjahr, W. 2001, 'Influence of Team Size And Defect Detection Technique On Inspection Effectiveness', *7th International Symposium on Software Metrics,* London, IEEE Computer Society, vol. pp.63-75.

Biffl, S. & Halling, M. 2003a, 'Investigating the defect detection effectiveness and cost benefit of nominal inspection teams'. *Software Engineering, IEEE Transactions on.* vol. 29, pp. 385-397.

Biffl, S. & Halling, M. 2003b, 'Investigating the defect detection effectiveness and cost benefit of nominal inspection teams'. *Software Engineering, IEEE Transactions on IEEE.* vol. 29, pp. 385-397.

Boardman, A. E., Greenberg, D. H., Vining, A. R. & Weiner, D. L. 2006 *Cost Benefit Analysis: Concepts and Practice (3rd Edition),* Prentice Hall, Upper Saddle River, NJ.

Boehm, B. & Egyed, A. 1998, 'Software Requirements Negotiation: Some Lessons Learned', *20th International Conference on Software Engineering,* Kyoto, Japan, IEEE Computer Society, vol.

Damian, D. E. H. 2000 Challenges in Requirements Engineering. *Computer Science Technical Report.* Calgary, The University of Calgary.

Dorfman, M. 1997 Requirements Engineering, in Thayer, R., H. & Dorfman, M. (Eds.) *Software Requirements Engineering.* Second Edition ed. Washington, IEEE Computer Science Society, pp.7-22.

Farris, P. W., Neil, T. B., Pfeifer, P. E. & Reibstein, D. J. 2010 *Marketing Metrics: The Definitive Guide to Measuring Marketing Performance. Second Edition,* Pearson Prentice Hall, New Jersey.

Farris, P. W., Neil, T. B., Pfeiffer, P. E. & Reibstein, D. J. 2006 *Marketing Metrics: 50+ Metrics Every Executive Should Master,* Pearson Education, Inc., New Jersey.

Gilb, T. & Graham, D. 1993 *Software Inspection,* Addison-Wesley, Great Britain.

Grunbacher, P. & Syeff, N. 2005 Requirements Negotiation, in Aurum, A. & Wohlin, C. (Eds.) *Engineering and Managing Software Requirements.* Berlin, Springer-Verlag, pp.143-158.

Halling, M., Biffl, S. & Grünbacher, P. 2003, 'An economic approach for improving requirements negotiation models with inspection '. *Requirements Engineering.* vol. 8, pp. 236-247.

Hofmann, H. F. & Lehner, F. 2001, 'Requirements engineering as a success factor in software projects'. *Software, IEEE.* vol. 18, pp. 58-66.

Höst, M., Regnell, B. & Wohlin, C. 2000, 'Using Students as Subjects—A comparative study of students and professionals in lead-time impact assessment'. *Empirical Software Engineering.* vol. 5, pp. 201 - 214

Jones, C. 1996 *Applied Software Measurement: Assuring Productivity and Quality,* McGraw-Hill, New York.

Juristo, N., Moreno, A. M. & Silva, A. 2002, 'Is the European Industry Moving Towards Solving Requirements Engineering Problems?'. *Software IEEE.* vol. 19, pp. 70-77.

Kotonya, G. & Sommerville, I. 1997 *Requirements Engineering Process and Techniques,* John Wiley & Sons, New York.

Pressman, R. S. 2005 *Software Engineering A Practitioner's Approach 6th Edition,* McGraw Hill, New York.

Prest, A. R. & Turvey, R. 1965, 'Cost-benefit Analysis: A Survey.'. *The Economic Journal.* vol. 75, pp. 683-735.

Price, J. & Cybulski, J., L. 2006, 'The Importance of IS Stakeholder Perspectives and Perceptions to Requirements Negotiation', *Australian Workshop on Requirements Engineering,* Adelaide.

Robinson, R. 1993, 'Cost-benefit analysis'. *BMJ.* vol. 307, pp. 924-926.

Sommerville, I. 2004 *Software Engineering 7th Edition,* Addison-Wesley, U.S.

Tichy, W. 2001, 'Hints for reviewing empirical work in software engineering'. *Empirical Software Engineering.* vol. 5, pp. 309-312.

Verner, J., Cox, K., Bleistein, S. & Narciso, C. 2005, 'Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the U.S.'. *Australasian Journal of Information Systems.* vol. 13, pp.

Waters, D. 2008 *Quantitative Methods for Business, 4th Edition,* Financial Times Press.

Zowghi, D. & Coulin, C. 2005 Requirements Elicitation: A Survey of Techniques, Approaches and Tools, in Aurum, A. & Wohlin, C. (Eds.) *Engineering and Managing Software Requirements.* Berlin, Spriner-Verlag, pp.19-41.