AN APPROACH TO ESTIMATE THE SAVINGS FROM NEGOTIATION BASED ON COST – BENEFIT ANALYSIS MODEL

DR. SABRINA BINTI AHMAD
NOOR AZILAH BINTI DRAMAN@MUDA
DR. AZAH KAMILAH BINTI DRAMAN @ MUDA
ZAHRIAH BINTI OTHMAN
(The 5th Malaysian Software Engineering Conference (MYSEC 2011), 13-14 December 2011, The Puteri Pacific, Johor)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2011 5th Malaysian Conference in Software Engineering (MySEC)

An Approach to Estimate the Savings from Negotiation based on Cost-Benefit Analysis Model

Sabrina Ahmad
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Malaysia
sabrinaahmad@utem.edu.my

Noor Azilah Muda
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Malaysia
azilah@utem.edu.my

Azah Kamilah Muda
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Malaysia
azah@utem.edu.my

Zahriah Othman
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka
Malaysia
zahriah@utem.edu.my

Abstract—**This paper presents an approach to estimate the savings from implementing negotiation in requirements elicitation process. The aim of implementing negotiation is to minimize the possibility of introducing defects during the creation of requirements and to decrease later effort required to fix requirements defects. An empirical evaluation study is adopted through a role play experiment to evaluate the benefit of exercising negotiation. The net-gain and the return-on-investment show positive value which suggest that negotiation activities worth an investment. Based on the return-on-investment of 197 percent in average, this paper suggests that negotiation is a useful prevention activity to inhibit defects from occurring during the creation of requirements.**

Keywords-component; empirical study, cost-benefit analysis, requirements elicitation, negotiation

I. INTRODUCTION

Project Managers aim to deliver a product of sufficient quality on time and within budget. In line with that, research has been done to reduce the defects by detecting and fixing the defects early [1-5] to better improve overall quality; both in the software development process and the end product. However, Boehm claimed that [5] current software projects spend about 40 to 50 percent of their effort on avoidable rework. Such rework consists of effort spent fixing software difficulties that could have been discovered earlier and fixed less expensively or avoided altogether. In order to reduce the effort of rework, inspections have been a well researched area to detect and to remove defects. The effort is claimed and proved to save rework effort in later project phases and to reduce the overall effort of the project [1-3, 6].

This paper attempts to show that negotiation is able to reduce the number of defects and save rework later in the project. Negotiation improves requirements quality to provide a higher quality of input [7] for development and project planning. The improvement is achieved through ability to identify conflicts, to share the meaning of keyword [8], to share perspectives, views, and expectation [9, 10] on the requirements, to assess the system feasibility, to justify the requirements needs [11, 12] and to prioritize the requirements [13, 14] in order to produce better quality requirements.

Following section I, section II provides an overview of related works. This is followed by section III which gives background on the cost of defects. Next, section IV explains the approach of cost-benefit analysis adapted in this research to estimate the benefit of negotiation. This is then followed by section V which elaborates on the empirical evaluation study. Section VI provides the analysis of the study and Section VII concludes the paper.

II. RELATED WORKS

It is basic knowledge that all efforts in software development incur cost as they involve human-effort. Suggesting requirements engineering process to add extra effort like negotiation or inspection obviously increase the development cost. How does extra effort save project cost? In order to answer this question, few researches [3, 4, 6] sought to calculate net-gain resulting from defect detection efforts through document inspection. The researchers investigated the effectiveness of defect detection and the cost value gained from the effort through empirical study. The net-gain was calculated based on effort benefit where the cost from rework to fix the defects is saved. The approach to determine the benefit for a defect and the benefit distribution was clearly discussed. The role play experiment exercised in Halling's research [4] and the method to calculate the net-gain are adapted to this research.

This paper is not suggesting that negotiation is better than inspection. Both efforts have different roles in reducing defects in software requirements. There exists fundamental difference between negotiation and inspection. Inspection is a validation and verification activity designed to catch defects within

978-1-4577-1531-0/11/$26.00 ©2011 IEEE

written documents before the defects propagate through a development process. Negotiation during the RE phase is designed to prevent defects occurring in the first place. The effectiveness of negotiation is measured in the reduction in number of defects needing to be discovered and resolved through validation, verification and testing. At some stage, these two efforts complement each other.

This paper is suggesting that 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.

III. THE COST OF DEFECTS

In order to reveal the amount of benefit gained from negotiation, the cost of the defects needs first to be discovered. According to Raffo et al [15] the cost of defects can be divided into five components;

1. *The cost of preventing defects* – these resources are expended in preventing defects from occurring.
2. *The cost of searching for defects* – these resources are expended in looking for defects that may have occurred.
3. *The cost of isolating and verifying defects* – these resources are expended to isolate and verify the defect as well as to record, track and establish the disposition of an anomaly once it is detected.
4. *The cost of fixing defects* – these resources are expended to correct defects that have been found, and determined to require correction.
5. *The cost of defect occurrences* – defects that "slipped though" the defect detection process, or defects that were found during the search activity but not fixed and are subsequently encountered after delivery, usually will have some measureable impact associated with them.

This paper refers to (4), the cost of defects means the cost of fixing defects. Negotiation is an effort to prevent defects from occurring and therefore the cost to exercise negotiation is part of (1), "a cost to prevent defects". Benefit is gained if (4) "the cost of fixing defects" is more than (1) "the cost of the negotiation" after the standard requirements process is taken into account.

The cost of defects is best determined by the exact cost spent to fix the defects throughout the software life-cycle. The cost to fix defects is not fixed but depends on the severity of the defects, the phase in which the defects surfaced and is due to the increasing cost to fix defects the longer the defects remain hidden in the development process. Estimation of full cost and benefit obtained is feasible only when the full software process is exercised and the end product is available. Hence, the cost of defects are only known when the software system is complete and technically at the end of the software life cycle.

Since a full cycle of software development was not available in this research, the exact cost to fix the defects could not be measured in this way. Therefore, an assumption of estimation is made based on theory, literature and several similar researches. Each project will have its own cost to fix the defects and there will be no 'one standard cost' because the cost depends on the activities undertaken in the project and when the project commences tracking and fixing defects [16]. The longer defects stay in the development process the more it will cost the software project to fix them. Therefore, an effort to reduce the number of defects, which may propagate into the later development process phase, is an investment to save the software project cost.

IV. THE APPROACH

A. 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 on economic terms, one can then calculate net gains and base decisions on these economic values. In line with the cost-benefit analysis used in both Biffl [3, 6] and Halling[4] researches; 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 paper, 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 [3]. 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 paper is presented 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.

Here, defect severity is considered based on the likely impact of a defect on further development [4]:

- 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 caused 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, 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 on 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 [17].

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. Defects may slip into later development stages and thereby increase the risk/cost to the project. This research is motivated by benefit distribution used by Halling [4].

TABLE I. EXPECTED BENEFIT IN HOURS PER DEFECT AVOIDED DURING NEGOTIATION

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

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 paper is based on the understanding of three defined different severity of classes of defects. 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.

B. Evaluation Criteria

This sub-section describes evaluation criteria for negotiation performance which are divided into negotiation benefit, negotiation cost, net-gain and return on investment.

1) 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. The negotiation benefit is defined as the number of defects avoided multiplied by the benefit based on Table 1. The benefit is always in a low severity level and in a best case scenario to stay in a conservative side to assume low benefit for all the defects avoided [4].

$$NegotiationBenefit = defects \times benefitPerDefect$$

2) 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 [4].

$$NegotiationCost = staffHours$$

3) Negotiation Effectiveness and Efficiency

The ***effectiveness*** of 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 [4, 6].

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

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

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

4) Net-gain

The ***net gain*** [4] 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.

$$NetGain = NegotiationCost - NegotiationBenefit$$

5) 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 [18]).

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

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

V. THE EMPIRICAL EVALUATION STUDY

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.

A. 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 The University of Western Australia. This was a system to enable students to register their choice of courses units. The students were third year, fourth year and master computer science students with software engineering knowledge background. Particularly, they were equipped with the negotiation theory and concept through formal lecture before the exercise.

B. 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, 20 minutes 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) or u (undecided). Next, a second chance was given to re-negotiate and another 20 minutes was given. This was the opportunity for the groups to consider more views from the stakeholders, to explore the requirements rigorously and to carefully make a better group decision. Again, during the negotiation process and whenever agreement was achieved, the team leader recorded 1 (agree-to-have) or 0 (agree-not-to-have) or u (undecided) in the consensus sheet. After the negotiation was over, an individual stakeholder had an opportunity to record their own say. In the individual sheet, every stakeholder can identify the requirements which they really think they wanted.

C. 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. Host [19] observed no significant differences between students and professionals for small tasks of judgment. According to Tichy [20], 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.

VI. THE ANALYSIS

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

A. 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 40 candidate requirements prior to the negotiation. The total number of defects found during the negotiation by the six groups is given in Table II.

Table II shows negotiation effectiveness for the six groups exercising negotiation and this indicated satisfying negotiation performance for all groups. The lowest effectiveness in this case is 22% achieved by G3 while the highest is 83% effectiveness achieved by G6. On average, the mean effectiveness for all groups is 55%. This can be represented as a triangle distribution. Table III indicates negotiation effectiveness achieved in 20 minutes and 40 minutes negotiation. In average, the figure shows very low effectiveness in 20 minutes negotiation but shows that the performance increases in 40 minutes negotiation. Overall performance shows satisfying achievement to detect defects through negotiation.

TABLE II. NEGOTIATION EFFECTIVENESS

	G1	G2	G3	G4	G5	G6
Total number of defects	18	18	18	18	18	18
Total number of defects detected	7	10	4	12	11	15
Defects detected effectiveness	39%	56%	22%	67%	61%	83%

TABLE III. PERFORMANCE INCREASE IN EFFECTIVENESS

	G1	G2	G3	G4	G5	G6	Av
20 minutes effectiveness	6%	0%	0%	33%	17%	72%	21%
40 minutes effectiveness	39%	56%	22%	67%	61%	83%	55%
Performance increase	33%	56%	22%	34%	44%	11%	33%

B. *Negotiation Efficiency*

Negotiation efficiency is defined as the number of defects found per person-hour. The total effort is 40 minutes negotiation which involved five participants each group. Hence, the total effort based on staff hour is 3.3 hour effort per group. Table IV shows the negotiation efficiency achieved by six groups exercising negotiation. In average, 10 defects were found by 3.3 hours effort and as for the efficiency, 3 defects were found per-hour.

TABLE IV. NEGOTIATION EFFECIENCY

	G1	G2	G3	G4	G5	G6	Av
Number of defects found	7	10	4	12	11	15	10
Total effort	3.3h	3.3h	3.3h	3.3h	3.3h	3.3h	3.3h
Negotiation efficiency	2.1	3.1	1.2	3.6	3.3	4.5	3.0

C. *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. Even though there is a mixture of severity level of defects as defined in Table I, the negotiation benefit here is assumed in low benefit value. Table V shows the net-gain value which is calculated by the difference between negotiation benefits and negotiation costs. Negotiation cost is direct negotiation cost invested to negotiate in person-hour by five participants. Negotiation benefit is calculated based on the number of defects found and excluded in the agreement times 1 hour benefit each. The average net-gain achieved here is 6.5 hour. Overall, the result shows that the cost of negotiation is lower than the benefit obtained from negotiation which revealed positive value in the net-gain for all the groups.

TABLE V. NET-GAIN IN HOURS

	G1	G2	G3	G4	G5	G6	Av
Negotiation cost	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Negotiation benefit	7	10	4	12	11	15	10
Net-gain	3.7	6.7	0.7	8.7	7.7	11.7	6.5

D. *Return on Investment*

Return on investment (ROI) is the net-gain per invested cost unit. Table VI shows the return on investment for an hour 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 used here is very low. Still, the net-gain and the ROI shows positive value which suggest that negotiation activities worth an investment.

TABLE VI. RETURN ON INVESTMENT

	G1	G2	G3	G4	G5	G6	Av
Negotiation cost (h)	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Net-gain (h)	3.7	6.7	0.7	8.7	7.7	11.7	6.5
ROI (%)	112	203	21	264	233	355	197

VII. CONCLUSION

As a conclusion, this paper provides an empirical framework for estimating the benefit of negotiation. Empirical data was used to quantify the likely return-on-investment for introducing a formal negotiation phase within RE. Based on the empirical data, this paper suggests that negotiation is a useful prevention activity to inhibit defects from occurring during the creation of requirements. This activity is especially useful when it involves multiple stakeholders with different roles and priorities. It is general knowledge that the role of negotiation is to achieve an agreement but while working together to reach an agreement, further benefits are obtained. For example, the requirements are refined into a feasible piece of functionality which is assessed to be achievable within project's constraints and reveal tacit knowledge among the stakeholders to develop understanding. This process therefore produces better quality requirements in which defects such as inconsistency, infeasibility, incomplete, incorrect and incomprehensible are detected and removed from the requirements.

This effort yields economic benefit to the software project in which the unnecessary cost of fixing defects later in the software development process 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.

REFERENCES

1. Megen, R. and D.B. Meyerhoff, *Costs and benefits of early defect detection: experiences from developing client server and host applications.* Software Quality Journal, 1995. **4**(4): p. 247-256.
2. Oliver, L., *Cost-effective Detection of Software Defects through Perspective-basedInspections.* Empirical Softw. Engg., 2001. **6**(1): p. 81-84.
3. Biffl, S., B. Freimut, and O. Laitenberger, *Investigating the cost-effectiveness of reinspections in software development*, in *Proceedings of the 23rd International Conference on Software Engineering.* 2001, IEEE Computer Society: Toronto, Ontario, Canada.
4. Halling, M., S. Biffl, and P. Grünbacher, *An economic approach for improving requirements negotiation models with inspection* Requirements Engineering, 2003. **8**(Number 4): p. 236-247.
5. Boehm, B. and V.R. Basili, *Software Defect Reduction Top 10 List.* Computer, 2001. **34**(1): p. 135-137.
6. Biffl, S. and M. Halling, *Investigating the defect detection effectiveness and cost benefit of nominal inspection teams.* Software Engineering, IEEE Transactions on, 2003. **29**(5): p. 385-397.
7. Ahmad, S., *Requirements Negotiation: Making System Stakeholders' Multiple-Views One.* Journal of Communication and Computer, 2011. **8**(4): p. 290-299.
8. Grünbacher, P. and B. Boehm. *EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation.* in *8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering.* 2001. Toronto, Canada.
9. Price, J. and J. Cybulski, L. *The Importance of IS Stakeholder Perspectives and Perceptions to Requirements Negotiation.* in *Australian Workshop on Requirements Engineering.* 2006. Adelaide.
10. Boehm, B., et al. *Software requirements as negotiated win conditions.* in *Requirements Engineering, 1994., Proceedings of the First International Conference on.* 1994.
11. Pruit, D.G., *Achieving Integrative Agreement*, in *The Negotiation Sourcebook*, I.G. Asherman and S.V. Asherman, Editors. 2001, HRD Press: U.S. p. 187-196.
12. Boehm, B. and A. Egyed. *Software Requirements Negotiation: Some Lessons Learned.* in *20th International Conference on Software Engineering.* 1998. Kyoto, Japan: IEEE Computer Society.
13. Berander, P. and A. Andrews, *Requirements Prioritization*, in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Editors. 2005, Springer-Verlag: Berlin. p. 69-91.
14. Hatton, S. *Software Requirements Prioritisation: The Client's Perspectives.* in *Fifteenth University of Western Australia, School of Computer Science & Software Engineering Research Conference.* 2006. Yanchep, Western Australia: CSSE, University of Western Australia.
15. Raffo, D., et al. *Understanding the role of defect potential in assessing the economic value of process improvements* in *Second Workshop on Economics-Driven Software Engineering Research.* 2000. Limerick, Ireland: ACM.
16. Rothman, J. (2002) *What Does It Cost to Fix a Defect?* StickyMinds.com Weekly Column.
17. Basili, V.R., *The Experience Factory and its Relationship to Other Improvement Paradigms*, in *Proceedings of the 4th European Software Engineering Conference on Software Engineering.* 1993, Springer-Verlag.
18. Waters, D., *Quantitative Methods for Business, 4th Edition.* 2008: Financial Times Press.
19. Höst, M., B. Regnell, and C. Wohlin, *Using Students as Subjects—A Comparative Study ofStudents and Professionals in Lead-Time Impact Assessment.* Empirical Software Engineering, 2000. **5**(3): p. 201 - 214
20. Tichy, W., *Hints for reviewing empirical work in software engineering.* Empirical Software Engineering, 2001. **5**(4): p. 309-312.