KISDI
KOREA INFORMATION SOCIETY DEVELOPMENT INSTITUTE
Asian Journal of Information and Communications

A practical approach for evaluating the software projects based on a request for proposal focusing on non-functional requirements

Rezaali Mosayyebi*
Mazandaran University of Science and Technology

Homayun Motameni**
Sari Branch, Islamic Azad University

Abstract

Before the beginning of the software development project which is based on contract between software user and developer company, the user company provides a RFP. This term includes project domain current situations and constraints, and all conditions such as desired initial system requirements specification. Also, the documents of developer companies' capabilities are asked for their proposals to fulfill the requirements and compete with each other. This paper renders an evaluation model of proposals based on functional and nonfunctional requirements(FRs and NFRs). In the meantime, technics and management capabilities of developer company, price and other contract conditions such as ownership agreements are included. The implementation shows effectiveness of the proposed methodology.

Keywords: request for proposal (RFP), non-functional requirement (NFR), quality attributes evaluation, evaluation model

1. Introduction

For a software project contract between software user and developer company, the user company provides a request for proposal(RFP) or software requirements specification (SRS). The RFP contains organization's initial system requirements such as business requirements, system architecture requirements, functional requirements(FRs), non-functional requirements(NFRs), license agreement, bidding qualification of suppliers, contract conditions and so on (Saito, Monden, and Matsumoto, 2012). Many of the problems in a software project are caused by faults in the requirements that properly the RFP or SRS should determine. It is said that items of re-work due to faults in the requirements account for 30~50% of the total cost of development, resulting in cost overruns (Boehm and Papaccio, 1998).

The study conducted by Price Waterhouse shows that the cost of software quality varies between 38 percent and 49 percent of the total cost of development. It excludes the cost of testing and the resulting anomalies, thereby reducing the cost of software quality, estimated to be between 40 percent and 55 percent of the project cost (Laporte, Nabil and Mikel, 2012).

According to Charette (2005), "Studies have shown that software specialists spend about 40-50 percent of their time on avoidable rework rather than on what they call value-added work, which is basically work that's

* Corresponding author: rmosayyebi@ict-tcm.ir
** motameni@iausari.ac.ir

done right the first time. Once a piece of software makes it into the field, the cost of fixing an error can be 100 times as high as it would have been during the development stage" (Charette, 2005).

If the RFP is ambiguous, the solution and the proposal will not be clear. The process to provide RFP and to review, analyze and evaluate a written proposal is an important and time-consuming task.

Furthermore, the expenses for correction of defects discovered in the latter half of the lifecycle of software development become more massive than the expenses for correction during the first half. For these reasons, it is important to ensure quality of description content of documents that created in upper process of software development such as RFP or SRS (Terawaki and Tamai, 2013). So this fact can be generalized for the proposals which have been written by developer companies. However, there is no clear evaluation criteria to make sure that basic user requirements are written neither in the RFP, nor in the developer company written proposals. The RFP itself should have a clearly defined section that outlines what the user wants the developers to provide in their proposals and be specific on the order in which their responses should be structured. This orderly and controlled structure will aid the user company's evaluation team as it reviews and traces the materials from each proposal. The degree of success and failure of a software system depends upon the level and quality of services it provides, as required by its users and stakeholders (Ambreen, Ikram, Usman and Niazi, 2016). A lot of problems happen due to low quality of the user company's RFP and developer company's proposal, such as (1) gap between user's intended requirements and developer's understanding, (2) ambiguity of project's domain, (3) ambiguity of each quality attribute coverage and satisfaction level, (4) neither user company nor developer company satisfy cause the intrinsic ambiguity, (5) evaluation problem for evaluator team.

In this paper we render an evaluation model of proposals based on functional and nonfunctional requirements, technics and management capabilities of developer companies, price and other contract conditions such as ownership agreements. This helps user companies and organizations in defining the evaluation process for the RFP's proposals. The evaluation model offers a suggested structure to the evaluation that allows input from a wide range of developers and specific subject matter experts, permits flexibility and weighted scoring in appropriate areas and provides a defensible process for determining the optimum desired proposal.

The rest of this paper is organized as follows: Section 2 summarizes the related works, Section 3 provides construction proposals' evaluation model, Section 4 introduces case studies of proposals' evaluation, and finally Section 5 provides summary and future work.

2. Related work

2.1 Standards and guidelines

Several standards or guidelines about FRs, NFRs or software quality requirements, and metrics focusing on evaluating RFPs as well as proposals have been published.

- **The software quality standard ISO/IEC 9126: 2001** (ISO/IEC, 2011a) divided into four parts: (1) quality model (2) external metrics (3) internal metrics (4) quality in use metrics. Ideally, the internal quality determines the external quality and external quality determines quality in use. This standard stems from the GE model for describing software quality, presented in 1977 by McCall et al., which is organized around three types of quality characteristic:

. Factors(to specify): they describe the external view of the software, as viewed by the users.
. Criteria (to build): they describe the internal view of the software, as seen by the developer.
. Metrics(to control): they are defined and used to provide a scale and method for measurement.

ISO/IEC 9126 was replaced by ISO/IEC 25010:2011 - Systems and software Quality Requirements and Evaluation (SQuaRE).

- **The software quality standard ISO/IEC 25010:2011** (ISO/IEC , 2011b) defines: (1) A quality in use model composed of five characteristics that relate to the outcome of interaction when a product is used in a particular context of use (2) A product quality model composed of eight characteristics that relate to static properties of software and dynamic properties of the computer system. The model is applicable to both computer systems and software products. The characteristics provide consistent terminology for specifying, measuring and evaluating system and software product quality. They also provide a set of quality characteristics against which stated quality requirements can be compared for completeness.

ISO/IEC 25012 contains a model for data quality that is complementary to this model. The scope of application of the quality models includes supporting specification and evaluation of software and software-intensive computer systems from different perspectives by those associated with their acquisition, requirements, development, use, evaluation, support, maintenance, quality assurance and control, and audit. The models can, for example, be used by developers, acquirers, quality assurance and control staff and independent evaluators, particularly those responsible for specifying and evaluating software product quality (ISO/IEC, 2011b).

- **The Non-Functional Requirements Grades Usage Guide** (Information-Technology Promotion Agency Software Engineering Center, 2010) composed of the “Usage Manual” and the “Description Manual”. The first manual indicates how to utilize non-functional requirements grades for visualization of system infrastructure outsourcer requirements. The objectives of second manual are to provide the background behind the creation of the non-functional requirements grades for visualization of system infrastructure outsourcer requirements, and to provide detailed information regarding non-functional requirements grades tools. Both of manuals are targeted primarily at persons responsible for placing or receiving orders, and who are involved in the provision, proposal, or determination of non-functional requirements during the requirement definition phase or similar phases of the development of information systems such as corporate business systems.

- **IEEE Std 830-1998 Recommended Practice for Software Requirements Specifications (SRS)** (IEEE, 1998) describes recommended approaches for the specification of software requirements. It is based on a model in which the result of the software requirements specification process is an unambiguous and complete specification document.

- **The Carnegie Mellon Software Engineering Institute™ (SEI)** (Gross, 2009) has initiated an effort to compile publicly available recommendations for RFP content. This paper was developed in response to Task 2.2.2 of the FY09 Strategic Software Improvement Plan (SSIMP), which is the implementation plan for the Army Strategic Software Improvement Program (ASSIP). Task 2.2.2 seeks to “define and communicate the software engineering and management events and deliverables necessary to be included in the RFP or the contract to support successful acquisition of software intensive systems.”

- **A user view guideline provided by the Information Promotion Agency, Japan (**Software Engineering Center, Information-Technology Promotion Agency, 2008) for FRs to concenter users' and developers' point of view.
- **The public guidelines for evaluation of software products quality** (Iranian National Standards Organization, 2000)
- **The software development and production standards (**The Iranian ICT Guild organization, 2008) composed of two parts. The first part includes about RFP, proposal, contract, competition, evaluation, etc. The second part includes the standard artifacts such as Test Plan, Transition Plan, Project Quality Plan, Project Management Plan, Configuration Management Plan, etc.

2.2 Evaluation of requirements(FRs and NFRs)

In Misaghian and Motameni (2016) by modeling a three-order tensor, was decided to consider the simultaneous effect among FRs, NFRs and stakeholders that have different preferences on requirements. To define the quality requirements and to check the quality attributes carefully is necessary for bringing good quality software and ensuring quality of the service (Terawaki and Tamai, 2013). The paper proposed a framework that measured the quality attributes in the requirement document (SRS, RFP and Proposal). In Thakurta(2013), Galster and Eberlein (2011) and Cleland-Huang, Settimi, Zou and Solc (2006), the researchers focus on prioritization of requirements. To avoid the ambiguity of user requirements, the Quality Function Deployment method by Kou et al.(2008) has been proposed. The research in Li et al.(2014) proposed a modeling language for NFRs that views NFRs as requirements over qualities, mapping a software related domain to a quality space. Also, they offered a methodology for systematically refining informal NFRs elicited from stakeholders, resulting in unambiguous, de-idealized, and measurable requirements. The proposal is evaluated with a requirements dataset that includes 370 NFRs crossing 15 projects. Several studies have focused on NFRs (Cleland-Huang, Settimi, Zou and Solc, 2006; Kiewkanya and Muenchaisri, 2005; Galster and Eberlein, 2011; Andrea, 2003; Sugumaran and Drake, 2006) in an early stage of development. The research in Naoki, Takashi and Takao (2008) prevents from ambiguous contracts. Also in articles such as Tomonori et al. (2008), tools to estimate the percentage of quality-related aspects in the requirements document using natural language processing has been proposed.

2.3 Evaluation of RFP and proposals based on FRs and NFRs

It is difficult to prove FRs comprehensive evaluation existed in the RFP but it is simple for proposals. The FRs existed in the proposals according to RFP are prioritized, weighted and normalized easily. Our method to evaluate NFRs is exactly similar to Saito, Monden and Matsumoto (2012). A simple evaluation model of NFRs in RFP has been proposed in Saito, Monden and Matsumoto (2012), mainly focusing on the user maintenance and operation issues. The main parts of rendered model are as follows: NFR categories, NFR metrics, description level grading and weight to each NFR. We used this suitable model for proposals' NFRs evaluation. Because each NFR metric has different importance, in Saito, Monden and Matsumoto(2012) to compute a score for each NFR abstraction level (high, middle and low), was weighted. The numbers of "yes" answers to a question "it is important to user companies?" in Ministry of Economy, Trade and Industry, Mitsubishi Research Institute Inc (2010) was used as a weight for each NFR metric since a high number means more important for user organizations. As a result in Saito, Monden and Matsumoto (2012) was constructed a "RFP Evaluation

Table" that we use as part of "Proposals' Evaluation Model".

It is necessary about the proposals that the points respect the NFRs should be normalized such as FRs.

3. Construction proposals evaluation model

The first step is to determine the major components of the RFP to be scored and publish that high-level evaluation method in the RFP document itself. The Iranian ICT Guild organization (IIG) in The Iranian ICT Guild organization (2008) proposed a method for the RFP's evaluation, but we added other factors such as FRs, NFRs and Demonstration. A sample evaluation breakdown could be as shown in Table 1:

Table 1. Major components

1	Management Capabilities	20%
2	Technics Capabilities	25%
3	Functional Requirements(FRs)	30%
4	Non-Functional Requirements(NFRs)	20%
5	Demonstrations	5%

The percentages above can vary based on the preference of the user company. A wide range of stakeholders should contribute to the evaluation of the vendors and their proposed solutions in the demonstrations sessions. Not all participants need to be involved in every component but it is wise, depending on the size of the company, to have a core team of 5-7 individuals. The sample management and technics capabilities factors, that we have used the sources in Iranian National Standards Organization (2000) and The Iranian ICT Guild organization (2008) could be as shown in Tables 2, 3:

Table 2. The proposed management capabilities

	Management Capability	Max Score
1	ICT Guild organization (IIG) membership	5
2	Related subject: 1.software 2.e-business 3.web 4.telecom 5.etc {each subject has 1 point}	5
3	Previous participation {each project has 5 points}	15
4	Similar projects {each project has 5 points}	40
5	Pilot or test installation site visit	10
6	Vendor company's organizational structure {each key experts such as project manager, analyst, designer, programmer 1 point; Established record 1 score every 2 years up to 5 points}	15
7	Native point: {providing evidence of the establishment of the company in province 5 points; an office or agency in the province 5 points}	10
	Total	100

Table 3. The proposed technics capabilities

	Technics Capability	Max Score
1	work breakdown structure	15
2	work breakdown vision to do; each stage artifacts	12
3	project proposed structure	11
4	Project schedule	8
5	Project expert considered: for each specialty intended awarded 1 point (maximum 7 specialty)	7
6	Amendments	7
7	Standards and tools used	10
8	quality control and assurance process	8
9	Project management plan and process	8
10	Project control process	7
11	Specifically, the project staff	7
	Total	100

The percentages above in Tables 2, 3 can vary based on the preference of the user company, contract and project.

Now we have all of the points plus the proposed prices.

So here we are able to obtain adjusted price for each proposal that will determine the final winner company and it's proposal.

If i is impact factor, t is technical points, p is proposed price, then L will be adjusted price in the following formula:

$$L=(100\times p)/(100-[i\times(100- t)])$$

For example, suppose the following three companies, participated in a software development competition, and all of points according to the model have been computed, assuming i=0.70, therefore proposal C will be selected.

Table 4. Example of final selection in evaluation model

Company	Proposed Price	Technical Point (normalized)	Adjusted Price
A	1000$	40	1724$
B	1200$	60	1668$
C	1500$	90	1613$

4. Case study of proposals evaluation

As a case study, 20 proposals of 5 telecommunication projects RFPs were evaluated by the proposed model. As a result, we have confirmed that the model can identify relative optimum desired proposals and indicate the effect of each quality attribute coverage and satisfaction level for agreement between software user and developer company to trading price and quality off. The latter is simple with managing quality attributes or NFRs to choose, reject, and vary the individuals satisfaction levels.

5. Summary and future work

This paper rendered an evaluation model of proposals, mainly focusing on NFRs. As a result, we have confirmed that the model can identify relative optimum desired proposals and indicate the effect of each quality attribute for agreement between software user and developer company. As a future work, we plan the pricing and measurement of the software proposals.

References

Ambreen, T., Ikram, N., Usman, M. and Niazi, M. (2016). Empirical research in requirements engineering: trends and opportunities. *Requirements Engineering*, 1-33.

Andrea, J. (2003). An agile request for proposal (RFP) process. *In Proceedings of the Agile Development Conference, 2003. ADC 2003,* 152-161. IEEE.

Boehm, B. W. and Papaccio, P. N. (1998). Understanding and controlling software costs, *IEEE Transactions on Software Engineering*, *14*(10), 1462-1476.

Charette, R. N. (2005). Why software fails. *IEEE Spectrum, 42*(9), 42-49.

Cleland-Huang, J., Settimi, R., Zou, X. and Solc, P. (2006, September). The detection and classification of non-functional requirements with application to early aspects. *14th IEEE International Requirements Engineering Conference (RE'06)*, 39-48, IEEE.

Galster, M. and Eberlein, A. (2011). Facilitating software architecting by ranking requirements based on their impact on the architecture process. *18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems*, Las Vegas, NV, 232-240, IEEE.

Gross, C. (2009). *Incorporating Software Requirements into the System RFP: Survey of RFP Language for Software by Topic, v. 2.0.* Software Engineering Institute, Carnegie Mellon University.

IEEE (1998). *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications (SRS).*

Information-Technology Promotion Agency Software Engineering Center (2010). Common Frame 2010. Ohm Inc.

Iranian National Standards Organization (2000). *The Public Guidelines for Evaluation of Software Products Quality.*

ISO/IEC (2011a). ISO/IEC 9126-1 Software Engineering – Product Quality - Part 1: Quality Model.

_____ (2011b). ISO/IEC 25010 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) -System and software quality models.

Kiewkanya, M. and Muenchaisri, P. (2005, February). Measuring maintainability in early phase using aesthetic metrics. In *Proceedings of the 4th WSEAS International Conference on Software Engineering, Parallel & Distributed Systems,*1-6. World Scientific and Engineering Academy and Society (WSEAS)

Kou, L., Shin-ichiro, Y., Nobuaki, I, Tomoyuki, T., Ichirou, U., Naoto, N., Hiroaki, S., Shun, K., Yasunobu, K., Kazuo, K. and Yumi, K. (2008). Proposal of RFP Model and Utilization Method based on Stakeholder Value -Report of the Study Group on Applying QFD to Project Planning. *Conference of the Society of Project Management Spring. 2008 fiscal year.*

Laporte, C. Y., Nabil, B. and Mikel, D. (2012). Measuring the cost of software quality of a large software project at bombardier transportation: a case study. *Software Quality Professional, 14*(3), 14-31.

Li, F. L., Horkoff, J., Mylopoulos, J., Guizzardi, R. S., Guizzardi, G., Borgida, A. and Liu, L. (2014). Non-functional requirements as qualities, with a spice of ontology. *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, 293-302.

Ministry of Economy, Trade and Industry, Mitsubishi Research Institute Inc (2010). Product Quality Metrics WG Activities in 2010 Software Metrics Advancement Project.

Misaghian, N. and Motameni, H. (2016). An approach for requirements prioritization based on tensor decomposition. *Requirements Engineering*, 1-20.

Naoki, B., Takashi, H. and Takao, I. (2008). A procedure and a vital point for a IT systems contract to avoid troubles in system integration. Nikkei BP.

Saito, Y., Monden, A., and Matsumoto, K.(2012). Evaluation of non functional requirements in a request for proposal (RFP) in *Seventh International Conference on Software Process and Product Measurement, IEEE.*

Software Engineering Center, Information-Technology Promotion Agency (2008). Outsourcer view guidelines. Ver1.0 outline.

Sugumaran, V. and Drake, J. (2006). Quantifying quality of non-functional quality attributes using customer survey metrics. *Proceedings of the Fifth Workshop on Software Assessment 22.*

Terawaki, Y. and Tamai, T. (2013). A practical approach to quality requirements handling in software systems development. *The Eighth International Conference on System(ICONS 2013).*

Thakurta, R. (2013). A framework for prioritization of quality requirements for inclusion in a software project. *Software Quality Journal, 21*(4), 573-597.

The Iranian ICT Guild organization(IIG) (2008). The guidelines for RFP.

Tomonori, S., Shunichi, S., Naoyuki, K., Akira, O., Haruhiko, K. and Kenji, K. (2008). Design of a tool for measuring quality ratio in a software requirements specification. The Institute of Electronics, Technical Report of IEICE.