A Conceptual Framework for Software Change Effort Estimation during Ongoing Software Development

Haslina Md Sarkan, Mohd Nazri Kama

Abstract

Change effort estimation (CEE) is the process of predicting the work and the hours needed when there is a software change request. Software changes that are not correctly addressed, not complete and not consistent with other artefacts will introduce delay to the project and exceed the original budget. Requirement changes during software development phase are delicate to handle due to the fact that the software artefacts are still undergoing development. Any change would affect the components under development. Software Project Manager (SPM) needs evidence and justifications to decide whether to accept or not a change request. This paper aims to investigate a suitable effort estimation approach that can be applied to requirement changes during an ongoing software development. A conceptual model of the influential effort estimation methods is proposed. It is hoped that the evidence and results from the proposed model is able to convince the client and the developers involved to accept the SPM’s decision.

Keywords: Change Effort Estimation, Requirement Change, Estimation Methods, Effort Prediction, Software Development

1. Introduction

Effort estimation is an ongoing activity throughout a project lifecycle [1], thus a continuous challenge to software developers. A high level estimate is necessary to determine the viability of a project when preparing for effort estimation at the initial stage. Then, a more thorough analysis and estimate must be done and this estimate will have to be constantly refined from time to time. An estimate should not be presented without having any detailed requirements or analysis to back it up. Software effort estimation is not an easy task for both the client and the development team. If the project cost has been underestimated, the project will exceed its total budget.

* Corresponding author. E-mail address: haslinams@utm.my
This will initiate other problems such as software with poor quality or underdeveloped functions and most likely project delay [2]. On the other hand, if a project is overestimated, we will be confronted with risks of the project having redundant resources [3]. Inaccurate estimation is most of the time caused by unsuitable estimation technique applied to the project. In the real world, there is no one perfect estimation model for all types of software project. Therefore, an accurate estimation is necessary to manage requirement changes for a certain type of project [1][4] and this issue has encouraged the author to pursue this topic.

2. Related Works

Effort Estimation in Software Development

Most work related to effort estimation in software projects was done during the planning stage [5] and some focused on effort estimation based on early phase development activities [6]. These estimates are likely to be inaccurate because at the early stages, there are not enough information about the project yet. During the software development phase, the classes in the class artefacts have not been developed completely, unlike in software maintenance phase. This is because some classes in the class artefact are still under development or are partially developed. [7] worked on an impact analysis approach for the software development phase. [8] later extended this work by automating the approach called CIAT (Change Impact Analysis Tool). An automated prototype tool was developed based on CIAT where there is the element of integration between static and dynamic analysis techniques that can overcome the challenges on impact analysis implementation using static analysis and dynamic analysis techniques. A new change effort estimation model using static and dynamic analysis technique has been introduced more specifically for ongoing software development to address the problem of accepting or not request for changes [9]. Their work is based on algorithmic approach of software change effort estimation. However, algorithmic approach even though is most of the time accurate, at the same time it can be very rigid.

3. Literature review

Conceptual framework

Conceptual framework is the basis for a research problem [10]. This paper aims to propose a conceptual framework that can answer our research question which is “What are the effort estimation methods that can be applied to requirement changes during an ongoing software development?”. Linked to the problem statement, the conceptual framework makes a connection to present a specific research question that drives the investigation being reported [11].

Software Requirement Change in ongoing Software Development

Requirement changes during software development phase are delicate to handle due to the fact that the software artefacts are still undergoing development. Any change would affect the components under development. More effort would be needed to cater for the changes. Dateline and budget are compromised. On the other hand, the client’s request
for change must be attend to. Software Project Manager (SPM) needs evidence and justifications to decide whether to accept a change or not. Client and project manager during ongoing software development phase are the main stakeholders.

No matter how well software requirements have been prepared in advance, it is not a surprise if there will still be changes in the requirements. We cannot set a rigid specification at the beginning of the project. Need to be more agile in order to be better prepared. The activities that are usually spared from client’s interference are the software design and implementation phases as they are very technical. Therefore less change request is expected then. However, as soon as the client starts to have a hand trying out the software, the change requests will start coming in [12].

*Software Impact Analysis*

SWEBOK V3 describes impact analysis as the manner in which to conduct, cost effectively, a complete analysis of the impact of a change in existing software [13]. [14] defined Impact Analysis as the process of identifying probable outcomes of change or estimating what is needed to be modified in order to realize a change. Basically it can be described as the identification of the software artefacts that would likely be affected by a change.

*Different types of Effort Estimation Approach*

Algorithmic estimation models are made up of mathematics and statistics equations. They are fixed and predefined. An example, COCOMO II [15] is derived from formal method and uses diverse parameters together with a formula to estimate effort needed in software development. The model consists of several components or issues such as non-sequential and rapid development process models [16]. It takes quantitative and qualitative weighted characteristics as input and effort estimation as the output. Effort is expressed as Person Months (PM) which is the quantity of time one person spends working on the software development project for one month. This number is not taking into account holidays and vacations. It considers weekend time off [8].

Non-algorithmic models characteristics include learning, inferring and analyzing past projects. Analogy Based Estimation [17] is a simple and flexible. Therefore it is popularly used to hybridize effort estimation models. The way it works is by estimating current project compared to previous projects that are finished. The similarity level of 2 projects are compared and measured. If the selected projects have remarkable high or low productivity, the estimates should be adjusted towards productivity values a number of average projects. The size of the adjustments depends on the anticipated accuracy of the estimation model [18].

4. **Methodology**

The methodology chosen to come out with the conceptual framework is based on the following figure:
Figure 1: Steps to generate the Conceptual Framework

The first step is to pick a topic and then formulate a research question. The interest is on the existing available effort estimation techniques and looking into their pros and cons. The research question that the author would like to ask is, “What are the suitable effort estimation techniques that can be applied to requirement changes during an ongoing software development?”

The second step is to identify the suitable keywords to probe on the research question. The following step is to decide where to obtain the relevant evidence needed to answer the above question. This search process was done using the electronic literature databases by manual search [19].

Step number four is to isolate the important variables found during literature and to figure out how they are interconnected. In some papers, the abstracts were well written and contained the variables which are useful for the research. If these are not available, go to the methodology section or the results and discussion paragraph to quickly identify the variables of the study and the significant findings.

Lastly the conceptual framework can be generated by mixing the variables from the electronic articles read. The reference for the construction of the framework is the problem statement. This study attempts to answer the above question that no other researchers have explained yet and it should contribute to the knowledge gap.

5. Results

The conceptual framework in Figure 2 is the result obtained. The stakeholders perspectives, the project risk aspects, the change control and change management process have been taken into consideration. The software impact analysis approach for handling the software requirement change has been identified to be of important approach to answer the research question. From there, the estimation for the time, cost and effort can be done. Two approaches in software effort estimate are available and the study indicated that the combination of non-algorithmic approach can help to lessen the rigidness of the
algorithmic estimation approach. The new approach will eventually be evaluated and will propose a decision that can help the SPM to better decide whether to accept the change or not.

![Diagram of Conceptual Framework for Change Effort Estimation during ongoing Software Project]

**Figure 2:** Conceptual Framework for Change Effort Estimation during ongoing Software Project

6. Conclusion

During the software development phase, the classes in the class artefacts have not been developed completely because some classes in the class artefact are still under development or are partially developed. This paper presents a conceptual framework that integrates software effort estimation and software impact analysis in order to solve the software change requirement problem during ongoing software development. This framework will be able to help the software project manager to make the best decision on whether to accept or not a requirement change based on the evidence generated from this framework. The future work will be to implement this framework and to study its effectiveness when implemented on an ongoing software project.

References


