Software engineering and their conflicts in development of software

Godfrey D. Beray1, Sabah Mohammed2)

Abstract

In software engineering there are many different uses that will be mainly helpful in the development of suitable software that is required by the particular user in order to achieve the reliability of the user. So natively we used the technique called we use an adaptive testing strategy for testing software components. This strategy (AT_RLSEc with c indicating components) applies a recursive least squares estimation (RLSE) method to estimate parameters such as failure detection rate. It is different from the genetic algorithm-based adaptive testing (AT_GA) where a genetic algorithm is used for parameter estimation. By using this method we will not be getting the desired result so there will be some defects in the software. To overcome these defects we use qualitative and quantitative approaches, the points until the separately generated results are interpreted. It will show efficiency of the conflicts and risks that are presented that are occurred by using the existing approach. So it will be helping the developers to give perfect software that is required by the user.

Keywords: adaptive testing strategy, recursive least squares estimation, genetic algorithm-based adaptive testing, Qualitative and quantitative approaches

1. Introduction

Software Engineering is the study of design, development and maintenance of software. In other words it is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software and an engineering discipline that is concerned with all aspects of software production. Communication skills, team dynamics, working with a "customer", and creativity are also important factors in the software engineering. It is important because of the large expensive software systems [1-3].

Identifying and resolving development processes of sharing project by software developers arise in collaborative development and can slow progress and decrease quality of the assurance development of shared project. Traditionally developed and design crystal tool as a speculative analysis in real time application development. Crystal, a publicly available tool that helps developers identifies, manage, and prevent conflicts.

Received (February 20, 2013), Review request (February 21, 2013), Review Result (1st: March 08, 2013)
Accepted (June 10, 2013)

1Western Institute of Technology, Philippines
email: jetdario@gmail.com

2(Corresponding Author) Department of Computer Science Lakehead University Ontario-CANADA
email: sabah.mohammed@lakeheadu.ca

ISSN: 2383-5281 AJMAHS
Copyright © 2013 SERSC
Crystal uses speculative analysis to make concrete advice unobtrusively available to developers. Qualitative and Quantitative approaches have typically been combined by using them side-by-side or sequentially, until the point when the separately generated results are interpreted and conclusions drawn. In this we propose to develop a mixed method to describe the analysis of the qualitative and quantitative methods. The mixed methods research purpose most frequently served by integration of analyses is initiation, that is, to be provocative and bring fresh perspectives through contradiction and (intended or unintended) discovery of paradox. Experimental results show efficiency of the conflicts and risks present in the shared project development [3-7].

2. Risk Management in Software Engineering

The management of a risk is the important future in throughout the software development life cycle. A risk is a potential future harm that may arise from some present action, such as, a schedule slip or a cost overrun. “Risk in itself is not bad; risk is essential to progress, and failure is often a key part of learning. But we must learn to balance the possible negative consequences of risk against the potential benefits of its associated opportunity.” Risk management is a series of steps whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful software operation or a major source of expensive rework [8][9].

3. The Risk Management Process

The risk management process can be divided into two phases. Those are risk assessment and risk control. The risk assessment further broken down into risk identification, risk analysis, and risk prioritization. Like that risk control also divided into risk planning, risk mitigation, and risk monitoring.

![The Risk Management Cycle](Fig. 1)
4. Software Risk Management

There could be risk associated with the every software project, the main goal is to identify and manage those risks. The most important risk management tasks are risk index, risk analysis, and risk assessment [10][11].

1) Risk Index: Risk index is the multiplication of impact and probability of occurrence. Risk index can be characterized as high, medium, or low depending upon the product of impact and occurrence. Risk index is very important and necessary for prioritization of risk.

2) Risk Analysis: The risk analysis is used to identify the high risk elements of a project. The main purpose of risk analysis is to understand risks in better ways and to verify and correct attributes. A successful risk analysis includes important elements like problem definition, problem formulation, data collection.

3) Risk Assessment: It integrates risk management and risk analysis. Risk assessment requires correct explanations of the target system and all security features. It is important that risk deferent levels like performance, cost, support and schedule must be defined properly for risk assessment to be useful.

5. Strategies for Risk Management

Throughout the software development process, there are various strategies for risk management could be identified and developed. The risk strategies could be divided into three classes namely careful, typical, and flexible. Generally, careful risk management strategy is proposed for new and inexperienced organizations whose software development projects are connected with new and unproven technology; typical risk management strategy is well-defined as a support for mature organizations with experience in software development projects and used technologies, but whose projects carry a decent number of risks; and flexible risk management strategy is involved in experienced software development organizations whose software development projects are officially defined and based on proven technologies.

6. Related Work

The identification of the various factors that have an effect on software development is of prime concern to software engineers. The specific focus of this paper is to analyze the relationships between the frequency and distribution of errors during software development, the maintenance of the developed software, and a variety of environmental factors. These factors include the complexity of the software, the developer's experience with the
application, and the reuse of existing design and code. Such relationships can provide an insight into the characteristics of computer software and the effects that an environment can have on the software product. Such relationships can also improve the reliability and quality with respect to computer software. In an effort to acquire knowledge of these basic relationships, change data for a medium-scale software project were analyzed.

Developing software is a relatively new area of enterprise that bears little resemblance to other engineering disciplines. Although the term software engineering is widely used throughout the business, the act of creating a new piece of software can hardly be compared to the design and construction of a new building or bridge. Computer scientists are still struggling after 30 years to define software engineering and to find the right combination of techniques, procedures, and tools that assure success in development of large complex systems.

7. Conflict

Conflict is a natural disagreement resulting from individuals or teams that differ in attitudes, beliefs, values, or needs. As human beings interact in organizations, differing values and situations create tension. One prominent scholar of conflict listed the following issues involved in conflicts:

- Control over resources;
- Preferences and nuisances in which the tastes or activities of one party impinge upon another;
- Values, when there is a claim that a value or set of values should dominate;
- Beliefs, when there is a dispute over facts, information, reality, and so forth;
- The nature of the relationship between the parties.

The traditional view of conflict was that it was a negative phenomenon and a serious threat to effective team performance. However, this is not a universally held opinion amongst conflict researchers, and it has been challenged: Scholars have argued that more focus should be placed on the form the conflict takes. The point is that conflict per se need not be a negative force. Indeed, some have argued persuasively that when positive conflict is recognized, acknowledged, and managed in a proper manner, personal and organizational benefits can accrue.

8. Constructive and Destructive Conflict

Conflict has been given a bad name by its association with disruption. However, as was mentioned in the previous section, several researchers have argued that conflict need not be a negative force and that it is often the case that it is the form the conflict takes that determines how much damage is caused that is, whether it is a constructive or destructive conflict.

Constructive conflict is characterized by cooperation and flexibility. The principal focus is on trying to
achieve a solution between struggling parties that is mutually satisfactory to everyone. However, destructive conflicts are more concerned with power struggles and personal antagonisms and are characterized by domination, escalation, retaliation, com petitions, and inflexibility. When a conflict spirals out of control, it runs the risk of becoming destructive. When this happens, participants lose sight of their initial goals and focus on hurting the adversary.

**Existing System:** CAT theory tries to mimic the usual assessment procedure followed by a human teacher. That is, it first gives the student an item of medium difficulty. If the student answers correctly, it then administers an item that is a little more difficult and if not, it administers a less difficult item. This process should be repeated until the teacher considers that he/she has enough evidence to determine the student’s knowledge level. In CAT theory, this process has been automatized. Items are posed one by one. After posting an item, a temporary student knowledge level estimation is achieved. In terms of this estimation, the next item to be posed is chosen in such a way that this estimation will be more accurate. In more precise terms, an adaptive test can be seen as an iterative algorithm that starts with an initial estimation of the student's knowledge level, and comprises the following steps:

1. All the items that have not been administered yet are examined to determine which the best item to ask is next, according to the current estimation of the student’s knowledge level.
2. The item is asked, and the student responds.
3. According to the answer, a new estimation of the knowledge level is computed.
4. Steps 1 to 3 are repeated until the test stopping criterion defined is met.

IRT postulates that there is a relationship between the student’s knowledge level and the probability of successfully answering an item. This interdependent relationship is probabilistically expressed by means of a function called Item Characteristic Curve (ICC). Accordingly, this function collects, for each knowledge level, the probability that a student with this level will correctly answer the item. If this probability function is available for every item of a test, the student’s knowledge can be directly inferred. In CAT theory, IRT is used to estimate the student’s knowledge level, in order to determine the next item to be posed, and to decide when to finish the test. This theory ensures that the student knowledge estimations obtained do not vary in terms of the items used in the estimation process. The models most commonly used as ICC functions are the family of logistics models of one (1PL), two (2PL) and three parameters (3PL). All of them can be expressed by the following equation:

\[ P(u_i = 1 | \theta) = c_i + (1 - c_i) \frac{1}{1 + e^{-(\theta - a_i) / b_i}} \]

where \( c_i \) is the guessing factor, \( a_i \) is the item difficulty and \( a_i \) is its discrimination factor. The guessing
factor is the probability that a student with no knowledge at all will answer the item correctly. The difficulty represents the knowledge level in which the student has the same probability of passing or not the item, besides the guessing factor. The discrimination factor is a value proportional to the slope of the curve and represents a measure of how the item contributes to estimating the knowledge level. The formula just shown in Equation 1 expresses the 3PL model. When the guessing factor is always assumed to be zero, the 2PL model is obtained. If, in addition, we consider the discrimination factor equal to 1, we obtain the 1PL model.

**Proposed System:** By combining the qualitative and quantitative approaches, the points until the separately generated results are interpreted. It will show efficiency of the conflicts and risks that are presented.

9. Qualitative and Quantitative Approach

The multiple research methods and tools of qualitative experimental and non-experimental are essential for researchers. The quality of a program is limited when we are not using by combining both the methods. The elements of qualitative and quantitative approaches are combined into a unique design to undertaking as a mixed method. The way in which the mixed methods might be differentiated at which the elements of qualitative and quantitative approaches are integrated together. The purpose of using both methods to finding the corroborative evidence from different methods.

10. Strategies for Integration

There are four strategies for combing of both qualitative and quantitative approach. Those are: (a) Data Transmission (b) Typology Development (c) Extreme Case Analysis (d) Data Consolidation.

**Data transmission:** The one form of data is transformed into other form is known as data transmission.

**Typology Development:** Classification of data from one set of data is applied to another set is known as typology development.

**Extreme Case Analysis:** The outliers or residuals revealed by one analysis are explored using alternative data or methods are known as Extreme Case Analysis.

**Data Consolidation:** To create variables for use in further analysis is known as data consolidation.

Combination of mixed method analysis is most obvious when data from one type is used in analysis of other type. The strategies of integration might be used in the context of expansion, development and complementarity. But the integration with corroboration is inconsistent. The popular association of mixed methods with corroboration and consequent lack of consideration of integrative strategies; and the view that integration or synthesis of results is an intellectual or ideologically driven activity. To achieve integration of
data analysis, it requires the capacity to visualize what might be possible to set the new paths. Integration is greatly helped by data handling technology to facilitate the process

11. Two Major Routes to Integration in Analysis

Propose in terms of data handling, there are two major routes to integration that underlie the various strategies are

1) Combination of data types with in an analysis, which is used for both statistical analysis and comparison of coded narrative material. This could occur through by combining both numerical and textual data. For example a combination of survey and interview.

2) Conversation of data from one type to another type for analysis. The conversation of qualitative codes to codes used in a statistical analysis through the contribution of qualitative analysis.

12. Using Software to Combine Numeric and Text Data for Analysis

The data management is to combine mixed forms of data and procedures for working with them. The advent of text-handling spreadsheets and databases and, in particular, of text analysis software, has heralded solutions to these data management problems, and opened up new possibilities for more rigorous and/or deeper analysis of this type of data. They have not necessarily solved the theoretical issues which could arise when different forms of data are combined.

13. Results

Here we show that how our existing system will be working and how it will be giving the conflicts the software. So we show that our proposed system will be saying that this technique will be giving the desired software and also without any of the defects in it. So here in his results we say that proposed approach is far better when compared to the native approach.

14. Conclusion

In this paper we show that in software engineering there is need of giving the desired software that will be mainly helpful for the good relation between the user and the developer. Natively they will be using adaptive testing strategy that will be not satisfying the needs and there will be some conflicts that will be raised in the system. by using our proposed approach we will be rectifying those conflicts and giving a good software as a result.
References


