Application of Evolutionary Algorithm to find the trade-off between Complexity of Software and its Deliverability

EXECUTIVE SUMMARY

The term software complexity can be categorized in two ways. One is the code complexity which is not visible to the user and is the second one is the user interface complexity which is visible to the user. This research work is specifically about the second one that is user interface complexity aspect of any software.

The complexity of software with respect to the user varies from person to person. The same software at the same time could be a difficult to a person to work upon, very difficult for another person and at the same time could be simple for another person. This complexity or the comfort level varies because of social, economical and technical reasons. For example, an online shopping from a website could be a simple task for a person who has some prior knowledge about computer and internet but at the same time it is a complex procedure for those who are not familiar with the internet and computers. Similarly, if talk about the ERP software, that are being used by a lot of companies are having a different complexity level among the user in the same organization.

The high complexity of usage affects the market status of the software as the user finds it difficult to operate or to work upon and as a result, the software fails to deliver its maximum value to the organization which is the deliverability factor of the software.

It is the usability feature which plays a major role in having the product more sellable, but at the same time it should cater to all high level needs of the consumer. Although, there are number of companies developing certain guidelines about software development process, the major focus is on user-centred application development, software is evaluated with various tools like Cognitive tools and Complexity Matrices to find out the degree of acceptability amongst the users.

The deliverability of the software can further be described as the usability aspect of the software which is inversely proportional to the complexity of the software. The main problem arises to maintain a balance between complexity and deliverability, as both the quantities are very much inter-related, it is very difficult to raise the deliverability without increasing the complexity of the system.

This research is basically focuses on the user interface complexity of the software, various parameters that affects the complexity of the software and its trade-off relationship with the deliverability along with the usability parameters.

There are many researches and work done that shows that software's with very high complexity are very low in deliverability and hence are not popular in the market. That is, as the complexity of the software raises, the deliverability drops significantly, but in this research work, this trade-off relationship between the complexity and deliverability has been established with the concept of evolutionary multi-objective optimization which is novelty of this work and further there is framework developed using the fuzzy rule based systems (FRBS) with the help six identified parameters of software usability.

The framework has been verified using the hypothesis testing and a mathematical model has been developed for the support of the framework using rough set theory.

Step 1- Study of various software complexity issues, software deliverability issues and usability aspects of software.

In this step, the analysis has been done of the basic concepts of software complexity and by referring the various literatures available by the various researchers as well as big companies; the identification has been done about the fundamental criteria to analyze the complexity and deliverability of any software from user point of view. There are various models and schemes developed for checking the software quality improvement in terms of Flow of Data, Mean Time to Repair (MTTR) in addition to the Mean Time between Failure (MTBF), but there exists a huge gap in terms of usability of the software.

Usability of the software refers to the ease of use in driving the desired result. The ISO/DIS defined the term usability as “Degree to which a software package can be utilized by a specific user to attain specific objectives with maximum efficiency, satisfaction as well as effectiveness in a precise usage circumstance”.

As functionality of the software increases the deliverability value to the client also increases but at the same time, complexity also increases. High software complexity leads to various issues such as lack of Adoption, more end-user training, more software Technical support, less likability of software, low user’s performance and lack of customer satisfaction etc.

Complexity analysis involves breaking down a user task into a set of constituent steps and then calculating a complexity metric for each step in the task relative to the type of user.

The software deliverability can be considered as the degree of the usability factor provided to the user of the system by the software. The software deliverability should be high in order to attain maximum value from the software. The business value of any software is highly affected by the software deliverability factor which later imposes several constraints on the software developers.

The term software usability is actually the level of comfort or the ease with which a user can work on the software. As the discussion has been done earlier that the software complexity varies from person to person and from software to software in a well defined and constrained scenario, the software usability also varies accordingly. Higher the complexity, lower will be the usability aspect of that particular software product.

Step 2- Study of Evolutionary multi-objective optimization to establish the trade-off between software complexity and its deliverability.

In this step, various research papers and literature that focuses on the EMO has been analyzed and studied thoroughly in order to achieve the objective of this research work. One cannot identify a single point of solutions to optimize each objective simultaneously.

The job of handling multiple objective problems is known as multi-objective optimization. The trade-off surface's convexity is based on the fact that in what manner the objectives are scaled. As a result, the look for the best or an optimal solution is discarded from the observation in the case of one objective problem. In general, simplifying the multi-objective problems can be seen by decision making as well as searching. The primary step towards solving a multi- objective problem is the Pareto Optimality.

Step 3- Establishment of trade-off between software complexity and deliverability using EMO.

Here, there are two conflicting situations: software complexity and deliverability. In this particular case there is a need to increase up to maximum the deliverability and software's usability and minimize the software's complexity. A set of software and two set of questionnaires [Annexure 2, 3] has been developed for private and government financial institutions with varying range of complexity level. The software's complexity level has been performed by the process of complexity analysis. This particular software application has been used by three individual banks in the city named Lucknow.

The data has been collected from the employees after filling the questionnaires over a period of time which is basically the ratings the experiences that they faced during working on that software and then this data has been feed or entered into the JAVA based open tool known as 'GUAJE' which works on the basics of EMO and the results that are generated are promising. After analysis the results obtained from the tool it is concluded that software with higher usability factor or lower complexity level are much popular among the user which results into the higher acceptability of that particular software.

Step 4- Development of a framework using FRBS for the quantification of software complexity and its usability.

In this step, a framework has been developed using fuzzy rule base systems. The development of FRBS framework starts with Fuzzy Inference System (FIS). The input to FIS may be fuzzy or crisp but the output from FIS is always a fuzzy set as discussed in 3-Block Diagrams of expert systems. The basic step in FIS is to convert the crisp set into fuzzy input. This input is fed to the Rule Base which consists of Knowledge Extractor to generate the complex output set. Later on it is defuzzified to produce the crisp output of a particular event. Since, the calculation of the trade-off value for a software module is needed; the process begins with the basic building block of categorizing the clusters of software module.

Three rules have been identified that are applied to the framework. Also, there are six parameters of software usability that has been identified in the previous steps are used as inputs and the result has been taken in the form of software popularity.

This framework basically quantifies the value of software complexity, usability and popularity. The rule description of software complexity and usability has been done using Mamdani FIS. Based on the linguistic set, the output is classified into three broad categories: High, Low and Moderate ranging from 0, 1 and 2.

Step 5- Verification of the framework using hypothesis testing and the mathematical model in support of the model using rough set theory.

In this step, hypothesis testing has been used for the verification of the framework that has been developed. Also, there has been a mathematical model for the support of the framework using the rough set theory. The hypothesis testing is a very prominently used method of verification that is used in statistics.

In statistical hypothesis testing, a statistical inference is carried out based on the data that has been gathered from a research or survey carried out. If the occurrence of the result is predicted as unlikely according to the pre-calculated threshold probability also referred to the significance level, then the result is called as statistically significant in statistics. Ronald Fisher was the person who initiated the concept of "test of significance". The tests of significance are used to determine that which outcomes of a research will direct to a denial for a pre-specified significance level of the null hypothesis. This provides contribution in deciding whether the results contain sufficient information or not in order to cast disbelief on predictable insight, to establish the null hypothesis, considering the fact that the usual perception has been applied.

In order to analyze the relationship between the complexity of the software based on the various parameters like context shift, navigational guidance, input parameters and system feedback, The concept that is used here is rough set theory. Since two decades this approach is frequently used in the analyzing the relationship between various parameters.

Rough set methodology uses the concept of decision table consisting universe of discourse showing a relationship represented by two types of attributes: condition attribute and decision attribute. Basically it gives a concept of relationship between attributes called lower and upper approximation.

Step 6- Conclusion and future scope.

Complexity of the software varies from person to person. In an observation, it has been analyzed that when the software complexity increases, the usability aspect of the software drops significantly. The novelty of the current work is use of soft computing techniques in analyzing the trade-off between the complexity and the deliverability of the software. This research work has also involves the development of a framework using Fuzzy Rule based System (FRBS) for quantification of software complexity and usability aspects. In the final phase, the development of Fuzzy Inference System (FIS) by means of Expert Knowledge Base was done. In future, the work can be extended by enhancing the framework by identifying more usability aspects of the software and the application of the framework can be possible in various types of software to analyze the complexity aspect of the software to determine its future market potential.