

Review paper on Predicting Software maintainability using fuzzy logic

¹ Neha Prashar, ²Chhavi Rana
¹Student, ²Assistant Professor
¹UIET, MDU, Rohtak, Haryana

Abstract - Ascertaining the excessive time, efforts and budget spent in the maintainability phase of a software development Maintainability has always drawn the attention of researchers. Software Maintainability is the measure of ease with which a software upgrades, enhances or debugs itself. The software which adapts itself quicker to the upgrades or changes is more maintainable compared to the softwares which takes more time to adapt themselves. The relation between the amount of time taken by the developer to change the software and maintainability is inverse.

Introduction

If we google the term software maintainability, the most common definition found is "The degree to which an application is understood, or enhanced. Understanding software maintainability allows organizations to identify improvement areas as well as determine the value supplied by current applications or during development changes" The definition sounds a bit complex. Practically speaking, we have a body and a mind. It is easy to change our mind than body. But for some people changing mind is harder than changing their body. Similarly, some software are harder to change compared to others. Like we say some things are more flexible than others, some people adapt themselves more easily in new environment compared to others. So, what happens when a piece of software needs to be changed we need to change the laws, infrastructure and entire regulations. In this competitive world, software will fall down if it will not adapt to users' changing demands. This is why software maintainability matters. Maintainability is the measure how easy or difficult it is for a software r change. We study for the things which makes the software difficult to change. Software is considered unmaintainable when the costs of keeping it operational are very high

SOFTWARE EVOLUTION:

It can be further categorised into:

TRADITIONAL MAINTENANCE

When the software module is delivered to client, and client after using the product find some bugs in it. The effort than taken to repair that module by the developer is called traditional or corrective maintenance.

ADAPTIVE MAINTENANCE

In the rapidly developing digital world, any upgrade to modify the software product in the new environment forms adaptive maintenance.

PERFECTIVE MAINTENANCE

There are clients that want certain changes in the software according to their personal requirements, the modification of module in that case forms perfective maintenance.

PREVENTIVE MAINTENANCE

This is the most important type of maintenance .While developing the software module, there are certain bugs that remain hidden but later become effective and hinder the work later. Modification of module to correct those faults is called preventive maintenance.

NEED OF SOFTWARE MAINTENANCE

- To evacuate bugs
- Change the structure
- To execute improvements
- To diminish time and endeavors spent in looking after programming
- To lessen the costs as it frames the 90% of the product advancement cycle
- To structure billions of unstructured lines of codes

FUZZY LOGIC

Before defining how is fuzzy logic used in the maintainability, it is important to understand what fuzzy logic is. Contrary to conventional logic where the truth table strictly contains two values true and false i.e 0 and 1. Fuzzy logic use partial values i.e range of values between 0 and 1 .For example Is water hot? The answer to the question can have several values like

No, Slightly hot, Very hot etc.

fuzzy logic is also called multivalued logic.

Why Fuzzy Logic: _____

- It can work with non linear systems.
- Output produced is smooth function
- Any logical system can be fuzzified.
- It is robust.
- There is no limit on the number of inputs processed.
- The main advantage is that it does not need lots of data to train .interpretability and simplicity, as it is used to “compute with words”.
- Simplicity of fuzzy logic is the another big advantage.
- When a user need to add new rules , all he has to add new rules instead of retraining the entire system.
- Fuzzy Logic allows you to model in a more intuitive way complex dynamic systems

How is Fuzzy Logic used

1. Determine the settled criteria: what would I like to control? What do I need to do to control the framework? What sort of output I need ? What are the ways a framework can fizzle?
2. Discover the connection amongst information and yield
3. Utilizing the run base structure of FL, separate the control the issue into a progression of IF X AND Y

At that point Z decides that characterize the coveted framework yields reaction for given framework input conditions the number and many-sided quality of standards relies upon the quantity of info parameters that are to be prepared furthermore, number fluffy variable related with every parameter. Utilize one variable and one subsidiary.

4. Make FL enrollment works that characterize the importance of info/yield terms utilized as a part of the tenets.
5. Make the essential pre and post preparing FL schedules if executing in programming, generally program the principles into the FL equipment motor.
6. Test the framework, assess the outcomes, tune the principles and participation capacities, and retest until acceptable outcomes are acquired.

LITERATURE REVIEW

UTTAMJIT KAUR AND GAGANDEEP SINGH

Real maintenance of a software begin after delivering it to the client. He stressed on the important points regarding software maintenance i.e.: database size, system age, maintenance budget, system size, staff size or restructuring for change. Also they presented several ways to reduce cost and efforts involved in software maintenance. Another important point to ensure is upgrading and continuity of the services to the clients.

JANE HUFFMAN HAYES et. al.

Introduced the observe-mine-adopt paradigm that assists organisations in making improvements to their software developments processes without committing to and undertaking large scale sweeping organisational process improvement, Specially, the approach has been applied to the improving the software practices focused on maintainability. OMA has been studied experimentally using two project studies and a Web - based health care system which is mentioned by a large industrial software organisation.

R Malhotra

She observed the trends in the field of software maintainability for years. Based on her studies she declared that usage of "machine learning algorithm " in predicting software maintainability has increased since 2005. Also there has been a significant increase in the use of the public dataset. Moreover, further experiments should be done on the use of open source datasets with hybrid algorithms.

Warren Harrison

When there is a bug in the software or when a software needs to be updated, there are two methods : either the software has to be modified or the developer should rewrite the entire module. To determine which of the above mentioned method is suitable for any particular software module, Warren developed a software maintenance model based on objective decision rule. Identification of change prone modules at an early stage was suggested in his paper.

Megha and Harish Mittal

The ease with which a software can be modified against faults, bugs and how quickly it adapts itself to new environment forms important attributes in software maintainability. Undeniably, more than 60 % of the entire time and efforts required in developing any software module spends in maintainability, leaving alone the expensive phase of this step. The major focus in this paper was producing a model that can significantly reduce the time and efforts of the maintenance phase of any software development cycle.

Ritika Vern

Maintainability is an essential attribute for all the software quality models. Maintainability is not only the most expensive phase of the development cycle but also it requires great efforts. Although its evaluation is complex several techniques like fuzzy technique, AHP technique were mentioned in this research paper.

Scott. L. Schneberger and Ephriam R. Mclean

trusted that the biggest single life cycle PC cost has been for keeping up data framework programming. PC framework has changed from incorporated PC models. Two diametrics which were engaged in his paper were segment effortlessness and framework many-sided quality. Littler segments are anything but difficult to manage contrasted with the general framework. This

paper offers a point of view on estimating and investigating the rising innovation of appropriated registering, including suggested zones of particular research required in light of support.

Conclusion

This paper reasons that upkeep is a tedious and costly period of SDLC. We have looked into crafted by numerous scientist which comprehends the distinctive methods to keep up the product. We likewise reasoned that deliberate change of upkeep requires information of the current issues, the capacity to tailor existing approaches, and a promise to screen the adequacy of these approaches.

References

- [1] C. van Koten 1 and A.R. Gray „An application of Bayesian network for predicting object-oriented software maintainability“ in 2005 Department of Information Science,University of Otago, P.O.Box 56, Dunedin, New Zealand
- [2]. K.K. Aggarwal et. al. „Measurement of Software Maintainability Using a Fuzzy Model“ Journal of Computer Sciences 1(4):538-542, 2005

