Research Proposal: Digging deeper into Software maintainability

Gautam Kumar

January 25, 2016

1 Introduction

Software maintenance can be defined as any modification to the product after delivery to the customer. Maintenance is considered the most expensive portion of a project's life-cycle and is known to take up around 40% to 80% of the total costs of the project according to a paper by Robert Glass [Gla].

A 1999 study[ZP] after analysing 32 factors found that Software complexity, Programmer skill, testability and test coverage to the major factors affecting Software reliability and maintainability.

Considering the critical role played by maintainance in a software project's lifecycle I would like to focus my research on method and metrics for evaluating Software models and architectures to asses their maintainability and related quality attributes.

2 Description

My research into Sotware maintainability would be focussed on evaluating existing software models and architectures using commonly used metrics such as HPMAS described in the paper "Using Metrics to Evaluate Software System Maintainability"[CALO]

A paper of interest in this regard is "Long-term Life Cycle Impact of Agile Methodologies" [KMLS+] Grace Lewis mentions that maintainability as a quality attiribute has to be baked into the system because of the small role that software architecture plays in agile methodologies.

3 Relevance

Maintainability is a core quality attribute in many software models and has far reaching consequences into the success of any project. I feel that this research aligns well with some of the learning objectives of the class such as "Research best practices and methods to educate a technical audience.” and “Create & evaluate models and other artifacts used to express software design ”. I hope to better understand the value and effects using various software quality models at the end of this research work.

Annotated Biblography

[CALO] Don Coleman, Dan Ash, Bruce Lowther, and Paul Oman. Using metrics to evaluate software system maintainability. 27(8):44–49.

[DH] Sayed Mehdi Hejazi Dehaghani and Nafiseh Hajrahimi. Which factors affect software projects maintenance cost more? 21(1):63–66.

> This paper analyses the effects of various factors such as Software Experience, Application understanding, Programming language etc on the maintenance costs of a project.

[Gla] R.L. Glass. Frequently forgotten fundamental facts about software engineering. 18(3):112–111.

> This paper offers an interesting analysis of the relationship between various attributes of software project such as Reliability, Maintainability, Efficiency and Complexity.

[KMLS^{+}] M. Kajko-Mattsson, G.A. Lewis, D. Siracusa, T. Nelson, N. Chapin, M. Heydt, J. Nocks, and H. Snee. Long-term life cycle impact of agile methodologies. In *22nd IEEE International Conference on Software Maintenance, 2006. ICSM ’06*, pages 422–425.

> This paper offers unique perspective of analysis into Agile methodologies by summarising the positions of proponents and opponents of Agile who participated in a panel discussion.

[RJ] Ekaterina Razina and David Janzen. Effects of dependency injection on maintainability. pages 7–12.

> This paper provides context into the relationship between testability and maintainability. A common factor with both attributes as mentioned is lower coupling of software components which eases testability and while also improving maintainability.

[ZP] Xuemei Zhang and Hoang Pham. An analysis of factors affecting software reliability. 50(1):43–56.

This paper presents 32 factors that affect maintainability & reliability of software applications and offers analysis which finds that software complexity, programmer skill and testing effort to be some of the most significant factors that affect maintainability and reliability.