**Name: Gaurav Kumar Singh**

**Registration Number: 19BCE2119**

**Course: Software Engineering (E2)**

**Digital Assignment 2**

**Paper-I**

An industrial case study on the use of UML in software maintenance and its perceived benefits and hurdles *(2018) Ana M. Fernández-Sáez & Michel R. V. Chaudron & Marcela Genero*

**Abstract**

UML (Unified Modelling Language) is one of the key modelling languages in the field of software engineering. UML provides a standardized way to visualize the design of the system. UML’s origin is co-related to the OOPs (Object Oriented Programming). It consists of several diagramming techniques and methods which make it easier for the developers to visualize various aspects such as scale, actors, timeline etc. of the project. All this directly impacts the software maintenance strategy of the project as well. Software maintenance takes one of the biggest shares from the timeline for development of required software. Perception of purpose of UML may vary from person-to-person. UML can be considered as communication tool, overview, prototype guideline, testing guideline or can be for one’s own understanding as well.

Scale of software projects can grow exponentially in very small amount of time and after a certain threshold it becomes very difficult to keep track of all the developed and developing modules of the project, therefore proper maintained documentation of the project is important to be able to maintain the software more easily. UML is one of the most efficient ways to quickly and accurately visualize different aspects of the project/ module. It requires proper experience and training to be able to develop well formulated UML for its referencing to be effective to be used for software maintenance of the project.

**Paper-II**

Analysing Forty Years of Software Maintenance Models *(2017) Valentina Lenarduzzi, Alberto Sillitti, Davide Taibi*

**Abstract**

Software maintenance mechanism has naturally evolved in the last forty years to adapt to the changing software development needs, models, programming languages, scale etc. Software maintenance tools are using increasingly advanced prediction models to cater to this rapidly growing industrial sector. In this paper, the author uses ISO/IEC 25010:2011 definition and reports on models for analyzability, modifiability, modularity, reusability, and testability using literature reviews following systematic snowballing.

The following points were concluded by the end of the research:

1. Continuous increase in number of papers: This implies software maintenance is still an increasingly popular research topic. More number of software maintenance models are being proposed everyday each with their own set of pros and cons.
2. Continuous increase of the complexity of the models: The normalized complexity of every sequentially proposed software maintenance model is increasing due to discovery of improved mathematical techniques, visualization techniques, advanced deep mining and neural network approaches which increase the reliability and adaptability of the software but is still limited in its usage.
3. Increase in the size of data analyzed: Scale of software projects substantially increase every decade due to availability of enormous open-source codes which are very well maintained by the community concerned and modified for organizations own needs.
4. Limited evolution of metrics adopted: Despite of proposal of innumerous amount of software maintenance models, the analysis metrics have not changed much in the last four decades.

**Paper-III**

A Review on Software Maintenance Issues and How to Reduce Maintenance Efforts *(2015)* *Uttamjit Kaur, Gagandeep Singh*

**Abstract**

Software maintenance has many challenges and requires a lot of efforts to be done properly. This paper suggests some issues and problems encountered during the process of software maintenance. Software maintenance is often time consuming and costly.

Software maintenance serves the following purposes to the project:

1. Updating the software
2. Software upgrades
3. Continuity of Service

Maintenance costs are mostly by-products of software improvements rather than corrections. Maintenance of software in general is a tedious task since it requires one to have thorough knowledge of the source-code in question of updating, upgradation or integration.

It opens another entry of risk which is cost estimates of product. Since, the maintenance cost is tough to predict due to its volatile nature in general. Cost estimate is a very common risk to have in software projects. Maintenance issues could be concerned with staff turnover, cost, deadline/timeline, product quality, database size, documentation quality etc.

Maintenance issues can be minimized through:

1. Eliminating Dead Code
2. Decreasing Turnovers
3. Reducing unnecessary and redundant complexity from project
4. Re-evaluation of software maintenance model to stay up-to-date
5. Quality testing
6. Bug Elimination
7. Increasing understandability of documentation

**REFERENCES**

* *Ana M. Fernández-Sáez & Michel R. V. Chaudron & Marcela Genero (2018)* An industrial case study on the use of UML in software maintenance and its perceived benefits and hurdles
* *Valentina Lenarduzzi, Alberto Sillitti, Davide Taibi (2017)* Analyzing Forty Years of Software Maintenance Models
* *Uttamjit Kaur, Gagandeep Singh (2015)* A Review on Software Maintenance Issues and How to Reduce Maintenance Efforts

**IMAGES**





