ANALYZING THE IMPACT OF UML DIAGRAMS ON SOFTWARE MAINTENANCE

Dr. Yashwant Waykar,
Assistant Professor,Department of Management Science,
Dr.Babasaheb Ambedkar Marathwada University, Aurangabad.
email id- waykar.yashwant@gmail.com

Dr. Kaveri Lad,
Assistant Professor,
Department of Management Science,
Dr.Babasaheb Ambedkar Marathwada University, Aurangabad.

Abstract –

The main objective of software maintenance is to make the software system operational according to the user requirements and fix errors which are present in it. The maintenance process also involves adding new features and functionalities using latest technologies to the existing software system. Because of the non-functioning of the software or incompatibility of hardware with the software, an error occurs in the software. Implementation of UML is classical, theoretical & standardized approach and every software development has to implement this approach. This Study ensures the usage of UML approach in software industry, its applicability in different types of software development like web, application, embedded system, mobile app development etc. from the study it is concluded that UML implementation improves all the qualitative properties such as covering requirement, Correctness, Modularity, Testability, and Understanding Ability of the final software product. It results in significant impact on maintenance which definitely reduces the maintenance efforts.

Keywords – UML, Maintenance, Analysis, software

Introduction –

Over a period of time, With respect to the changing user's requirements, the developed software system may need some changes. In Such cases the maintenance becomes essential. In the software maintenance process it includes a set of software engineering activities that occur after the software has been delivered to the user. The main objective of software maintenance is to make the software system operational according to the user requirements and fix errors which are present in it. The maintenance process also involves adding new features and functionalities using latest technologies to the existing software system. Because of the non-functioning of the software or incompatibility of hardware with the software, an error occurs in the software. When software maintenance is to be done on a small part of the software code, software patches are applied to it. These patches are used to fix errors only in the software code that contains errors. When the technical problems occur in software or in hardware or increasing in the cost then the software maintenance affects on these things. In this chapter we will see how software maintenance assists the present software system to accommodate changes according to the new requirements of users.

Basics of Software Maintenance

Software does not wear out or get tired. However, it needs to be upgraded and enhanced to meet new user requirements. For such modifications in the software system, software maintenance is performed. IEEE defines maintenance as 'a process of modifying a software system or component after delivery to correct faults, to improve performance or other attributes or to adapt the product to a changed environment.' It is intended to ensure that the software is convenient to modify after system is deployed and delivered.

In addition, there are further intentions for software maintenance, these are as follows:

1. **Providing continuity of service:** The software maintenance process focuses mainly on error solving, recovering from failures such as, hardware failures, or incompatibility of hardware with the software, and adjusting changes in operating systems and hardware

2. **Supporting mandatory upgrades:** Software maintenance supports the upgradations if necessary in a software system. Upgradations may be required if there any changes in government standards or in regulations. For example, If a web-application system has been developed with multimedia capabilities then changes in countries where the video (on the Internet) screening is restricted. With the help of other software existing in the same category, the need for upgradation can also be realized.

3. **Improving the software to support user requirements:** For enhancing the functionality of the software, to improve performance, or to customize data processing functions as desired by the user then requirements may be requested. With the help of Framework which is provided by software maintenance, we can accomplish changes which are required can be accommodated.

4. **Facilitating future maintenance work:** Software maintenance also helps in future maintenance work, which may include reconfiguration of software code and databases used in the software.

Implementation of UML -

Implementation of UML is classical, theoretical & standardized approach and every software development has to implement this approach. This Study ensures the usage of UML approach in software industry, its applicability in different types of software development like web, application, embedded system, mobile app development etc. Depending upon the type of software development there can be some variation in the approach. According to this study, it identified the actual ground reality about implementation of UML. The Ideal scenario about this study is that all the projects must implement UML but practically whether it can be possible or not is to be identified through the study. While developing the software IT companies has to focus on lots of issue regarding the projects like Time constraint (i.e. deadline of project), cost, expertise, manpower in terms of efforts required, complexity of the project etc. By considering all these vital factors companies proceed with the software development. This Study wants to ensure about the implementation approach adapted by the IT companies in software development. Approach can be fully implemented, partially implemented

or not implemented at all. This implementation study wants to identify the usage percentage, benefits of usage, hurdles if any during implementation of UML.

SCOPE AND LIMITATION OF THE STUDY–

Scope of the study is limited only to some important aspects of UML. Collected samples are related only to IT companies located in Maharashtra region. Where major of the IT companies are located these cities of Maharashtra region are considered for the study. For sample collection, 5 cities from Maharashtra region are selected. Not all IT companies are contacted in these cities; only selected IT companies are considered as a part of this study.

RESEARCH METHODOLOGY -

Data collection -

The research is based on the primary as well as secondary data.

Primary Data -

Primary data has been collected by organizing surveys and discussions with IT employees in different areas of Maharashtra. Only 5 cities in Maharashtra were considered for the study because of the presence of IT sector in these cities. With respect to the objectives of the study well formulated questionnaire was prepared. For conducting the survey, online and offline mode of data collection was used. In online mode assistance of Google Forms, Face book and LinkedIn has taken. In offline mode personally questionnaires were circulated & responses were collected in hardcopy format. Also lots of discussions were made to analyze responses of IT employees on different aspects of the study.

Secondary Data –

Secondary data related to the concept of unified modeling language, UML's various implementation techniques, detailed study of UML diagrams, innovative research from different corners of the world, etc. Data which are related to many other important aspects of the study has been collected.

Following are some of important ways through which secondary data was collected –

1. Online Journals (EBSCO, IEEE CS etc)
2. Online Databases
3. Digital Libraries (UGC INFONET)
4. Printed National & International Journals
5. Conference Proceedings
6. Websites
7. Ph.D Thesis

Sample Collection -

Before sample selection the main task was to identify the cities for research purpose in Maharashtra where IT companies are located. Pune and Mumbai are the major IT hubs of

Maharashtra, where the development of the IT companies is predominantly active and the manpower is heavily related to that area.

Nagpur, Nasik and Aurangabad samples were considered for selection as IT cities in these cities are growing late. Many big names in the IT sector come to these respective development centers in these cities. Therefore, these cities have been selected for research purposes. After selection of cities next important task was the sample size was considered to be suitable for accurate conclusions about the next important work study.

The primary hurdle in the search process is because the total number of data related to the IT sector is not available. To overcome this, the researcher selected a sample size based on time-tested research design techniques which included 95% confidence level with 5% margin of error. Accordingly, the optimal sample size was generally more than enough and 384 samples are required for the exact minimum size required.

However, in this case 704 samples were considered for research because many statistical theories have indicated that the exact ratio of the sample is much larger than that required. Furthermore, this sample size shows the number of employees employed in different companies in the selected cities of Maharashtra.

704 employees from **183** IT companies viz. IBM, Accenture, Amdocs, Infosys, Cognizant, Tieto, Persistent, Tech Mahindra, Syntel, Cybase, Google, L&T, Hexaware, Bitwise, Capgemini, Atos Origin, Ericssion, HSBC, ORACLE, Mind Tree, TCS, Sunguard, HCL, Mastek, Wipro and lot more.

Simply random sampling technique was used for the study. For the purpose of data Collection, The s/w developers, analysts, Team leaders, Project managers, consultant, DBA, Tester & other persons related to the s/w development across Maharashtra were contacted.

There are several issues like privacy, confidentiality of projects and related information, security etc. Because of this reason It is very difficult task to collect data from the IT companies & their employees. Surveys were done through different ways like Online and Offline modes. Online data collection has been done using Google Forms, Facebook and LinkedIn. In offline mode, IT personnel were contacted individually and data was collected through the questionnaire.

Following table shows the distribution of **704 samples** in different cities of Maharashtra –

Table No 1:

Region- wise distribution of samples

Sr. No	Name of City	No. of Samples
1	Pune	310
2	Mumbai	249
3	Nagpur	56
4	Nasik	50
5	Aurangabad	39
Total		**704**

Source – Field Survey

Use of UML in different activities / phases of software development -

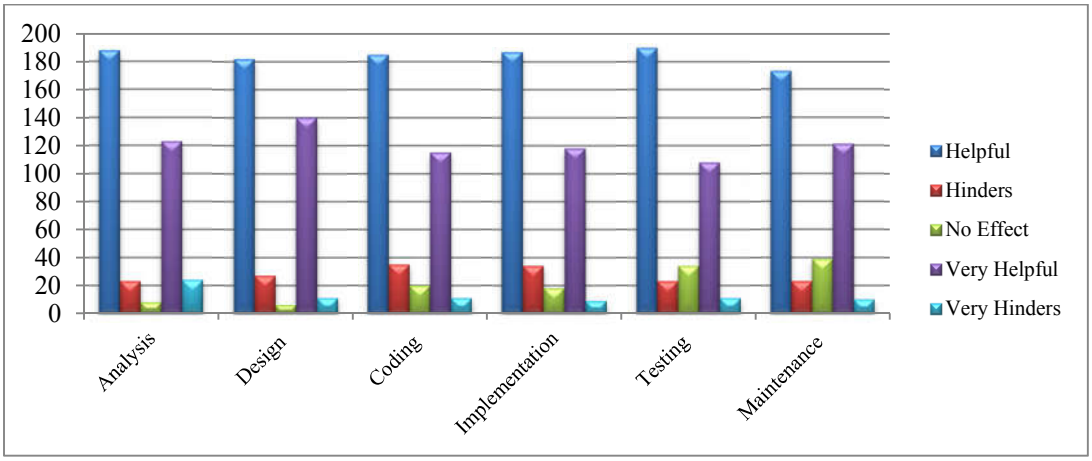
The Software development is divided into several phases, such as analysis, design, coding implementation, testing and maintenance. researchers are comparing the usage of UML in each phase and identifying whether it is helpful or hinders the productivity of the employees.

Table No.2:
Use of UML in different activities of software development

Sr. No	Use of UML	Analysis	Design	Coding	Implementation	Testing	Maintenance
1	Very Hinders	24	11	11	9	11	10
2	Hinders	23	27	35	34	23	23
3	No Effect	8	6	20	18	34	39
4	Helpful	188	182	185	187	190	173
5	Very Helpful	123	140	115	118	108	121
	Grand Total	**366**	**366**	**366**	**366**	**366**	**366**

Source: – Field Survey

Figure No. 1:
Use of UML in different activities of software development



Source: – Table No.2

From Table No 2 it is clearly identified that in all the phase of software development more number of responses are in the row of helpful and very helpful as 188-182-185-187-190-173 and 123-140-115-118-108-121 respectively. From Figure 1 it depicts that big taller bar in blue and violet colour highlight the helpful and very helpful nature of UML in all the phase of software development. On the contrary side, Table 2 also indicates that in all the phase of software development very less number of responses are in the row of hinders and very hinders as 23-27-35-34-23-23 and 24-11-11-9-11-10 respectively. From Figure 1 it depicts that smaller bar in red and sky blue colour highlight the hinder and very hinders nature of UML in all the phase of software development.

Above analysis UML diagrams are really helping in performing several tasks in each phase of SDLC. It reflects the scope & range of UML diagrams. This study found that UML implementation is helpful in being more productive to the employee in all the phases of software development.

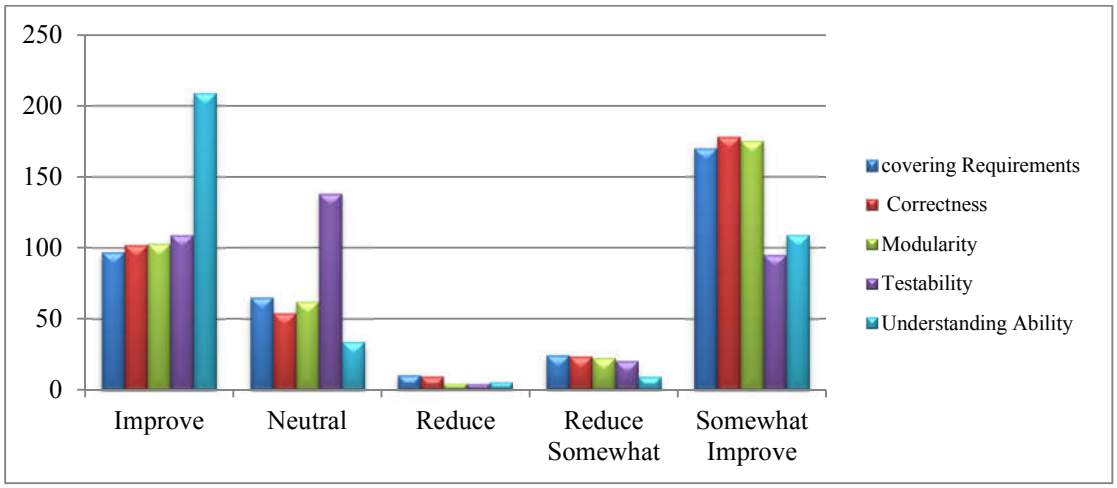
Influence of UML on different qualitative properties of the final Software product -

Table No. 3:
Influence of UML on different Qualitative Properties

Sr. No	Scale of UML Influence	covering Req.	Correctness	Modularity	Testability	Understanding Ability
1	Improve	97	102	103	109	209
2	Neutral	65	54	62	138	34
3	Reduce	10	9	4	4	5
4	Reduce Somewhat	24	23	22	20	9
5	Somewhat Improve	170	178	175	95	109
	Grand Total	**366**	**366**	**366**	**366**	**366**

Source: – Field Survey

Figure No. 2:
Influence of UML on different Qualitative Properties



Source: – Table No. 3

From Table 3 it is clearly identified that in all the qualitative properties of final software product more number of responses are in the row of improve and somewhat improve as 97-102-103-109-209 and 170-178-175-95-109 respectively. From Figure 2 it depicts that big taller bar in all colour highlight improve and somewhat nature of UML for all the qualitative properties of the end product. On the contrary side, From Table 3 it is clearly identified that in all qualitative properties of the end product very less number of responses are in the row of reduce and reduce somewhat as 10-9-4-4-5 and 24-23-22-20-9 respectively. From Figure 2 it depicts those smaller bars in all colour highlight reduce and reduce somewhat nature of UML in all qualitative properties of the end product.

Above analysis proves that UML implementation improves all the qualitative properties such as covering requirement, Correctness, Modularity, Testability, and Understanding Ability of the final software product.

FINDINGS AND CONCLUSIONS –

From **Data Analysis and Interpretation** following important conclusion can be made.

1. UML diagrams surely helpful in performing maintenance activity. Maintenance is majorly done on the basis of design & other documentation of the system so indirectly it is also proved that UML helps in generating the documentation of the system.

2. UML implementation improves all the qualitative properties such as covering requirement, Correctness, Modularity, Testability, and Understanding Ability of the final software product.

REFERENCES –

1. Ambler, S. (2005). *The Elements of UML(TM) 2.0 Style*. Cambridge: Cambridge University Press.

2. Fernández-Sáez, A. M., Genero, M., & Chaudron, M. R. (2013). Empirical studies concerning the maintenance of UML diagrams and their use in the maintenance of code: A systematic mapping study. *Information & Software Technology*, *55*(7), 1119-1142. doi:10.1016/j.infsof.2012.12.006

3. WAYKAR, Y. (2016). A STUDY OF UML DIAGRAMS IN SOFTWARE DEVELOPMENT (PUBLISHED THESIS). DR.B.A.M.U.AURANGABAD.

4. Terese Helene Haug. (2007). *A Systematic Review of Empirical Research on Model-Driven Development with UML* (Master's Thesis, UNIVERSITY OF OSLO).Retrieved from https:// www.duo.uio.no/bitstream/handle/10852/9635/Haug.pdf?sequence=4

5. Felderer, M., & Herrmann, A. (2015). Manual test case derivation from UML activity diagrams and state machines: A controlled experiment. *Information & Software Technology*, *61*1-15. doi:10.1016/j.infsof.2014.12.005