Importance and Impact of Class Diagram in Software Development

Abdul Hafeez*, Mansoor Ahmed¹, Muhammad Furqan², Wasi-Ur-Rehaman³ and Imtiaz Husain¹

¹Department of Computer Science, SMI University; karachi.akhkhan@smiu.edu.pk
²Department of Information Technology, Higher Technology College Muscat; muhammadfurqan@hct.edu.om
³Techway System, Musacat; wasi@gmail.com

Abstract

Background/Objectives: This work mainly focuses on the role of UML Class diagram in different phases of software development cycle and shows the industry responses regarding its impact and importance. Methods/Statistical Analysis: we conducted industry survey and collected data from software practitioners. Participants were asked about their usage of UML for modeling software artifacts and most frequent diagram they use. Findings: Result shows that UML Class diagram is most important element of UML and it performed highly significant role in software development. It is most frequently used diagram. Application/Improvements: Usage of UML models during the software development improve the quality of software and UML Class diagram is most important part of UML and help practitioners to perform analysis and as well as design in a systematic way.

Keywords: Software Class Development, Software Class Diagram, Software System

1. Introduction

UML is an industry standard and significantly used visual modeling language for the modern software system. It hides the complexity of system and provides an appropriate level of abstraction. It is used in the software industry for software specification, analysis, design and documentation. Nowadays, it is also used for code generation. UML offers a number of diagrams for dealing various aspects of software modeling. It uses simple diagrammatic notations for describing software. Therefore, customer can also easily understand the specification and design of the system. Currently, it is also used in engineering, ontology development, DBMS and other discipline of technology. In Model Driven Architecture, UML considers as a core and the entire process of the MDA move around the UML Models.

2. Research Hypothesis

Research suggests that class diagram is the most important part of UML. It performs an important role in software analysis and design. It describes the system through concepts, their relationship and constraint over concept. It’s also a key ingredient of the MDA process. Therefore, we advanced the following research hypothesis:

H0: The UML Class diagram is considered as a core diagram which is used in the different activities in the software development.

H1: Class diagram is not an important artifact in software development.

For determining importance and impact of class diagram in software development, an industrial survey was performed and investigated:

*Author for correspondence
• Which UML diagram mostly use in software development (TCD).
• Is Class diagram improves the understanding of business domain (ICDIUBD)?
• Is Class diagram improves the communication among the team members (ICDICAT)?
• Is Class diagram gives a blueprint of system to maintenance engineer to get a general idea of how the software is structured before investigating the code (ICDBIC)?
• Is Class diagram assists development of building other UML diagrams (ICDAUML)?

3. Research Methodology

In the survey, data were collected from (N = 71) software practitioners. The sample consisted of 13 (18.8%) international software practitioners and 58 national software practitioner (81.2%). Survey questionnaire consisted of three sections. In Section 1 practitioners were inquired about their demographic information (e.g. Name, email, Work Location, Position). In Section 2 they were asked about their usage of UML for modeling software artifacts and most frequent diagram they use. Finally, in Section 3 they were asked their use of different diagrams.

4. Research Results

The sample consisted of 32 (45.07%) software developers, 11 (15.49%) Software Analysts, 12 (16.90%) Team Leads, 9 (12.68%) Project Managers, 4 (5.63%) Software Designer and 3 (4.23%) Software Testers as shown in Figure 1.

In the sample 2 practitioners (2.82%) have more than 10 years of experience, 7 practitioners (9.86%) have 5 to 10 years of experience, 19 practitioners (26.76%) have 3 to 5 years of experience and 43 practitioners (60.56%) have 1 to 2 years of experiences as shown in Figure 2.

As per result shown in Figure 3, majority of the practitioners (89.73%) use UML for modeling software artifacts. Only (11.27%) stated they do not use UML.

As can be seen in Figure 4, in response of most frequent diagram for modeling they use. (47.89%) reported class diagram, (29.58%) reported use case diagram, (11.27%) reported Activity diagram, (4.23%) reported component and deployment diagram, (2.82%) reported state chart diagram, (1.41%) reported sequence diagram and (0.0%) reported communication, composite structure, timing, and interaction overview. Our hypothesis is, the UML Class diagram is considered as a core diagram which is used in the different activities in the software development. Whereas, alternate hypothesis is class diagram is not an important artifact in software development. Correlation analysis was performed to observe the effectiveness of UML Class diagram. The Table 1 shows that the class diagram was found to be highly significant and integral diagram which improves the understanding of business domain as well as improve the communication among the team members with r = 0.008 at P < 0.05. Whereas, class diagram is very helpful in maintenance phase and help out maintenance engineers to get a general idea of how the software is structured before investigating the code with r = 0.066 at P < 0.01 and also assist the development of building other UML diagrams (r = 0.012; P<0.05).
5. Conclusion

UML Class diagram is most important element of UML and it performed highly significant role in software development. It is most frequently used diagram. It improves the communication among the team member, gives a blueprint of system to maintenance engineer to get a general idea of how the software is structured before investigating the code and assists development of building other UML diagrams.

6. References


