



## ORIGINAL RESEARCH PAPER

## Information Technology

# ANALYZING THE RELATIONSHIP BETWEEN IMPLEMENTATION OF UML AND SOFTWARE COMPLEXITY

**KEY WORDS:** UML, software development, complexity

**Dr. Yashwant Waykar**

Assistant Professor, Department of Management Science Dr.B.A.M.University, Aurangabad.- 431001 (Maharashtra) - Corresponding Author

### ABSTRACT

UML stands for Unified Modeling Language. UML is a modeling language vastly being used to visualize, specify, document, and construct the documentation of the system. It is used to model the system (software) along with non system (other than software's) also. "A picture can speak more than thousand of words" is the basis for UML as it is a pictorial form of the system. It is been useful in the different phase of software development life cycle.

It is identified that as the level of complexity of software increase the effectiveness of UML implementation also increases. Also as complexity of software decrease the effectiveness of UML also decrease. Complexity and UML effectiveness are directly proportional to each other. Implementation of UML is always depends upon the complexity of software. The effectiveness of UML also increases with the complexity of software.

### INTRODUCTION:

UML stands for Unified Modeling Language. UML is a modeling language vastly being used to visualize, specify, document, and construct the documentation of the system. It is used to model the system (software) along with non system (other than software's) also. "A picture can speak more than thousand of words" is the basis for UML as it is a pictorial form of the system. It is been useful in the different phase of software development life cycle like analysis, design, implementation etc. UML diagrams helps for better understanding of the system, their functionalities, different entities, relationships etc. UML basically designed to perform different activities; one of the important activity is to have better understanding of the system to the developer, analyst, business person & to everyone who is directly or indirectly associated with the system.

History of UML starts in way back 1990's. Booch Methodology, Object Modeling Technique (OMT) by Rumbaugh, Object Oriented Software Engineering (OOSE) by Ivar Jacobson was the object oriented modeling approach. Later on by combining these approaches UML had been invented in 1996-97. Further UML was given to OMG (Object Modeling Group) for the Standardization. The first draft of UML 1.0 was being released in 1997. Lots of versioning of UML has been taken place from 1997 to 2013 and several new things been added, changes been incorporated to the UML. currently the UML 2.5 version is available in market.

Mainly all UML diagrams are broadly categorised into two classes i.e. Structural & Behavioural Diagrams.

### Structural Diagrams:

Structural Diagrams models the structural or static part of the system. It represents physical & conceptual elements, which are helpful in better understanding of the static nature of the system; it shows complexities involved the system in the easiest & pictorial format. Structural diagrams include

- Class Diagram
- Object Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram
- Profile Diagram
- Package Diagram

### Behavioural Diagrams:

Behavioural Diagrams models the dynamic nature of the system i.e. the behaviour of the system. It helps in understanding the functionalities of the system. Behaviour Diagram includes:

- Use Case Diagram
- State Diagram
- Activity Diagram
- Interaction Diagrams –

Sequence Diagram,  
Communication Diagram, Interaction overview,  
Timing Diagram

Each diagram in UML gives different perspective to look at the system & better understanding of the system. UML diagrams are closely associated with the SDLC phases like:

- Analysis Phase:
  - Use Case Diagram
  - Activity Diagram
- Design Phase:
  - Class Diagram
  - State Diagram
  - Interaction Diagrams
  - Sequence Diagram
  - Communication Diagram
- Implementation Phase:
  - Component Diagram
  - Deployment diagram

### SCOPE AND LIMITATION OF THE STUDY–

Scope of the study is only restricted to few of the important aspects of UML. The Collected samples belong to the IT companies located in Maharashtra region only. Important cities in Maharashtra region are considered for the study where major of the IT companies are located. 5 cities are selected for sample collection. From these cities all the IT companies are not been contacted; only selected IT companies are considered for the study. For the study primary data is collected during the year 2014-15. Secondary data used does not belong to specific time period as the nature of the research does not demand such data.

### RESEARCH METHODOLOGY -

#### Data collection -

The research is based on the primary data & secondary data.  
Primary Data -

Primary data was collected by conducting survey and discussions with the IT employees from different regions of Maharashtra. Only 5 cities in Maharashtra were considered for the study as the presence of IT sector in these cities. Considering 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 gathered 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 is related to the concept of Unified Modeling Language, different implementation techniques of UML, detailed study of UML diagrams, innovative researches from different

corners of the world etc. Along with this data related to several other important aspects of the study had collected. Following were few of the 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 -

Prior to sample selection the main task was to identify the cities for research purpose in Maharashtra where IT companies are located. Pune and Mumbai are major IT hubs in Maharashtra where software development is prominently active and huge amount of manpower is associated with that sector. Nagpur, Nasik and Aurangabad were considered for sample selection since IT sector in these cities has been rapidly increasing of late. Lot of big names in IT sector come up with their respective development centres in these cities. Hence, these cities were selected for research purpose.

After selection of cities next important task was to decide the sample size deemed appropriate to draw accurate conclusions about the study. Lack of available data regarding the total number of employees engaged in the IT sector was the primary hurdle in the research process. 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 sample size thus selected was more than sufficient in general & more over the min size is of 384 samples required for accurate results. However, in this case 704 samples were considered for research as several statistical theories suggest that larger the sample size than required (Min size) accurate are the results. Further, this sample size represents the scale of employees employed in different companies across selected cities in Maharashtra region.

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

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

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

DATA INTERPRETATION –

From Table No.1, it is identified that total 704 respondents are considered for the study. This study is broadly categorized into two major parts such as respondents who uses the UML & respondents who won't use UML. It is observed that 366 (52%) respondents used UML in software development whereas 338 (48%) respondents never used UML. These are quite surprising numbers in terms of UML usage.

Effectiveness of UML implementation for different type's of software -

Through this part, study identifies effectiveness of UML in different types / nature (as per complexities) of software development. Here researchers are comparing the UML effectives among all types of software with an intent to identify pattern.

Table No 2: Effectiveness of UML implementation for different software

Sr. No	Level of Effectiveness	Nature of Software			
		Small Scale	Mid Scale	complex	Highly Complex
1	Effective	75	208	203	97
2	Highly Effective	10	23	132	250
3	Less Effective	177	122	24	16
4	Not Effective	104	13	7	3
Grand Total		366	366	366	366

Source: – Field Survey

Table 2 shows the figures indicating the responses for highly effective from small scale to highly complex software development are increasing steadily from 10 – 23 – 132 -250. Table 2 denotes that the figures indicating the responses for less effective from small scale to highly complex software development are decreasing steadily from 177 –122 – 24 -16.

Hypothesis Testing-

Hypothesis 1: UML is mostly preferred for complex software development.

H0: Effectiveness of UML implementation and nature of software are not related with each other.

H1: Effectiveness of UML implementation and nature of software are related with each other.

For this analysis, the significance level is 0.05. Using sample data, we will conduct a chi-square test for independence.

Applying the chi-square test for independence to sample data, we compute the degrees of freedom, the expected frequency counts, and the chi-square test statistic. Based on the chi-square statistic and the degrees of freedom, we determine the P-value.

Expected Frequencies –

Table No :3 Expected frequencies of level of effectiveness of UML

Sr. No	Level of Effectiveness	Nature of Software				Total
		Small Scale	Mid Scale	complex	Highly Complex	
1	Effective	145.75	145.75	145.75	145.75	583
2	Highly Effective	103.75	103.75	103.75	103.75	415
3	Less Effective	84.75	84.75	84.75	84.75	339
4	Not Effective	31.75	31.75	31.75	31.75	127
Grand Total		366	366	366	366	1464

Source: – Table No. 2 & calculated values

Table No :4 P-Value calculation for level of effectiveness of UML

O	E	O-E	(O-E)2/E
75	145.75	-70.75	34.34
208	145.75	62.25	26.59
203	145.75	57.25	22.49
97	145.75	-48.75	16.31
10	103.75	-93.75	84.71
23	103.75	-80.75	62.85

132	103.75	28.25	7.69
250	103.75	146.25	206.16
177	84.75	92.25	100.41
122	84.75	37.25	16.37
24	84.75	-60.75	43.55
16	84.75	-68.75	55.77
104	31.75	72.25	164.41
13	31.75	-18.75	11.07
7	31.75	-24.75	19.29
3	31.75	-28.75	26.03
			<b>898.05</b>

Source: – Table No 3 & calculated values

$\chi^2 = 898.05$

$N = 9$

We reject  $H_0$  because  $898.05 > 16.92$ . We have statistically significant evidence at  $\alpha = 0.05$  to show that  $H_0$  is false or that effectiveness of UML diagram and nature of software are not independent so we accept  $H_1$  as effectiveness of UML diagram and nature of software are dependent (related with each other),  $p < 0.05$ .

#### CONCLUSION:-

It can be clearly interpreted that as the level of complexity of software increase the effectiveness of UML implementation also increases. Also as complexity of software decrease the effectiveness of UML also decrease. Complexity and UML effectiveness are directly proportional to each other. Implementation of UML is always depends upon the complexity of software. The effectiveness of UML also increases with the complexity of software. Therefore it is identified that UML is mostly preferred for complex software development.

#### REFERENCES:

1. WAYKAR, Y. (2016). A STUDY OF UML DIAGRAMS IN SOFTWARE DEVELOPMENT (PUBLISHED THESIS). DR.B.A.M.U.AURANGABAD.
2. HADAR, I., & HAZZAN, O. (N.D.). ON THE CONTRIBUTION OF UML DIAGRAMS TO SOFTWARE SYSTEM COMPREHENSION. THE JOURNAL OF OBJECT TECHNOLOGY,