

Improving Reliability of Object Oriented Design: Defect Mitigation Perspective

SUMMARY of THESIS

Submitted to
Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow



For the Degree of
Doctor of Philosophy
In
INFORMATION TECHNOLOGY

By:

Pawan Kumar Chaurasia

Enrollment No. 960/14

DEPARTMENT OF INFORMATION TECHNOLOGY
BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY
(A CENTRAL UNIVERSITY)
VIDYA VIHAR, RAEBARELI ROAD, LUCKNOW-226 025 (U.P.), INDIA

2017

1. Introduction

With the development of electronic gadgets, people have become highly dependent on these gadgets in our daily lives. The gadget revolution is accelerated by technological advancement in software as well as hardware. Today, software has fully captured our modern society. It is embedded in mobiles, home appliances, aviation, automobiles, industries and military. Software makes every work easier, faster and accurate etc with less cost. It has become a driver for everything from elementary items to genetic engineering. Every work can be done easily with the help of software based devices. This is the reason software has crept into every industry including telecommunication, petroleum, banking, railway, hospital, pharmaceutical etc. All these are highly dependent on software for their basic functioning.

Modern computer industry and electronic systems typically contains a significant amount of software. With the development of the industry, the size and complexity of software is also grown during the past decades and these trends will continually persuade in the future. Software is a part of system solution that can be executed on a computer in the form of instruction. Due to increased dependency on software, it is the responsibility of the developer to provide quality and reliable software to the society¹. In view of quality, software reliability is becoming more important. Software is used in many applications in various fields. Software development is badly affected by strict product delivery deadlines. This is one of the reasons for the improper behavior or failure of the software. Software failure affects seriously the field of aviation, telecommunication, industry, handheld devices, home appliances etc. Even a single defect in the software may affect billions of people life. With the increase in the dependency on software, the possibility of defects and hence software failures increases. These failures cause economic damage as well as loss of life. In other words, we can say that software can also kill people and society. Literature survey reveals that failure of software led to serious effects on business, economy and society². Even the most powerful business organization such as Microsoft is fearful from “killer bugs”³. Hence, the reliability of computers systems has become a major interest for our society.

There happened many casualties due to software failure in the field of aviation etc. and many hazards made people lost their lives⁴. L. Lee wrote in his book that

various patients have lost their lives in 1985 and 1986 due to software errors occurred in Therac-25 radiation therapy machine². In 1990, a fault in a switching system software caused heavy long distance networks and another series of local phones equipment failure². In October 1992, the Computer Aided Dispatch (CAD) system of London Ambulance service broke down right after its installation, deactivating the capability of world's largest ambulance service, which handles more than 5000 requests daily to transport patients in emergency^{2, 22}. In 2003, the cause of North East blackout at New York has been a programming error. The failures occurred when multiple systems tried to assess the same information at the same time and acquired the busy signals. Actually, the software could allow one system presidency at a time. Unfortunately, due to several requests at a time the software could not perform properly. The several incidences have forced the industry to develop reliable software. Though, it is not possible to develop 100 percent reliable software the developers and researchers have been making an effort to improve reliability of software under process.

Identification and fixing of software defects is important for improving the quality of the software. It is the most expensive activity in software development¹⁵. Software reliability has become an important component for software industry. With the growth in the demand of software, industry also advances in technology development, which makes software easily available to the society in minimum cost with faster speed. Software failures occur due to errors or bugs in the software at initial level of the design phase of the software. With the increased growth of software development and its easy availability to the user, failures have also increased in the same manner. Therefore, strict quality assurance is required for the execution of software to validate whether the software application behaves as required by the user¹⁶. Software quality is one of the important factors of software reliability. Software quality is affected by the development cost, delivery, schedules and user satisfaction. Software defect's cost billions of dollars for the industry²⁴. Therefore, these defects need to be identified and removed at early stage of software development to improve the reliability of the software²¹.

2. Need and Importance of Software Reliability

From literature survey it is found that various hazards have occurred claiming lives of people and assets because of error, faults, failure, defects and bugs in software. In 1985 and 1986, a lot of patient lost their lives because of software errors in Therac-25 radiation therapy machine¹. One of the incidence happened in 1992 in London Ambulance service, provide the largest ambulance service in emergency situation to patient, broken down after its installation¹. In aviation industry, every year several airline crashes due to misinterpretation and un-compatible work between software and pilot¹. In 1990, a small fault in a switching system newly released network, and traced to software problems occurred during the summer¹. All the above hazards occurred because of unreliable software and cause of defect in software. The object oriented design defects happened in Ariane-5 software reuse the specification from the Ariane-4, but the path defined in Ariane-5 was substantially different. Testing is not performed by the tester, which had never been performed on Ariane-5 flight and the result is shown in figure 1 Ariane-5 Disaster⁵. These hazards occurred because of reuse of code and applicable of OOPs concept without test the design of the software.



Figure 1: Ariane-5 Disaster

Design defects play an important role in software development lifecycle⁶. High quality and reliable software is difficult to achieve in large and complex object oriented programs¹⁰. These defects come from internal design of software and their effect on quality factors like understandability, maintainability, reusability and ease of implementation^{9, 11}. These defects occurred at the design level are known as code smells^{9, 12} or anti-patterns¹³. It is difficult to address every defect for software developer. Even a single defect in software made the whole system crash. It is difficult to maintain software after deliver to the customer. If defects are not exposed

and mitigated during software design, can obtain huge cost in terms of time, money and effort after implementation. From the above software failure shows that a single defect may cause the huge lost and people lost their lives. Therefore, it is required that the defects should be mitigate at the design level for improving reliability of object oriented design.

3. Relevant Issues

Today, internet, mobile and computer are the important role of every people in daily life. Everyday people perform his task by using computers and internet. Every person has a computer and deals with the use of software and services depend on them²⁵. Business of different areas uses the internet for transaction of payment, purchase, sales and transportation of business with partners. From academic to health, business to medical everyone use the internet and computer and they are not alone. All these items run upon which they can belief. More dependency on the system makes the system more error prone and unreliable¹⁷. This makes the system more typical to understand and increase the no of defects of the system. Due to reliability many other episodes also occurred. Early stage of software development is necessary to detect defect and measure the reliability of the software.

Twenty first century is the revolution of object-oriented technology. Object oriented concepts have changed the viewed that used to assess the reliability of software. Object oriented technology applied various constructs like cohesion, coupling, inheritance, encapsulation and polymorphism^{20, 23}. In design phase of the software development the programmer has an authority of using the coupling, cohesion, inheritance and polymorphism for the design of reliable software. High reliable software makes the system defect free, easy to understand.

Based on the discussion of the above problem there may be large set of research challenges that may need to focus. Some of the relevant issues discussed which are as follows:

- What are the factors that directly influence the reliability of an object oriented software design?
- Is design defect a reliability factor?

- Is there any standard framework available to develop reliability improvement of object-oriented design.
- Is there any OO design metric or model available, which can be used in the early phase of software development life cycle to measure its reliability?
- Can we develop an approach that may be used in the design phase to estimate the reliability of an object oriented software?
- Can effort of estimating the design defect of OOD be reduced by using good measurement tools?
- Is the proposed approach to quantify object oriented design defect help to compare the reliability of various alternative designs?
- Can a proposed approach be useful without assuring theoretical and empirical validity?
- How general are the lessons learned in this study? Can they be applied in situations involving other metrics and models, or to organizations, which have different operational contexts?

4. Research Problem

A design should be as simple as possible. Design is the main frame of the software. It is easy to incorporate reliability with design during software development life cycle (SDLC). Similarly, if the reliability features ignored, the final design of the software may be complex¹⁷. Object oriented software frequently modified during development or software evolution. Design defects are highly effect the reliability. If design defect increased, it affects reliability. These design defects can manage through the implementation of object oriented concepts. It has been use as one of the key factor for solving the software crisis problem¹⁸. Defects of software design can checked by software design constructs like inheritance, encapsulation, polymorphism, cohesion and coupling. High coupling and inheritance, increases the design defects of the software design and when cohesion increases, design defects decreased. Therefore, inheritance and coupling have a positive impact while cohesion has negative impact

on software design. Reliability of the object-oriented design can increase by controlling the design defects.

It comes from the discussion that defect mitigation in the software design leads to the development of high reliability of final products¹⁹. Using metrics, defects may be measure and mitigate. A metrics based approach may assess defects and their expansion for object oriented design. With these approaches, reliability of the object oriented software design at early stage of software development can improved. There is not any approach used for early stage of software development to measure the defect. It is highly demand to measure the design defects for object- oriented software used in the design phase. In view of that, the researcher has developed a problem in order to detect defect and estimate reliability of object oriented design by controlling the defects of software design. From critical review and found that there is not any framework to measure design defect to improve the reliability of the software design. Therefore, researcher has proposed framework can mitigate the design defects of object- oriented design under this title:

<p style="text-align: center;">Improving Reliability of Object Oriented Design: Defect Mitigation Perspective</p>
--

5. Objectives of the Research

To achieve the most general objective of working out of an approach for improving reliability of object-oriented design defects, following objectives are set away:

- To critically review and analyze the literature survey on software reliability, object oriented design constructs and object oriented design defects.
- To consider the demand, importance and implications of identifying reliability and design defects in the early stage of software development life cycle.
- To identify the factors which affect the reliability of object-oriented design.
- To study OOD constructs such as inheritance, cohesion and coupling to address design defects.

- To consider the relation between object oriented designs construct with object-oriented design defects.
- To measure which of the object-oriented design constructs has positive and which one has negative impact on design defect.
- To establish the relation between OOD defects and reliability.
- To develop a design defect mitigation methodology to be used in the design phase to estimate the rigidity of object oriented design and provide guidelines to improve the reliability of the object oriented design.
- To create a framework for improving reliability of the software for each of the identified object oriented design constructs.
- To develop design defect and reliability estimation model to validate the proposed framework and model.

6. Major Findings

From the exhaustive literature survey on software reliability of object oriented design construct and design defect the researcher has made the following contributions:

- ***Development of Reliability Improvement Framework:*** The reliability estimation framework for an object-oriented design proposed in chapter 3. The framework mitigates the defects by using the design constructs at the design phase of object-oriented software. Reliability estimation framework for OOD³⁰ contains four phases.
 - Identification Phase
 - Mapping Phase
 - Measurement Phase
 - Improvement Phase

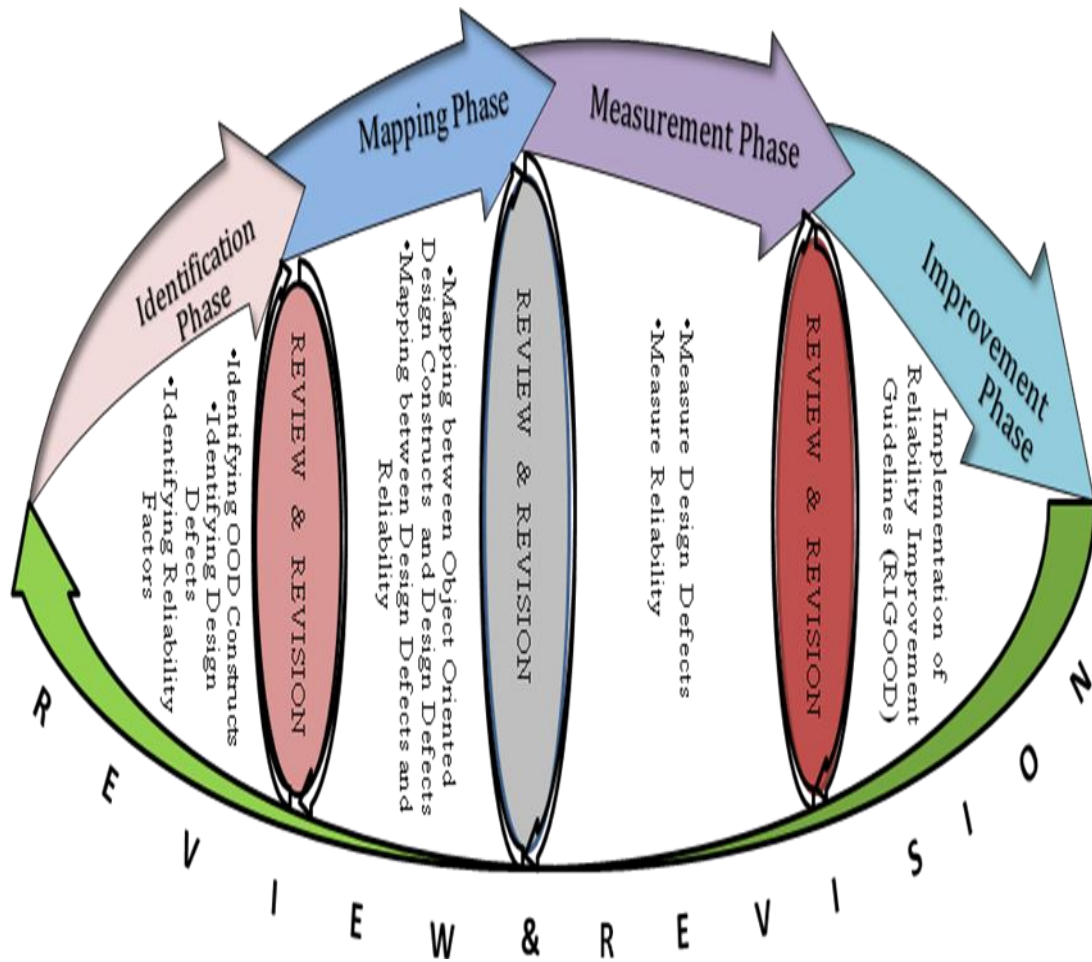


Figure 2: Framework for Reliability Improvement of Object Oriented Design

Object oriented design constructs, reliability factors and design defects have been placed at the identification phase. Second phase is the mapping phase, mapping between object-oriented design constructs and design defects, mapping between design defect and reliability and their impact on each other, have been established. In measurement phase, measurement of design defect and reliability have been done by using the rigidity estimation model and reliability estimation model. In the last phase, reliability improvement guidelines are implemented for object-oriented design. Finally, the whole approach is revised on the basis of suggestions received from the reviewer and experts.

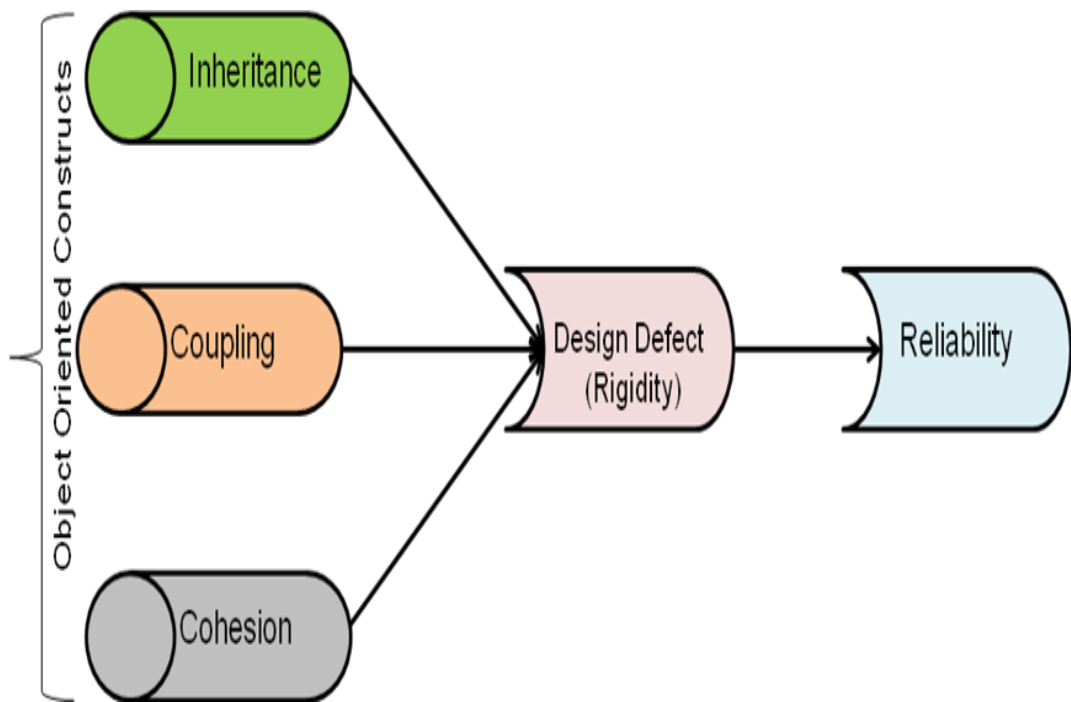


Figure 3: Mapping between OO Design Construct, Design Defect and Reliability

Table 1: Impact of OO Design Constructs on Design Defects

Design Constructs → Design Defects ↓	INHERITANCE	COHESION	COUPLING
Rigidity	↑	↓	↑
Fragility	↑	↓	↑
Immobility	↑	↓	↑
Needless Complexity	↑	↓	↑

- ***Development of Rigidity Estimation Model:***

Rigidity estimation model estimates the defect of an object oriented design. It correlates the defect with object oriented design constructs including inheritance, cohesion and coupling. The identified object oriented design metrics are used to compute defect of an object oriented design. As discussed earlier in chapter-4, that object oriented design metrics affect design defect whether it is positive or negative as shown in table. High value of DIT and CBO increase the rigidity defect and high value of CAMC decrease the rigidity defect which increases the reliability of the object oriented design as discussed in chapter-5. These implications are made while analysing the rigidity estimation model (RiEM).

$$\mathbf{Rigidity\ (R)=0.0579+0.1244(DIT)+0.4126(CBO)+0.2892(CAMC)}$$

‘The high value of rigidity of an OO Design, increases the rigidity defect of design’

- ***Development of Reliability Estimation Model:***

Reliability estimation model maps the reliability with defect of an object oriented design. The proposed reliability estimation model is used to compute reliability of an object oriented design. The model is discussed thoroughly in chapter 4. It was found that defect affects reliability of the design, i.e. presence of defect decreases the reliability of the object oriented design. The implications are made by analyzing the reliability estimation model (ReEM).

$$\mathbf{Re = 1.021 - 0.3651(Ri)}$$

‘The high value of reliability of an OO design, decreases the rigidity defect of design’

- ***Development of Reliability Improvement Guidelines for OO Design (RIGOOD):***

Reliability improvement guidelines for object oriented design have been proposed from the analysis of the metrics (DIT, CBO and CAMC) and models (RiEM), (ReEM). ^{RI}G1 and ^{RI}G2 states that a more reliable design can be developed by

keeping the value of DIT and CBO for the object oriented design closure to its minimum limit to 0. ^{RIG3} and ^{RIG4} states that a more reliable object oriented design can be developed by keeping the value of CAMC close to its maximum limit i.e. approaching to 1. In the light of the proposed reliability improvement guidelines ^{RIGs}, changes are made to the designer so that a more reliable object oriented design can be produced by minimizing rigidity defect of the OOD.

RIG001: Keep the value of DIT for the object oriented design closure to its minimum limit value to 0.

RIG002: Keep the value of CBO for the object oriented design closure to its minimum limit value to 0.

RIG003: Keep the value of CAMC for the object oriented design closure to its maximum limit value to 1.

RIG004: Keep the value of rigidity estimation model for the object oriented design lies between 0 and 1.

RIG005: Keep the value of reliability estimation model for the object oriented design lies between 0 and 1.

In the light of above guidelines, the following characteristics are made by the designer in order to improve the reliability of object oriented design for developing more reliable software.

- Keep the inheritance of object oriented design as low as possible.
- If class is inherited, make sure that attributes and methods are not used unnecessary.
- If there is inheritance in classes, make assurance that data members and methods are not used unnecessarily by the relative classes.
- Acquiring of data members and methods by some other methods must be avoided until necessary.
- Try to avoid coupling between classes as much as possible.

- The attributes defined in class should not be passed as parameter to some method of a simple class.
- Any simple class should not have a return type of a complex class until required.
- Too much method makes a system complex and increases the probability of defects.

7. Validation

Validation is an ongoing process and hence there are degrees of validity more evidence increase the validity of an approach. For collecting more and more evidences for validating of the proposed work, a tryout is carried out by adopting the pre tryout. For collecting more and more manifest for the proposed framework, a tryout is carried out by adopting the pre tryout.

The tryout contains ten object oriented designs. These designs are developed by the B.Tech, M.C.A and M.Sc students in partial fulfilment of their final semester project. All the projects were developed in a software organization at UPTECH, Consultancy, Pvt. Ltd, Hazratganj, Lucknow. The size of the object oriented design ranges is 15 to 35 classes.

(i) ***Collection of Metrics and Model Values:*** Object oriented designs were used to analyse the values of proposed object oriented design metrics and models. For each object oriented design metric and model values were collected. A tabular presentation of the metric values for all the ten designs has been given in table-2.

Table 2: Collection of Metric Values for Ten Designs

Metrics → Designs ↓	DIT	CBO	CAMC	Defect (Rigidity)	Reliability
Design 1	0.4561	0.9678	0.5678	0.3497	0.8933
Design 2	0.3472	0.5748	0.6573	0.1482	0.9669
Design 3	0.5671	0.8256	0.4241	0.3464	0.8945
Design 4	0.6721	0.9281	0.2426	0.4543	0.8551
Design 5	0.1218	0.7861	0.2781	0.3170	0.9053
Design 6	0.1416	0.9192	0.3456	0.3548	0.8915
Design 7	1.2357	0.8921	0.3452	0.4799	0.8458
Design 8	0.8256	0.8857	0.5641	0.3629	0.8885
Design 9	1.3278	0.8264	0.6721	0.3697	0.8860
Design 10	0.7856	0.7862	0.5468	0.3219	0.9035

(ii) **Improving Reliability of Object Oriented Design:** After collection of all the object oriented design metric values, rigidity and reliability values, each of the ten designs is thoroughly analysed keeping in mind for the object oriented design defect. By using the reliability improvement guidelines for object oriented design (^{RI}GOOD) proposed in the study, rigidity within each design has been mitigate and hence values of reliability is improved as much as potential. All the process are possible with the help of refactoring as the aim to develop alternate designs having the same functionality but less rigidity or much reliable.

(iii) **Recollection of Metrics and Models Values:** To assess whether the proposed ^{RI}GOOD guidelines is able to mitigate the rigidity and improve the reliability in the design. Taken for the experiment or not, the value of three metrics and models were again collected for all the ten refactored design. The values are shown in table-3

Table 3: Recollection of Metric Values for Ten Designs

Metrics → Designs ↓	DIT	CBO	CAMC	Defect (Rigidity)	Reliability
Design 1	0.4123	0.8236	0.5678	0.2848	0.9170
Design 2	0.3172	0.4721	0.6982	0.0902	0.9881
Design 3	0.4071	0.7832	0.5432	0.2746	0.9207
Design 4	0.5221	0.8267	0.2136	0.4022	0.8742
Design 5	0.1107	0.6827	0.3341	0.2567	0.9273
Design 6	0.1226	0.8276	0.4456	0.2858	0.9167
Design 7	1.0357	0.7856	0.4452	0.3821	0.8815
Design 8	0.6256	0.7821	0.6254	0.2776	0.9197
Design 9	1.0278	0.7934	0.7221	0.3043	0.9099
Design 10	0.6856	0.5723	0.6456	0.1926	0.9507

8. Hypothesis Testing

It is used to conduct a quantitative research to attempting answer a research question or set of hypothesis. A null hypothesis reflects that there is no significance relationship between two or more parameters whereas alternate hypothesis supports the relationship²⁴. Rejection of a null hypothesis provides a stronger base to accept the relationship or to accept the alternate hypothesis. As this study relates that design defect with object oriented design constructs inheritance, cohesion and coupling following null and alternate hypothesis were made for the purpose of validation of the proposed reliability improvement framework.

- **Null Hypothesis (H₀₁):** Rigidity defect of OO design cannot be mitigating by minimizing inheritance of OO design.

- **Alternate Hypothesis (H₁₁):** Rigidity defect of OO design can be mitigating by minimizing inheritance of OO design.
- **Null Hypothesis (H₀₂):** Rigidity defect of OO design cannot be mitigating by minimizing coupling of OO design.
- **Alternate Hypothesis (H₁₂):** Rigidity defect of OO design can be mitigating by minimizing coupling of OO design.
- **Null Hypothesis (H₀₃):** Rigidity defect of OO design cannot be mitigating by maximizing cohesion of OO design.
- **Alternate Hypothesis (H₀₃):** Rigidity defect of OO design can be mitigating by maximizing cohesion of OO design.
- **Null Hypothesis (H₀₄):** Rigidity defect of OO design cannot be mitigating by minimizing rigidity of OO design.
- **Alternate Hypothesis (H₀₄):** Rigidity defect of OO design can be mitigating by minimizing rigidity of OO design.
- **Null Hypothesis (H₀₅):** Reliability of OO design cannot be maximized by mitigating rigidity defect of OO design.
- **Alternate Hypothesis (H₀₅):** Reliability of OO design can be maximized by mitigating defect of OO design.

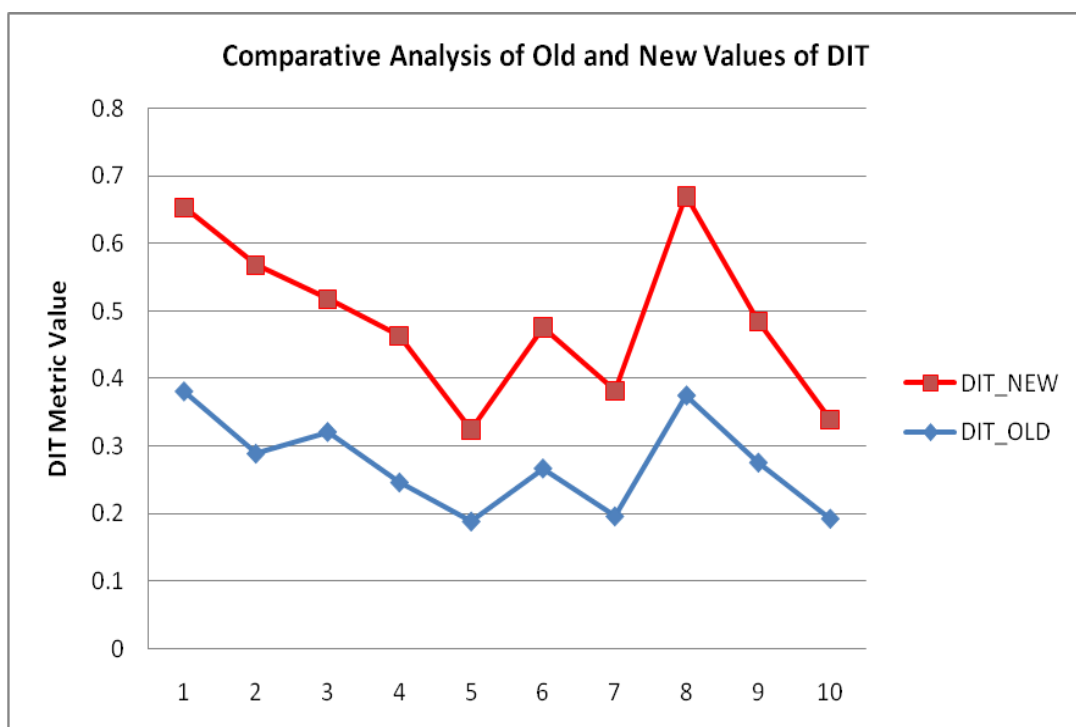


Figure 4: Comparative Analysis of Old and New Value of DIT Metric

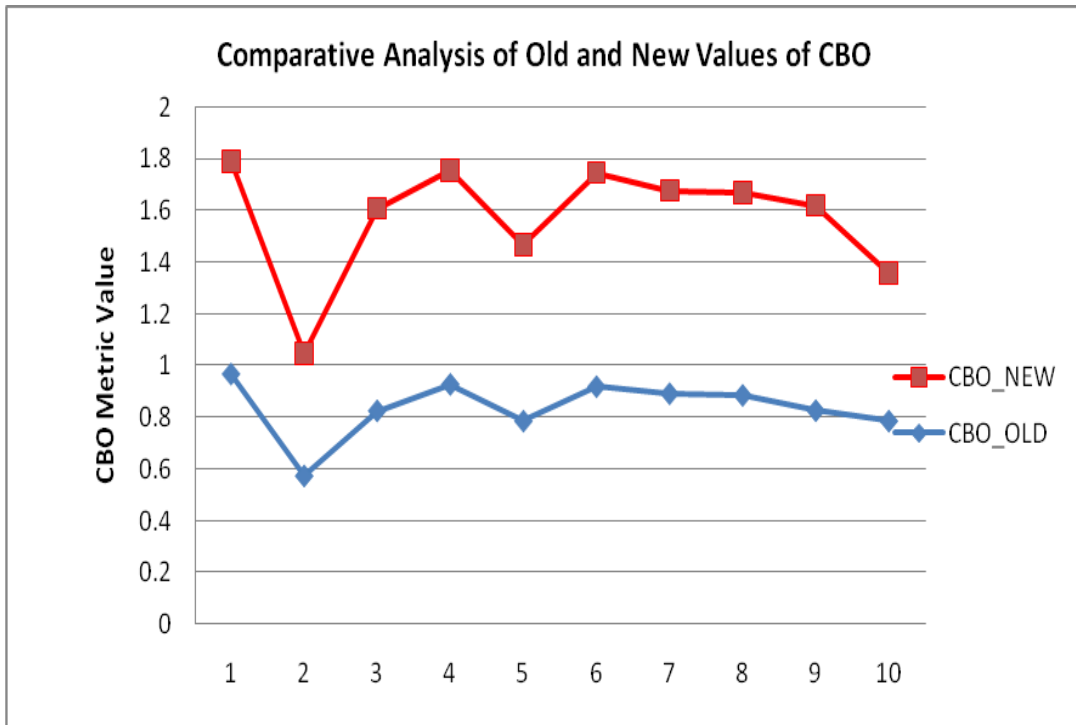


Figure 5: Comparative Analysis of Old and New Value of CBO Metric

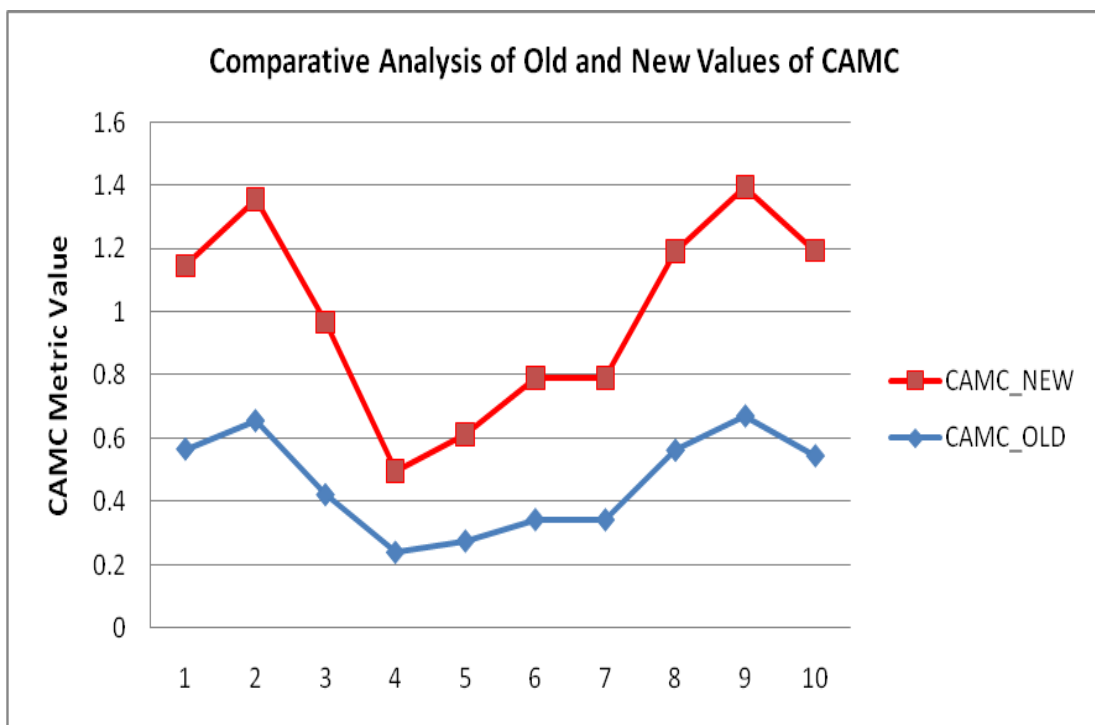


Figure 6: Comparative Analysis of Old and New Value of CAMC Metric

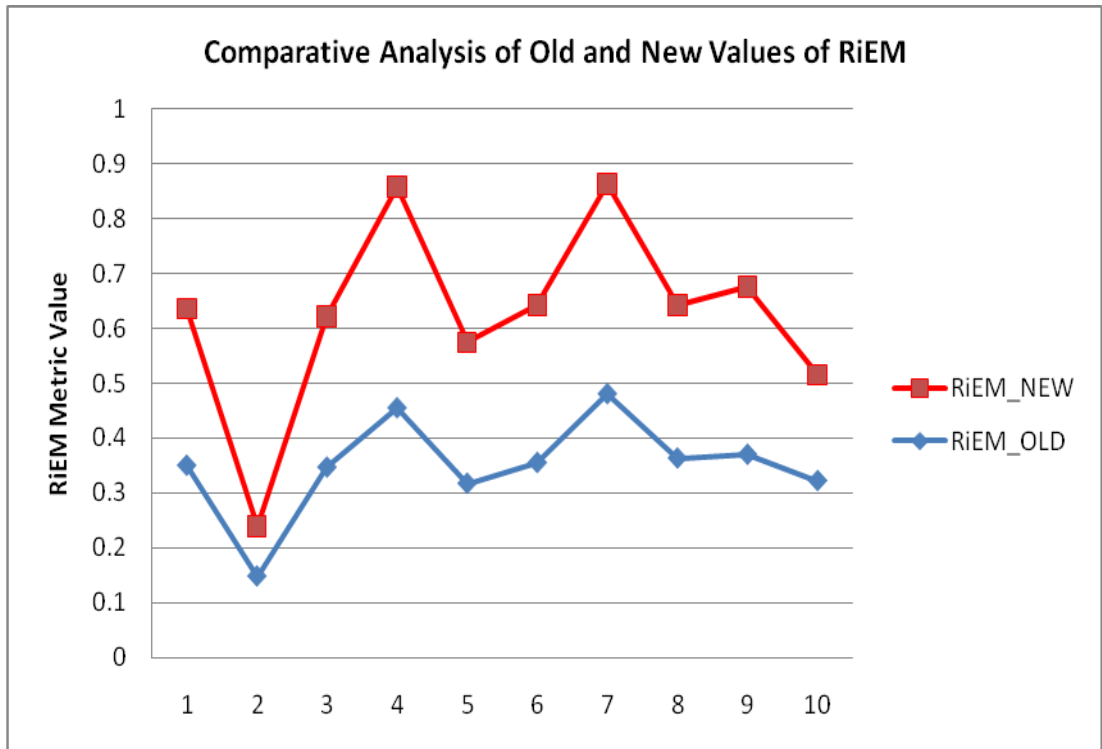


Figure 7: Comparative analysis of Old and New Value of Rigidity Metric (RiEM)

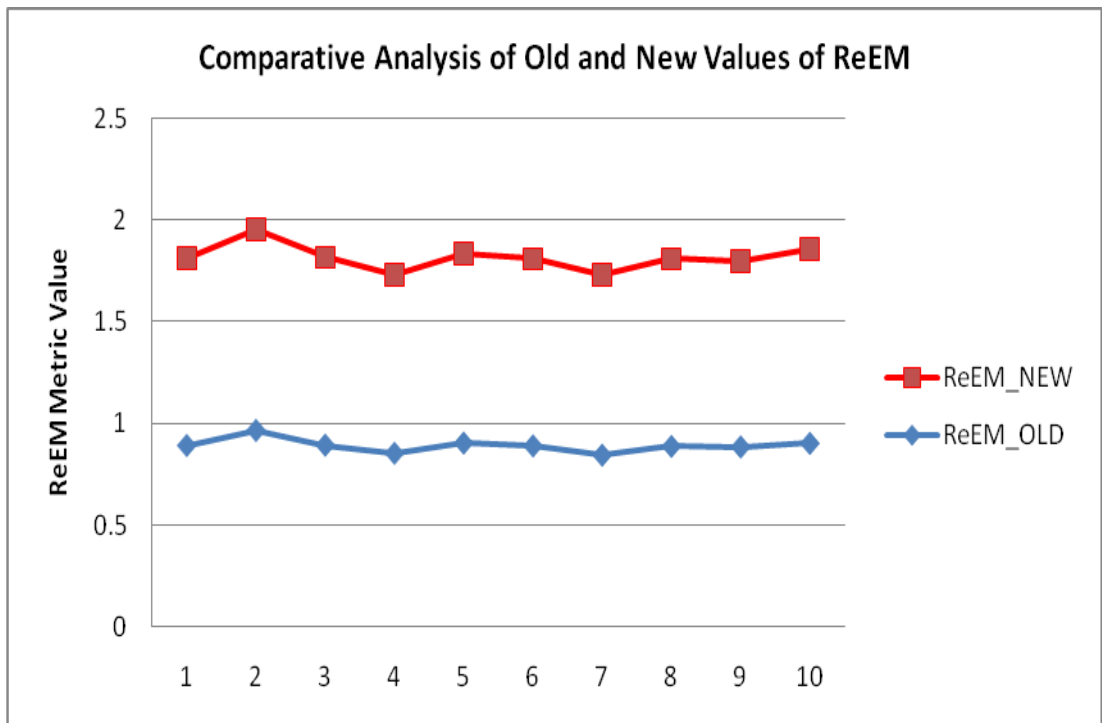


Figure 8: Comparative analysis of Old and New Value of Reliability Metric (ReEM)

9. Significance of the Findings

The role of software in society always increases and leads a series of tragedy and conditions caused by its failures. Unreliability of the software shows that the presence of defect in the software may be simple a matter of life and death at the time when software fails to perform. Various incidents are discussed in chapter 2. There is not any methodology to measure the accuracy of failures and design defects in the software of the design phase remains a difficult problem²³. Still, there is not any accurate definition “What are the factors that directly affect the software reliability”. It is difficult to find the precise direction to estimate software reliability measurement becomes important for software industry, developers and users²⁶. Therefore, a study carried out on improving reliability of OOD by mitigating design defect, with its direct contribution to the field of knowledge, may prove to be significant directly or indirectly in terms of following:

- It may help to improve both the design and architecture information of the software system, which in turn may ease the process of development and maintenance as well.
- It may help to identify and mitigate the design defect in the software design at the early stage of software development life cycle to produce reliable software.
- It may help to quantify the reliability of software and furnish the cost estimation of a software project, which facilitate the estimation and planning of new activities.
- It may help to determine the involvement of object oriented technology with the mitigation of design defects.
- It may also help the designer to develop various alternative designs of the same object oriented software.
- It may also help to determine which one of two versions of object oriented software design is more reliable.
- It may help to find out which object-oriented design has more defects among designs of different software.

Moreover, it is also observed that the contribution of this successful proposed study may provide to be specifically significant in the manner:

- The proposed model based approach may promote and enable the developers to come out with good reliability models with better acceptability.
- The proposed model may provide the basis for the development of new modified or refined approaches.
- The proposed model may find a place among the measurement tools for reliability evaluation of object-oriented software.
- The proposed methodology may encourage other researchers to undertake the development of other new model based approaches.
- The proposed approach may be used to set the benchmark value for reliability estimation by the organization.

10. Future Direction

Research is an on-going process. Reaching one milestone promote the way to the next. As a future research plan, there may the following tasks to be performed:

- The methodology proposed for the computation may be automated so that the models computation can be done automatically and hence not to become tedious for large software.
- Researcher is planning to conduct more experiments on industrial data to draw more concrete conclusions about the implications of the metrics and models proposed.
- The reliability estimation framework for an object oriented design defect can be modified so that it can help to mitigate the defect and improve the reliability of any software design.
- Different execution of the reliability estimation framework for an object oriented design defects are possible. Researchers may implement the same framework by choosing another set of reliability factors and object oriented design constructs to improve reliability.

11. Limitations

Every field have its own advantages and limitations. No work is perfect. In this research, design defects have taken as the main factor of software reliability. When the OOD defects increases reliability of software decreases. Object oriented design is broader area of research, so it is not possible to take all the aspects of object-oriented

design. In addition to successful completion of the research work, the study organized with the following limitations:

- The approach can only be applied to measure and mitigate OOD defects and to improve reliability of object oriented design.
- Due to unavailability of corporate data, the proposed research work is validated with a small set of data.
- The research focuses only on rigidity (a factor of reliability).

12. Thesis Outline

A thesis of the research has been prepared to reflect the detailed study on the research problem and aforesaid research problems.

Chapter -1: Introduction

First chapter is the introduction part of the thesis. The chapter starts with the introduction of reliability and importance of reliability in software development. Some incidents are discussed about cause of software failure in introduction; that needs to conduct the research on software reliability. Second part is introduced about the design defects. The research question is generated related to software reliability at design level. On the basis of these research effects, research problem is formulated. Then objectives of research are framed. At last, each research have its own limitation is discussed.

Chapter-2: Present Approaches

This chapter is associated with the presently existing approaches for mitigation of software defects and improving reliability methods of object oriented software are discussed. An elaborated review on reliability estimation model and design defect from last two decades is presented. The detail review of some significant existing models from the year 2000 to 2014 is introduced.

On the basis of the review, it is recommended that reliability is affected inversely when design defect of software increases. Design defect is identified as a key factor to reliability through literature survey and review. Therefore, there is requirement of reliability estimation by taking the design defect into consideration. In

addition, reliability improvement of an object oriented design with object oriented design constructs is not addressed to design defect.

Chapter-3: Proposed Framework for Reliability Improvement

This chapter presents a framework for improving reliability of object oriented design in respect of object oriented design defect. The chapter covers defect mitigation can be controlled by the object oriented design constructs like inheritance cohesion and coupling. The assumptions of the framework are presented. A framework for reliability improvement of an object oriented design is proposed. This framework consist of four phases including identification phase, mapping phase, measurement phase and improvement phase.

Chapter-4: Execution of the Framework

Implementation of the framework is presented in this chapter. According to the framework design defect is a key factor of reliability. In first phase of the framework, object oriented design defects, object oriented design constructs and reliability factors are identified. In the second phase, mapping between object oriented designs construct and design defects are established and then object oriented design defects and reliability is established.

To establish the relation, Depth of Inheritance Tree (DIT) is used for Inheritance, Coupling between Objects (CBO) is used for coupling and Cohesion among Methods in Class (CAMC) is used for cohesion. Rigidity Estimation Model (RiEM) and Reliability Estimation Model (ReEM) are proposed to measure the design defect and reliability of object oriented design.

Chapter-5: Validation of Framework

This chapter presents the theoretical and empirical validation of the proposed framework. As an experimental validation, pre tryout is carried out on an object oriented design. On the basis of the reliability improvement framework, metrics and models values are computed according to the requirement. On the base of reliability improvement guidelines of object oriented design (^{RI}GOOD) design defect of the object oriented design is mitigate.

After redesign, the metrics and model values are recomputed. It is then verified that the DIT and CBO values are minimized and CAMC values are maximized. After dealing with ^{RI}GOOD on ReEM, the value of design defect was found less than the values of previous design. For the tryout purpose, ten object oriented designs are accepted. The same process is reproduced and is resolved that the approach performs well in this experiment also. Various statistical studies and hypothesis test carried out for the acceptability of the framework.

Chapter-6: Conclusion and Future Work

It is the last chapter of the thesis. In this chapter, major research findings along with the other findings are presented. The research questions of the first chapter are treated one-by-one in this chapter. Significance of the research is also discussed in details. Future plans for extending the study are discussed.

References

1. M. R. Lyu, "Handbook of software reliability engineering", IEEE Computer Society Press, Los Alamitos, California, Washington Brussels Tokyo, April 1996, pp. 1-26, ISBN: 0070394008.
2. S. R. Dalal, M. R. Lyu, and C. L. Mallows, "Software Reliability", Available at: <http://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat05035/abstract>
3. M. L. Shooman, "Yes, software reliability can measure and predicted", Proceedings of the 1987, Fall Joint Computer Conference on Exploring Technology, IEEE Computer Society, 1987, ISBN: 0-8186-0811-0.
4. G. J. Myers, "The art of software testing", Wiley New York, 1979.
5. IEEE Standard Glossary of Software Engineering Terminology, ANSI/IEEE Standard 729, 1991.
6. J. D. Musa, "Software reliability: measurement, prediction, application", Professional Edition, McGraw Hill Publishers, New York 1988.
7. J. D. Musa, "Software reliability engineering: more reliable software faster and cheaper", McGraw Hill, 2004.
8. J. L. Lions, "Ariane-5 Flight-501 Failures-Report by the enquiry Board", 2010, Available at: <http://www.di.unito.it/damiani/ariane5rep.html>.
9. M. Fowler, "Refactoring: Improving the Design of Existing Code", Addison-Wesley, New York, USA, ISBN-13: 9780201485677, pp. 431.
10. T. Mens, and T. Tourwe, "A Survey of Software Refactoring", IEEE Transactions, Software Engineering, Vol. 30, pp. 126-139.
11. Y Shu, H. Liu, Z. Wu, and X. Yang, "Modeling of software faults detection and correction process based on the correction log", Information Technology, Vol. 8, pp. 735-742.
12. A. J. Riel, "Object oriented design heuristics", Addison-Wesley Longman Publishing Co. Inc. Boston, MA, USA, pp. 379, ISBN-13: 9780201633856.
13. W.J. Brown, "Anti-Patterns: refactoring software", Architecture and Projects in Crisis Wiley, USA, 1998, pp. 309, ISBN-13: 9780471197133.
14. R. Subramanyam, M.S Krishnan, "Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects", IEEE Transactions on Software Engineering, 2003, Vol. 29, No. 4, pp. 297-310.
15. M. Fowler, "Refactoring: Improving the design of existing code", Addison-Wesley, New York, USA, 1999.

16. A. Bertolino, "Software testing research: achievements, challenges, dreams", Future of Software Engineering, 2007. FOSE '07, 23-25 May 2007, pp. 85 – 103.
17. C. Wohlin, "Estimation of Software Reliability Growth Models", Blekinge Institute of Technology, Sweden, 2007, pp. 1-3
18. F. B Abreau and R. Carapuca, "Candidate metrics for Object Oriented software within a taxonomy framework" Journal of Systems and Software, Vol. 26, No 1, July 1994, pp.1-6.
19. D. Arora, P. Khanna, A. Tripathi, S. Sharma and S. Shukla, "Software quality estimation through object oriented design metrics" IJCSNS, International Journal of Computer and Network Security, Vol.11, No.4, April 2011, pp. 100-104.
20. R. A. Khan, "Quality Estimation of Object Oriented Code: A Design Metrics Perspective" Ph.d Thesis, 2004, 2004, pp. 437-444.
21. P. K. Chaurasia, R. A Khan, "Software Models for Software Quality", National Conference on Information Security Challenges (NCISC-2016), organized by Department of Information Technology, BBAU, Lucknow 24th February 2016.
22. P. K. Chaurasia, "Security Breaches from Categorization of Software Errors", National Seminar on Emerging Trends and Advancements in Cyber Security, Organized by Department of Computer Application, Integral University, Lucknow on 04 April 2016.
23. A. Yadav, R. A. Khan, "Reliability Estimation Framework-Complexity Perspective", Computer Science and information Technology, Vol. 2 no. 5, 2012, pp. 97-104.
24. R. A. Khan, A. Agrawal, "Software Engineers: A Practitioners Approach", Alpha Science International, May 2014.
25. F. Huang, and B. Liu, "Software defect prevention based on human error theories", Chinese Journal of Aeronautics", Vol. 30, Issue 3, June 2017, pp.1054-1070.
26. D. Amara, L. B. A. Rabai, "Towards a new framework of software reliability measurement based on software metrics", Procedia Computer Science, Vol. 109, 2017, pp. 725-730.
27. Y. Ligu, R. Stephen, K.C. Syhach and S. Ramaswamy, "Coupling measurement in multi kernel based software with its application to Darwin", International Journal of Intelligent Control and systems, Vol. 13, No.2, 2008, pp: 109-119.
28. A. Nicolaescu, H Lichter and Yi Xu, "Evolution of Object Oriented Coupling Metrics: A Sampling of 25 Years of Research" IEEE/ACM 2nd International workshop on Software Architecture and Metrics (SAM), May 2015, pp. 48-54.

29. L. Shanmugam & L. Florence, "An Overview of Software Reliability Models", IJARCSSE, Vol. 2, Issue 10, October 2012, pp. 36-42.
30. P. K. Chaurasia, R.A. Khan, "Reliability Improvement Framework: Defect Mitigation Perspective", International Journal of Advance Research in Computer Science, Vol.8, No.9, Nov-2017. (Accepted)

13. List of Publications (Journal and Conferences)

1. P. K. Chaurasia and R. A Khan, "Classification of Software Requirement Errors: A Critical Review", International Journal of Computer Applications (IJCA), Vol. 132, No-07, pp. 09-15, 2015.
2. P. K. Chaurasia and R. A. Khan, "Relationship between Object Oriented Design Constructs and Design Defects", International Journal of Computer Application and Management (IJCAM), Vol.7, No. 9, pp: 35-38, 2017. ISBN 2231-1009. Available at: http://ijrcm.org.in/article_info.php?article_id=8018
3. P. K. Chaurasia, R.A. Khan, "Reliability Improvement Framework: Defect Mitigation Perspective", International Journal of Advance Research in Computer Science, Vol.8, No.9, Nov-2017. (Accepted)
4. P. K . Chaurasia, R. A. Khan, "Exploring Object Oriented Design Metrics", International Conference on Nano-Science and Nano-Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, 22-24 Sep, 2017.
5. P. K Chaurasia, R. A. Khan, "Identification of Error Prone Path for Improving Software Testing: Genetic Algorithm Approach, Conference Proceeding National Conference on Recent Advances in Chemical and Material Sciences, 23-24, February, 2015, ISSN 978-93-84224-257, pp 52-54, organized by MMMUT, Gorakhpur.
6. P. K. Chaurasia, R. A Khan, "Software Models for Software Quality", National Conference on Information Security Challenges (NCISC-2016), organized by Department of Information Technology, BBAU, Lucknow 24th February 2016.
7. P. K. Chaurasia, "Security Breaches from Categorization of Software Errors", National Seminar on Emerging Trends and Advancements in Cyber Security, Organized by Department of Computer Application, Integral University, Lucknow on 04 April 2016.