Defects Avoidance in E-commerce Projects using Rayleigh Method

R. Sudarshan, S. K. Srivatsa

Abstract: With the advent of e-commerce business, many software providers have started developing e-commerce solutions. However, the emphasis has to be given on developing defect free e-commerce portals. The current work is produced based on quantitative analysis done on five e-commerce projects. The defect data (discovery, fix and regression) are fed into the development process of the subsequent projects that helped in preventing the similar defects. A data analysis is done using Rayleigh Method to estimate the number of latent defects which may seep to the customers. This enabled in reduction in the count of User Acceptance Test defects, in the projects developed subsequently there by greatly decreasing the cost of development.

Index Terms: defect origin and detection, defect removal efficiency, defect metric analysis, e-commerce Web portal.

I. INTRODUCTION

Software reliability models are utilized to assess a software product's reliability or to approximate the number of dormant defects when it is released to the customers. The measured parameter under study is the number of defects in specified time intervals (weeks, months, etc.) as per Kan [2]. In this work, Dynamic Reliability Model namely Rayleigh Model is used since it could be adopted for the entire development process.

A. Rayleigh Model

This model is found to be appropriate for envisaging dependability of software product. It envisages the likely value of defect density at different phases of life cycle of the project. This is made possible by the parameters (total number of defects and peak of the curve in terms of unit of time in the development life cycle). These factors help in drawing the required curve. The Probability Distribution Function (PDF) is given as [4]:

\[ F(t_a) = f(CD, t_p, t_a) \]

where \( t_p \) is the peak of the curve, \( t_a \) is actual time unit and \( CD \) is cumulative defect density.

Process control can be established using these predictable values of defect density at different stages of testing phase.

II. SOFTWARE DEFECT ANALYSIS

This examination is built on the study of defects that are captured in different phase of the software development. They included Requirements Analysis, Design (High Level and Low Level), Coding (with Unit Testing), Integration Testing and User Acceptance Testing. Effective defect elimination reduces development cycle time, thus increasing the product quality [5].

A. Methods Adopted for Defect Removal

In all the projects static code analysis tools were used to ascertain the defect levels. All projects included reviews of requirements documents, design documents, test cases and code driven by check lists. Peer inspection processes (design inspection and code inspection) were also introduced to eliminate any possible defects. Test cases were reviewed and inspected by the developer. About 70% of defects that test cases were intended to “catch” were uncovered prior to test-case execution. Testing was chiefly carried out to check defect removal, not to discover defects. All defects were traced during the course of the development in all phases (injection, detection, and fixing of defects). The development and testing teams implemented detailed planning for every upcoming code release cycle and aligned with the overall schedule [6]. Development team used their authentic data to plan succeeding work and came to an agreement on the schedule, process adopted, and resources required. This majorly helped to hit the delivery schedule without compromising the quality [7].

B. Origin and Discovery of Defects

Table I shows the techniques implemented for discovering defects within a specific stage [8].

| Table I: Defect origin and discovery methods adopted |
|----------------------------------------------------|
| Defect Source | Defect Discovery Techniques |
| Requirements defects | Prescribed requirements reviews |
| Design defects | Prescribed design reviews |
| Coding defects | Static analysis Unit testing |
| Bad fixes | Re-inspection after the fix Regression Testing |
| Test case defects | Formal test case inspections |
| Integration test defects | Re-inspection after the fix Regression Testing |
| Document defects | Formal document inspections |

Analysis of Defect Data

The current work has been carried-out on five e-commerce projects that are developed in sequence by adopting ATG e-commerce development platform. They are denoted as Project A, Project B, Project C, Project D and Project E. The size of each project was around 90 Executable Kilo Line of Code (EKLOC) and in terms of effort it was around 3000 person days [9]. Time scale with reference to formula (1) that is adopted in this study for conducting non-linear regression analysis is given in Table II.
Defects Avoidance in E-commerce Projects using Rayleigh Method

Table II: Development Phase/Time Scale Factor

| Development Phase          | Time Scale |
|----------------------------|------------|
| Requirements Analysis      | 0.05       |
| Design                     | 0.15       |
| Construction               | 0.25       |
| Unit Testing               | 0.35       |
| Component Testing          | 0.45       |
| Integration Testing        | 0.55       |
| User Acceptance Testing    | 0.65       |

Defect density at Life cycle stage \( t = T_{UT}(At \ Unit \ Testing) \) is given by \( K \times Y \) \( (at \ t = T_{UT}(At \ Unit \ Testing)) \). Consequently, defect density is projected at any stage of the project by replacing the value of “t” and getting the “Y” axis value, multiplied by K factor.

The criticality of the defect is established on the impact of it, in terms of schedule, risks, debugging time etc. Whereas some defects are insignificant, others have a major bearing on functionality. To regulate the effects of such defects, a “severity factor” has been introduced to fine-tune the defect tally to be more accurate for the metrics purposes.

In this work, the source of a defect is mapped to the developmental phase to which it belongs to. Each defect is given a weight based on the criticality of impact on the operation of the software. Table II lists the classification of defects based on severity that is adopted in the software development organization.

Table II: Classification of defect data based on criticality

| Type of Criticality | Weightage Factor | Remarks |
|---------------------|------------------|---------|
| Show Stopper        | 10               | These are the extremely severe defects, which have the potential to halt or already halted business system. |
| Critical            | 8                | These are severe defects, which have not stopped the application, but have seriously hampered the performance of many business functionality. |

High | 7 | These defects have an impact on the functionality of some modules |

Medium | 5 | These are defects which have an adverse effect on the general transactions |

Low | 2 | These defects are primarily related to Look and Feel feature (cosmetic) |

Every project’s defect data (logged for Inspection and Testing) is presented in Tables III to VII [9]. A matrix method is adopted by cross-classifying defect data mapped to the stage in which they are found and the stage in which they are introduced [2].
### Table III: Defect data of Project A

| Phase       | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total | Show Stopper | Critical | High | Medium | Low | Weighed Number of Defects |
|-------------|-----------------------|--------|------------------------|---------------------|-------|--------------|----------|------|--------|-----|--------------------------|
| Analysis    | 122                   | 90     | 104                    | 112                 | 112   | 21           | 11       | 35   | 11     | 54  | 296                      |
| Design      | 18                    | 90     | 108                    | 118                 | 118   | 27           | 18       | 33   | 30     | 0   | 295                      |
| Code with UT| 78                    | 109    | 243                    | 823                 | 841   | 24           | 45       | 81   | 117    | 162 | 2076                     |
| Integration Testing | 10          | 12     | 363                    | 459                 | 445   | 45           | 72       | 81   | 281    | 365 | 3728                     |
| UAT         | 231                   | 112    | 212                    | 565                 | 587   | 93           | 81       | 56   | 155    | 180 | 3105                     |
| Total       | 469                   | 332    | 818                    | 459                 | 2078  | 210          | 227      | 286  | 594    | 761 | 10410                    |

### Table IV: Defect data of Project B

| Phase       | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total | Show Stopper | Critical | High | Medium | Low | Weighed Number of Defects |
|-------------|-----------------------|--------|------------------------|---------------------|-------|--------------|----------|------|--------|-----|--------------------------|
| Analysis    | 261                   | 100    | 362                    | 622                 | 622   | 26           | 26       | 45   | 63     | 83  | 1440                     |
| Design      | 21                    | 100    | 120                    | 240                 | 240   | 6            | 12       | 27   | 39     | 45  | 630                      |
| Code with UT| 48                    | 63     | 251                    | 362                 | 362   | 25           | 42       | 75   | 108    | 112 | 1875                     |
| Integration Testing | 65          | 73     | 260                    | 247                 | 467   | 54           | 109      | 112  | 180    | 228 | 2572                     |
| UAT         | 155                   | 18     | 90                     | 261                 | 261   | 18           | 31       | 61   | 49     | 102 | 1304                     |

### Table V: Defect data of Project C

| Phase       | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total | Show Stopper | Critical | High | Medium | Low | Weighed Number of Defects |
|-------------|-----------------------|--------|------------------------|---------------------|-------|--------------|----------|------|--------|-----|--------------------------|
| Analysis    | 342                   | 207    | 240                    | 561                 | 581   | 27           | 30       | 66   | 107    | 112 | 1751                     |
| Design      | 33                    | 207    | 240                    | 561                 | 561   | 12           | 31       | 54   | 51     | 92  | 1185                     |
| Code with UT| 36                    | 45     | 154                    | 255                 | 255   | 13           | 21       | 40   | 71     | 90  | 1113                     |
| Integration Testing | 50            | 18     | 300                    | 310                 | 360   | 37           | 72       | 87   | 160    | 205 | 2765                     |
| UAT         | 63                    | 4      | 45                     | 112                 | 112   | 9            | 13       | 31   | 27     | 32  | 610                      |
| Total       | 524                   | 274    | 302                    | 1490                | 1490  | 98           | 107      | 278  | 416    | 551 | 7404                     |

### Table VI: Defect data of Project D

| Phase       | Requirements Analysis | Design | Coding | Integration Testing | Total | Show Stopper | Critical | High | Medium | Low | Weighed Number of Defects |
|-------------|-----------------------|--------|--------|---------------------|-------|--------------|----------|------|--------|-----|--------------------------|
| Analysis    | 405                   | 295    | 260    | 460                 | 460   | 25           | 36       | 72   | 125    | 146 | 1959                     |
| Design      | 35                    | 295    | 260    | 460                 | 460   | 11           | 27       | 52   | 59     | 111 | 1207                     |
| Coding with UT| 35            | 133    | 205    | 470                 | 470   | 14           | 19       | 35   | 59     | 78  | 468                      |
| Integration Testing | 45          | 15     | 250    | 160                 | 410   | 27           | 55       | 58   | 145    | 185 | 2211                     |
| UAT         | 51                    | 6      | 35     | 92                 | 92    | 7            | 11       | 27   | 22     | 25  | 507                      |
| Total       | 569                   | 285    | 418    | 1432                | 1432  | 84           | 148      | 244  | 410    | 545 | 6872                     |

### Table VII: Defect data of Project E

Retrieval Number: D8538118419/2019©BEIESP
Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

![IJRTE Logo](https://www.ijrte.org)
DATA ANALYSIS BY ADOPTING RAYLEIGH MODEL

As discussed in Section II, the Rayleigh method is applied on the data shown in Table III to Table VII, for the Project A, Project B, Project C, Project D, and Project E respectively. The graphs generated using SAS (Analytics Software & Solution) is given in Figure 1 to Figure 5.

Figure 1: Rayleigh plot of Project A

Figure 2: Rayleigh plot of Project B

Figure 3: Rayleigh plot of Project C

Figure 4: Rayleigh plot of Project D

Figure 5: Rayleigh plot of Project E

Figure 6 gives the comparison study of the defects of all the five projects in different phases of development.

Figure 6: Rayleigh plot of consolidated data of all the five projects

The area of graph for each project is increasingly reduced indicating the lesser defects being produced from Project A to Project E.
III. COST OF FIXING DEFECTS

According to Ljubomir Lazic [10] the cost of defect removal is calculated as follows:

\[ TRC = RD \times CDR \tag{2} \]

Where TRC is Total Removal Cost, 
RD is Removed Defects and 
CDR is Average Cost of Defect Removal (for each phase).

Based on Equation 2 the TRC is calculated for each project and is shown in Tables VIII through XII. As per the industry standard the billing rate to fix a defect is $50. The TRC is multiplied by this factor to arrive at the actual cost to fix the defects in each phase.

Table XIII gives the summary of Total Defects and the Total Cost to fix them for each project.

Table XIII: Total Defects and Total Cost to Fix

| Project | Total Defects | Total Cost to fix the defects (in $) |
|---------|---------------|-------------------------------------|
| A       | 2078          | 19647                               |
| B       | 1706          | 12232                               |
| C       | 1490          | 7962                                |
| D       | 1432          | 6793                                |
| E       | 1340          | 5255                                |

The total cost has shown a steady decline from Project A (19 lakh $) to Project E (5 lakh $) marking a significant improvement. This is shown in Figure 7.

IV. CONCLUSION

Defects origin and their prevention from each stage, namely, Requirements Analysis, Design, Coding (with Unit Testing), Integration Testing and User Acceptance Testing was analyzed for e-commerce project development. Defect data was collected at each stage. The defects were classified according to the origin of defects (as per the phases). Further, they were treated with a Weight, based on their severity of impact. It was observed with the help of Rayleigh graphs that review of Design (High Level and Low Level) was the major contributor for considerable amount of defects. The process of review of these documents was improved from Project A to Project E. This helped in a progressive improvement in achieving lesser number of defects, from the initial project to the later project. Hence the cost of fixing defects showed a decline, impacting positively on the development of the projects.
## Table VIII: Total Cost Removal for Project A

| Phase                | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total Defects | Average Cost of removing a detected defect (cost units) | Total Cost |
|----------------------|-----------------------|--------|------------------------|---------------------|---------------|--------------------------------------------------------|------------|
| Requirements Analysis| 132                   |        |                        |                     | 132           | 1                                                     | 6600       |
| Design               | 18                    | 90     |                        |                     | 108           | 2.5                                                   | 13500      |
| Code with UT         | 78                    | 108    | 243                    |                     | 429           | 6.5                                                   | 139425     |
| Integration Testing  | 10                    | 12     | 363                    | 459                 | 844           | 16                                                    | 675200     |
| UAT                  | 231                   | 122    | 212                    | 459                 | 565           | 40                                                    | 1130000    |
| Total                | 469                   | 332    | 818                    | 459                 | 2078          |                                                        | 1964725    |

## Table IX: Total Cost Removal for Project B

| Phase                | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total Defects | Average Cost of removing a detected defect (cost units) | Total Cost |
|----------------------|-----------------------|--------|------------------------|---------------------|---------------|--------------------------------------------------------|------------|
| Requirements Analysis| 261                   |        |                        |                     | 261           | 1                                                     | 13050      |
| Design               | 21                    | 108    |                        |                     | 129           | 2.5                                                   | 16125      |
| Code with UT         | 48                    | 63     | 251                    |                     | 362           | 6.5                                                   | 117650     |
| Integration Testing  | 63                    | 23     | 360                    | 247                 | 693           | 16                                                    | 554400     |
| UAT                  | 153                   | 18     | 90                     |                     | 261           | 40                                                    | 522000     |
| Total                | 546                   | 212    | 701                    | 247                 | 1706          |                                                        | 1223225    |

## Table X: Total Cost Removal for Project C

| Phase                | Requirements Analysis | Design | Code with Unit Testing | Integration Testing | Total Defects | Average Cost of removing a detected defect (cost units) | Total Cost |
|----------------------|-----------------------|--------|------------------------|---------------------|---------------|--------------------------------------------------------|------------|
| Requirements Analysis| 342                   |        |                        |                     | 342           | 1                                                     | 17100      |
| Design               | 33                    | 207    |                        |                     | 240           | 2.5                                                   | 30000      |
| Code with UT         | 36                    | 45     | 154                    |                     | 235           | 6.5                                                   | 76375      |
| Integration Testing  | 50                    | 18     | 303                    | 190                 | 561           | 16                                                    | 448800     |
| UAT                  | 63                    | 4      | 45                     |                     | 112           | 40                                                    | 224000     |
| Total                | 524                   | 274    | 502                    | 190                 | 1490          |                                                        | 796275     |
Table XI: Total Cost Removal for Project D

| Phase                | Requirements Analysis | Design | Coding | Integration Testing | Total Defects | Average Cost of removing a detected defect (cost units) | Total Cost |
|----------------------|-----------------------|--------|--------|---------------------|---------------|--------------------------------------------------------|------------|
| Requirements Analysis| 405                   |        |        |                     | 405           | 1                                                      | 20250      |
| Design               | 35                    | 225    |        |                     | 260           | 2.5                                                    | 32500      |
| Coding with UT       | 33                    | 39     | 133    |                     | 205           | 6.5                                                    | 66625      |
| Integration Testing  | 45                    | 15     | 250    | 160                 | 470           | 16                                                     | 376000     |
| UAT                  | 51                    | 6      | 35     |                     | 92            | 40                                                     | 184000     |
| Total                | 569                   | 285    | 418    | 160                 | 1432          |                                                        | 679375     |

Table XII: Total Cost Removal for Project E

| Phase                | Requirements Analysis | Design | Coding | Integration Testing | Total Defects | Average Cost of removing a detected defect (cost units) | Total Cost |
|----------------------|-----------------------|--------|--------|---------------------|---------------|--------------------------------------------------------|------------|
| Requirements Analysis| 427                   |        |        |                     | 427           | 1                                                      | 21350      |
| Design               | 45                    | 252    |        |                     | 297           | 2.5                                                    | 37125      |
| Coding with UT       | 27                    | 31     | 120    |                     | 178           | 6.5                                                    | 57850      |
| Integration Testing  | 39                    | 11     | 200    | 139                 | 389           | 16                                                     | 311200     |
| UAT                  | 27                    | 4      | 18     |                     | 49            | 40                                                     | 98000      |
| Total                | 565                   | 298    | 338    | 139                 | 1340          |                                                        | 525525     |
REFERENCES

1. https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, para. 2, page 4 [Accessed: Aug 22, 2018]
2. Stephen. H. Kan, “Metrics and Models in Software Quality Engineering”, Second Edition, Pearson Education, 2004, pp 164-165, pp 187-188.
3. Capers Jones, “Software Engineering Best Practices – Lessons from Successful Projects in Top Companies” 2010 Edition, McGraw-Hill, pp 124-128.
4. https://www.scientificbulletin.upb.ro/rev_docs_arhiva/full18273.pdf, page 57, [Accessed: Oct 31, 2019]
5. B. Clark and D. Zubrow, “How good is the software: a review of defect prediction techniques”, Software Engineering Symposium, IEEE Computer Press, Pittsburgh, PA, 2001, pp. 1–35
6. https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, para. 3, page 13 [Accessed: Dec 22, 2017]
7. R. Sudarshan, Dr. S. K. Srivatsa. (2017, December). “A study on Defect Prevention in E-commerce Web Sites”. IJETCS [Online], pp. 163-167. [Accessed: https://www.ijetcs.org/Volume6Issue6/IJETCS-2017-12-23-49.pdf]
8. R. Sudarshan, Dr. S. K. Srivatsa. (2019, July), “A Quantitative Analysis of Defect Prevention in E-commerce Projects”. IJRTE [Online], pp. 1335-1361. Available: https://www.ijrte.org/wp-content/uploads/papers/v8i2/B2004078219.pdf
9. R. Sudarshan, Dr. S. K. Srivatsa. (2018, September). “A study on Stages of Defects Injection and possible methods of their avoidance in E-commerce Web Sites”. IJAEM [Online], pp. 14-24. Available: https://www.ijaem.org/Volume7Issue9/IJAEM-2018-09-23-5.pdf
10. http://www.comsis.org/pdf.php?id=218-0909 [Accessed 2 November 2019]

AUTHORS PROFILE

Sudarshan R is a Bachelor of Engineering in Computer Science, and M. Tech in Information Technology. He has over 19 years of experience in IT industry that includes many CMMI companies. He has experience in the area of Software Delivery & Software Quality Assurance. He is pursuing his Phd in VELS University, Chennai.

Dr. S.K. SRIVATSA received the Bachelor of Electronics and Telecommunication Engineering degree from Jadavpur University, Calcutta, India. Master’s degree in Electrical Communication Engineering and Ph.D from the Indian Institute of Science, Bangalore, India. He was a Professor of Electronics Engineering in Anna University, Chennai, India. His current research activities pertain to computer networks, Design and Analysis of algorithms, coding Theory and Artificial Intelligence & Robotics. He has produced seventy PhDs’ and is the author of over 750 publications.