International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 6, November- December 2017 ISSN 2278-6856

A study on Defect Prevention in E-commerce Web Sites

Sudarshan R[1] , Dr. S. K. Srivatsa[2]

[1]Research Scholar, Department of Computer Science and Engineering, VELS University, Chennai 600 117, India

[2]Former Senior Professor, Department of Computer Science and Engineering, Anna University, Chennai 600 025, India

Abstract

E-Commerce portal development includes a number of integrations like Vendor Information, Buyer Information and Authentication, Inventory, Payment Gateways, Delivery Logistics etc., hence it becomes very important to minimize defects at the end of each phase of developmental activity. This paper is an attempt to look at one of the methods to minimize the defects in E-Commerce sites.

Keywords: E-Commerce, Defects, Cohesion, Dependency Matrix

1. INTRODUCTION

E-commerce is a transaction of buying or selling online. It is a methodology of modern business which addresses the need of business organizations, vendors and customers to reduce cost and improve the quality of goods and services while increasing the speed of delivery. [1] In India gradually the trend of doing electronic business is increasing and there are several software service companies claiming to develop and deliver E-Commerce web portals.

2. DEFECT PEDAGOGY

Defect: An imperfection or deficiency in a work product where that work product does not meet its requirements or specifications and needs to be either repaired or replaced, as per IEEE Std. 1044-2009.

According to Software Engineering Institute, a defect in software engineering parlance is any flaw or imperfection in a software work product or software process [2]. A software work product is any artifact created as part of the software process. Defects are injected when a product is created (design and code) and defects are removed during review, inspection, and test activities. These defects when unattended would cause failure to the product and risk to the users. The increasing use of software by all spheres of society has put more onuses on the defect free software product. When a defect gets through during the development process, the earlier it is diagnosed, the easier and cheaper is the rectification of the defect. The end result in prevention or early detection is a product with zero or minimal defects [2].

The Systems Sciences Institute at IBM has reported that the cost to fix a defect found after product release was four to five times as much as one uncovered during design, and up to 100 times more than one identified in the maintenance phase as shown in Figure 1 [3].

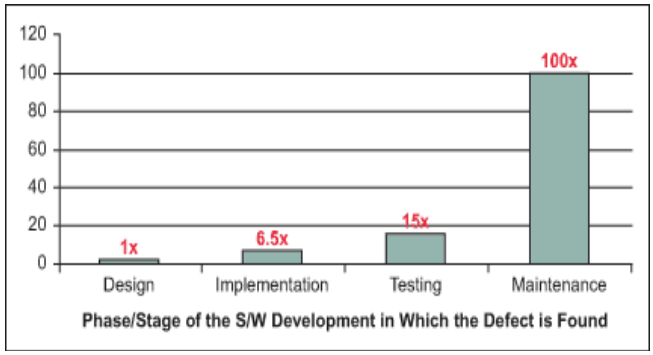

Figure 1 Fixing Cost of a Defect in Different Stages

Since the E-Commerce portal development includes a number of integrations like Vendor Information, Buyer Information and Authentication, Inventory, Payment Gateways, Delivery Logistics etc., it becomes very important to minimize defects at the end of each phase of developmental activity.

3 E-COMMERCE SYSTEM DEVELOPED USING RATIONAL UNIFIED PROCESS (RUP) MODEL

The **Rational Unified Process** (**RUP**) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM [4].
The important modules of E-commerce system developed using Rational Unified Process (RUP) Model are [4]:

1. Membership & Account Management
2. Catalog
3. Search
4. Browse & Shop
5. Pricing & Promotions
6. Order Management
7. Checkout & Payment
8. Shipping & Delivery
9. Integrations

Salient feature of each of the modules is listed below [5].

3.1 Membership & Account Management

- Membership Accounts can be created, managed and searched.
- Create and view any returns

- View previous orders including status of orders and line items
- View account and invoice information including the ability for customers to pay for invoices by any of the online payment option or Cash On Delivery
- Manage, view and print recent transaction statements
- Request "email me" when back in stock
- Quick reorder from previous orders
- Create and save baskets

3.2 Catalog

- CSV and XML import and export of products, product data
- Image management, auto resizing and generation of thumbnails
- Manage master products, variations, related products and multiple price lists
- Integration with 3rd party warehousing system feeding live stock levels
- Define different product types such as gift vouchers, subscriptions, memberships, digital products and events

3.3 Search

- Fast site-wide searches returning accurate results from all content
- Multi-faceted search allows customers to sort, apply and remove filters on the results
- Sorting (price, name, new products, star rating, pre-defined metrics – or other fields specific to the context)
- Pagination (customer can also choose the number of products to display at any one time)
- Filtering (customer can filter the list of products by applying filter options)
- Promotional areas for ads and banners. These can be set to display against specific search terms and can be configured to display offers, products or even content pages

3.4 Browse & Shop

- Multi-faceted and layered navigation for advanced filtering and sorting based on user defined criteria for search results and for product categories
- Multiple product images with high resolution product image zooming capability
- Stock availability information is fetched and displayed from different vendor sites
- Product comparison information is facilitated by referring different vendor sites
- Product reviews and ratings are obtained from different sites and presented to the customer
- Relevant promotions and related product cross-sells information is fetched and displayed on product pages and in the shopping basket

3.5 Pricing & Promotions

- Handles complex pricing models that includes gross/net pricing, previous price, tax structure etc.
- Personalized shipping pricing
- Supports quantity based pricing
- Configurable rules based promotions and offers
- Session specific offers; can be triggered by a customer filing out a feedback form, referring a friend etc
- Discounts include, % off an order, % off a product(s)
- Buy x get y Free
- Free promotional products
- Free shipping
- Bundles: Buy 2 get 1 Free, 3 for Rs. 900 etc
- Product inclusions and/or exclusions based on the pattern of search
- Allowing Coupons / vouchers
- Set start and end dates and timing for promotions

3.6 Order Management

- Powerful search to find orders quickly and easily based on multiple criteria
- Manage status and append notes to orders down to line item level
- User defined order workflows for handling different payment and shipping methods
- Print orders, invoices and packing slips
- Process returns and raise credit notes
- Report on abandoned baskets / orders
- Audit payment details such as paid or unpaid, payment method, receipt number, receipt value, transaction reference

3.7 Checkout & Payment

- Customizable, streamlined and guided checkout processes facilitating member and non-member checkouts
- Integrated with leading payment service providers
- Sophisticated 'My Account' feature with management of personal details, delivery addresses, saved baskets, active and past orders
- Shipping cost management

3.8 Shipping & Delivery

- Weights / Units delivery calculations with upper and lower thresholds
- Integration with courier pricing matrixes and shipment tracking
- Exceptions for specific products which can have separate costs and different workflows for fulfillment
- Add delivery surcharges for individual expensive items that require additional cost to deliver

3.9 Integrations

- Integrates with 3rd party applications such as ERP or CRM systems

- Integrates with multiple secured payment gateways
- Integrates with online chat software applications
- Integrates with Google sitemaps for location tracking
- Automatically updates vendor product stock levels

4 A POSSIBLE METHOD OF DEFECT PREVENTION IN E-COMMECRCE SITES

Currently, as per the survey, the existing practice is to detect the defect at the end of the coding stage by way dynamic testing which is a latent method. Instead, at the end of the each stage like Requirement Gathering, Design (both High Level and Low Level), Creation of wire frames, if the defects are detected, and corrected, then, the code developed on such relatively less defect artifacts, will have lesser defects [2].

A reduction in known software defects is not necessarily a predictor for either improved security or reliability. However, reduction in known software defects and reliability are the two major parameters of the software quality [6].

4.1 A Real Time Case Study

A study on three different sized software projects was conducted by Carol Woody, et. al of Software Engineering Institute, Carnegie Mellon University, to determine the steps to be taken to minimize the defects in each phase and enhance the software quality[7].

The software projects in this study were of size 2.8 Million Lines of Code (MLOC), 1.3 MLOC, and 0.9 MLOC. The largest of the three projects included a large body of legacy code that was in test for an extended time (>8 months) because defect detection rates were not decreasing.

All projects included the use of static code analysis tools to establish defect levels. All projects included checklist-driven reviews of requirements, designs, code, and test cases. Peer inspection processes (design inspection and code inspection) were also introduced to eliminate any potential defects.

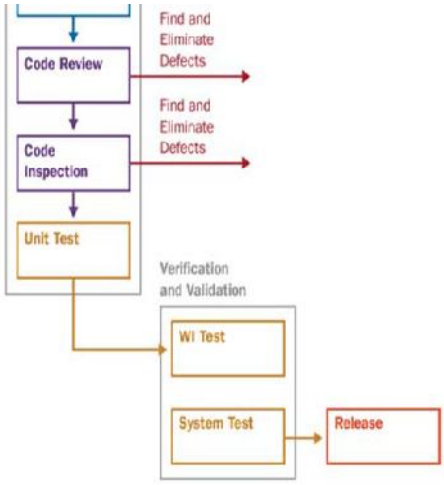

Figure 2 Defect Elimination at the end of each phase

The general workflow for the projects was similar to that shown in Figure 2 for quality and security, where each process step includes peer reviews and expert inspection [8].

Test cases were reviewed by the developer and inspected by peers. In practice, 80% of defects that test cases were designed to identify were removed prior to test-case execution. Testing was used to confirm defect removal, not to find defects. All defects were tracked throughout the projects in all phases (injection, discovery, and fix data).

The teams performed detailed planning for each upcoming code release cycle and confirmation planning for the overall schedule. Developers used their actual data to plan subsequent work and reach agreement with management on the schedule, process used, and resources required so that the plan could proceed without compromising the delivery schedule.

For the development phases, Table 1 lists the percentage of defects removed by phase (as recorded) for the three projects, for Code and Design Defects only [9].

Table 1: Percentage of Defects Removed by Phase for Three Projects

Phase	Project 1 0.9 MLOC	Project 2 1.3 MLOC	Project 3 2.8 MLOC
Static Analysis	4%	6%	9%
Unit Test	8%	17%	26%
Personal Review	24%	23%	27%
Peer Inspection	38%	60%	80%

4.2 Case Study Inference

Personal review and peer inspections provided the primary means of defect removal. Over 90% of the defects had been removed prior to test [10]. This removal may have positively affected the high test yields because

1. Large numbers of tests could be developed and run
2. Tests were able to focus on appropriate types of defects, as the "static" defects had already been removed.

The steps required to achieve lower levels of defects in test included design review and inspection (to reduce injections), personal reviews, and peer inspections.

The root cause analysis for identified defects indicated that most common defects associated with "Level 1," the highest severity category that would lead to a product recall, were found to have been injected during the coding phase. For these projects, a focus on coding quality was determined to be an effective strategy for reducing the risk of released safety-critical defects.

This type of root cause analysis is useful because designers and developers have a tendency to repeat the same types of mistakes, so defect removal efforts should be focused to offset these injections as much as possible.

The teams in this study developed and followed quality plans supported by explicit resource plans and schedules and demonstrated by metrics. The plans included personal reviews, peer inspections, static analysis, and a rigorous test case development process, including code coverage analysis and inspection of the test cases.

The teams were able to execute their plan by executing corrective measures throughout the development cycle. The results suggest that products with very low levels of defects are associated with a disciplined approach to planning quality activities, especially static defect removal tasks. The omission of quality practices, such as inspections, can lead to defects that can exceed the capabilities of existing code analysis tools [11].

5 CONCLUSION

Defect prevention typically depends on analysis that extends beyond the practice of dynamic testing. The projects analyzed in Section 4.1, pointed to a disciplined lifecycle approach with quality defect identification and removal practices combined with code analysis to provide the strongest results for building a defect free software system.

The various modules of an E-commerce website (as discussed in Section 3), have to be first analyzed for the "Dependency" on one another, to understand the sequence of artifacts (such as Design, Unit Test Cases) that have to be developed. For example, *Browse & Shop* module functionality depends on *Search* function, *Search* module depends on *Catalogue* and so on [12].

Once all these dependencies are determined for all the modules, then the developmental activities like creation of Design Document, Unit Test Cases and the Static Analysis on them could be sequenced based on the logic flow. By this method several unknown scenario of flow that may cause defects could be unearthed and fixed or marked for future fixing.

References

[1] B. Clark, D. Zubrow, "How good is the software: a review of defect prediction techniques", in: Software Engineering Symposium, IEEE Computer Press, Pittsburgh, PA, 2001, pp. 1–35.
[2] http://www.isixsigma.com/industries/software-it/defect-prevention-reducing-costs-and-enhancing-quality [Accessed: Dec 22, 2017]
[3] https://www.ibm.com/developerworks/rational/.../1251_bestpractices_TP026B.pdf [Accessed: Dec 22, 2017]
[4] https://www.redtechnology.com/ecommerce-platform [Accessed: Dec 22, 2017]
[5] https://www.redtechnology.com/downloads/ecommerce-platform-feature-list.pdf [Accessed: Dec 22, 2017]
[6] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, para. 2, page 4 [Accessed: Dec 22, 2017]
[7] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, para. 3, page 13 [Accessed: Dec 22, 2017]
[8] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, page 14 [Accessed: Dec 22, 2017]
[9] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, page 15 [Accessed: Dec 22, 2017]
[10] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, Section 3.4.2 page 15 [Accessed: Dec 22, 2017]
[11] https://resources.sei.cmu.edu/asset_files/TechnicalNote/2014_004_001_428597.pdf, para. 1 page 15 [Accessed: Dec 22, 2017]
[12] Kenneth C Laudon, Carol Guercio Traver, "E-Commerce – Business, Technology, Safety, 10th Edition, 2014, PEARSON

AUTHOR

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 and Software Quality Assurance. Currently 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 Ph.D's and is the author of over 750 publications.