IOSR Journal Of Humanities And Social Science (IOSR-JHSS)
Volume 19, Issue 1, Ver. VII (Jan. 2014), PP 106-112
e-ISSN: 2279-0837, p-ISSN: 2279-0845.
www.iosrjournals.org

Impact of Automation Techniques on Cost of Quality

Muhammad Danish Ali[1]

[1](*Hamdard Institute of Education and Social Sciences, Hamdard University, Pakistan*)

Abstract : *For software companies in business of custom software development faces a high threat on quality, specifically for large enterprise & complex systems. It is very time consuming and difficult to test every feature of application whenever new changes are implemented. Testers usually work in a fire fighting mode and do not get enough time to test and execute regression testing. Cost in defect identification, fixing and verification increases after every phase of software development life cycle, which also increases de-motivation in employees. In order to evaluate automation techniques and their effects on cost of quality, a study is conducted at Systems Limited (one of pioneer Software House of Pakistan).*

Keywords : *Basic Tools of Quality, Cost of Quality, Software Performance Testing, Software Quality Assurance, Software Test Automation*

I. Introduction

Unlike any other engineering domain, software development is usually considered as most volatile discipline. And the reason behind is that there isn't any hard code regulations that a software engineer should follow to develop the required product to solve any problem statement; in other words,
"Software is produced by people, not machines"

With the rapid growth of software the negative impact of its defects is also increasing. The most vital activity for quality assurance is to ensure that when software is delivered to its customers or released to the market, it should be with minimal number of defects. And these remaining defects should not impact the actual working of the product/solution.

Creating a perfect, faultless software that is free from any flaw or defect in condition or quality is a dream that every software engineer desire to achieve someday and which is a focus of I.T Companies as well developing software solutions but they don't call it perfect but that they are known for producing quality software, because a perfect thing exists theoretically but not practically. A product's quality can be improved to most desirable characteristics but there is no measure to call it a "PERFECT SOFTWARE". The main impact comes when companies need to maintain the "quality" of their product.

"If you don't care about quality, you can meet any other requirement" [1]

1.1 Cost of Quality

The "cost of quality" is the cost associated with a product or service not making up to the quality standards. Therefore every time the work is redone to enhance the quality of product, the directly proportional cost of quality increases. Obvious examples include the reworking of a contrived item, retesting of an assembly, rebuilding of a tool. In short, any cost that would not have been spent if quality were perfect comes under the category of "Cost of Quality".

Quality costs are the total of the cost incurred by financing for the prevention of nonconformance to requirements, appraising a product or service for conformance to requirements, failing to meet the standard requirements.

1.1.1 Prevention Costs

The idiom "Prevention is better than cure" highlights this category. The cost invested to make sure that efforts are being taken to design a better quality product and minimize the occurrence of appraisal and failure costs.it is concerned with the overall quality plan and the numerous specialized plans

Examples are the costs of new product reviews, quality planning, conduction of surveys for supplier capability, evaluations related to process capability, quality audits and quality improvement team meetings, implementation of Quality improvement projects, planning of Quality education and training

1.1.2 Appraisal Costs

Appraisal costs are linked with assessing, estimating or auditing products or services to guarantee conformance to quality standards and performance requirements. In short it is mainly concern with the test and inspection of the product.

These include the costs of:

- Incoming and source inspection/test of purchased materials/services/products
- In-process and final inspection/test of product
- Auditing of product, process or service
- Standardization of measuring and test equipment
- Associated supplies and materials

1.1.3 Failure Costs

Failure costs results from products or services not conforming to requirements or customer/user requirements. Failure costs are distributed into internal and external failure categories.

1.1.3.1 Internal Failure Costs

The failure which occurs before the shipment or delivery of the product or the furnishing of a service to the customer is termed as Internal Failure and the costs associated as Internal Failure Costs.

Examples are the costs of scrap and rework, re-designing, re-inspection, re-testing, material review, downgrading of product due to quality issues.

1.1.3.2 External Failure Costs

The most critical cost is the external Failure cost that occurs after transport or delivery of the product and during or after furnishing of a service to the end user.

Examples are the costs of resolving customer complaints, conducting customer surveys to identify maximum defects, customer returns, fulfilling warranty claims ,dealing with the downgrading of product quality .

1.2 Total Cost of Quality:

The total of all the above costs is “total cost of quality”. This exemplifies the difference between the actual cost of a product or service and what the abridged cost would be if factors like the possibility of substandard service, failure of products or defects in their manufacturing are removed. [2]

1.3 Software Quality Issues:

This section describes in detail the issues encountered in the manufacturing of quality software as well as introduction of quality standards during manufacturing process to ensure conformance of requirements and reduction in cost of quality is achieved. Quality from software and core IT related point of view has been defined very specifically by IEEE (Institute of Electrical and Electronics Engineers):

i. “The totality of features and characteristics of a software product that bear on its ability to satisfy given needs: for example, conform to specifications.”
ii. “The degree to which software possesses a desired combination of attributes.”
iii. “The degree to which a customer or user perceives that software meets his or her composite expectations.”
iv. “The composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customer". IEEE Standard [3]

The main issue lies in the gap between the manufacturer and the customer, all the issues occur due to the lack of understandings during Requirement Gathering process, Planning, deployment and testing.

Let’s analyze issues related to each area of SDLC (Software Development Life Cycle) which results in the discovery of defects during testing process and requires re-work to confirm that a quality product will be delivered at user end.

Table 1: Software Quality Issues Occurred during SDLC Phases.

S.No.	Software Development Life Cycle Phases	Software Quality Issues Occurred
1.	Requirement Gathering and Preparation of Software Requirements Specifications also preparing Project Time Line.	• Increase in the cost of quality due to non-conformance to requirements • Time and labor constraints for product manufacturing increases.
2.	Project/Product/Service Development	• Confusion due to incomplete requirements results in lesser quality product • Hiring of inexperienced staff increases appraisal cost and failure cost • Not effectively Unit tested by developer’s increases defects availability.
3.	Testing of Product/Service	• An effective test plan and cases are not being developed for the product. • Frequent defects increases time and labor for Regression testing. • Manual regression testing takes time and effort and increases appraisal cost.

4.	Deployment and Maintenance Activities	• Defects increases internal failure cost for deployment. • External failure cost increases due to defects.

1.4 Automation Tools & Techniques

Rapid growth of I.T industry has made life easier for human kind but this also brings increasing complexity in code written for software products .the higher the complexity rate the defect rate increases also and the debugging process becomes harder to proceed manually. The need of automated solution to reduce the burden of testers and to make them focus on the core theme of product can result in a more effective product and the reduction of errors occurring and reduction of total quality cost.

"Quality software remains a key business differentiator. Businesses and the IT organizations supporting them have no choice with regard to quality initiatives and the need to address the debilitating costs of software defects," said Melinda Ballou, program director for IDC's Application Life-Cycle Management research. "Organizations should evaluate current process and organizational approaches to software quality in combination with automated approaches for code analysis and testing to help enable more secure, successful, better managed software implementations."

Ben Chelf, CTO of Coverity. "Development organizations today need innovative tools that automate the tedious process of defect detection so their valued developers can stay focused on delivering new features and functionality instead of debugging problems left over from previous releases." [4]

II. Research Methodology:

This is a quantitative research that could help in understanding the impact of automation techniques/tools on quality. Seven Basic tools will be used for initial understanding of problem that leads to increase in cost of quality. It will be a working project that will help comparing manual testing and using automation tool. This analysis will help in deciding what techniques/tools to be used in order to eliminate defects and improve quality of product.

2.1 Target Population & Sample Design:

This research is specifically targeted to first software house of Pakistan **Systems Limited** (SL).

2.2 About Systems Limited (Research Target):

Systems Limited is one of the leading IT and IT enabled Services provider in the region. Established in 1977, providing solutions to cross section of sectors including Government, Banking, Insurance, Manufacturing, Pharmaceutical, Distribution, and Textile & Apparel.

Our +2,000 employees all over the world possess strong expertise in IT Consultancy, Business Process Automation, ERP Solutions, Business Intelligence, Human Capital Management, Business Process Outsourcing and Document Management Systems.

Since there are plenty of enterprise applications that are in either development or maintenance phase and it won’t be possible to cover all of them. Therefore, for *sample*, following three applications of different domain are taken into account for this study;

- SysHCM (Human Capital Management System) – Systems Limited’s one of the best seller product
- Oracle Enterprise Business Suit (EBS) – Oracle Product & Systems Limited provide implementation consultancy to various client
- ECSI – Customized Project for a client

2.2.1 Usage of SysHCM, Oracle EBS and ECSI

Rational behind using SysHCM is that this product is running at 8 clients and approximately 4 implementations are underway. And with passage of time its complexity has increased, due to which the number of defects increases with every implementation. This in turn has increased the cost of quality and management is concern to reduce this cost otherwise there is a chance that revenue goes down.

Although Oracle EBS is not a proprietary product of SL, but when it comes to implementation of its modules like Inventory Management, Human Resource Management, Financials, etc., it is quite a repetitive verification effort at each client.

ECSI application is developed specifically for a client, and the reason to consider it as sample is to identify whether the testing effort increases/decreases with automation for customized application or not.

From each application couples of frequently used test scenario were taken as bench mark.

2.3 Design of the Data Analysis

Interviews based on focus group of testers, developers, managers & customer support members are conducted to understand the weak link of quality. Data will be design on basis of most prominent quality concerns shown by tester or client, number of reported defects by tester or client, nature of defects, time require to develop test data, time require to execute test cases manually, etc. The feedback derived from the focus group is the basis for hypothesis formation. Since, Systems Limited has implemented quality management systems (like ISO 9001 and CMMI level III) and has strong processes areas therefore this data can also easily be accessed from various tools like Bug Reporting Tool, Knowledge Base Portal, document management system etc.

2.4 Data Analysis Using Basic Tools of Quality:

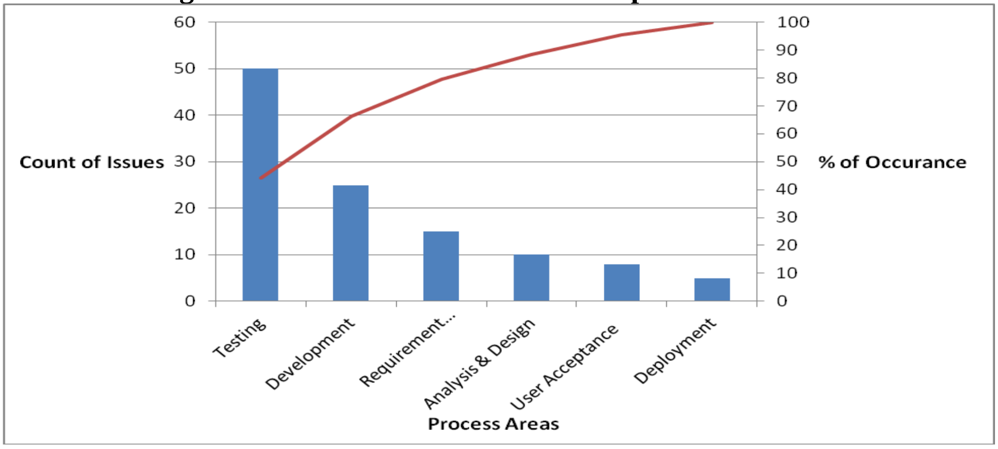
From the knowledge base of Systems Limited and experience shared by project team members, various data has been derived. Below is the average defect count taken from Issue Management Tools (HP Quality Control and JIRA) running at Systems Limited. It elaborates that in which software development process area most defects are observed. Usually defects are categorize in 3 types Critical (show stoppers/business process related), Major (Functional issues) and Minor (cosmetics issues)

Table 2 : Average Defect Count per Process Area

S.No	Process Area	Critical	Major	Minor
1	Requirement Gathering	15	23	22
2	Analysis & Design	8	14	33
3	Development	25	40	41
4	Testing	50	80	30
5	User Acceptance	10	15	35
6	Deployment	5	7	10

From the above data in order to identify the weak link in process area, critical issues are separated and applied Pareto chart or 80/20 rule. Refer to below chart as it clearly shows that development, requirement gathering and testing need to be closely monitored so that the count of defect may decrease.

Fig. 1: Pareto Chart for Defect Count per Process Area



2.5 Cost of Quality Control:

Furthermore, to analyze the Cost of Poor Quality on an average project development at systems limited, SysHCM’s workflow module is taken as bench mark. All data is based on focus group findings and general practices, however actual data varies with respect to project's complexity. Refer to below Table 3;

Table 3 : Average Cost of Quality

Cost of Poor Quality (COPQ)		
SysHCM - Workflow module Implementation at Systems Limited (SL)		
Average Project length:	Approximately 1 month or 176 Hrs	
Average Work Hour cost:	RKR 1000/- per Hour	
Resources:	1 Business Analyst	80 Hrs
	4 Developers	160 Hrs
	1 Testers	80 Hrs
	Total Man Hours:	320 Hrs
Total Resources Cost:	PKR 320,000.00	
Infrastructure Cost: *(Servers, Machine, Software, Support staff etc)*	PKR 150,000.00	
Total Minimum Cost	**PKR 470,000.00**	
Appraisal Expense		
Test	15 Hrs	
Process Controls	10 Hrs	
Inspections	10 Hrs	
Internal Failure Costs		
Scrap or Rework	10 Hrs	
Redesign/Engineering Change	10 Hrs	
Re-Inspection	5 Hrs	
External Failure Costs		
User reported issues	20 Hrs	
User reported modifications	20 Hrs	
Non-Value Added Work		
Documentation for change/issues	20 Hrs	
Internal Meetings	20 Hrs	
Late Sittings	40 Hrs	
Total Expense/Failure Cost	**PKR 180,000.00**	
Actual Cost (Min. Cost + Expense Cost)	**PKR 650,000.00**	
COPQ = Actual Cost - Min. Cost		**PKR 180,000.00**
COPQ In Percentage:		**38.30%**

Moreover, various frequently used test scenarios were obtained from all the three applications and manual effort is measured in hours taken to complete the quality control/testing activities. Below table shows how much effort is required to prepare test data, execute functional/regression testing.

Table 4 - Average Manual Effort (in Hours)

Average Manual Effort (in Hours)						
Application	**Frequently Executing Test Scenarios**	**Test Data Preparation**	**Smoke Testing**	**Functional Testing**	**Regression Testing**	**Total Effort**
SysHCM	Administration Setups	12	4	16	12	44
	Employee profile creation	10	4	24	12	50
Oracle EBS	HR module - profile creation & assignment	10	6	20	8	44
	Inventory Search & Setup	6	2	16	6	30
ECSI	Client Creation	4	1	10	2	17
	Create Product	4	1	8	1	14

However, load or performance testing is never possible manually, since it require hundreds of users to work concurrently on same application. Above data design and analysis will provide the assistance in identifying the variance of cost after the implementation of automation tool is

2.6 Manual Effort in Cost:

Considering the effort applied while performing testing manually, it can be converted in to cost. Each team member working on project is charged to client on hourly basis at Systems Limited. For this study we will take bench mark of ***Rs. 1000 per hour.*** See below table for effort to cost conversion before applying any automation tool.

Table 5: Manual Effort in Cost

Application	Frequently Executing Test Scenarios	Total Effort	Total Cost (Rs. 1000 per hour)
SysHCM	Administration Setups	44	44000
	Employee profile creation	50	50000
Oracle EBS	HR module - profile creation & assignment	44	44000
	Inventory Search & Setup	30	30000
ECSI	Client Creation	17	17000
	Create Product	14	14000

III. Results

3.1 Implementation of Test Automation Tools & Summary of Effort:

Various test automation tools (like ***Selenium, Oracle OATS and HP Load Runner***) were implemented against the test cases that were taken as bench mark. Below is the summary of testing effort in hours.

Table 6: Average Automation Testing Effort (in Hours.)

Average Automation Testing Effort (in Hours.)					
Application	**Frequently Executing Test Scenarios**	**Automation script development**	**Total Execution**	**Script Maintenance**	**Total Effort**
SysHCM	Administration Setups	16	0.5	4	20.5
	Employee profile creation	20	0.5	6	26.5
Oracle EBS	HR module - profile creation & assignment	24	0.5	6	30.5
	Inventory Search & Setup	16	0.6	6	22.6
ECSI	Client Creation	36	0.33	8	44.33
	Create Product	16	0.1	6	22.1

From the above data it clear that automation can help in reducing the effort as compare to manual testing.

3.2 Automation Effort in Cost:

Referring to the cost of quality table listed above, automation effort can be converted from hours in to cost. Since, there is an hourly rate of human resource working on a project and from the total effort obtained after implementation of automation; it can be easily converted in to cost. See below table for automation cost.

Table 7: Automation Effort in Cost

Application	Frequently Executing Test Scenarios	Total Effort	Total Cost (Rs. 1000 per hour)
SysHCM	Administration Setups	20.05	20050
	Employee profile creation	26.05	26050
Oracle EBS	HR module - profile creation & assignment	30.5	30500
	Inventory Search & Setup	22.6	22600
ECSI	Client Creation	44.33	44330
	Create Product	22.1	22100

IV. Conclusion:

Below is the conclusion derived from the study, which depicts the picture of pros and cons of usage/preference of Automation Tools and Techniques over Manual Testing methods.

Table 8 below illustrates the pros and cons related to Automated and Manual Testing:

S.No.	Automation Tools and Techniques	Manual Testing Tools and Techniques
1.	Reduces time and efforts.	Takes maximum time and efforts.
2.	Reduction in Cost for Long Term Testing of products	Increases Cost and Labor for Long Term Testing of products. Effective for short term Testing.
3.	Quick and effective testing through scripts which needs to be written once but runs endless times for testing.	Steady Testing not effective due to repetitive testing of same bug during regression testing.
4.	Interesting as the tester can explore different techniques.	Old methods are followed in repetitive manner results in a boring job for tester.
5.	Team Collaboration since the results can be reviewed by Team members by signing into the testing page. Results in better quality product.	Single tester is responsible for test execution and result verification results in no team collaboration for effective product.

Automation is not feasible for applications that are not stable, as the cost of maintenance is higher.

References:

Web References:

[1] Brown, 2011, 16 Great Quotes for Software Testers. 14 January 2011. *Mike Brown: Blog*. Available from: <http://blog.utest.com/16-great-quotes-for-software-testers/>. [14 January 2011].

[3] IEEE Standard (IEEE Std. 729-1983) http://standards.ieee.org/findstds/standard/729-1983.html

Book References:

[2] ASQ Quality Costs Committee, 1999, *Principles of Quality Costs: Principles, Implementation, and Use, Third Edition*, ed. Jack Campanella, ASQ Quality Press, pages 3–5.

Research Paper References:

[4] Melinda-Carol Ballou, 2008, "Improving Software Quality to Drive Business Agility", IDC white paper, June 2008, pp. 5-6.