JCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-4, Issue-4 E-ISSN: 2347-2693

A Comparison Report on Efficient Types of Testing

Thulasiram K[1*], Ramakrishna.S[2] and Chitti Babu.P[3]

[1,2]Dept.of Computer Science, S.V.University, India
[3]Dept. of MCA, APGCCS, India, Country

Available online at: www.ijcseonline.org

Received: Mar/27/2016 Revised: Apr /08/2016 Accepted: Apr/19/2016 Published: Apr/30/2016

Abstract: In this paper, by comparing various types of testing, a framework was created for an efficient and cost effective testing. For the research, one of the largest digital marketing companies was considered. The company wanted to analyze their business to identify which website is generating more revenue. To analyze the business, the company has developed an analytical tool. To ensure the efficiency of the product and to compete with other analytical tool vendors in the market, they keep enhancing the product and release it frequently. To test these analytical product for frequent releases the company came up with an idea to compare the types of testing and decide the right testing methodology that will give the best output with less turnaround and reduced cost. In this approach four types of testing were considered, them being Priority based, Critical based, Risk based and Resource based testing.

Keywords—Comparision testing,priority based testing,risk based testing,critical based testing,resource based testing

I. INTRODUCTION

One of the largest digital marketing companies in United States of America developed an analytical tool and to compete in the market with other vendors they frequently release it once in six months with some enhancements to the product. To reduce the cost and time in testing the product, with a frame work is considered as shown in the Figure

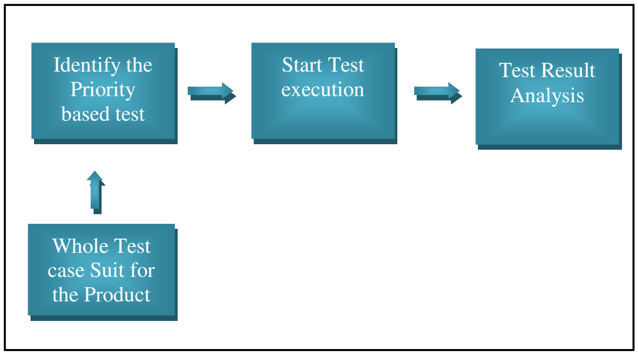
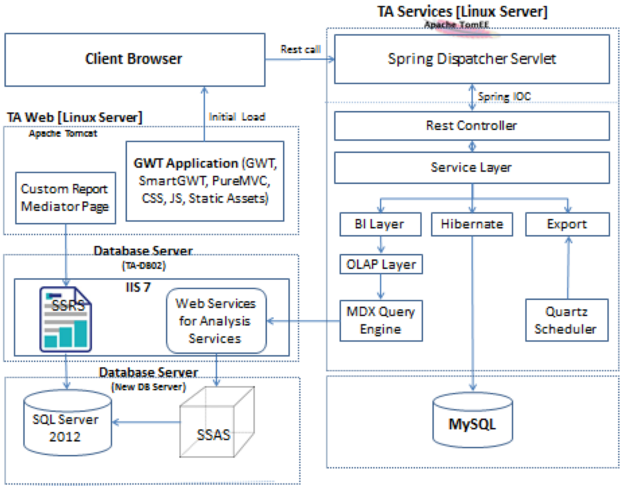

Identify the Priority based test
Start Test execution
Test Result Analysis
Whole Test case Suit for the Product

Figure 2: Frame work for Cost Effective testing

This paper contains comparison of four different types of testing and identifying the best fit for the analytical product.

The architecture and overview of the product is as follows: figure-1.

The application is developed using pure MVC (Model view controller) architecture. The use of the MVC pattern in web applications exploded in popularity after the introduction of spring framework for Java. It has strong emphasis on rapid deployment. **It** increased its popularity outside the traditional enterprise environment in which MVC has long been popular. MVC web frameworks now hold large market shares relative to non-MVC web toolkits.



Architecture Figure 1

The analytical tool usually extracts data from the ad servers, which will be segregated and mapped using analysis services tool. These data will be converted in the form of data sets. Using these data sets digital marketers create–reports. The report contains data which is used by the marketers to analyze their business like which site is giving good revenue.

In general, the SDLC process contains the following phases, which are: requirement gathering, design & analysis, development, testing and maintenance. The testing phases contain the following processes: test planning, test development, test execution and evaluation

© 2016, IJCSE All Rights Reserved

of results. Those processes can be represented as follows: Figure 2.

In General Software Testing Process has been widely used as a way to help engineers develop high-quality systems. It is an important to support quality assurance by gathering information about the nature of the software being studied. These activities consist of designing test cases, executing the software with those test cases, and examining the results produced by those executions. Generally more than fifty percent of the cost of software development is devoted to testing with the percentage for testing critical software being even higher. As software becomes more pervasive and is used more often to perform critical tasks, the importance of its quality will remain high. Unless engineers can find efficient ways to perform effective testing, the percentage of development costs devoted to testing may increase significantly. Software testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. It also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of implementation of the software. Software testing can be implemented at any time in the development process; however, the best test effort is employed after the requirements have been defined and coding process has been completed. Software engineers generally save test suites that they develop so that they can easily reuse those suites later as the software evolves. Cost effective-based techniques are methods of prioritizing test cases based on costs, such as cost of analysis and cost of prioritization [1].

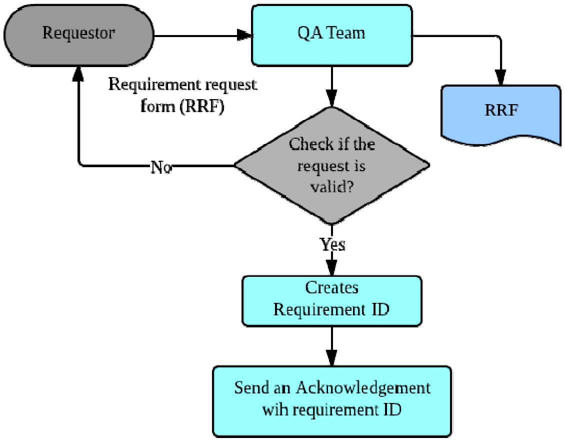

Requestor
QA Team
RRF
Requirement request form (RRF)
Check if the request is valid?
No
Yes
Creates Requirement ID
Send an Acknowledgement wih requirement ID

Figure 3: Process flow: Requirement request process

Software testing has been proven that testing, analysis, and debugging costs usually consumes over 50% of the costs associated with the development of large software systems.
This section describes the requirement request process that helps identifying the business needs and opportunities. Requirements are documented to a level of detail that is sufficient for the development of Analytical Product. Every requirement (New requirement, Enhancement and Defect) must be always requested through Requirement Request Form (RRF) and to be sent to the QA Team [3].

QA team verifies the RRF to ensure the requirement is valid. When the requirement is found valid, then requirement ID is created in Mantis and the same will be communicated to the requestor with RRF. If the requirement is not convincing, justification for rejection or deferred will be communicated with RRF. QA Team must ensure RRF is attached to every corresponding requirement in Mantis [16].

Quality Assurance (QA) ensures the delivery of stable product. The goal of testing is to verify accuracy and completeness of the deliverables, starting from requirements until the product is released. Testing takes place parallel to the development.

Once Sprint plan is created, testing team identifies test scenarios. Test cases will be developed in excel sheet. After confirmation from development team, testing team starts executing test cases and updates the test case status as “Pass” that is if it is functioning as per the requirement, else it will be updated as “Fail” in the bug tracking tool. Once all the testing completes with 100 % accuracy, testing team confirms to Production.

Software Applications follow a process of SDLC models based on the organization and the project to execute. In current trend, most of the organizations are interested to run their projects using the agile methodology, where there is a possibility of change in the requirements at any moment of the phase or iteration/cycles. The Application requirement specifications need not be frozen before moving to design.

Consider the below example for the test summary report of the analytical product for 6 months release [8].

S.No	Testing metric	Test cases execution status
1	Number of Requirements	55
2	Number of test cases	1020
3	Total number of test cases Executed	1020
4	Number of test cases passed	770
5	Number of test cases Failed	250
6	Number of Defects found	170
7	Critical Defect count	35
8	High Defect count	45
9	Medium Defect count	55
10	Low Defect count	35
S.No	Testing metric	Test cases execution status

Defect Density: Number of Defects identified per 100 lines of code [OR] No. of defects identified per module etc.)

In the above summary Defect Density is calculated as (170 / 55) = 3

Defect Removal Efficiency (DRE):_(No. of Defects found during QA testing / (No. of Defects found during QA testing +No. of Defects found by End user)) * 100

DRE is used to identify the test effectiveness of the system. Suppose, during Development & QA testing, 170 defects were identified and after the QA testing, during Alpha & Beta

testing, 60 end user / client defects were identified, which could have been identified even during QA testing phase then the DRE will be calculated as, DRE = [170 / (170 + 60)] * 100 = [170 /230] * 100 = 73%

Defect Leakage: Defect Leakage is the Metric which is used to identify the efficiency of the QA testing i.e., how many defects are missed / slipped during the QA testing.

Defect Leakage **= (**No. of Defects found in UAT / No. of Defects found in QA testing.) * 100

Suppose, during Development & QA testing, we have identified 170 defects and after the QA testing, during Alpha & Beta testing, 120 end user / client defects were identified, which could have been identified even during QA testing phase then user / client defects were identified, which could have been identified even during QA testing phase then

Defect Leakage = (60 /170) * 100 = 35%

Defects by Priority: This metric is used to identify the no. of defects identified based on the Severity / Priority of the defect which is used to decide the quality of the software.

% of Critical Defects = No. of Critical Defects identified / Total no. of Defects identified * 100

From the data available in the above table, %of Critical Defects = 35/ 170 * 100 = 20%

%of High Defects = No. of High Defects identified / Total no. of Defects identified * 100

From the data available in the above table,% of High Defects = 45/ 170 * 100 = 26%

%of Medium Defects = No. of Medium Defects identified / Total no. of Defects identified * 100

From the data available in the above table, % Medium Defects = 55/ 170 * 100 = 32% [17].

% of Low Defects = No. of Low Defects identified / Total no. of Defects identified * 100 From the data available in the above table, % of Low Defects = 35/ 170 * 100 = 20%

II. CRITICAL BASED TESTING

Consider an example, for analytical product, the agency / publisher wants to analyze the data of their business, he / she considers the raw data and puts the data into the analytical tool to analyze. In this example tester has to concentrate mainly on validating the server data as well as the tool data. Consider this as critical point for the business to analyze.

The process of testing the application with critical test cases that covers all client requirements: Here let us consider the Analytical application will have addition of new features and enhancements for every release. Based on the new features, the number of test cases will also increase. Whenever testing the latest deployment of the application needs to execute all the test cases which cover all clients' requirements[10].

		2013	2014	2015
S.No	Test cases execution status	Test cases execution status	Test cases execution status	Test cases execution status
1	Number of Requirements	56	117	162
2	Number of test cases	1074	2062	3077
3	Total number of test cases Executed	587	1075	1570
4	Number of test cases passed	307	627	837
5	Number of test cases Failed	208	307	407
6	Total Number of Defects found QA + Client	312	517	710

Here QA Lead / Business Analyst decide based on the business needs. Testers start executing critical test cases.

Critical level approach

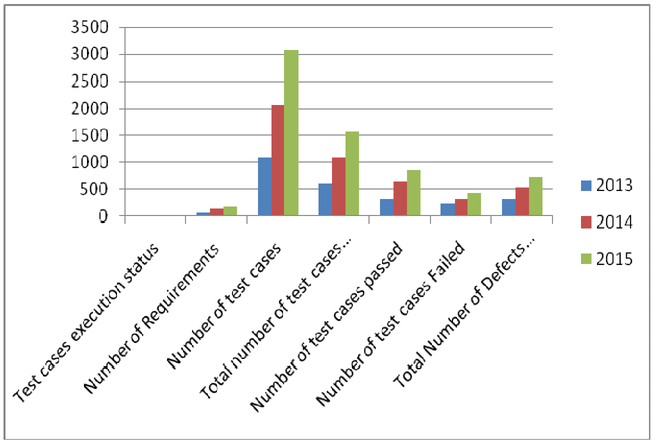


Figure 4

In the above analysis, every year the test cases increase but the old functionality test cases do not change, If those critical level test cases are executed then conclude that the critical features of the analytics product has been tested i.e business needs are fulfilled.

III. PRIORITY BASED TESTING

In the Priority based testing approach testers only concentrates on the priority i.e on enhancements, new features and patch releases etc [4].

Software tests have to be repeated often during development cycles to ensure quality. Every release of the software may be tested on all supported operating systems and hardware configurations. Testing is costly and time consuming whereas; the new testing process provides the solution for improving the effectiveness of software testing. Thus, Priority based testing plays an important role.

In this phase of testing, tester defines priority for various modules of application under test. The priorities will be used to execute tests over the specified modules in a specific order. High priority modules will be tested prior to the low priority modules. Once the testing process for a particular module is over, the test report is developed. The testing process continues till the test report for each module under test is developed successfully.

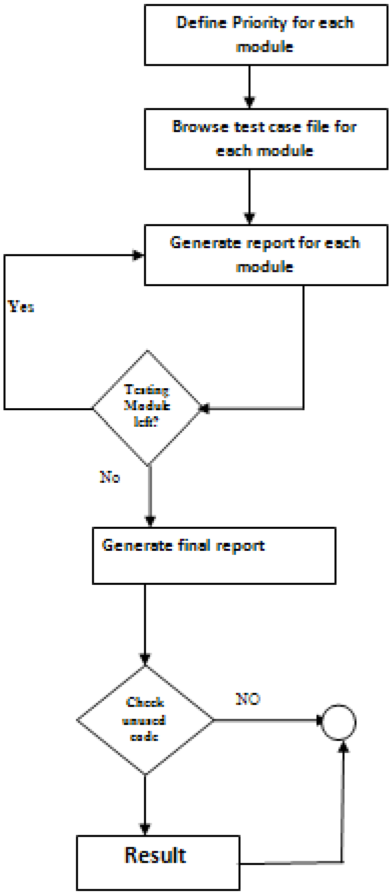

Define Priority for each module
Browse test case file for each module
Generate report for each module
Yes
Testing Module left?
No
Generate final report
Check unused code
NO
Result

Figure 5: Flow Chart of Priority based Testing

Here the focus is on the software testing with the priority of the user so that the user can run only less number of tasks so that the time in the execution takes less time and the rest of the test cases with the Equivalence classes will be helpful in generating sufficient number of prioritized test cases to be executed

Advantages/Benefits of Priority based testing:

- Improved quality – All of the Priority based functions of the application are tested. Real time clear understanding of Product Priority
- Gives more focus on Priority of the business Product instead of the functionality of the information system.
- Associates the Product Priority to the requirement identify gaps.
- Testing always concentrates on the most important matters, first with optimal test delivery, in case of – limited time, money and qualified resources. With the time and resources 100% testing can be completed. So we need to determine a better way to test the analytics product.

© 2016, IJCSE All Rights Reserved

		2013	2014	2015
SNO	Test cases execution status	Test cases execution status	Test cases execution status	Test cases execution status
1	Number of newly added Features	27	48	67
2	Total Number of test cases	1077	2200	3120
3	Number of test cases for newly added features	407	607	820
4	Number of test cases passed	410	620	870
4	Number of test cases passed	323	527	745
5	Number of test cases Failed	112	107	212
6	Total Number of Defects found QA	17	12	17
7	Total Number of Defects found in Regression	11	23	29

Priority based testing

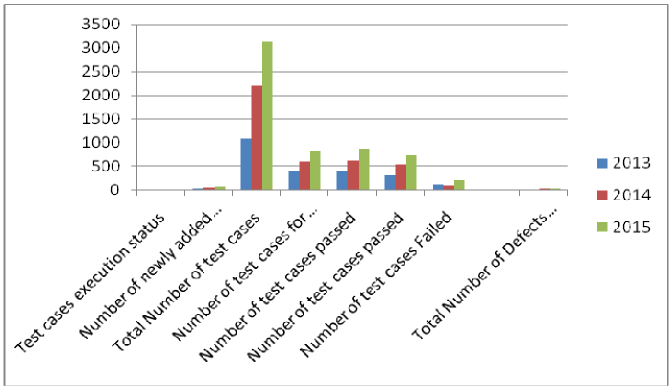


Figure: 6

In the above analysis, Priority based testing defects count found is very less in enhancements as well as in regression testing

IV. Risk Based Testing

Risk based testing is a testing process with unique features. It is basically for those project and application that is based on risk. Using risk, Risk based testing prioritizes and emphasizes the suitable tests at the time of test execution. In other word, Risk is the chance of event of an unwanted outcome. This unwanted outcome is also related with an impact. Sometimes it is difficult to test all functionality of the application or it is might not possible. Using Risk based testing the functionality which has the highest impact and probability of failure can be tested [6].

It is better to start risk based testing with product risk analysis. The numerous methods used for this are,

- Clear understanding of software requirements specification, design documents and other documents.
- Brainstorming with the project stakeholders.
- Risk-based testing is the process to understand testing efforts in a way that reduces the remaining level of product risk when the system is developed,
- Risk-based testing applied to the project at very initial level, identifies risks of the project that exposes the quality of the project. This knowledge guides to testing planning, specification, preparation and execution.
- Risk-based testing includes both mitigation (testing to give chances to decrease the likelihood of faults, especially high-impact faults) and contingency (testing to know work-around to create the defects that do get past us less painful).
- Risk-based testing also includes measurement process that recognizes how well we are working at finding and removing faults in key areas.
- Risk-based testing also uses risk analysis to recognize proactive chances to take out or avoid defects through non-testing activities and to help us select which test activities to perform.
- Risk which test activities to perform.

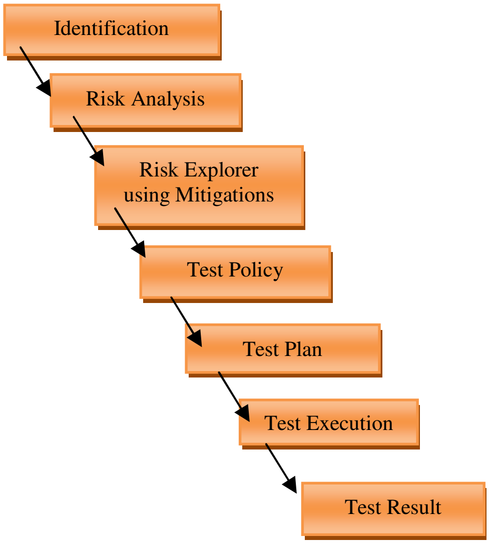


Figure 7:

Major processes to execute the Risk-based testing are described below:

- Process 1 – Describe all requirements in terms of Risk involved in the project
- Process 2 – In terms of risk assessment, prioritize the requirements
- Process 3 – Plan and define tests according to requirement prioritization
- Process 4 – Execute test according to prioritization and acceptance criteria.

		2013	2014	2015
SNO	Test cases execution status	Test cases execution status	Test cases execution status	Test cases execution status
1	Number of Requirements	50	160	150
2	Number of risk impact test cases	1090	2080	3090
3	Total number of risk impact test cases Executed	570	1070	1580
4	Number of test cases passed	340	680	800
5	Number of test cases Failed	50	170	210
6	Number of test cases not executed	157	370	560

Risk based approach

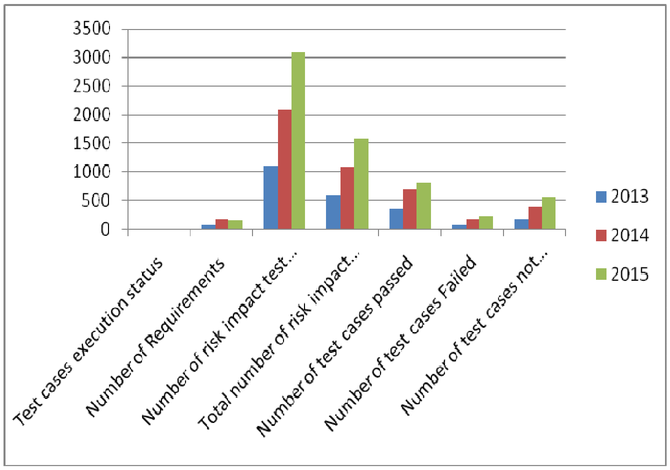


Figure 8:

In the above analysis, the failed test cases are nearly 30 % of the total number; here tester executes those test cases that highly impact the analytics product.

V. Resource Based Testing:

Resource based testing In this process we analyze Resource usage, which shows information on the usage of each resource that is allocated in testing process simulation [18]

Using this analysis to help in resource planning, it enables you to see how each resource is allocated to different testing activities across the process. In addition to showing how a resource uses its time to accomplish one or more activities, this analysis shows which resources are prone to shortages and which activities delay as a result. You can use this information to determine where additional resources are required.

For each resource used by a process during the simulation run, the analysis displays the following summary information:

		2013	2014	2015
SNO	Test cases execution status	Test cases executio n status	Test cases execution status	Test cases execution status
1	Number of Requirements	57	128	157
2	Number of risk impact test cases	1030	2070	3080
3	Total number of risk impact test cases Executed	520	1080	1560
4	Number of test cases passed	307	650	870
5	Number of test cases Failed	50	110	220
6	Number of test cases not executed	150	340	550
7	Time taken for execution of the test cases with one resource	10 Days	15 Days	20 Days
8	Time taken for execution of the test cases with	5 Days	7 Days	10 Days

	two resources			
SNO	Test cases execution status	Test cases execution status	Test cases execution status	Test cases execution status
1	Number of Requirements	57	128	157
2	Number of risk impact test cases	1030	2070	3080

Resource based approach

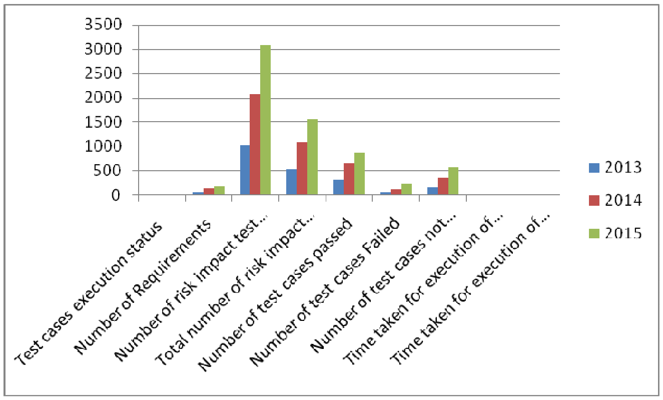


Figure 9:

In this analysis, results are organized by resource. Activity resource allocation analysis displays a similar set of information, organized by activity instead of by resource.

VI. Conclusion

As the size and complexity of software is continually growing, testing becomes very difficult and tedious task. But on the contrary, Priority based testing can help to improve efficiency of the testing process and results in to efficient cost effective testing report for the analtics product of the digital marketing companies.

The proposed model allows the user to define priority for each of the modules under test. After each of the modules is tested effectively, the corresponding testing report is generated and is available for the review while other modules are being tested. The priority based model helps to minimize the user intervention to a large extent, which makes it highly efficient and less time consuming. Hence it concludes that Priority based testing is the best among compare all other types of testing.

References

[1] T. Mens and T. Tourwe, " A survey of software refactoring," Software Engineering, IEEE Transactions on, vol.30, no. 2, pp. 126-139, Feb 2004

[2] http://www.webopedia.com/TERM/S/static_routing.html, date last viewed: 2016-1-17.

[3] A. Leitner, I. Ciupa and B. Meyer,"Reconciling Manual and Automated Testing: the AutoTest Experience", Proc. 40th Hawaii International Conference on System Sciences, pp: 261-271, January 2007.

[4] M. Ali, S. Mushtaq and N. Arshad., "Priority-Based Automated Testing of Web Application", Proc. IEEE Conference on Communication and Information Technology,2012.

[5] Felix Redmill in „Professional Tester", April 2003. www.professional-tester.com

[6] Tom DeMarco and Tim Lister, "Waltzing with Bears: Managing Risk on Software Projects", 2003.www.riskbasedtesting.com

[7] IEEE Standard 1044.1-2002: Guide to Classification for Software Anomalies

[8] IEEE Std. 16085 Standard for Software Engineering - Software Life Cycle Processes -Risk Management You find them at sales@ieee.org

[9] Information Systems Examinations Board. Practitioner Certificate in Software Testing - Guidelines and Syllabus. Version 1.1. British Computer Society, Swindon, 2001

[10] Gerrard P and Thompson N. Risk-Based E-Business Testing. Artech House, Norwood,2002.

[11] Kahneman D and Tversky A. Prospect Theory: An Analysis of Decision Under Risk.Econometrica, 47, 263-291, 1979.

[12] Redmill. Risk Analysis - a subjective process. Engineering Management Journal, 12, 2,91-96, 2002.

[13] Moseley, C. E., Gettings, R. M., & Cooper, R. (2002). Havingityourway: Under-standing state individual budgeting strategies . Washington, DC: National Associa-tion of State Directors of Developmental Disabilities Services.

[14] Campbell, E. M., Fortune, J. R., Frisch, J. K., Heal, L. W., Heinlein, K. B., Lynch, R. M., et al. (2005). Predictors of expenditures in western states. In R. J. Stancliffe and K.C. Lakin (Eds.), Costs and outcomes of community services for people withIntellectual disabilities (pp. 175–201). Baltimore, MD: Paul H. Brookes Publishing Co.

[15] Kaner, Cem and Walter P. Bond "Software Engineering Metrics: What Do They Measure and How Do We Know?."

[16] Weinberg, Gerald M. "Quality Software Management, Volume 2: First-Order Measurement.".

[17] P. Chitti Babu and K.C.K. Bharathi, "Assessment of Maintainability factor", International Journal of

© 2016, IJCSE All Rights Reserved

Computer Science Engineering and Information Technology Research, Vol. 3, Issue.3,29-42, 2013

[18] C.Mallikarjuna and P.Chitti Babu,” A Report on the Analysis of Software Maintenance and Impact on Quality Factors”, International Journal of Engineering Sciences Research-IJESR, Vol.05,1485-89,2014

AUTHORS PROFILE

Dr.S.RamaKrishna has obtained M.Sc.,M.Phil., Ph.D., Degrees from Sri venkateswara University, Tirupathi and M.Tech (IT) from Punjabi University. He has guided 17 Ph.D & 15 M.Phil scholars and currently he is guiding 7 Ph.D.&3 M.Phil scholars in ComputerScience in Sri venkateswara University, Tirupathi, Andhra Pradesh. He is working as a Professor &BOS Chairman in Computer Science Department, Sri venkateswara University, Tirupathi, Andhra Pradesh with the experienceof 27 years.He has published 4 books and 90 journalpublications as his credits.

Dr.P.Chitti Babu has received his Master’s Degree in Computer Applications (MCA) from Sri venkateswara University, Tirupathi. He has completed M.Phil in Computer Science from Alagappa University and M.Tech in Computer Science & Engineering from Acharya Nagarjuna Univeristy, Guntur, India. He has successfully received his Ph.D (Computer Science) form Sri venkateswara University. Presently he is working as a Professor & Principal, Annamacharya PG College of Computer Studies, Rajampet, Andhra Pradesh. He has a total of 16 years of experience. He has published 25 papers in International and National Journals. He is a life member of CSI, IACSIT, IAENG and ISTE. Presently he is guiding 2 students for research under Ph.D programme.

Thulasiram K, received his M.C.A degree in Computer Science from Sri venkateswara University during 2001-2004.He is Research Scholar, Dept of Computer Science, Sri venkateswara University,Tirupati. Intests in autamating the products and going to implement Automation testing using enhanced Genetic Algorithm.