

Validation Of Usability Evaluation Models Through Experimental Setup And Empirical Study

P. Ashok Reddy

Sr. Asst. Professor of Comp. Applications LBRCE-
Mylavaram, Krishna Dt., AP, India

Dr. M. Babu Reddy

Asst. Professor of Computer Science Krishna University,
Machilipatnam, AP, India

Dr. K. Rajasekhara Rao

Director, Usha Rama College of Engineering &
Technology, Vijayawada, AP, India

Abstract: In the entire software life cycle, it is evident that the maintenance phase occupies major role as the organizations are using the software for years together with up to date modifications and enhancements subject to the ever changing business dynamics. In light of the importance given on maintenance activity, a model has been developed to understand the basic characteristics of maintenance viz. Understandability, Usability and Modifiability and an effort has been made to result in suitable predictions of maintenance effort required for the chosen software application. In this paper, empirical study has been carried out on Internet banking applications and the proposed maintenance prediction model has been verified using various Usability aspects.

Keywords: Maintainability, Software Maintainability Models, Maintainability Process Models, Usability, Understandability, Modifiability.

I. INTRODUCTION

Maintenance is the ability of an item to be maintained; this ability stems from the aggregate of all design features which promote serviceability [1].

Maintainability measures the simplicity and pace with which a system can be repaired to working condition after a failure occurs. Maintainability is also defined as the ease with which modification can be made to a software system. These changes are required for defect removal, revision of the system to meet new requirements, enrichment of functionality, alteration or deletion of existing functionality or corrected when deficiencies occur and can be rectified, or action taken to minimize further maintenance costs.

Maintainability encompasses the following things

- ✓ An operational measure of effectiveness
- ✓ A design characteristics
- ✓ An Exclusive engineering knowledge for effective design

- ✓ A cost driver
- ✓ A Well planned product life-cycle

Section 2 of this paper highlighted the basic outline of Maintainability model both in product aspect and process aspect, in section 3 the outline of maintenance model is described, section 4 is design metrics for web application were presented and in section 5 of this paper an empirical evidence has been presented for maintenance prediction using internet banking application and an effort has been carried out to through some useful light in the lines of maintenance effort prediction.

II. OUTLINE OF THE MAINTAINABILITY MODEL

The outline of the maintainability model framework considers both the product and process aspects.

A. PRODUCT ASPECT

The proposed paper focuses on product maintainability model. It is called as Software Maintainability Model (SMM). The SMM can be applied to conventional software systems as well as to the systems developed according to new and emerging approaches. It covers the following parts:

a. COMMON CHARACTERISTICS

Irrespective of their differences, many software systems hold the same common preset characteristics that may be useful throughout the system. Here, the SMM model would identify these common criteria, define, implement, and validate them. These criteria would also apply to different software system types spanning from embedded real time systems to web based applications.

b. VARIABLE CHARACTERISTICS

Some characteristics are volatile. They may vary with the software types, product types, process or technology used. One must identify these differences and identify the differences amongst them, and suggest how they may be cross-compared, evaluated and calibrated [6].

c. QUALITATIVE AND QUANTITATIVE CHARACTERISTICS

Some characteristics would be quantitatively measured. On the other hand, some characteristics would have to be assessed qualitatively, for instance, the layout of code. Here, there is a need to define the qualitative and quantitative models for assessing maintainability characteristics.

d. TRACEABILITY CHARACTERISTICS

A maintainability model would never be complete if it is not complemented with traceability. The ability of evolving one artifact strongly depends on the possibility to trace it to its previous and successive artifacts, and to the artifacts on lower or higher system levels. Traceability is a fundamental qualification for linking a variety of system levels and for understanding of the system during development, evolution, and maintenance.

e. MATURITY LEVELS

The maintainability characteristics should be assigned to a number of product maturity levels. These levels would give a extent of assurance in the superiority of a software product, and help the organizations improve their product maintainability in a step-wise manner. Hence, there is a need to divide SMM model into several maturity levels.

f. TAXONOMY OF MAINTAINABILITY DEFECTS

The list of negative and constructive examples (patterns and anti-patterns) would help to locate defects linked with maintenance. This means that the architecture of design

models need to be studied to identify the bad patterns existed in design models. For improving maintainability, it is essential to identify and remove these anti-patterns.

g. SAMPLES OF MAINTAINABLE SOFTWARE SYSTEMS

To facilitate the product comparisons, samples of maintainable and unmaintainable software should be provided and that maintainability can be influenced in each software lifecycle phase, all the system document levels, beginning with the enterprise modeling and requirements phase and finishing with the testing one, would be candidates for defining the above-mentioned characteristics.

B. PROCESS ASPECTS

A model of product maintainability must be complemented with process models whose common task would be to manage maintainability throughout the whole software life-cycle. These are called as Maintainability Process Models (MPM)

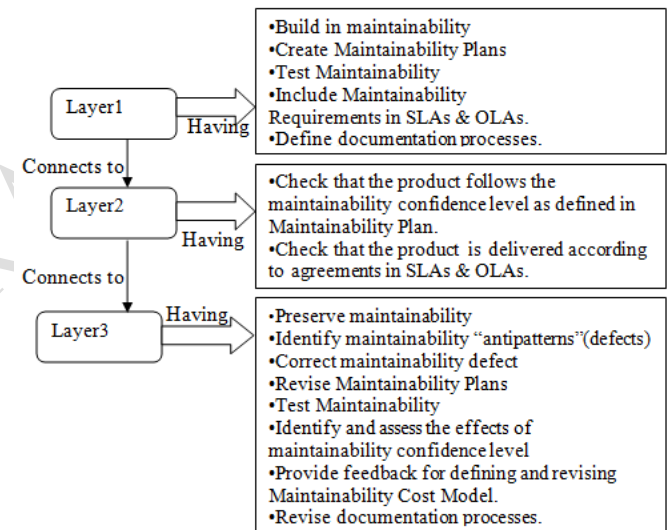


Figure 1: Identifying the role of maintainability within the life-cycle processes

As depicted in Figure 1, the role of the MPM models would be to build in, monitor and preserve maintainability into software, and systematically ensure that the maintainability-carrying properties are satisfied at all levels of system documents. Here, processes can be assumed to be run in parallel with the development and evolution processes, as suggested by the standard ISO/IEC 14764:1999 [7].

Just as the SMM proposes creating models of and metrics for product maintainability, the MPM suggests creating models of and metrics for observing various processes and their competence to build in or protect maintainability during the product life-cycle.

III. SUGGESTIBLE MAINTAINABILITY MODEL

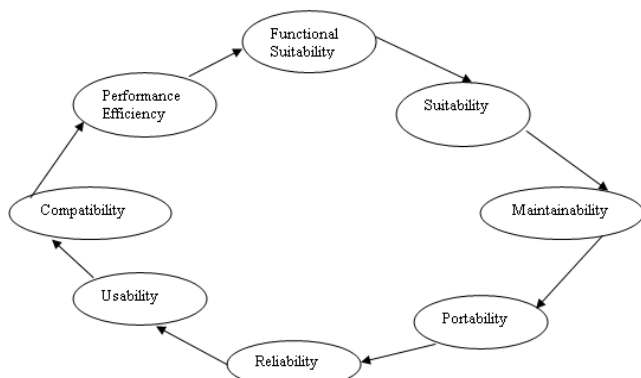


Figure 2

IV. DESIGN METRICS FOR WEB APPLICATION

Metric	Type Description
Size	Total number of server pages (NServerP) Total number of client pages (NClientP) Total number of web pages (NWebP)=(NServerP + NClientP) Total number of form pages (NFormP) Total number of form elements (NFormE) Total number of client components (style sheet and JavaScript components)(NClientC)
Structural Complexity	Total number of link relationships (NLinkR) Total number of Submit relationships (NSubmitR) Total number of builds relationships (NbuildsR) Total number of forward relationships(NForwardR) Total number of include relationships(NIncludeR) Total number of use tag relationships(NUseTagR)
Control Coupling	Number of relationships over number of web pages: $WebControlCoupling = (NLinkR + NSubmitR + NbuildsR + NForwardR + NIncludeR + NUseTagR) / NWebP$ [11]
Data Coupling	Number of data exchanged over number of server pages: $WebDataCoupling = (NFormE / NServerP)$
Reusability	Number of include relationships over number of web pages: $WebReusability = (NIncludeR / NWebP)$

Table 1

V. USABILITY EVALUATION-PREDICTION THROUGH EMPIRICAL EVIDENCE

For the past few years, various usability techniques were proposed by the researchers and these techniques are being incorporated into the design phase of applications. Among all these techniques, user testing and heuristic analysis are the

most frequently used ones[Tax,Lier & Bishu-2009].Heuristic evaluation depends on the set of identified heuristics where the selected set of experts examine user interface for design problems by judging its compliance with the identified heuristics[12].Normally heuristic evaluation will be carried out by multiple evaluators so as to improve the effectiveness of the method[13] but using more examiners by exceeding a threshold limit does not produce good results[12].

Heuristic evaluation covers all the major and global problems; however it uncovers most of the minor problems, which will be well addressed by user testing. Hence, it is advisable to use both the evaluation methods with an iterative design change between the usability testing and heuristic evaluation [12].

More specifically, priority must be given to the user needs and it is necessary to observe the user needs and it is necessary to observe the users while performing real transactions so as to spot out the feelings of the users. Nielson (2012) supports an idea of observing the users when they are using the system to improve the system performance. Normally, users may not be in a position to spot out minute design problems and they will be kept in loop of wasting time on accomplish desirable tasks.

EMPIRICAL STUDY

- TS₁ → Usability Testing Scenarios
- TS₂ → Functional Test Scenario
- TS₃ → Compatibility Test Scenario
- TS₄ → Database test Scenario
- TS₅ → Security Test Scenario
- TS₆ → General Test Scenario
- $UQ = \sum TS_{1i} + TS_{2j} + TS_{3k} + TS_{4l} + TS_{5m} + TS_{6n}$

Where
 $1 \leq i \leq 20$ $1 \leq j \leq 24$ $1 \leq k \leq 6$ $1 \leq l \leq 24$ $1 \leq m \leq 17$
 $1 \leq n \leq 6$
 Each TS_{rs} is measured within the range of 4 to 1 i.e.,
 4- Strongly Agree 3- Agree 2- Disagree 1-Total disagree

The purpose of this empirical study to came up with concrete evidence and to consolidate the findings for obtaining a general understanding of the user experience and opinion on the usage of internet banking and web based applications.

This study is also aimed at arriving guiding solutions towards the improvement of identified usability aspects. The collected data on user responses is analyzed to come up with suitable solutions to research questions in connection to usability aspects.

RESPONSE COLLECTION

As the target group comprising of the users who had the experience of using more than 2 internet banking sites response collection task is witnessed with greater complexity. Apart from accessing the relevant target group for expensive as there is involvement of human resources in identifying their usage experiences through various sources and analyze the data. Hence a limited sample of size 30

Respondents were selected from various organizations/sectors. The entire data collection process was carried out with in span of 40 days.

Out of the selected 30 respondents along with the end users, the people who were involved in design, development and testing were also targeted as the design and development decisions influences the usability.

The interview questions were designed to capture the real feelings of the respondents about the usage of internet banking web sites. However the user experience is also to be considered as per the complexity level carries with the period from which the user is using the applications. According to the user experience, appropriate scaling need to be done with capturing responses from the inexperienced end users. Scaling factor of 1.3 is taken as most of the users will get familiarity after 3 years of usage.

The questioner was clustered into six test scenarios which cover various aspects of usability, functionality, database usage, security, interoperability and portability. The corresponding respondents gave clear and correct response to their domain scenario.

Test Scenario	Category Sub Label	Details of Usage Aspect
TS ₁	TS _(1,01)	Accuracy Content
	TS _(1,02)	Total Consistency in font usage for text and buttons
	TS _(1,03)	Alignment of text and fields in the content
	TS _(1,04)	Accurate error messages corresponding to the concerned filed in the content
	TS _(1,05)	All fields in the content should be linked with tool tip text
	TS _(1,06)	Ability to progress to home page from every page
	TS _(1,07)	Separate the color specification (Probably gray color) for disabled fields
	TS _(1,08)	Checking of broken links and images.
	TS _(1,09)	Authorization message must be displayed for any kind of update and delete operation.
	TS _(1,10)	Verify the site on different resolutions available (640 x 480, 600x800 etc.?)
	TS _(1,11)	Check the end user can run the system or not
	TS _(1,12)	Check the tab should work properly and tab functionality with in the page
	TS _(1,13)	Scroll bar be supposed to appear only if necessary.
	TS _(1,14)	Information revisit even if error message need to be displayed on the page
	TS _(1,15)	Title must display on every web page title/page header

	TS _(1,16)	Verify Key board shortcuts are working or not
TS ₂	TS _(2,01)	Test the existing mandatory fields should be validated or not
	TS _(2,02)	Test the asterisk sign should exhibit for all the mandatory fields in the page.
	TS _(2,03)	Check the system should not show the error message for optional fields.
	TS _(2,04)	Examination that leap years are validated properly & do not cause errors/miscalculations.
	TS _(2,05)	Test the numeric fields should not accept the alphabets and proper error message should display.
	TS _(2,06)	Test for negative numbers if allowed for numeric fields.
	TS _(2,07)	Test division by zero should be handled properly for calculations.
	TS _(2,08)	Test the max length of every field to ensure the data is not truncated.
	TS _(2,09)	Test the pop up message ("This field is limited to 500 characters") should present if the data reaches the utmost size of the field.
	TS _(2,10)	Check that a confirmation message should display for update and delete operations.
	TS _(2,11)	Test the quantity of values should present in currency format.
	TS _(2,12)	Analysis all input fields for special characters.
	TS _(2,13)	Assessment the timeout functionality.
	TS _(2,14)	Test the Sorting functionality in the content.
	TS _(2,15)	Test the functionality of the buttons accessible.
	TS _(2,16)	Examination if any functionality fails the user gets redirected to the custom error page.
	TS _(2,17)	Analysis all the uploaded documents are opened properly.
	TS _(2,18)	Test the user should be able to download the uploaded files.
	TS _(2,19)	Check the email functionality of the system.
	TS _(2,20)	Test the java script is properly functioning in different browsers (IE, Firefox, Chrome, safari and Opera).
	TS _(2,21)	Test to see what happens if a user deletes cookies while in the site or not.
	TS _(2,22)	Test to see what happens if a user deletes cookies after visiting a site or not.
	TS _(2,23)	Check all the data inside combo/list box is arranged in chronological

		order.
TS ₃	TS _(3,01)	Test the website in dissimilar browsers (IE, Firefox, Chrome, Safari and Opera) and make sure the website is displaying properly.
	TS _(3,02)	Test the HTML version individual used is compatible with appropriate browser versions.
	TS _(3,03)	Test the images demonstrate correctly in different browsers.
	TS _(3,04)	Test the fonts are working in dissimilar browsers.
	TS _(3,05)	Investigation the java script code is usable in different browsers.
	TS _(3,06)	Test the Animated GIF's across dissimilar browsers.
TS ₄	TS _(4,01)	Validate the database name: The database name should match with the specifications.
	TS _(4,01)	Ensuring exact match between the requirement and database attributes like: name of the database, tables, columns etc.....
	TS _(4,02)	Acceptance of NULL values by mandatory columns only
	TS _(4,03)	Ensuring the existence of primary key and if required foreign key for each table
	TS _(4,04)	Substantiate the Stored Procedure:
	TS _(4,05)	Test whether the Stored procedure is installed or not.
	TS _(4,06)	Authenticate the Stored procedure name
	TS _(4,07)	Verify the available parameter names, types and number of parameters.
	TS _(4,08)	Analysis the parameters if they are required or not.
	TS _(4,09)	Test the stored procedure by deleting some parameters among them.
	TS _(4,10)	Test while the output is zero, the zero records should be precious.
	TS _(4,11)	Test the stored procedure by writing easy SQL queries.
	TS _(4,12)	Check whether the stored procedure returns the values or not
	TS _(4,13)	Test the stored procedure through sample input data.
	TS _(4,14)	Verify the activities of each flag in the table.
	TS _(4,15)	Corroborate the data gets properly saved into the database after the each page compliance.
	TS _(4,16)	Verify the complete data if the DML (Update, delete and insert) operations are performed.
	TS _(4,17)	Ensure the total length of every field: The field length in the back end and front end must be the same.

	TS _(4,18)	Verify the database names of QA, UAT and construction. The names should be same.
	TS _(4,19)	Authenticate the encrypted data in the list.
	TS _(4,20)	Verify the database size. Also test the response time of each query executed.
	TS _(4,21)	Check the data which is displayed on the front end and make sure it is same in the back end or not.
	TS _(4,22)	Corroborate the data validity by inserting the invalid data items in the database.
	TS _(4,23)	Verify the list of Triggers.
TS ₅	TS _(5,01)	Check the web page which contains important data like password, credit card numbers, secret answers for security question etc be supposed to submit via HTTPS (SSL).
	TS _(5,02)	Verify the significant information like password, credit card numbers etc should display in encrypted data.
	TS _(5,03)	Verify password regulations are implemented or not on all authentication pages like Registration page , forgot password, change password.
	TS _(5,04)	Corroborate if the password is changed the user supposed or not is able to login with the old password.
	TS _(5,05)	Check the error messages in the page should not display by important information.
	TS _(5,06)	Verify if the user is logged out from the system or user session was expired, the user should not be able to find the way of site.
	TS _(5,07)	Validate to access the secured and non secured web pages directly without login.
	TS _(5,08)	Verify the "View Source code" option is disabled and should not be visible to the user.
	TS _(5,09)	Confirm about the user account gets locked out if the user is entering the wrong password by several times.
	TS _(5,10)	Authenticate the cookies should not store password items.
	TS _(5,11)	Verify whether any functionality is not working or not , the system should not display any application, server, or database information.
	TS _(5,12)	Check about the SQL injection attacks.
	TS _(5,13)	Corroborate the role of user and their rights. For Example The requestor should not be able to access by admin page.

	TS _(5,14)	Verify various necessary operations are written in log files, and that information should be traceable.
	TS _(5,15)	Verify the values of session and information of cookie is in an encrypted format
	TS _(5,16)	Compelling for password reset in regular intervals of time
	TS _(5,17)	Verify the application for Brute Force Attacks are formed or not
TS ₆	TS _(6,01)	To resolve the performance, stability and scalability of relevance under unlike load conditions.
	TS _(6,01)	To verify if the present architecture can support the application or not at peak user levels.
	TS _(6,02)	To check which configuration sizing provides the best presentation level.
	TS _(6,03)	Validate the application and infrastructure bottlenecks.
	TS _(6,04)	Find out if the latest version of the software adversely have an impact on response time.
	TS _(6,05)	To assess the product and/or hardware to determine if it can handle projected load volumes or not.

Table 2: Broad Spectrum of Usability Aspects

For all the usability aspects mentioned in the table, from 30 users with different spectrum of experience who are operating 2 or more internet banking applications, responses on their general observations on particular usability criteria is collected and tabulated. Subject to the user experience, for the responses other than 4 i.e. ‘Strongly agree’ a scaling factor of 1.33 is applied while recording and quantifying the user responses. The following table summarizes the response of the heterogeneous group of users on various aspects of specific test scenario.

Usability Criteria	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	TSUQ _i	TSUQ _{ih}
TS _(1,01)	4	4	3	4	4	4	3	4	4	4	4	3	4	4	4	3.8	1
TS _(1,02)	4	4	4	4	4	4	4	4	4	3	3	4	4	4	4	3.86	1
TS _(1,03)	3	4	3	4	3	4	4	4	4	3	3	4	3	4	4	3.6	1
TS _(1,04)	3	3	4	3	3	3	4	4	4	3	4	3	4	3	4	3.46	1
TS _(1,05)	4	3	4	4	3	3	4	4	4	4	4	3	3	4	4	3.66	1
TS _(1,06)	3	2	3	3	2	2	1	1	2	2	2	1	2	1	2	1.93	0
TS _(1,07)	3	3	4	3	4	4	4	4	3	3	4	4	4	4	3	3.6	1
TS _(1,08)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1
TS _(1,09)	3	4	3	3	4	4	4	4	3	4	4	4	4	3	3	3.6	1
TS _(1,10)	3	3	4	3	4	3	4	4	3	4	4	3	4	4	4	3.6	1
TS _(1,11)	2	3	3	3	3	2	2	3	2	2	2	2	3	2	3	2.46	0
TS _(1,12)	2	2	3	2	1	2	2	1	1	2	2	1	2	2	2	1.8	0
TS _(1,13)	3	3	4	4	4	4	4	4	3	4	3	3	4	3	4	3.6	1
TS _(1,14)	3	3	3	2	3	3	4	4	4	3	3	3	4	4	4	3.33	1
TS _(1,15)	3	4	3	3	3	3	3	4	4	4	4	4	3	3	4	3.46	1
TS _(1,16)	4	4	3	3	3	3	3	3	4	4	4	3	4	3	4	3.46	1
UVQ _i	3.1	3.	3.3	3.18	3.18	3.31	3.43	3.	3.2	3.3	3.06	3.3	3.1	3.5			
	25	25	75	7	7	2	7	25	5	12	2	75	87				
UVQ _j	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
																WUQ	0.903

Table 3: Net Banking for active members (Minimum usage is for past 3 years)

Here

T₁ → Threshold value to draw conclusion on the acceptability of specific usability aspect of the cohesion test scenario.

If TSUQ_i ≥ T₁, TSUQ_{ih}=1, otherwise TSUQ_{ih}=0
Here TSUQ_i = ∑1/15*TS_{ij} for all i= 1 to 16 and j= 1 to 15

T₂ → Threshold value to draw conclusion on the acceptability by a specific user by considering all the aspects of the chosen test scenario.

If UUUQ_i ≥ T₂, UUUQ_{iv}=1, otherwise UUUQ_{iv}= 0
Here UUUQ_i = ∑1/16 * TS_{ij} for all 1 ≤ i ≤ 16 and 1 ≤ j ≤ 15
Final usability quotient will be calculated as WUQ = (∑ UVQ_{jv} + ∑ TSUQ_{ih})/(UB_i+UB_j) for all 1 ≤ i ≤ 16 and 1 ≤ j ≤ 15.
Here, UB_i and UB_j are Upper bound values of rows and columns.

Hence for the selected test scenario (TS₁), upper bound of usability quotient (UBUQ) is 240 as there are 16 usability aspects on which user responses were collected from 15 experienced users. After normalization, the optimum value for WUQ is ‘1’. The ratio between the extracted WUQ and the optimum value of WUQ gives the overall view of the Usability of the chosen application.

Based on the fraction of WUQ in upper bound of usability quotient (UBUQ), final conclusions were drawn and the considered internet banking applications can be classified into 4 groups based on the final WUQ value.

After performing the similar empirical study on the remaining scenarios TS₂, TS₃, TS₄, TS₅ and TS₆, the extracted WUQ values are ranging from 0.64 to 0.86 which is an indication that there is still some scope for further improvement of usability.

VI. CONCLUSION

By keeping the importance of maintenance in view, the present work emphasizes the consideration of various characteristics of usability of Internet banking applications though an empirical study. Experienced users were chosen as target group to study the usability aspect as it is one of the essential parameters of maintainability and this paper thrown some useful light in the lines of maintainability effort prediction and to highlight the pressing requirements of the software product to optimize the maintenance effort.

REFERENCES

- [1] Source: web.utk.edu/~kkirby/IE591/Maintainability
- [2] Cem Kaner, Walter P. Bond. “Software Engineering Metrics: What Do They Measure and How Do We Know?” 10th International software metrics symposium, Pages 1-12. IEEE Computer Society Press, 2004.
- [3] G. DiLucca, A. Fasolino, P. Tramontana, and C. Visaggio. “Towards the definition of a maintainability model for web applications”. In Proceeding of the 8th European Conference on Software Maintenance and Reengineering, pages 279– 287. IEEE Computer Society Press, 2004.
- [4] E. Mendes, N. Mosley, and S. Counsell. “Web metrics - estimating design and authoring effort”. IEEE Multimedia, 08(01):50–57, 2001.

- [5] Anil Kumar Malviya, Laxmi Shanker Maurya “Some Observation On Maintainability Metrics And Models For Web Based Software System” Volume 3, No. 5, May 2012, ISSN-2229-371X.
- [6] Koscianski A, Candido Bracarense Costa J, Combining Analytical Hierarchical Analysis with ISO/IEC 9126 for a Complete Quality Evaluation Framework, International Symposium and Forum on Software Engineering Standards, 1999, pp. 218-226.
- [7] ISO/IEC 14764: Information technology – Software Maintenance, Ref. No. ISO/IEC 14764:1999(E).
- [8] Franch Z, Carvalho, J P, A Quality-Model –Based Approach for Describing and Evaluating Software Packages, Joint Int. Conf. on Requirements Engineering, 2002, pp. 1-8.
- [9] Mira Kajko-Mattsson , Arie van Deursen et al...”A Model of Maintainability – Suggestion for Future Research”
- [10] Ilja Heitlager, Tobias Kuipers and Joost Visser “Practical Model for Measuring Maintainability – a preliminary report”
- [11] Anil Kumar Malviya, Laxmi Shanker Maurya “Some Observation On Maintainability Metrics And Models For Web Based Software System” Volume 3, No. 5, May 2012, ISSN-2229-371X.
- [12] Manzari And Trinidad-Christensen “User-Centered Design Of A Web Site” Information Technology And Libraries | September 2006
- [13] Hix, Deborah, Hartson, H. Rex, Nielsen, Jakob (1994): A Taxonomy for Developing High Impact Formative Usability Evaluation Methods. In ACM SIGCHI Bulletin, 26 (4) pp. 20-22.
- [14] Mohd Nazir, Raees A. Khan and Khurram Mustafa “A Metrics Based Model for Understandability Quantification “
- [15] D.Srinivasulu “Evaluation of Software Understandability Using Software Metrics “
- [16] Anil Kumar Malviya, Vibhooti Singh “Some Observations on Maintainability Estimation Model for Object Oriented Software in requirement, Design, Coding and Testing Phases” by, IJARCSSE with ISSN no:2277 128X, Page No: 174-178.
- [17] Northrop, L. Achieving Product Qualities Through Software Architecture Practices.
- [18] <http://www.sei.cmu.edu/architecture/cseet04.pdf> (2004).