INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATION AND MANAGEMENT

CONTENTS

CHIEF PATRON

PROF. K. K. AGGARWAL

Chancellor, Lingaya's University, Delhi
Founder Vice-Chancellor, Guru Gobind Singh Indraprastha University, Delhi
Ex. Pro Vice-Chancellor, Guru Jambheshwar University, Hisar

PATRON

SH. RAM BHAJAN AGGARWAL

Ex. State Minister for Home & Tourism, Government of Haryana
Vice-President, Dadri Education Society, Charkhi Dadri
President, Chinar Syntex Ltd. (Textile Mills), Bhiwani

CO-ORDINATOR

MOHITA

Faculty, Yamuna Institute of Engineering & Technology, Village Gadholi, P. O. Gadhola, Yamunanagar

ADVISORS

DR. PRIYA RANJAN TRIVEDI

Chancellor, The Global Open University, Nagaland

PROF. M. S. SENAM RAJU

Director A. C. D., School of Management Studies, I.G.N.O.U., New Delhi

PROF. S. L. MAHANDRU

Principal (Retd.), Maharaja Agrasen College, Jagadhri

EDITOR

PROF. R. K. SHARMA

Professor, Bharti Vidyapeeth University Institute of Management & Research, New Delhi

CO-EDITOR

MOHITA

Faculty, Yamuna Institute of Engineering & Technology, Village Gadholi, P. O. Gadhola, Yamunanagar

EDITORIAL ADVISORY BOARD

DR. RAJESH MODI

Faculty, Yanbu Industrial College, Kingdom of Saudi Arabia

PROF. PARVEEN KUMAR

Director, M.C.A., Meerut Institute of Engineering & Technology, Meerut, U. P.

PROF. H. R. SHARMA

Director, Chhatarpati Shivaji Institute of Technology, Durg, C.G.

PROF. MANOHAR LAL

Director & Chairman, School of Information & Computer Sciences, I.G.N.O.U., New Delhi

PROF. ANIL K. SAINI

Chairperson (CRC), Guru Gobind Singh I. P. University, Delhi

PROF. R. K. CHOUDHARY

Director, Asia Pacific Institute of Information Technology, Panipat

DR. ASHWANI KUSH

Head, Computer Science, University College, Kurukshetra University, Kurukshetra

DR. BHARAT BHUSHAN
Head, Department of Computer Science & Applications, Guru Nanak Khalsa College, Yamunanagar
DR. VIJAYPAL SINGH DHAKA
Head, Department of Computer Applications, Institute of Management Studies, Noida, U.P.
DR. SAMBHAVNA
Faculty, I.I.T.M., Delhi
DR. MOHINDER CHAND
Associate Professor, Kurukshetra University, Kurukshetra
DR. MOHENDER KUMAR GUPTA
Associate Professor, P. J. L. N. Government College, Faridabad
DR. SAMBHAV GARG
Faculty, M. M. Institute of Management, Maharishi Markandeshwar University, Mullana
DR. SHIVAKUMAR DEENE
Asst. Professor, Government F. G. College Chitguppa, Bidar, Karnataka
DR. BHAVET
Faculty, M. M. Institute of Management, Maharishi Markandeshwar University, Mullana

ASSOCIATE EDITORS

PROF. ABHAY BANSAL
Head, Department of Information Technology, Amity School of Engineering & Technology, Amity University, Noida
PROF. NAWAB ALI KHAN
Department of Commerce, Aligarh Muslim University, Aligarh, U.P.
DR. ASHOK KUMAR
Head, Department of Electronics, D. A. V. College (Lahore), Ambala City
ASHISH CHOPRA
Sr. Lecturer, Doon Valley Institute of Engineering & Technology, Karnal
SAKET BHARDWAJ
Lecturer, Haryana Engineering College, Jagadhri

TECHNICAL ADVISORS

AMITA
Faculty, Government M. S., Mohali
MOHITA
Faculty, Yamuna Institute of Engineering & Technology, Village Gadholi, P. O. Gadhola, Yamunanagar

FINANCIAL ADVISORS

DICKIN GOYAL
Advocate & Tax Adviser, Panchkula
NEENA
Investment Consultant, Chambaghat, Solan, Himachal Pradesh

LEGAL ADVISORS

JITENDER S. CHAHAL
Advocate, Punjab & Haryana High Court, Chandigarh U.T.
CHANDER BHUSHAN SHARMA
Advocate & Consultant, District Courts, Yamunanagar at Jagadhri

SUPERINTENDENT

SURENDER KUMAR POONIA

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing, Human Resource Management, General Management, Banking, Insurance, Corporate Governance and emerging paradigms in allied subjects like Accounting Education; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Monetary Policy; Portfolio & Security Analysis; Public Policy Economics; Real Estate; Regional Economics; Tax Accounting; Advertising & Promotion Management; Business Education; Business Information Systems (MIS); Business Law, Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labor Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; Public Administration; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism, Hospitality & Leisure; Transportation/Physical Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Digital Logic; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Multimedia; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic and Web Design. The above mentioned tracks are only indicative, and not exhaustive.

Anybody can submit the soft copy of his/her manuscript **anytime** in M.S. Word format after preparing the same as per our submission guidelines duly available on our website under the heading guidelines for submission, at the email addresses: **1** or info@ijrcm.org.in.

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____

THE EDITOR

IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF _____.**

(e.g. Computer/IT/Engineering/Finance/Marketing/HRM/General Management/other, please specify).

DEAR SIR/MADAM

Please find my submission of manuscript titled '_____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication anywhere.

I affirm that all author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:

Designation:

Affiliation with full address, contact numbers & Pin Code:

Residential address with Pin Code:

Mobile Number (s):

Landline Number (s):

E-mail Address:

Alternate E-mail Address:

2. **MANUSCRIPT TITLE:** The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS:** The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ABSTRACT:** Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

5. **KEYWORDS:** Abstract must be followed by list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT:** Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of the every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS:** All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS:** All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT:** The main text should follow the following sequence:

INTRODUCTION

REVIEW OF LITERATURE

NEED/IMPORTANCE OF THE STUDY

STATEMENT OF THE PROBLEM

OBJECTIVES

HYPOTHESES

RESEARCH METHODOLOGY

RESULTS & DISCUSSION

FINDINGS

RECOMMENDATIONS/SUGGESTIONS

CONCLUSIONS

SCOPE FOR FURTHER RESEARCH

ACKNOWLEDGMENTS

REFERENCES

APPENDIX/ANNEXURE

It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed 5000 words.

10. **FIGURES &TABLES:** These should be simple, centered, separately numbered & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS:** These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES:** The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio," Ohio State University.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITE

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on December 17, 2011 http://epw.in/user/viewabstract.jsp

SOFTWARE DEFECTS IDENTIFICATION, PREVENTIONS AND AMPLIFICATION IN SDLC PHASES

BHOJRAJ HANUMANT BARHATE
ASST. PROFESSOR
BHUSAWAL ARTS, SCIENCE & P.O.NAHATA COMMERCE COLLEGE
BHUSAWAL

ABSTRACT

The present paper finds that to detect and cure software defects is one of the best band esteemed quality activity which saves time, cost and man power. It also delivers qualited and more user friendly software product to customers.

KEYWORDS
SDLC – Software Development Life Cycle.

INTRODUCATION

In software development area there is no any such software development company which develop 100% error free software. What is Software error? Occurrence of the technical difficulty in software is the error. But when any customer complaints about software then we can conclude about error. Software error is error -resulting from bad code in some program involved in producing the erroneous result[1]. To maintain software error of defects it is necessary to concentrate over phases of SDLC "Software defect" has many interchangeable names: software bug, error, fault, failure, crash and anomaly. The institute for electrical and electronic engineering (IEEE), the standards body regulating even the software profession, defines a software bug as a programming error that causes software to malfunction. IEEE sees defect as a product anomaly when software does not conform to customer expectation and specification. A crash is an extreme case of a software defect that stops the software from further working.

SOME SOURCES OF DEFECTS

Software defects can arise from misinterpreting customer requirements, poor programming habits, wrong programming logic, poor testing and even untested software implementation scenarios. For example, a customer specifying a blue background to a website and the developer producing a white background is a defect. Even though the website runs fine, it does not meet customer requirements.

DEFECTS IN SDLC

Defects can occur at any phase in the software development life cycle (SDLC). The SDLC phases are requirements analysis, systems design, program design, program implementation, program testing, system testing and maintenance. Thus since defects can occur at any phase, the defect life cycle involves quality assurance at every phase. Configuration management enables defects to be recorded and tracked to removal. Thus tools for configuration management and defect tracking will be required in the defect life cycle quality assurance.

1. Customers don't know what they want

Due to the Technical concern behind the concept software development the customer who developes software can't have any clear idea about the problem. If software developer identifies problem from customer then customer can take care about the clear understanding about problem. Enlist all the task that to be completed by software developer. List of all the task/functions are finalised here.

2. Requirements change during the course of the project

The second most common problem with software projects is that the requirements defined in the first phase change as the project progresses. This may occur because as development progresses and prototypes are developed, customers are able to more clearly see problems with the original plan and make necessary course corrections; it may also occur because changes in the external environment require reshaping of the original business problem and hence necessitates a different solution than the one originally proposed. Good project managers are aware of these possibilities and typically already have backup plans in place to deal with these changes.

To solve this problem, you should:

- Have a clearly defined process for receiving, analyzing and incorporating change requests, and make your customer aware of his/her entry point into this process.
- Set milestones for each development phase beyond which certain changes are not permissible -- for example, disallowing major changes once a module reaches 75 percent completion.
- Ensure that change requests (and approvals) are clearly communicated to all stakeholders, together with their rationale, and that the master project plan is updated accordingly.

3. Customers doesn't follow principles

The analysis in each phase may be extended due to some long activities. Due to the assurance by developer to the customer about quality developer tries to develop Quality software. Customer can't understand this difficulty and ask for final product again and again.

To solve this problem, you should:

- Convert the software requirements specification into a project plan, detailing tasks and resources needed at each stage and modeling best-case, middle-case and worst-case scenarios.
- Ensure that the project plan takes account of available resource constraints and keeps sufficient time for testing and quality inspection.
- Enter into a conversation about deadlines with your customer, using the figures in your draft plan as supporting evidence for your statements. Assuming that your plan is reasonable, it's quite likely that the ensuing negotiation will be both productive and result in a favorable outcome for both parties.

4. Communication gap between customer and Developer

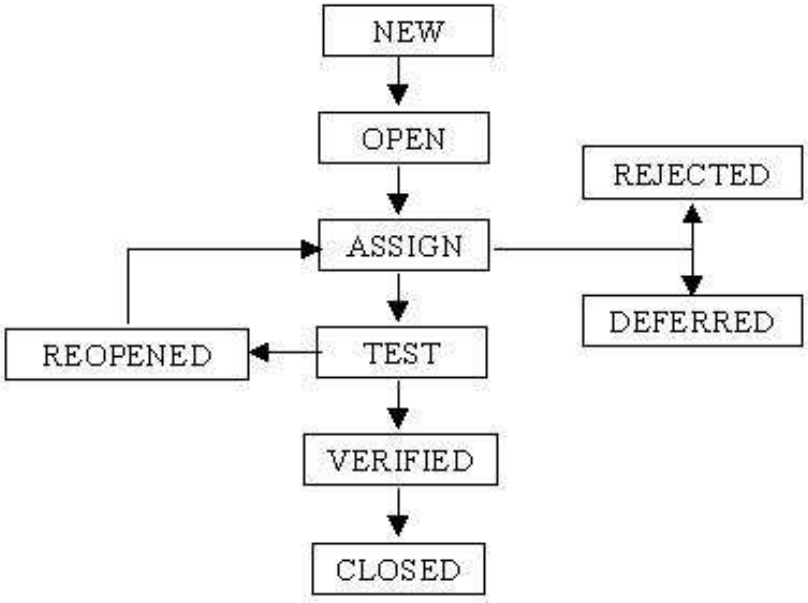
Often, customers and engineers fail to communicate clearly with each other because they come from different worlds and do not understand technical terms in the same way. This can lead to confusion and severe miscommunication, and an important task of a project manager, especially during the requirements analysis phase, is to ensure that both parties have a precise understanding of the deliverable and the tasks needed to achieve it.

To solve this problem, you should:

- Take notes at every meeting and disseminate these throughout the project team.
- Be consistent in your use of words. Make yourself a glossary of the terms that you're going to use right at the start, ensure all stakeholders have a copy, and stick to them consistently.

BEHAVIOR OF SOFTWARE DEFECTS

Basili and Boehm assembled a list of ten rules of thumb which main purpose was to highlight pitfalls in software engineering. Seven of these rules are directly related to how defects impact the project [Boehm and Basili, 2001]. First, it is 100 times more expensive to correct a defect after delivery than during the requirements or design phase. Second, 40 to 50 percent of the effort in the project is spent on rework which could have been avoided. Third, 20 percent of the defects results in 80 percent of the rework and 80 percent of the defects come from 20 percent of the modules while half of modules are almost free of defects. Developing high dependability software are often 50 percent more expensive than low dependability software, and 90 percent of the downtime comes from 10 percent of the defects. However, investing the 50 percent extra is well worth it if the software is to be maintained. Last, 40 to 50 percent of software contains defects which are nontrivial. The results described by Boehm and Basili in [Boehm and Basili, 2001] gives an impression of how software defects influence the total costs of software projects.



DESCRIPTION OF VARIOUS STAGES

1. New: When the bug is posted for the first time, its state will be "NEW". This means that the bug is not yet approved.
2. Open: After a tester has posted a bug, the lead of the tester approves that the bug is genuine and he changes the state as "OPEN".
3. Assign: Once the lead changes the state as "OPEN", he assigns the bug to corresponding developer or developer team. The state of the bug now is changed to "ASSIGN".
4. Test: Once the developer fixes the bug, he has to assign the bug to the testing team for next round of testing. Before he releases the software with bug fixed, he changes the state of bug to "TEST". It specifies that the bug has been fixed and is released to testing team.
5. Deferred: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
6. Rejected: If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "REJECTED".
7. Duplicate: If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "DUPLICATE".
8. Verified: Once the bug is fixed and the status is changed to "TEST", the tester tests the bug. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "VERIFIED".
9. Reopened: If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "REOPENED". The bug traverses the life cycle once again.
10. Closed: Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to "CLOSED". This state means that the bug is fixed, tested and approved.
While defect prevention is much more effective and efficient in reducing the number of defects, most organization conducts defect discovery and removal. Discovering and removing defects is an expensive and inefficient process. It is much more efficient for an organization to conduct activities that prevent defects.

BUG LIFE CYCLE IMPLEMENTATION GUIDELINES

- Make sure the entire team understands what each bug status exactly means. Also, make sure the bug life cycle is documented.
- Ensure that each individual clearly understands his/her responsibility as regards each bug.
- Ensure that enough detail is entered in each status change. For example, do not simply DROP a bug but provide a reason for doing so.
- If a bug tracking tool is being used, avoid entertaining any 'bug related requests' without an appropriate change in the status of the bug in the tool. Do not let anybody take shortcuts. Or else, you will never be able to get up-to-date bug metrics for analysis.

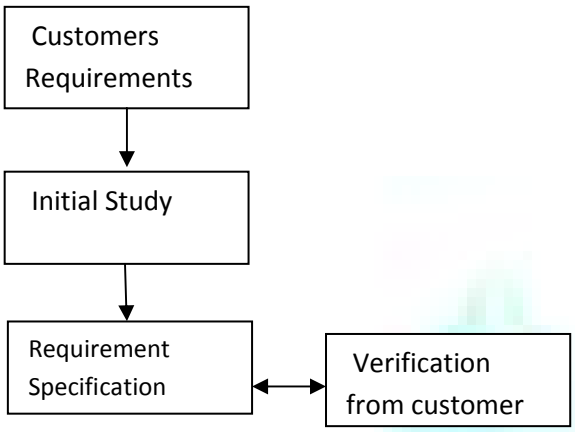
TECHNIQUES FOR SOFTWARE DEFECT PREVENTION AND IDENTIFICATION

A quality process should produce close to zero-defect software that meets the user requirements. In this article, we will go through step-by-step, the various practices/techniques that can help you prevent defects in your software, and how to catch them if they already exist.

TO KNOW CUSTOMER REQUIRMRNTS

When a customer provides you a requirement, it's not that you start working on it there and then (Figure 1, below). You first need to understand clearly what the customer wants. Once you have gone through the requirement, put down what you have understood. Then, get your understanding confirmed from the customer. Doubts, if any, in the requirement specification, must be clarified at this stage. Do not procrastinate or hesitate in asking your queries. If customers are many try to understand all the customers and collect common data or unambiguous data from that collection.

FIG. 1: REQUIREMENT WORKFLOW FOR CLARIFICATION



While working in development process there are many defects creeping in due to misinterpretation of the requirement specification. Fixing such defects at a later stage can prove costly. So, it is very important that you get your understanding verified from the customer.

PEER REVIEW

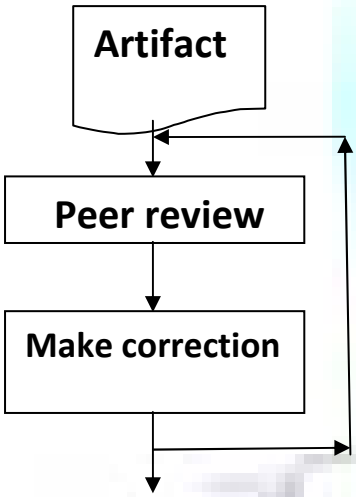
Software project development is one of the Team work activity .Functioning as a team is a skill. Delivering a high quality product is no responsibility of an individual. It's the entire team who is responsible for it. If the product fails, each team member is responsible for it.

Review is an important part of team work. Peer Review refers to participation of team members in the development and assessment of task/activity performed by an individual.

In this process, illustrated in Figure 2 below, a team member requests an artifact to be reviewed. The other team members then provide their review comments which may include corrections, suggestions and doubts on the artifact.

The artifact is then updated based on the review comments. This process is repeated till the artifact is up to satisfaction of all the team members. Of course, in case of any conflict it is the Project Leader who makes the final judgment.

FIGURE 2: HUNTING FOR ARTIFACTS



REVIEWS IN DIFFERENT SOFTWARE DEVELOPMENT STAGES

a. Requirement Specification Review. As we discussed in the last point, understanding the customer requirement is very important. So, while you prepare the Requirement Specification, get it reviewed by your team members. Ten people in a team may have ten interpretations of the specification. Discuss it out within your team, and then pass your understanding to the customer for verification.

b. Design Review. Once the requirement specification gets finalized, we move on to the design phase. In the design stage, you would now think about how to approach the problem.

As you will agree, review at this stage is vital. Selection of a wrong strategy can put the entire system in a miserable state. Reviews performed at this stage will help you in:

*Analyzing various strategies to solve the problem.

* Analyzing feasibility of each strategy.

* Knowing advantages/disadvantages of each strategy.

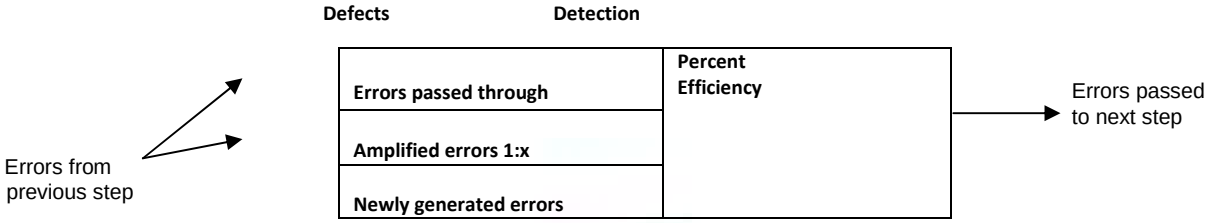
c. Code Review. Code Review involves examining the source code to find defects. While working on development of system software tools, I have been a witness to how code review can really help you find defects.

It is good to let someone in your team walk through you code, and do the review. All the members must review the code changes with respect to other modules and give their feedback in case some side-effect is suspected.

Many defects such as memory leaks, wrong passing of arguments, unreachable code, lack of readability, high complexity and maintainability issues can be identified via code review. Finding defects at the coding stage, and fixing them there and then, would prove to be less expensive than finding them in the testing stage.

A defect amplification model [IBM81] can be used to illustrate the generation and detection of errors during the design and code generation actions of a software process.

DEFECTS DETECTION



DEVELOPMENT STEP

Figure 2.4 illustrates hypothetical example of defect amplification for a software development process in which no reviews are conducted. As shown in the figure each test step is assumed to uncover and correct fifty percent of all incoming errors without introducing new errors (an optimistic assumption). Ten preliminary design errors are amplified to 94 errors before testing commences. Twelve latent defects are released to the field. Figure 2.5 considers the same conditions except that design and code reviews are conducted as part of each development step. In this case, ten initial preliminary design errors are amplified to 24 errors before testing commences.

Only three latent defects exist. By recalling the relative cost associated with the discovery and correction of errors, overall costs (with and without review for our hypothetical example) can be established.

To conduct reviews a developer must expend time and effort and the development organization must spend money. However, the results of the preceding or previous, example leave little doubt that we have encountered a “Pay now or pay much more lately” syndrome.

REFERENCES

1. Defect Reduction. Top 10 List. Barry Boehm, University of Southern California. Victor R. Basili, University of Maryland 3 Jan 2001 – available at www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.78.pdf
2. Rojer Pressman (1997), “Software Engineering A Practitioner’s Approach,”, McGRAW HILL INTERNATIONAL EDITIONS [ISBN 0-07-114603- 4ed,Chapter-8A
3. Software Defect Life cycle available at http://www.ehow.com/about_6682230_software-defect-life-cycle.html#ixzz1ZbokXLqz
4. Software Testing Fundamental available at http://softwaretestingfundamentals.com
5. SOFTWARE QUALITY & TESTING (MSIT - 32) by Dr. B.N. Subraya, Infosys Technologies Ltd., Mysore
6. Testing Articles available at http://www.exforsys.com/tutorials/testing

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Computer Application and Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mails i.e. **infoijrcm@gmail.com** or **info@ijrcm.org.in** for further improvements in the interest of research.

If you have any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-

Co-ordinator