International Journal of Advanced Smart Convergence Vol.8 No.4 207-213 (2019)
http://dx.doi.org/10.7236/IJASC.2019.8.4.207

IJASC 19-4-26

Record keeping system of bugs for software quality assurance

Ahmed Mateen*[1], Qingsheng Zhu[2], Salman Afsar[3]

*[1] *Ph.D., Scholar, College of Computer Science, Chongqing University, China.*
Lecturer, Department of Computer Science, University of Agriculture Faisalabad, Pakistan.
[2]*Ph.D., Professor, College of Computer Science, Chongqing University, China.*
[3] *Ph.D., Chairman, Department of Computer Science, University of Agriculture Faisalabad, Pakistan.*
ahmedmatin@hotmail.com

Abstract

Assistance for the quality assurance management and developers to provide high quality software products. Using a bug record keeping system is exceptionally important in software progress, and it is followed vastly by majority of software producing companies in modern era. Regular application of a bug record keeping system is very helpful in developing software systems. We developed this system which helps the software testing team to keep a complete record of their testing activities during the software testing, also increases your confidence in the software quality, class and worth. Our proposed system offers trouble-free and effortless approaches to acquire desired information about bug, also produces different kinds of reports like summary reports, detailed reports etc. It gives facility to create, delete and update any project. Our developed application system is designed by using visual c# at front end and sql server 2008 management studio express at back end.

Keywords: *Visual C#.NET, Visual Studio 2010, SQL Server 2008, Crystal Reports 10, Crystal Report Viewer*

1. Introduction

Computer plays an incredible role in modern era as it is making uprising changes in all fields of life like education, defense, science & technology, communication etc. Software market and its applications in business industry are the huge skill dependent marketplace in 21st century which have a substantial cost for their services roughly more than 200 billion [1]. Production of software provides many potential benefits for developing countries, including creation of jobs, skills and income. Computerized system is ultimate and crucial for handling complex issues [2]. Computer plays very important role in development, progress, enhancement and prosperity of any country. There are number of software houses in the world, but limited software houses are known and recognized, because they are engaged in producing quality conscious products

Manuscript Received: December.6, 2019 / Revised: December. 11, 2019 / Accepted: December. 13, 2019
Corresponding Author: ahmedmatin@hotmail.com
Tel: +86-18602399928
College of Computer Science. Chongqing University, Ph.D. Scholar, China.
Lecturer, Department of Computer Science, University of Agriculture Faisalabad, Pakistan

[3]. Every quality conscious software house must have a quality assurance section. Software testing team exists in quality assurance section (qas). Software quality is achieved with the help of quality assurance section which test the software against all its requirements. Quality assurance section process the buggy project for generating bug free project [4]. Software quality assurance (sqa) consists of a means of monitoring the software production procedures and schemes used to guarantee quality. Quality assurance makes sure the project will be completed based on the earlier agreed terms, standards and functionality required without deficiencies and potential troubles. It watches and attempts to improve the development process from the beginning of the project to ensure this [5]. It is oriented to "avoidance". The basic objective of bug record keeping system is to develop strategy and technique to report software bugs and summarizing the relevant information in a user's friendly graphical user interface (gui) environment for software quality assurance section. Quality assurance involves actions intended to guarantee that the development and maintenance procedure is sufficient to ensure a system will meet its goals. Software quality assurance only increases your confidence in the software quality [6].

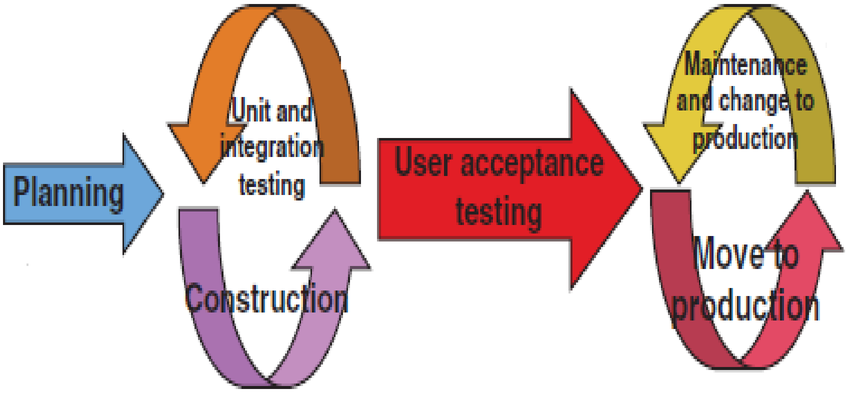
Planning
Unit and integration testing
Construction
User acceptance testing
Maintenance and change to production
Move to production

Figure 1. software quality assurance processes

In korean cuisine, seafood prepared by salting and fermentation is known as eojang or jeotgal, where degradation and decomposition of the main ingredients are inhibited during the fermentation process [7]. Moreover, decomposition of the protein in the raw materials by beneficial organisms produces a unique flavor and breaks down carbohydrates, lipids, and organic acids, thereby enhancing the flavor, making these fermented products important seasonings in kimchi and other korean foods [8]. Squid jeotgal is one well-known ready-to-eat seafood item that is very popular for its taste and flavor. Squid jeotgal is rich in lactic acid bacteria (lab) and yeast [9]. The consumption of squid jeotgal can inhibit somatic mutation, and this seafood exhibits anticancer and antioxidant properties due to the highly bioactive compounds. In recent years, fermented seafood enriched with different sub-ingredients, including garlic, hot pepper, and plant extract, has been produced to achieve variations in the flavor, aroma, and texture, and to increase the health function and nutritional value. Scientist reported that the addition of dietary fiber from rice bran can enhance the shelf-life of sausage. Low-salt squid combined with edible plant showed no change in the sensory quality but exhibited increased antioxidant and anti-diabetes activity. In our previous study, the anticancer activity of fermented squid jeotgal was improved by enrichment with soymilk. Soybean and its phenolic compounds may be used as potential functional sub-ingredients in fermented food. Isoflavones are the main phenolic compounds in soybean. Isoflavones are present in soy foods as aglycones as well as in the glycosylated forms [10]. The most abundant isoflavones are genistin and daidzin that are poorly absorbed in the human body, whereas the aglycone forms (daidzein, glycitein, and genistein) play important roles as antioxidants and inhibit cancer development and heart-related disease. Genistein and daidzein can be inter-converted via fermentation. Therefore, fermented foods enriched with isoflavones are potentially functional foods that can be used in higher functional products. The aim of the present study is to increase the food value of fermented squid by the addition of soy isoflavones or bioactive compounds produced during fermentation. The antioxidant capacity of enriched fermented squid has yet to be investigated. The effect of adding isoflavone on the microbial and sensory properties of fermented squid is also evaluated.

2. Methodology

Designing an application system that carries out all the demands of the clients and it is extremely crucial for software developing panel to understand and control all the chains and ladders of the desired system. The growth of an application system is made under several complex processes which are oriented as steps of a ladder named software development life cycle (sdlc). I used system development life cycle (sdlc) technique for developing the proposed system for software quality assurance section of a software house. The system development life cycle (sdlc) is an organized approach to develop bug record keeping system for software quality assurance section. System development life cycle depicts the way you systematize your activities. It consists of many steps and involves different persons. It comprises of software introductory study, software analysis, software design, software development, software implementation and software maintenance [11].

Before designing the bug, record keeping system, i made thorough analysis of the manual system for software quality assurance section. After that the object-oriented techniques are used to develop the system [12]. Testing techniques will be applied on every phase to make the system more secure and error free that enhances the consistency and precision of the system as well as knowing how records are sustained in the automated system [13]. The selection of right tool plays a key role in developing any software product. By analyzing and evaluating the nature of the system and requirements of the organization, i selected sql server 2008 management studio express as back end and visual c# as front end [14].

By considering the technicalities and complexities of the required system, the vital information about bugs are transformed into special reports [15]. Crystal reports is a business intelligence application used to design and generate reports from a wide range of data source [16]. Several other applications, including microsoft visual studio, at one time bundled an oem version of crystal reports as a general-purpose reporting tool. Crystal reports is a popular report writer and was especially so when microsoft bundled it with visual studio versions 2003 through 2008. Crystal reports 10 (for designing the bug reports) and crystal report viewer (used for viewing the bug reports) is used to design the different kinds of bug reports like complete status report, individual project bug report, total added projects for testing etc. Crystal reports allows users to graphically design data connection(s) and report layout [17]. In the database expert, users can select and link tables from a wide variety of data sources, including microsoft excel spreadsheets, sql databases, business objects enterprise business views, and local file system information. Fields from these tables can be placed on the report design surface and can also be used in custom formulas. Formulas can be evaluated at several phases during report generation as specified by the developer.

3. Result and discussion

The system evaluation is the process of' measuring the performance of the existing system with the new developed system, knowledge of the organizational structure was helpful in hardware decisions. Also present and future workload of transactions and the ability of hardware and software to handle the workloads adequately is evaluated. The evolution is not only necessary but also unavoidable to keep the system updated in term of business and economic environment as well as technological change in the electronic data processing. So the system which maintains accuracy, timeliness and completeness will be assessed as successful [10].

The problem of systems engineering is to design and implement a system that meets the needs of system stakeholders. Analysis of stakeholder needs results in a variety of system requirements. Software development process not only brings great benefits to the software development team but increase the quality of the software products as well. Appropriate tactics and techniques are used to build the bug record keeping application system for meeting client’s needs, requirements and specifications [8].

The new system has many implications and advantages over the existing system, some of the advantages are:

3.1. Generate voice sound

Record keeping system of bugs generates voice sound against specific actions or events to assist the user like generate sound against invalid input in text field, generate sound when record save, deleted etc. The system generates the voice sound against specific constraints/actions like login, save delete etc.

3.2. Login authentication

Login form of the system only allows or authenticates those operators/users which are added through the add operator form. Only added users can operate the system, this system protects itself against unauthorized internal or external access and intentional damage.

3.3. Detailed bug information

Bug information form consists of several important fields that should be filled against each bug/issue occur in the software. It provides detailed necessary attributes for storing vital investigation about bug, so project id, bug status and bug category fields are compulsory which must be filled against each bug/issue generated in the software product.

3.4. Bugs reporting

Bugs reporting is a complex and difficult process, bugs reporting varies from software house to software house. Different software houses use specific fields for bugs reporting. Developed system provides moderate fields for describing bug nature clearly.

3.5. Concise and complete

Designed bug reporting is not in perfect english, it is in technical terminologies, also emphasizes on conciseness and completeness.

3.6. Bugs reporting is in 3w’s format (rule of “3w”)

- What happened?
- Where it happened?
- Under which circumstances it happened?

Developed system generates all necessary and essential information about any bug detected by the software quality assurance and provides complete elements of the bug to developers for fixing point of view in order to gaining maximum confidence on software product.

3.7. Mail to developer

Designed system provides the opportunity to software quality assurance (software testers) e-mail the reported bugs to developers with the intent of time saving.

3.8. Help about forms

Developed system also provides the facility to operators for getting help about different forms of the system in an efficient and effective manner.

3.9. Integrity constraints

Integrity constraints and checks are implemented on every needed fields and places for preventing the wrong entry data/information.

3.10. Updating and deleting process

The new developed system provides the facility of updation and deletion the necessary/desired information against unique project identification under authorized access.

3.11. Add operator/user form

Add operator/user form used for adding the operators which can operate the system. Only administrator can add, delete and update the operator/user status.

3.12. Delete operator/user form

Delete operator/user form used for deleting the operators which can operate the system. When any one operator deleted, then it becomes disable i.e. It cannot operate the system further. Only administrator can add, delete and update the operator/user status.

3.13. Change operator/user password form

Change operator/user form used for changing the operator password, if any operator want to change his/her login password due to security reasons/threats. Only administrator can add, delete and update the operator/user status.

3.14. Add project for testing purposes

This system is mainly based on maintaining/tracking/recording bugs/issues which are occur in a project/system. Add project form used for adding project for testing purposes. All projects are added for testing and reporting the bugs.

3.15. Project information form:

Project information form provides the facilities, manipulation and amendments in the project profile like searching, updating and deleting.

3.16. Project information form (searching through combo box button)

Search combo box button of project information form used for searching or getting specific project information which already stored in the database against unique project id.

3.17. Project information form (update button)

Update button of project information form used for updating or editing specific project information which already stored in the database against unique project id.

3.18. Project information form (delete button)

Delete button of project information form used for deleting or removing specific project information which already stored in the database against unique project id.

3.19. Assign and de-assign project

The system should provide the capability to qa manager to assign and de-assign and edit a project.

3.20. Bug severity level

The system provides the capability to create, remove and edit the severity level.

3.21. Bug number

The system provides the capability to search based on bug number.

3.22. Detail status report

The system provides the capability to view and print the detail status report.

3.23. Bug creation

The system provides the capability to add new or existing bug.

3.24. Important bugs reporting fields

- Bug cycle (x1, x2, x3).
- Bug probability (low, medium, high).
- Bug status.
- Bug category (major, database, gui).
- Bug priority (must fix, should fix, fix when time, low priority).
- Bug severity (show stopper, high, medium, low).

System grants these fields which are used to depict the nature and environment of bugs.

4. Conclusion

To provide the information and manage the reported and fixed bugs daily bases is very beneficial for software quality assurance section of a software house. This system is very efficient, effective, fast and meets the requirements of the software house. It helps the software house to check previous record of reported bugs and added projects whenever they needed. Good coding styles are adapted in the developed system that works according to the requirements and customer needs, also system provides good coding format which intend to support and facilitate the organizational needs by providing any change could be made very easily and efficiently both internally and externally. The manual system was very slow in data processing and information retrieval. It was extremely difficult to handle a large amount of data about software product bugs manually and achieved valuable results. Through computerization such sort of problems can be overcome, and reliable results can be obtained. This management information system gives a number of facilities to insert, delete, edit, save, modify etc. The record in the user-friendly and gui (graphical user interface) environment.

With the help of developed system, the unexpected mistakes have been minimized and the chance of error also eliminated. The primary goal of this bug record keeping system of bugs is to made software quality assurance section more powerful and coordinated in their testing activities and track out the bug records in a well-defined, controlled and organized manner. Overall system is fully outfitted and equipped with all its ingredients and elements against the existing ones.

References

[1] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, "Natural language processing for requirements engineering: The best is yet to come," *IEEE software*, Vol. 35, No. 5, pp. 115-119, 2018 DOI: https://doi.org/10.1109/MS.2018.3571242

[2] A. Arora and M. Sinha, "Web Application Testing: A Review on Techniques, Tools and State of Art" *International Journal of Scientific & Engineering Research,* Vol. 3, No. 2, 2012. ISSN 2229-5518.

[3] K. Dheeraj, "An overview and analysis of automated testing tools: Ranorex, Test complete, Selenium" *International Research Journal of Engineering and Technology (IRJET),* Vol. 04, No. 10, pp. 1575-1579, 2017.e-ISSN: 2395- 006, pISSN: 2395-0072

[4] Li. Yuan-Fang, K. Paramjit, D. David and L. Dowe, "Two decades of Web application testing a survey of recent advances," *Information Systems*, Vol. 43, pp, 20-54, 2014. DOI:http: //dx.doi.org/10.1016/j.is.2014.02.001

[5] M. Mohamed, "Evaluation of automated web testing tools," *International Journal of Computer Applications Technology and Research,* Vol. 4, No. 5, pp, 405 - 408, 2015. ISSN: 2319–8656

[6] K. Ali, and X. Xiaoling, "a reliable and an efficient web testing system*," International Journal of Software Engineering & Applications (IJSEA),* Vol. 10, No. 1, pp. 727, 2019. ISSN: 0975-9018

[7] K, Cem , "Exploratory Testing," Florida Institute of Technology, Quality Assurance *Institute Worldwide Annual Software Testing Conference,* Orlando, FL, November 2006. NSF Grant EIA-0113539 ITR/SY+PE, Stanford, California 94305, USA.

[8] A. Kolawa and D. Huizinga, "Automated Defect Prevention: Best Practices in Software Management," *Wiley-IEEE Computer Society Press,* pp, 426, 2007. ISBN: 978-0-470-04212-0.

[9] C. Kaner, J. Falk, and V. Nguyen, "Testing Computer Software, 2nd Ed.," *John Wiley and Sons, Inc*. pp. 480 pages, 1999. ISBN 0-471-35846-0.

[10] P. Myers and J. Glenford, "The Art of Software Testing," *John Wiley and Sons,* 1979, ISBN 0-471-04328-1.

[11] Company, People's Computer, "Dr. Dobb's journal of software tools for the professional programmer". *Dr. Dobb's journal of software tools for the professional programmer (M&T Pub),* 12 (1-6): 116, 1987.

[12] D. Gelperin, and B. Hetzel (1988). "The Growth of Software Testing". *CACM 31 (6)*. ISSN: 0001-0782.

[13] J. Albahari and B. Albahari. "C# 4.0 in a Nutshell, *Fourth Edition," O'Reilly Media, Inc.,* 1005 Gravenstein Highway North, Sebastopol, CA 95472, pp, 1-15, 2010.ISBN: 978-1-449-322010-2

[14] J. Z. Gao, and T. Jacob, "testing and quality assurance for component-based software," *rtech house, inc. 685 canton street norwood,* u.s.a. pp, 439-440, 2003. ISBN: 978-1580534802

[15] M. L. Gillenson,, M. J. Racer, S. M. Richardson and X. Zhay, engaging testers early and throughout the software development process. *Journal of Information Technology Management, Vol,* 22, No, 1, pp, 1042-1319, 2011.

[16] Y. Jia and M. Harman. "An Analysis and Survey of the Development of Mutation Testing". *Journal of IEEE Transactions on Software Engineering,* Vol, 35, No, pp, 121-137, 2010.

[17] E. R. Vivoni, and R. Camilli, Real-time streaming of environmental field data. *Journal of Computers and Geosciences*, Vol, 29, No, 4, pp, 457-468, 2010.