

Total Health Care System

Tinku Aggarwal^{#1}, Mrs Mamta Yadav^{#2}

M.Tech Scholar CSE, Assitant Professor CSE

YCET, Narnaul (India)

Abstract— A TOTAL HEALTH Care System is an online facility by which user/doctor can view the diagnose information of a person related to his/her health for better analyzed medicine and medical treatments. This system helps in disease management and preventive services. It is an online service by which user can see their details and health measures by viewing their reports online. Lab tests Results, etc. each user can have his own rule and his own conditions and can decide if he wants to be alerted for the patient meeting the rule. These features helps user regarding their bright healthy future and precautions to be taken in future by viewing their report at any stage and at any time period. So this paper is concentrated around all the issues related to the is to achieve total participation of the user with their concerned doctor by means of suitable communication method and to automate all these features which provides excellent and easy to access means of communication medium between doctors and its patients.

Keywords — Introduction, Project Evaluation Techniques, Software Testing, Testing goals, Testing principles, Testing Limitations.

I. INTRODUCTION-

A TOTAL HEALTH CARE SYSTEM is an online facility by which user/doctor can view the diagnose information of a person related to his/her health for better analyzed medicine and medical treatments.

This system helps in disease management and preventive services. It provides very vital information like patients without an exam for certain time, patient with certain levels BP, patients who are taking a certain class of medication, patients who are suffering from a certain type of disease, screening and immunization information, Lab tests Results, etc.

Gause and Weinberg suggest that the analyst start by asking CONTEXT-FREE QUESTIONS. That is, a set of questions that will lead to a basic term and

understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of the first encounter itself.

The goal of the requirements gathering activity is to collect all relevant information from the customer regarding the product to be developed with a view to clearly understanding the customer requirements and weeding out the incompleteness and inconsistencies in these requirements. The requirements analysis activity is begun by collecting all relevant data

regarding the product to be developed from the users of the product and from the customer through interviews and discussions. The purpose of a preliminary investigation is to secure the innocent against hasty, malicious and oppressive prosecution, and to protect him from an open and public accusation of a crime, from the trouble, expense and anxiety of a public trial, and also to protect the state from useless and expensive trials. A preliminary investigation serves not only the purposes of the State. More important, it is a part of the guarantees of freedom and fair play which are birthrights of all who live in our country. It is therefore, imperative upon the fiscal or the judge as the case may be, to relieve the accused from the pain of going through a trial once it is ascertained that the evidence is insufficient to sustain a prima facie case or that no probable cause exists to form a sufficient belief as to the guilt of the accused.

Feasibility study is conducted to select the best system that meets the performance requirements. This entails an identification, description, and evaluation of the candidate system, and the selection of the best system for the job.

Many feasibility studies are disillusioning for both user and analyst. First the study often pre-supposes that when feasibility of the documents is being prepared, the analysis is in position to evaluate solutions. Second most studies tend to overlook the confusion inherent in the system development. Economic analysis is the most frequently used method for evaluating the candidate system. More commonly known as cost of Benefit Analysis, the procedure is to determine the benefits and savings that are expected from the candidate system and compare them with the costs. If benefit outweighs the cost then the decision is made to design, planned implementation otherwise further justification alterations are made in the proposed system. This project doesn't have many hardware requirements, thus, it requires less costing to install the software on the whole

II. PROJECT EVALUATION TECHNIQUES

PERT is a method to analyze the involved tasks in completing a given project, especially the time needed to complete each task, and identifying the minimum time needed to complete the total project. PERT makes use of tasks. Like milestone charts, it shows achievements. These achievements however are not task achievements. They are terminal achievements, called Events. Each activity/Task of the project is represented by a directional arrow

(more commonly known as arrow) pointing in the direction of progress in this project. The circles represent the beginning or completion of a task.

Advantages

- 1) PERT chart explicitly defines and makes visible dependencies (precedence relationships) between the WBS elements
- 2) PERT facilitates identification of the critical path and makes this visible
- 3) PERT facilitates identification of early start, late start, and slack for each activity,
- 4) PERT facilitates identification of early start, late start, and slack for each activity,
- 5) PERT provides for potentially reduced project duration due to better understanding of dependencies leading to improved overlapping of activities and tasks where feasible.

Limitations

- 1) There can be potentially hundreds or thousands of activities and individual dependency relationships
- 2) The network charts tend to be large and unwieldy requiring several pages to print and requiring special size paper
- 3) The lack of a timeframe on most PERT/CPM charts makes it harder to show status although colours can help.
- 4) When the PERT/CPM charts become unwieldy, they are no longer used to manage the project.

III. SOFTWARE TESTING

Testing presents an interesting anomaly for the software engineer. Now comes testing the engineer creates a series of test cases that are intended to ‘demolish’ the software that has been built. In fact, testing is the one step in the software process that could be viewed as destructive rather than constructive. Software Testing is the design and implementation of a special kind of software system: one that exercises another software system with the intent of finding bugs. Software testing is the process of executing a software system to determine whether it matches its specification and executes in its intended environment. Software testing is a process of verifying and validating that a software application or program meets the business and technical requirements that guided its design and development and works as expected and also identifies important errors or flaws categorized as per the severity level in the application that must be fixed. Software testing is also used to test the software for other software quality factors like reliability, usability, integrity, security, capability, efficiency, portability, maintainability, compatibility etc. Testing approach differs for different software’s, level of testing and purpose of testing. Software testing should be performed efficiently and

effectively, within the budgetary and scheduling limits. Due to large number of testing limitations like Exhaustive (total) testing is impossible, compromise between thoroughness, time and budget, it is impossible to be sure that we have removed each and every bug in the program. Following established principles can make testing easier and more effective, and can also ensure that testing goals are achieved to its maximum despite having certain limitations. They also ensure that a process is repeatable. Software testing is a very important quality filter and needs to be planned taking into account its goals, principles and limitations.

IV. TESTING GOALS

A principle is an accepted rule or method for application in action that has to be, or can be desirably followed. Testing Principles offer general guidelines common for all testing which assists us in performing testing effectively and efficiently. Principles for software testing are:

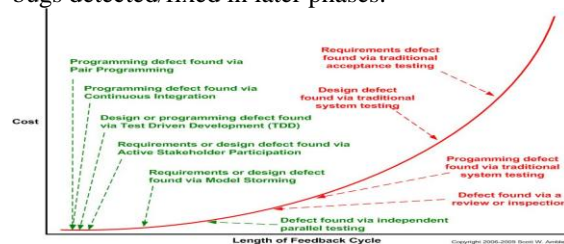
4.1 Test a program to try to make it fail

Testing is the process of executing a program with the intent of finding errors. Our objective should be to demonstrate that a program has errors, and then only true value of testing can be accomplished. We should expose failures (as many as possible)

4.2 Start testing early

If you want to find errors, start as early as possible. This helps in fixing enormous errors in early stages of development, reduces the rework of finding the errors in the initial stages.

Fixing errors at early phases cost less as compared to later phases. For example, if a problem in the requirements is found after releasing the product, then it would cost 10–100 times more to correct than if it had already been found by the requirements review. Figure 1 depicts the increase in cost of fixing bugs detected/fixes in later phases.



4.3 Testing is context dependant

Testing is done differently in different contexts. Testing should be appropriate and different for different points of time. For example, a safety-critical software is tested differently from an e-commerce site. Even a system developed using the waterfall approach is tested significantly differently than those systems developed using agile development approach. Even the objectives of testing differ at different point in software development cycle. For example, the objective of unit and integration testing is to ensure that code

implemented the design properly. In system testing the objective is to ensure the system does what customer wants it to do. Type of testing approach that will be used depends on a number of factors, including the type of system, regulatory standards, user requirements, level and type of risk, test objective, documentation available, knowledge of the testers, time and budget, development life cycle.

4.4 Define Test Plan

Test Plan usually describes test scope, test objectives, test strategy, test environment, risks and mitigation, schedule, levels of testing to be applied, methods, techniques and tools to be used. Test plan should efficiently meet the needs of an organization and clients as well. The testing is conducted in view of a specific purpose (test objective) which should be stated in measurable terms, for example test effectiveness, coverage criteria. Although the prime objective of testing is to find errors, a good testing strategy also assesses other quality characteristics such as portability, maintainability and usability.

4.5 Test for valid as well as invalid conditions

In addition to valid inputs, we should also test system for invalid and unexpected inputs/conditions. Many errors are discovered when a program under test is used in some new and unexpected way and invalid input conditions seem to have higher error detection yield than do test cases for valid input conditions. Choose test inputs that possibly will uncover maximum faults by triggering failures to make testing process more effective. not work under much more broader application.

4.6 Error Absence Myth

System that does not fulfill user requirements will not be usable even if it does not have any errors. Finding and fixing defects does not help if the system built does not fulfill the users' needs and expectations. In addition to positive software testing (which verify that system does what it should do), we should also perform negative software testing (which verify that system does not do what it should not do).

V. TESTING LIMITATIONS

Limitation is a principle that limits the extent of something. Testing also has some limitations that should be taken into account to set realistic expectations about its benefits. In spite of being most dominant verification technique, software testing too has following limitations:

1. Testing can be used to show the presence of errors, but never to show their absence. It can only identify the known issues or errors. It gives no idea about defects still uncovered. Testing cannot guarantee that the system under test is error free.
2. Testing provides no help when we have to make a decision to either "release the product with errors for meeting the deadline" or to "release the product late

compromising the deadline".

3. Testing cannot establish that a product functions under all conditions but can only establish, that it does not function properly under specific conditions
4. Software testing does not help in finding root causes which resulted in injection of defects in the first place. Locating root causes of failures can help us in preventing injection of such faults in future.

VI. CONCLUSION

This letter presents the concentrated views around all the issues related to the is to achieve total participation of the user with their concerned doctor by means of suitable communication method with the help of testing i.e. Software testing is a vital element in the SDLC and can furnish excellent results if done properly and effectively. Unfortunately, Software testing is often less formal and rigorous than it should, and a main reason for that is because we have struggled to define best practices, methodologies, principles, standards for optimal software testing. To perform testing effectively and efficiently, everyone involved with testing should be familiar with basic software principles, limitations and concepts. Already lot of work has been done in this field, and even continues today. Implementing principles in real world software development, to accomplish health goals to maximum extent keeping in consideration the research and knowledge limitations will validate the research and also will pave a way for future research.

REFERENCES

- [1] Antonia Bertolina, Software Testing Research and Practice, Proceedings of the abstract state machines 10th international conference on Advances in theory practice, 121, 2003
- [2] Dolores R. Wallace, Laura M. Ippolito, Barbara B. Cuthill, Reference Information for the Software Verification & Validation Process, DIANE Publishing, 1996.
- [3] Edward L. Jones —Grading student programs – a software testing, Proceedings of the fourteenth annual Consortium for Computing Sciences in Colleges, 2000
- [4] ASP .Net By Peter Norton
- [5] Mastering ASP.Net TECHMEDIA
- [6] Learn ASP.Net in 21 days TMH
- [7] Black book ASP.Net.
- [8] Nick Jenkins. — A Software Testing Primer || , 2008.
- [9] www.apollohospital.com
- [10] www.ministryofhealth.com
- [11] Salim Istaq et al., —Debugging, Advanced Debugging and Runtime Analysis—, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 246-249.
- [12] Shari Lawrence Pfleeger, —Software Engineering, Theory and Practice, Pearson Education, 2001.
- [13] Sheikh Umar Farooq and S.M.K. Quadri, Effectiveness of Software Testing Techniques on a Measurement Scale, Oriental Journal of Computer Science & Technology, Vol.3(1), 109-113 (2010).