



Survey on Acceptance Testing Technique

Ruchi Goel, Dr. Namita Gupta

CSE, Maharaja, Agrasen Institute of Technology, Sector-22, Rohini, New Delhi, INDIA

Abstract: Software testing is done to detect and correct software failures. Main purpose of testing is to check that software meets all specified requirements that guided its design and development. User Acceptance testing (UAT) is the key feature of software implementation. UAT is performed to ensure that the new system meets all the essential user requirements. It is the final testing activity performed by the customer to test for the completeness, correctness and consistency of the software.

Keywords: UAT, SDLC, STLC, OAT

I. INTRODUCTION

Testing measures the software quality. It is a process of identifying bugs in application and produce good quality software. Software testing is done with the objective of finding faults and defects and to prove that software does what is expected. Testing is a trade-off between budget, time and quality. Exhaustive testing is not always possible. Total time spend for software testing depends on the software delivery date and the budget allotted for testing. Incomplete software testing may lead to software failure after its installation resulting loss of money, loss of time and damage to business reputation and customer dissatisfaction. Testing activity should start in the early phases of the software development process to reduce the time spent to identify and correct the defects and hence to reduce the testing effort. Testing end-product doesn't indicate that product functions properly under all possible conditions but can only shows that it does not function improperly under specified conditions. Hence, the purpose of software testing is to examine and execute the developed software in real environment to detect the faults as early as possible during the software development process to deliver the product satisfying all customer requirements within the expected delivery schedule time and budget.

Verification and Validation play an important role in testing. Software testing includes both verification and validation activities. *Verification* means "Are we building the product right" [1]. It is a static activity and includes static methods like reviews, inspection, walkthroughs etc., to ensure that the software does what it is expected to do.

Validation means "Are we building the right product" [2]. It is a dynamic activity and includes execution of code to ensure that the software meets all users requirements. Validation includes functional, structural and Gray box testing techniques. In the next section, different testing activities involved during different phases of the software development lifecycle are discussed.

II. SDLC AND STLC

Testing is an important phase in SDLC (software Development Lifecycle). The process of testing the software in a well-planned and systematic way is known as software testing life cycle (STLC). STLC is most important part of SDLC life cycle. Final product cannot be released without running it through STLC process. SDLC process is prepared and done by the development team to work on the project in different phases which includes STLC. STLC process is done by the testing team to test the developed project. In classic waterfall model, testing starts very late as the last phase of SDLC. This increases the overall efforts and the testing cost. To reduce the testing cost, SDLC follows V-model. In V-model, testing starts from the first phase of the SDLC. Most of the errors or bugs in software are due to incomplete and inaccurate user requirements. It is assumed that requirements errors contribute about 25 to 70 percent of the total software errors [4]. Early defect detection and removal during initial phases of SDLC improves software quality.

Different software testing levels in V-model are component testing, Integration testing, System testing, Acceptance testing and Regression testing. Component testing and Integration testing are based on verification technique; System testing, Acceptance and Regression testing are based on validation technique.

III. LEVELS OF TESTING (V-MODEL)

V-model is known as verification and validation model. In v-shaped model sequential path is followed. Each path must be completed before the next begins. In v model the main advantage is time saving as testing activities like planning, test designing occur well before coding. V-model is best to use when requirements are clearly defined. On the left side software development activities are shown and on right side testing phased are shown

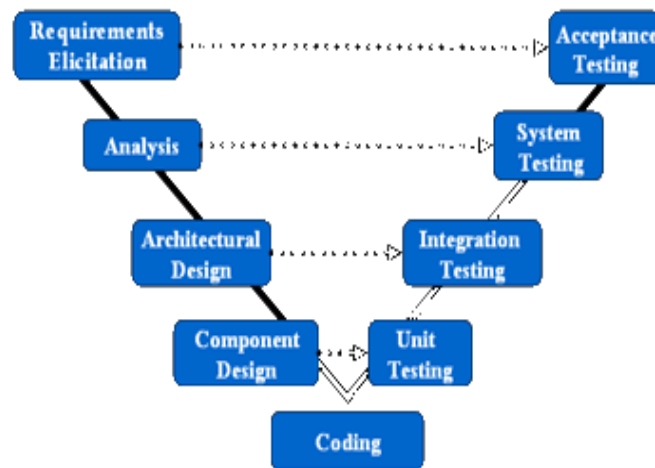


Fig: 1-V model of Testing

SDLC broadly has five phases:

A. Requirement gathering and analysis phase

Software development team gather the requirements from the client and document it as SRS document. During requirement gathering and analysis phase, static testing is done to check the completeness and consistency of the user requirements. SRS is treated as the legal document between developer and user, listing all the user requirements to be included in the final product. It includes both functional and non-functional requirements. Test cases are designed during this phase to perform system testing.

B. Design Phase

During this phase, modules corresponding to requirements listed in SRS are identified and interface between them is documented in SDD document. Components or modules which are constructed during design phase are checked to ensure that data exchanged between the modules is correct, complete and consistent. Integration testing is related to analysing the interface between the components in terms of data format, content, size, frequency, and their functionality.

Testing during design phase is important to check interfaces between GUI based application components. For example, university has different components like registration management, fee management, library management which are related to each other and share common data about students. These individual modules although successfully tested may show errors due to improper interface among them. Test cases related to integration testing are designed during this phase.

C. Implementation phase

Requirements are implemented as code by the programmers referring to SDD build during design phase. Dynamic testing is performed to test the code. Test cases build in the first two phases of SDLC are tested on executing the code to check the result and comparison of the expected output with actual output helps to identify the presence/absence of errors in the given code. Structural testing techniques are used to test each unit (loops, control statements, module) and their interfaces.

D. Acceptance, Installation and deployment Phase

Before the software is deployed at the client site, the end-product is tested to ensure that it is satisfying all the user requirements mentioned in SRS. This is known as Acceptance testing. Acceptance testing is performed when product is ready before delivery to customer. Acceptance test is a “contract” between the developers and the customer.

E. Unit Testing

Unit is small part of a program. It is a white box testing technique which is performed by developer. It is first level of testing where individual unit/component are tested. Main benefit is to check that each module is working correctly.

F. Integration testing

It is performed before system testing but after unit testing. It is done by developers or independent testers to check whether the units are interacting or not means to find fault between interacting units.

G. System Testing

It is black box testing technique to test the application on complete software product before launch to the market. In system testing both functional and nonfunctional requirements are verified. The main aim is to evaluate Business / Functional / End-user requirements.

H. Acceptance Testing

Acceptance criteria are the criteria that a system or component must satisfy in order to be accepted by the user, customer, or other authorized entity [IEEE,(1990)]. Before making the system for actual use Acceptance testing is done after completion of system testing..The main aim of acceptance testing is to gain confidence of the user that product is defect free and meet all the specified user requirements. Acceptance testing is necessary as the requirements which are changed during the project may or may not be communicated effectively to the developers.

There are many types of acceptance testing. During acceptance testing it is checked that product will be accepted by customers or not, means product will specify all user requirements or not Key component of this research is how types of acceptance testing different from each other.

In Acceptance Testing the major conclusions are taken as:

1. Accept the system as it is delivered
2. Accept the system after assuming the requested modifications has been made
3. Do not accept the system

IV. ACCEPTANCE TESTING TYPES

Acceptance testing is a type of software testing where system is tested for acceptability whether it will be accepted by user or not. Usually black box testing technique is used in this. Study showed that typical requirements specifications are 15% complete and 7% correct, and that it was not cost effective to complete or correct them [2]. Functional testing is black box type of testing which is done against business requirement of application. User can read functional specification document easily, no specific technical knowledge is required to understand the document. Functional requirements describe what and not how The non Functional Testing is done against non functional requirements.Nonfunctional requirement fall in areas as compliance, capacity.. Non-Functional Requirement is restriction that must be considered when designing the solution.e.g if we want to do shopping then we has to create a log in account. Non Functional testing includes Reliability testing, load testing, portability testing.

Broadly, acceptance testing is of following types:

A. Alpha testing

Alpha testing is a type of acceptance testing performed by the developers under controlled environment at software development site. Alpha testing is a kind of both black box as well as white box testing technique. Main aim of alpha testing is to improve the quality of product. Missing docs, crashes and plenty of bugs are expected in alpha testing

B. Beta testing

Beta versions are made available to the open public to increase the feedback and if any customer finds any bugs or faults, they report it back to the developers, who then recreate the problem and fix it before the release. This process helps identify and remove defects. Beta testing is also known as pre-release testing. Beta testing is a kind of black box testing technique. Fewer crashes and some bugs are expected in beta testing

C. User Acceptance testing (UAT)

UAT is formal means by which company confirms that now new system will actually meets essential user requirements. It is final testing performed after functional, system and regression testing completed. The main aim of UAT is to check the software against business requirements. This is done by end-users who are familiar with business requirements. UAT is carried out in separate testing environment with production like data setup. It is a kind of black box testing where two or more end users are involved. In UAT client know that what we want exactly rather than what we are expecting. User acceptance testing (UAT) is done by the users and application managers. Before the software goes live, this is the last option for the customer to test the software

and see whether it is fit for the purpose or not. The key deliverables of UAT are Test plan, UAT test cases, and UAT signoff, Testresults and error reports

D. Contract and Regulation Acceptance Testing

This type of Acceptance Testing aims to ensure that the acceptance criteria as specified in the contract document have indeed been met by the developed software. Regulation Acceptance Testing is performed when there exists specific regulations that must be adhered to, for example, there may be safety regulations, or legal regulations.

E. Operational Acceptance Testing

This form of acceptance testing (OAT) is commonly performed by a System Administrator and would normally be concerned with ensuring that functionality such as; backup/restore, maintenance, and security functionality is present and behaves as expected. Main aim of OAT is to get final commitment of handover of application to their owner and operation team

TABLE 1
Acceptance testing types

Type of acceptance testing	Tester category	Testing Environment	Nature of developed software	Source of Requirements	Nature of Requirements
Alpha	Developer	at developer site under controlled Environment	Specific to target client	legal document (Software Requirement Specification document)	Mainly functional requirements
Beta	Experienced user	in real working Environment	Customised	No legal document	Functional and non-functional requirements
UAT	Developer and target client	Under controlled Environment	Specific Client	Business specifications	Mainly functional requirements
Contract	Developer as per contract document	Under controlled Environment	Specific Client	legal document	Functional and non-functional requirements
Operational	System administrator	Under Controlled or real Environment	Any	No legal document	Mainly non-functional requirements

V. CONCLUSION

In this paper various acceptance testing techniques are considered and compared. Mostly people think that acceptance testing is done by user at user site, but in acceptance testing there are many testing categories and many testing environment depending on requirement of user.

REFERENCES

- [1] Trivedi Shivkumar Hasmukhrai, "Software testing Techniques" international Journal of Advanced Research in Computer Science and Software Engineering ,, Volume 2, Issue 10, October 2012 ISSN: 2277 128X.
- [2] Smith High , Adaptive J. Software Development, Information Architects, Inc. (2000), presentation at OOPSLA 2000 quoting a study by Elemer Magaziner.
- [3] Miller Roy W., Collins Christopher T.," Integrating User Acceptance Tests, Generation to Requirements", in IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore, Management
- [4] Jones, C., Software Challenges, Computer, Vol. 28, No. 10, October 1995.
- [5] IEEE Std. 610.12 – 1990. IEEE Standard Glossary of Software Engineering Terminology.
- [6] <http://www.softwarequalitysource.com/acceptance-testing>
- [7] Roger Pressman, titled Software Engineering - a practitioner's approach