



The Future Prospects of Software Testing

Kalpesh Parmar

Persistent Systems Limited,
Nagpur (MS), India

Bhaskar Y. Kathane

VMV College, Wardhaman Nagar,
Nagpur (MS), India

Abstract – *Information Technology (IT) has revolutionized the service industries in the last few years. There has been a substantial growth of Internet-based services both in pure Internet business and traditional companies. Software testing is a fast-changing industry that is experiencing global influence and trends as it starts to grow up from being a toddler to a teenager in terms of its maturity. There are many exciting developments on the horizon in terms of techniques, methodologies, economics, and academia, but how can any of us predict what they will be? This white paper details the user's postulation on how the software testing industry might develop in the next five years. Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Technology has been one of the most important factors for the development of mankind. Information and communication technology is the major advent in the field of technology which is used for access, process, storage and dissemination of information electronically and software testing is one of them.*

Keyword – *Software Testing, Verification, Validation, Regression, Black-box, White-box etc*

I. INTRODUCTION

Software Testing is the process of executing a program or system with the intent of finding errors. It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

The testing phase of software development sometimes gets the short shrift from developers and IT managers. Yet testing is the only way to determine whether an application will function properly at deployment. Without an effective testing strategy, companies sometimes blindly take on significant risks that go well beyond simply having poor functioning software. Organizational ramifications can be fierce, including the risk of:

- Customer/end user alienation, or brand injury,
- Competitive threat, and even
- Product revenue loss.

A further complication has to do with the dynamic nature of programs. If a failure occurs during preliminary testing and the code is changed, the software may now work for a test case that it didn't work for previously. But its behavior on pre-error test cases that it passed before can no longer be guaranteed. To account for this possibility, testing should be restarted. The expense of doing this is often prohibitive.

II. SOFTWARE TESTING

Software testing is the process of finding bugs or errors in the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development. So that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed can be implemented at any time in the development process.

Software Testing includes the following:

- Find out the bugs / errors in the software.
- Ensures the productivity and quality of the product.
- Validate and verify the functionality and usability of the software.
- Examination of code as well as execution of that code in various environments and conditions.
- Examining the aspects of code.

Testing can never completely identify all the defects within software. Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment. A study conducted by NIST in 2002 reports that software bugs cost the U.S. economy \$59.5 billion annually. More than a third of this cost could be avoided if better software testing was performed.

Testing Vs Debugging: The purpose of testing is to show that the program has bugs whereas debugging is finding out the bugs that led to the program's failure and correcting them.

A. To improve quality.

As computers and software are used in critical applications, the outcome of a bug can be severe. Bugs can cause huge losses. Bugs in critical systems have caused airplane crashes, allowed space shuttle missions to go awry, halted trading on the stock market, and worse. Bugs can kill. Bugs can cause disasters. The so-called year 2000 (Y2K) bug has given birth to a cottage industry of consultants and programming tools dedicated to making sure the modern world doesn't come to a screeching halt on the first day of the next century. In a computerized embedded world, the quality and reliability of software is a matter of life and death.

Quality means the conformance to the specified design requirement. Being correct, the minimum requirement of quality, means performing as required under specified circumstances. Debugging, a narrow view of software testing, is performed heavily to find out design defects by the programmer. The imperfection of human nature makes it almost impossible to make a moderately complex program correct the first time. Finding the problems and get them fixed, is the purpose of debugging in programming phase.

B. Software Testing Verification & Validation (V&V)

This topic indicated another important purpose of testing is verification and validation (V&V). Testing can serve as metrics. It is heavily used as a tool in the V&V process. Testers can make claims based on interpretations of the testing results, which either the product works under certain situations, or it does not work. We can also compare the quality among different products under the same specification, based on results from the same test. We cannot test quality directly, but we can test related factors to make quality visible. Quality has three sets of factors functionality, engineering, and adaptability. These three sets of factors can be thought of as dimensions in the software quality space. Each dimension may be broken down into its component factors and considerations at successively lower levels of detail. Table-1 illustrates some of the most frequently cited quality considerations.

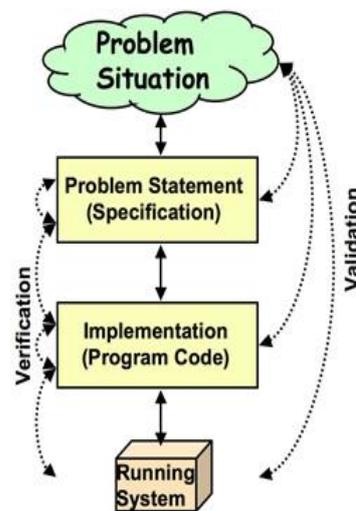


Fig-1 Software Testing Verification & Validation

TABLE-I : Quality considerations

Functionality (exterior quality)	Engineering (interior quality)	Adaptability (future quality)
Correctness	Efficiency	Flexibility
Reliability	Testability	Reusability
Usability	Documentation	Maintainability
Integrity	Structure	

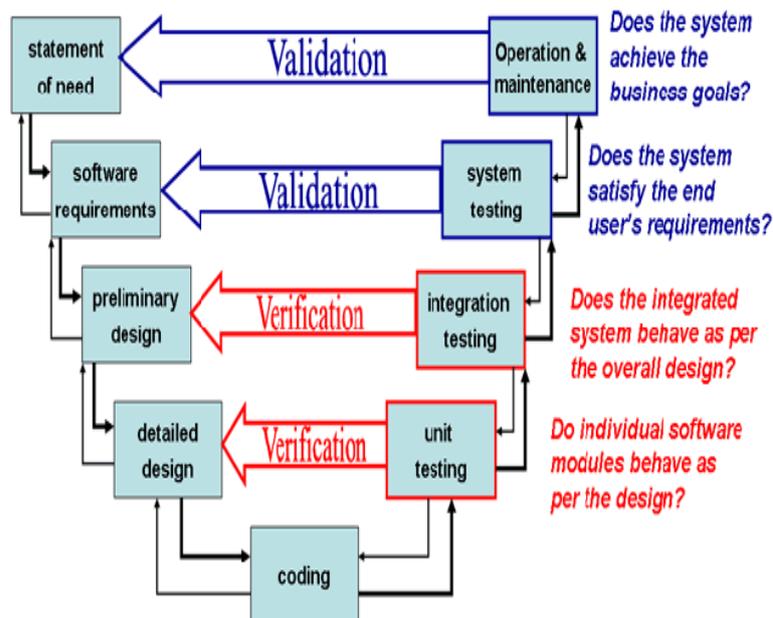


Fig-2: Dynamic Testing

C. Agile software Testing:

Agile Testing is a software testing practice that follows the principles of agile software development. Agile development integrates testing into the development process, versus having it as a separate phase. Testing therefore is an integral part of the core software development and actively participates throughout the software coding process. Agile Testing involves a cross-functional agile team actively relying on the special expertise contributed by Testers. This allows the combined team to better meet the project's defined business, software usability, quality, and timeline objectives. Agile teams use a "whole-team" approach to "bake in quality" to the software product. This approach allows the team to work at a sustainable pace because testing occurs in real time, allowing testers to collaborate actively with the development team and giving them an ability to identify any issues and transfer those into executable specifications that guide coding. Testing and coding are done incrementally and iteratively, building up each feature until it provides enough value to release to production.

D. Automated Testing-Regression:

Automation is a critical component of agile testing. It would be impossible to keep pace with the Agile development schedule otherwise. Automation is used to run regression testing. The combined team (Developers, Product Owners and Testers) usually predetermine, at the start of the project, which parts of the software will be tested using automation.

Continuous integration/builds, unit, functional and integration test execution as well as continuous or automated deployment are common areas where automation may work better than traditional tests.

The entire project team agrees upfront on which of the main flows will be automated. They also determine at this point how to prioritize defects identified by automation, and how to fix that during sprints.

Automated tests consist of unit tests, capable of verifying even the most segment of software. Furthermore, it does so rapidly. This makes it possible to execute the test set multiple times per day, per hour or even more frequently if needed.

The benefits of automation include:

- Allows re-use of tests.
- Enables faster execution for the most important test cases.
- Facilitates greater test coverage.
- Delivers higher test accuracy and identifies defects sooner.
- Facilitates regression testing.

III. IMPORTANT POINTS OF SOFTWARE TESTING

Software Testing is most important factors in software development. We consider the following are the important points of software testing:

- A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. Testing is non-trivial.
- Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.

- Not all software defects are caused by coding errors.
- A single defect may result in a wide range of failure symptoms.
- Bugs may arise due to incompatibility of hardware or software (OS, Browsers)
- Testing under all combinations of inputs and preconditions (initial state) is not feasible.
- Testing can be **Static** (examining the code and structure) and **Dynamic** (actually executing the program with test cases). Static testing, sometimes, is omitted.
- Software Testing is associated with **Verification** and **Validation**. Verification is process based and checks whether the software is correctly designed / functioning well or not?. Validation is product based and checks whether it is useful for customer or not?
- People who do testing are called **Software Testers**. The different roles in testing phase include: manager, test lead, test designer, tester, automation developer, and test administrator.
- Software can be viewed as an important part of **Software Quality Assurance (SQA)** process
- Software Testing methods are traditionally divided into **Black Box Testing** and **White Box Testing**. Black box testing treats the software as a 'black box', without any knowledge of internal implementation. Specification based testing, part of black box testing, tests the functionality of the software without considering the code. White box testing, in contrast to Black box testing, has access to the code, internal data structures and algorithms
- Testing can be done in different levels:
 1. **Unit Testing:** Tests the minimal software component or module. Each unit or module is checked whether it is functioning well independently or not?
 2. **Integration Testing:** Exposes defects in the interfaces and interaction between integrated components or modules. mal software component or module
 3. **System Testing:** Tests the system as a whole. Checks whether the completely integrated system meets its requirements or not?
 4. **Alpha Testing (Before Product Release):** Tests the software by potential end users or customers or an independent test team at developers site.
 5. **Beta Testing (Before Product Release):** Simulated after Alpha testing. The software is released to groups of people outside the programming team. End products are called Beta-versions.
- Other types of tests include: *Grey Box testing, Regression testing, Acceptance testing, Performance testing, Stability testing, System integration testing*, and many more.
- **Testing Tools:** Software that tests Software. Many testing tools are available in market. But, there is no universal tool that can test any software.

Testing Strategies offered:

- Risk-Based Testing
- Acceptance Testing
- Performance, Stress and Load Testing
- Installation and configuration
- Testing Security Testing
- API Testing and Unit Test
- Navigability testing
- Installation Qualification Testing
- DB testing Web services
- Web Compatibility
- Mutation test
- Integration Test
- Accessibility Testing

IV. CONCLUSIONS

- Software testing is a critical element in the software development life cycle and has the potential to save time and money by identifying problems early and to improve customer satisfaction by delivering a more defect-free product. Unfortunately, it is often less formal and rigorous than it should, and a primary reason for that is because the project staff is unfamiliar with software testing methodologies, approaches, and tools. To partially remedy this situation, every software professional should be familiar with basic software testing concepts, roles, and terminology.
- Software testing is an art. Most of the testing methods and practices are not very different from 20 years ago. It is nowhere near maturity, although there are many tools and techniques available to use. Good testing also requires a tester's creativity, experience and intuition, together with proper techniques.
- Testing is more than just debugging. Testing is not only used to locate defects and correct them. It is also used in validation, verification process, and reliability measurement.
- Testing is expensive. Automation is a good way to cut down cost and time. Testing efficiency and effectiveness is the criteria for coverage-based testing techniques.
- Complete testing is infeasible. Complexity is the root of the problem. At some point, software testing has to be stopped and product has to be shipped. The stopping time can be decided by the trade-off of time and budget. Or if the reliability estimate of the software product meets requirement.

ACKNOWLEDGEMENTS

We are very much thankful to Dr. Rahul Mohre and Dr. Ravindra Gharpure for their valuable inputs, constant guidance and their extensive support and encouragement in this work.

REFERENCES

- [1]. Software Body of Knowledge – Ironman Phase Documentation – Chapter 5
http://swebok.org/ironman/pdf/Swebok_Ironman_June_23_%202004.pdf
- [2]. Software Debugging, Testing and Verification, B. Hailpern and P. Santhanam
<http://www.research.ibm.com/journal/sj/411/hailpern.pdf>
- [3]. What Is A Good Test Case?, Cem Kaner, Florida Institute of Technology
<http://www.kaner.com/pdfs/GoodTest.pdf>
- [4]. Automated Testing, *Wikipedia [online]*, available from:
http://en.wikipedia.org/wiki/Automated_testing [Accessed 15th November 2009]
- [5]. Testing Tools, *ApTest [online]*, available from:
<http://www.aptest.com/resources.html#reqs> [Accessed 15th November 2009]