

Comprehensive Testing Methodologies in Software Development

June 2, 2000



ALLOFE | SOLUTIONS



<i>Comprehensive Testing Methodologies in Software Development</i> _____	3
<i>Abstract</i> _____	3
<i>Proposed Solution</i> _____	3
Requirements / Standards Testing _____	3
Unit Testing _____	3
System Testing _____	4
Exploratory Testing _____	4
Functional Testing _____	4
Security Testing _____	4
Usability Testing _____	4
Integration Testing _____	4
End-to-End Testing _____	4
Performance-Load Testing _____	4
Additional Testing _____	5
Acceptance Testing _____	5
New Functionality Testing _____	5
Regression Testing _____	5
<i>About AllofE</i> _____	5

Comprehensive Testing Methodologies in Software Development

Abstract

Staying ahead of the software development game requires implementing the latest and greatest of technological advancements. A lot of newly released technology not only comes with complex features and abilities, but also comes with its own set of “bugs” and incompatibilities. Software developers must become adept at testing new technology along with the old technology and existing applications. The software must be tested at various stages throughout the development process as well as by different types of user groups. An organization, therefore, needs a comprehensive testing methodology to ensure that a system is free from defects in development and meets all design and application requirements.

Proposed Solution

A comprehensive testing methodology involves using both accurate and inaccurate data to demonstrate that a product satisfies its requirements and, if the requirements are not met, to fix the specific differences between expected and actual results. Below are examples of strategies used during the system testing process.

Requirements / Standards Testing

Some of the objectives of requirements-driven testing are to ascertain:

1. Computational correctness.
 - Sales Tax
 - Price Calculation
2. Proper handling of boundary conditions, including extreme inputs and conditions that cause extreme output:
 - Text box length and database's table field should match.
 - Minimum/maximum character length for user name and password should be valid.
3. Proper behavior under stress or high load:
 - Large number of users.
 - Large amounts of data.
4. Consistency in interface design and layout:
 - When a user clicks to delete an item a pop-up window should appear asking if they really want to delete the item. This pop up window should be constant throughout the web site.
 - Screens with similar functionality should have similar look and feel.
 - Buttons that perform the same action should be named the same.

Unit Testing

Unit Testing is based on knowledge of the internal logic of an application's code. Tests are based on coverage of code statements, branches, paths, and conditions. Unit Testing is used to test particular functions or code modules during the development

phase. This is typically done by the programmer, as it requires detailed knowledge of the internal program design and code. Once the programmer has completed the unit test, the testers may then run detailed tests of major functionality logic.

System Testing

System testing is based on the requirements and functionality defined in the system design. It is usually performed on a module by module basis. While some overlap may occur, each module is taken separately and thoroughly tested for functionality, security, and usability.

Exploratory Testing

Exploratory Testing is often taken to mean a creative, informal software test that is not based on formal test plans or test cases. This can be done by testers that are learning the software or that do not have comprehensive knowledge of the system design.

Functional Testing

Functional Testing focuses on the functional requirements of the application and covers all combined parts of a system. This type of testing should be done by testers after the developer has tested the software.

Security Testing

Security Testing is testing how well the system protects against unauthorized internal or external access, willful damage, etc. This requires more advanced testing techniques and uses predefined test cases and procedures.

Usability Testing

Usability testing tests the user friendliness of the application and focuses on the end user or target group perspective. It also involves reviewing the look and feel of site to ensure consistency between pages, and other GUI standards.

Integration Testing

Integration testing is testing combined parts of an application to determine if they function together correctly. These can include modules within the application as well as individual applications or systems working together. This is especially relevant to client/server and distributed systems.

End-to-End Testing

End-to-End Testing is similar to Functional Testing; the 'macro' end of the test scale; involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

Performance-Load Testing

Performance/Load Testing is testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.

Additional Testing

Additional Testing includes post-implementation testing by the end users, enhancement and modification testing, and re-testing of errors after they have been fixed. This type of testing occurs throughout the development, implementation, and production stages of the system.

Acceptance Testing

Acceptance Testing is the final testing after Implementation and is done by the end user group. This is used to ensure that design requirements have been met and the system is usable by the key stakeholders.

New Functionality Testing

New Functionality Testing is continuous testing of an application as new functionality is added and requires that the various components of an application's functionality be independent enough to work separately before all parts of the program are completed.

Regression Testing

Re-testing after fixes or modifications to the system have been made. This is to ensure that the fix or modification has been done accurately.

About AllofE

AllofE (www.allofe.com) is a technology firm specializing in Enterprise Web products and custom applications across a number of vertical markets and industry segments that include Education, HealthCare, Financial, Transportation, Corporate, Construction & Engineering, Sports & Athletics, Printing & Publishing and Telecommunications.

AllofE's broad industry expertise extends to functional areas such as:

- Content Management
- Asset Tracking and Management
- Auction Management
- Scheduling and Resource Management
- Online Training and Certification Systems
- e-Procurement and e-Commerce
- Customer Relationship Management
- Job Costing and Tracking
- Project Management

In the academic sector, AllofE specializes in the areas of Web Content Management, Curriculum Management, Online Assessments/Testing, HR Applicant Tracking, Academic Calendars/Scheduling and Staff Development.

Allofe invests very heavily in R&D to keep pace with the latest trends in web technology and constantly trying to find the most efficient, cost-effective and high quality enterprise-wide solutions. Current research areas include:

- Rich Web Applications including effective UI design and AJAX
 - Data Mining and Analysis
-

- Cross Platform Applications
- Search algorithms and crawling, optimization strategies
- Data Mining and Warehousing
- Security at Functional and Object Level for Web based Applications
- Backend Systems Integration design for interoperability and scalability
- Component based Application Server architecture

For more information about AllofE please visit:

www.allofe.com

www.studio.allofe.net

www.e-curriculum.net
