QA Testing Boot Camp

Chapter 10 – Testing Types

Learning Outcomes

- Understand the types of testing associated with the various phases of the development cycle
- Understand functional testing
- Understand non-functional testing

Testing Types

Are introduced as a means of clearly defining the objective of a certain level of the development cycle

Testing Objectives:

- Functions
  - Non-functional characteristics
  - Architectural characteristics
  - Changes – intentional and unintentional

Four Testing Types

- Functional
- Non-functional
- Structural
- Change Related

Functional Testing

- Concentrates on authenticating the operation of the functions of component or system
- Verifies a specific action or function as specified in the SRS

Functional Testing (2)

- Techniques
  - Requirements Based Testing - Requirements are prioritized depending on the risk criteria. Tests are then tested according to the priority – most important first
  - Business Process Based Testing - Scenarios involved in the day-to-day business use of the system are described and tested

Non-Functional Testing

- Quality based testing
- The user does not directly see quality. They will, however, experience it (i.e. performance, look/feel, reliability, etc.).
- Is more extensive than Functional Testing
- Testing the environment as well as the software

Non-Functional Testing (2)

- Functionality testing
- Reliability testing
- Usability testing
- Efficiency testing
- Maintainability testing
- Portability testing
- Baseline testing
- Compliance testing
- Documentation testing
- Endurance testing
- Load testing
- Performance testing
- Compatibility testing
- Security and Penetration testing
- Scalability testing
- Volume testing
- Stress testing
- Recovery testing
- Internationalization testing
- Localization testing

Non-Functional Testing (3)

- **Functional Testing** - Verify that a software application performs and functions correctly according to design specifications
- **Reliability Testing** - Exercise an application so that failures are discovered and removed before the system is deployed
- **Usability testing** - Basically, the tester tests the ease with which the user interfaces can be used

Non-Functional Testing (3)

- **Efficiency testing** - Inspects the amount of code/testing resources required by a program to perform a particular function
- **Maintainability testing** - Defines how easy it is to maintain the system
- **Portability testing** - Testing the ease with which a computer software component or application can be moved from one environment to another

Non-Functional Testing

- **Baseline Testing** - Validation of documents and specifications on which test cases will be designed
- **Compliance testing** - Focus on the IT standards followed by the company
- **Documentation testing** - Documentation is any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results.
    - If the documentation is not correct, there will be major and costly problems

Non-Functional Testing

- **Endurance (Soak) testing** - Testing a system with a significant load, extended over a significant period of time, to discover how the system behaves under sustained use
- **Load testing** - Determine a system's behavior under both normal and peak conditions
- **Performance testing** - Determine how fast some aspect of a system performs under a particular workload

Non-Functional Testing

- **Compatibility testing** - Demonstrates how well a system performs in a particular environment that includes hardware, network, operating system, other software, etc.
- **Security testing/Penetration testing -** Seeks to check the vulnerability of the application or product software and determine how it will behave in the presence of a malicious attack
  - Confidentiality, integrity, authentication, availability, authorization and non-repudiation
- **Scalability testing -** Assess the ability of a system, a network, or a process to function well, when it is changed in size or volume in order to meet a growing need

Non-Functional Testing

- **Volume testing** - Determine system performance with increasing volumes of data in the database.
- **Stress testing** - Involves taxing the product beyond normal operational capacity, often to a breaking point, in order to observe the results.
  - Used to determine stability of the system

GRAND CIRCUS
DETROIT

Non-Functional Testing

Volume vs. Load vs. Stress Testing

Volume Testing = Large amounts of data

Load Testing = Large amount of users

Stress Testing = Too many users, too much data, too little time, and too little room

Non-Functional Testing

- **Recovery testing** - Performed to check how fast and well the application can recover after it has experienced a crash or hardware failure
- **Internationalization testing and Localization testing**
  - Internationalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes
  - Localization is a process of adapting internationalized software for a specific region or language by adding local specific components and translating text

Structural Testing

- Validates the structure of the system or component
- White Box Testing (chapter 13)
  - Also known as glass box or clear box testing
- Testers are required to have knowledge of the code implementation
- Concentrate on the “how” of the software solutions
- Used at all levels of testing

Change Related Testing

- Testing after a defect repair
- Often called Confirmation Testing
- Test must be executed exactly as the original test to ensure the defect was repaired adequately

Change Related: Regression Testing

- Verifies that modifications in the software or the environment have not caused any unintended adverse side effects and that the system still meets its requirements
- Executed whenever the software changes, either as a result of fixes or new or changed functionality
- Often mandatory when doing a maintenance cycle on the software

Change Related: Maintenance Testing

- Testing when the environment changes
- Two parts
  - Changes that have been made as a result of a correction to the system, a system extension, or the addition of features.
  - Regression testing to prove that the rest of the system has not been affected by the maintenance work.

Summary

A thorough survey was conducted of the various types of testing that can be performed at any level of the software development lifecycle.

Choosing the appropriate test coverage is integral to successful software development

QA Testing Boot Camp

Chapter 10 – Testing Types