Software Quality and Testing

TURKCELL TECHNOLOGY
Software Testing Types

20.10.2011

Quality Characteristics – 1

Categorization of software testing according to ISO 9126 (2001) which provides six perspectives of quality:

- Functionality, which consists of five sub-characteristics: suitability, accuracy, security, interoperability and compliance; this characteristic deals with functional testing.
- Reliability, which is defined further into the sub-characteristics maturity (robustness), fault-tolerance, recoverability and compliance
- Usability, which is divided into the sub-characteristics understandability, learnability, operability, attractiveness and compliance

Quality Characteristics – 2

- Efficiency, which is divided into time behavior (performance), resource utilization and compliance
- Maintainability, which consists of five sub-characteristics: analyzability, changeability, stability, testability and compliance
- Portability, which also consists of five sub-characteristics: adaptability, installability, co-existence, replaceability and compliance.

FUNCTIONAL TESTING TYPES
Objectives and Methods

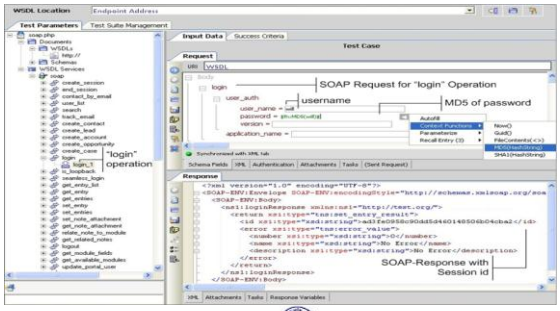
Functional Testing

- Testing the functionality according to the specified requirements
- Earlier functional testing can display the sofware maturity level and enable early bug fixing activities
- Functional testing can be done in any software testing level

Web Services Testing

- Ensuring functional quality for a set of services by eliminating error or failure
- Testing Web Services Description Language (WSDL) interfaces
  - WSDL/SOAP-based messaging
- Verifying compliance with Web Service standards (SOAP, HTTP(S), JSON, JMS, WS-Security, UDDI, etc.)
- Security perspective
  - Message security validation
  - Penetration test

Web Services Testing Example



Migration Testing

- Conduct testing for software version changes
  - Validation of the preserved functionality for newer versions
- Prepare data migration testing approach in order to copy of data from one system to another
  - Misspelled data
  - Duplicate data occurence
  - Null data presence
  - Missing data records
  - Aggregated data etc.

Maintenance Testing

- Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment. (IEEE 1219)
- The software product undergoes modification to code and associated documentation due to a problem or the need for improvement. The objective is to modify the existing software product while preserving its integrity. ISO/IEC [ISO95]

NON – FUNCTIONAL TESTING TYPES
Objectives and Methods

Various Types Non-Functional Testing

- Performance Testing
- Load Testing
- Stress Testing
- Compatibility Testing
- Security Testing
- Usability Testing
- Localization Testing

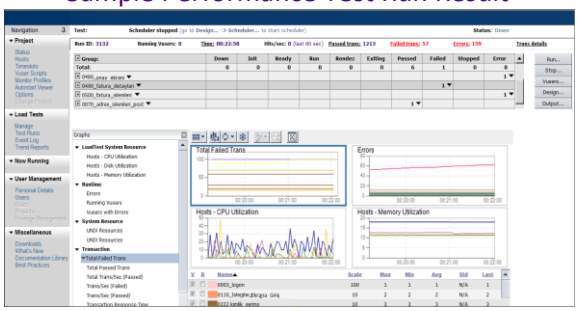
Performance Testing

- Ensure the system achieving the specified performance levels.
- Identify bottlenecks in the given network
- Investigate speed characteristics of system under test
- Determine how fast a given system performs under load
- Reduce overall costs by finding bugs at earlier stages
- Analyze average server measurements
- Demonstrate if network meets the performance criteria

Performance Metrics

- Decide measurement types for the real system
- Classify metrics by identifying related application components
- Identifying priorities on speed measurement
- Resource utilization requirements different platforms
- Identify concurrent usage definitions for each channel
- Latency, jitter, throughput, packet loss

Sample Performance Test Run Result



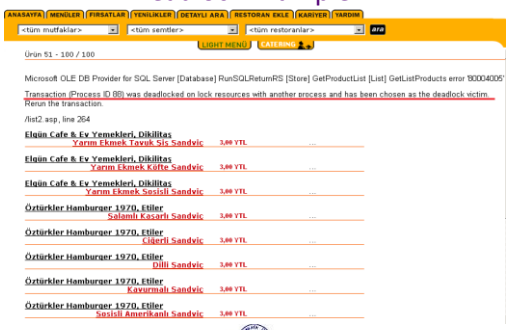
Load Testing

- Simulate actions of real users
- Create mix scenarios for various workloads
- Realistic workload on system interfaces
- Identify system responses for a period of time (Response time, Hits per second, throughput, resource utilization etc.)
- Determine end-to-end capacity
- Expose unexpected behaviour between subsystems
- Validate specification limits

Stress Testing

- Testing the system or entity as a perspective of excessive operational capacity
- Determine acceptable system behaviour under heavy load
- Detecting memory leaks, thread deadlocks, unresponsive software entities, data corruption, process anomalies, runtime errors, etc.
- Develop corrective action models in order to mitigate the risks

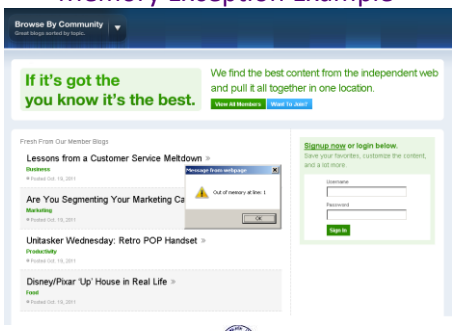
Deadlock Example



TURKCELL TEKNOLOJİ

TURKCELL AKADEMİ

Memory Exception Example



TURKCELL TEKNOLOJİ

TURKCELL AKADEMİ

Compatibility Testing

- Ensure the application's compatibility with the computing environment.
- Testing the application by considering different environments
- Generate a compatibility testing matrix according to technical requirements
- Consider backward/forward compatibility
  - Code written for version 2.3 may not work on version 3.1

TURKCELL TEKNOLOJİ

TURKCELL AKADEMİ

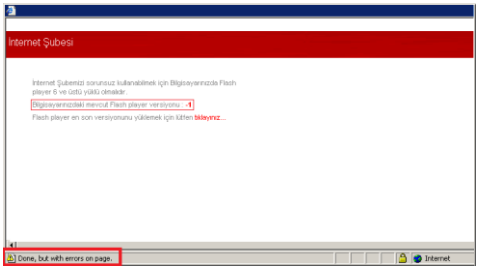
Compatibility Testing Matrix Items

- Operating System
  Microsoft Windows 98/Me/NT/2000/XP/Vista/Windows 7
  Unix/Linux (Pardus, Debian, Ubuntu, Gentoo, FreeBSD, CentOS, OpenSUSE)
  Android, iPad, iPhone
- Browser Compatibility
  Internet Explorer, Firefox, Google Chrome, Opera, Camino, Netscape, Safari, etc.
- Database
  Oracle, MSSQL, MySQL, DB2, Sybase, Postgre SQL, etc.
- Hardware Platform
  CPU, RAM, Hard Disk, Screen Resolution
- Network
  Bandwidth, speed, capacity

TURKCELL TEKNOLOJİ

TURKCELL AKADEMİ

Compatibility Testing Example



Security Testing-1

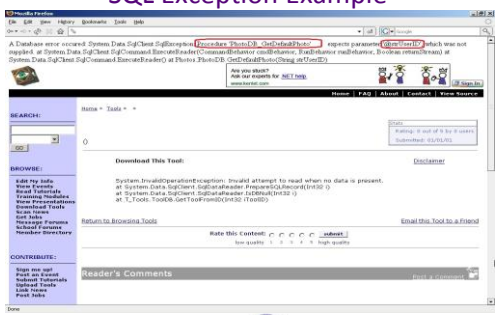
- Adopt routine security procedures to be performed for each release/build
- Probe vulnerabilities of the system by using security tools
- Automatic test genaration to simulate specific network attacks

Security Testing-2

Conduct security assessments:

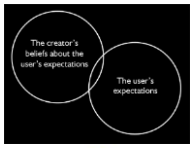
- Data Validation Errors (Client-side validation)
- Cross Site Scripting
- Buffer Overflows
- Improper File Access
- SQL, LDAP, XPATH, OS command injection
- Sensitive data in unsecured pages

SQL Exception Example



Usability Testing

- Evaluating product or service by considering the end-user point of view
- A part of system unit may function correctly however, may not provide ease of use
- Focus on the intended groups and increase overall user satisfaction

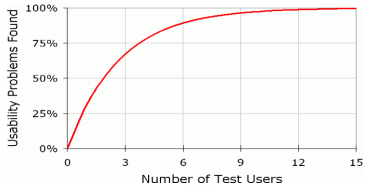


A Mathematical Model of the Finding of Usability Problems

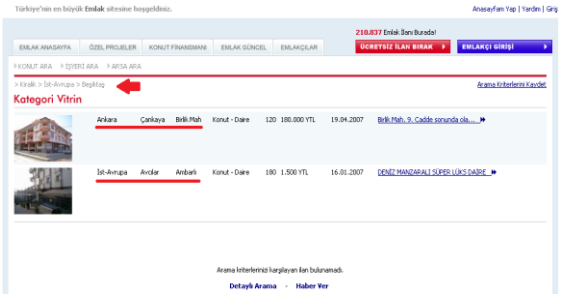
$N(1-(1-L)^n)$

N : the total number of usability problems in the design

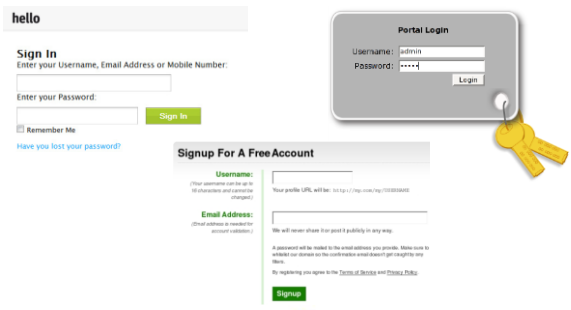
L: the proportion of usability problems discovered while testing a single user



Web Site Search Suggestion Example



Usability on Different Login Pages



Localization Testing

- Testing the customized software product according to targeted software market
- Discovering cultural or grammatical potential failure points
- Ensuring compatibility and functional consistency across all localized software versions
- Determining adaptation level of a software unit for multiple configurations on different locales

Localization Focus Items

- **Accuracy of translation**
- **Text direction**
  - **Bi-directional text :**
    Right-to-left (RTL)
    Left-to-right (LTR)
- **Date Time Parsing**

  Turkey format : 23.5.2011
  US format : 5/23/2011
- **Digits**
- **Text Encoding**

  Turkish Alphabet Dotted & Dotless I

Testing Types - Sample Question

Which of these is a functional test?

a) Measuring response time on an airline reservation system
b) Checking the effect of high volumes of traffic in a CRM software
c) Checking the on-line bookings screen information and the database contents against the customer bill information
d) Checking how easy the system is to use

Happy testing ☺

Thank you...