

# **Software Quality Assurance Training**

# **What covered in the Program**

- **Manual Testing**
  - **Creating Test Plans and Test cases**
- **Automated Testing**
- **Interview Preparation**
- **Job searching Skills**

- **Basics of Testing**
  - Requirement (SCOPE)
  - Time
  - Cost
  - Risk
  - Constraints
  - Assumptions

# Software Quality Assurance (SQA)

- What is it?
  - Ensure Software works according to the requirement.
- Why do we need it?
  - To avoid the defects during the production and ensure the customer satisfaction
- How do we achieve it?
  - By formal method of testing

# What is (SQA)?

- Quality Assurance
  - Ensuring the product conforms to specified requirements
  - Quality = “Fit for use”
  - Quality = Customer satisfaction
  - Acceptable Level of Quality People can use
  - May not be meeting all requirements

# Understand Software Testing

- Doing the right thing
- Doing it the right way
- Doing it right the first time
- Doing it on time without exceeding cost

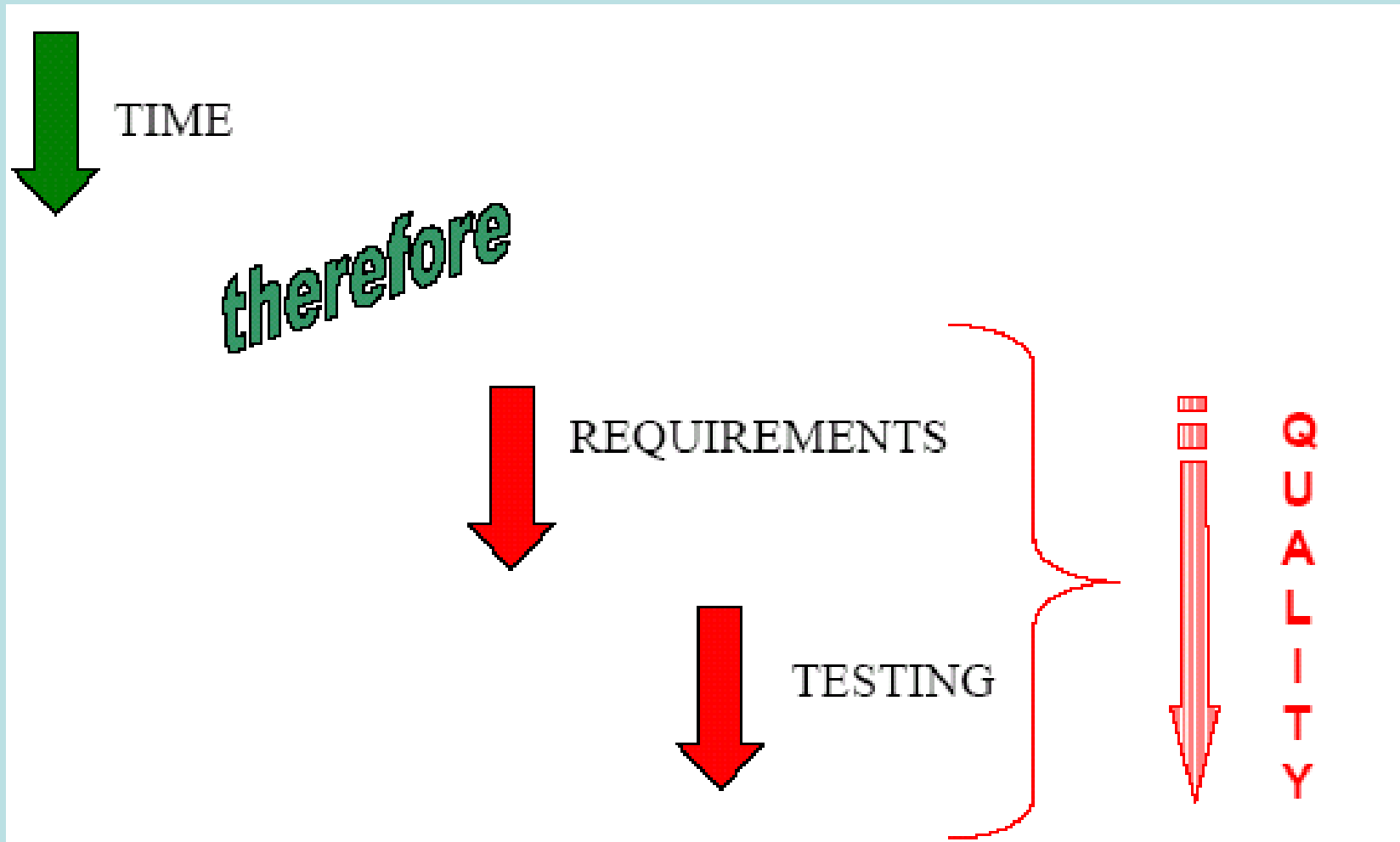
# Fit For Use

- Receiving the right product for their use
- Being satisfied that their needs have been met
- Meeting their expectations
- Being treated with integrity, courtesy and respect
- Product may be released even with out all requirements being met. It is based on SLA and fit for use by the User.

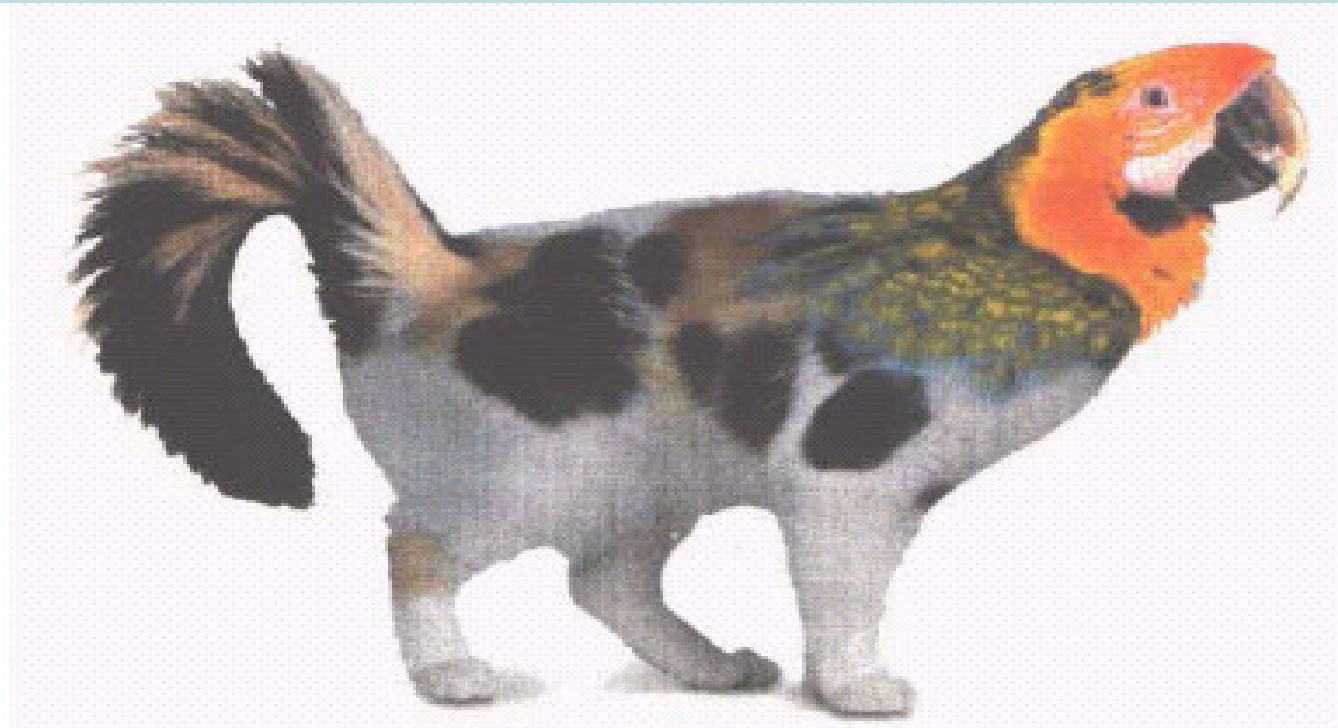
# Quality Assurance Ideals

- Quality Assurance is Valuable.
- Quality Assurance is everyone's responsibility. (Business analyst, Designer, Developer, Tester, Data Modeler, Application/system Architect)
- Quality Assurance is more than software testing.

# Requirement to Testing



# Output with out Software Testing



Paraskitty

- **Basics of Software Testing**

- **BOOKS**

- **The art of software testing - Glenford J Myers**
    - **The Complete Guide to Software Testing- Bill Hetzel**

# **Testing in the Software Development Life Cycle**

- **Software Testing Psychology**
- **Testing in Different Development Lifecycles**
- **The Fundamental Test Process**

# Software Development Life Cycle

- **Sequential**
- **(SDLC, WATER FALL)**
  - **REQUIREMENT**
  - **DESIGN**
  - **DEVELOPMENT**
  - **TESTING (User Acceptance Testing- BA is responsible this)**
  - **IMPLEMENTATION (DEPLOYMENT)**

# Agile

- **Collaborative in nature. Everybody works together in one room.**
- **All Activities same time**
  - **REQUIREMENT**
  - **DESIGN**
  - **DEVELOPMENT**
  - **TESTING (is conducted in conjunction with Development in short release cycles 3 to 6 weeks)**
  - **IMPLEMENTATION (DEPLOYMENT)**

# **Rational Unified Process(RUP)**

- **Collaborative in nature.**
- **All Activities in short interval. (Complete business involvement. Demos at reasonable interval. Show and tell development)**
  - **REQUIREMENT**
  - **DESIGN**
  - **DEVELOPMENT**
  - **TESTING (**
    - **After development in conjunction with business requirements. Often**
    - **Developer writes unit test cases before Application development and verify every unit test case before developing application to test.**
    - **QA can expand the unit test case.**
  - **IMPLEMENTATION (DEPLOYMENT)**

# Testing in the Software Development Life Cycle

- **Verification**

- Stated results are produced

- **Validation**

- Actual of stated results can be compared.

- **Prioritizing Tests**

# Test Management

- Roles and Responsibilities
- Team Organization
- Test Environments and Labs
  - For BA Make sure the test environment is available.
- Test Strategy and Planning
- Scope, Schedule, Cost and Risk Issues

# Testing Basics -1

- **Functional Testing**
  - Testing every functionality
  - Integration Testing
  - All functions are integrating together
- **BA is responsible for function testing in an Integrated environment**
  - Make sure all requirements are met
    - If not please comply the test deliverables with SLA.
      - » Because not all requirement will be delivered.

# Testing Basics - 2

## – **Acceptance Testing**

- **Based on Requirement and SLA**
- **User Acceptance Testing**
  - Test user flow of the system

## – **System Testing**

- Understanding the system level response of system
  - (Also includes Performance of server(s) running as well as network resources involved).
  - CPU, MEMORY, HARDDRIVE, DATABASE Performance
  - Network appliance performance
  - Application Server performance

# Testing Basics - 3

- **Regression Testing**
  - re-testing after fixes or modifications of the software or its environment.
- **Performance Testing(Load Testing)**
  - **Load Testing**
    - Average Load on the application
  - **Endurance Testing**
    - How long the application can run under normal load
  - **Stress Testing**
    - How much load application can take.
      - » (Monitor CPU, MEMORY, NETWORK Resources etc. tool like Site Scope will be used)

# Discovering the Test cases - 1

- **Read the requirement documentation and understand the requirement**
- **Software Design Documentation**
- **Talk to knowledgeable people and understand the application**
- **Navigate through application.**
- **Monitor issues during development**
- **Above all, apply the common sense.**

# Requirement Document

- **Steps**
  - **Flow charts**
  - **Database information**
  - **Interface point to external systems**
  - **SLA(service level agreement)**
  - **Use case documentation**
  - **Software architecture document**

# SLA(Service Level Agreement)

- **Application performance requirements**
  - **Ex: Login should happen within 2 seconds**
- **Boundary Condition Requirements**
  - **Ex: Website should be able to sell 25000 books a day**
  - **Ex: Employee database can store up to 100000 employees**
- **Testing Requirement**
  - **Software Release Matrix**
    - **Ex: 0 Sev 1 defects, 0 Sev 2 defects, 10 Sev3 defects**
      - **Sev1. Major functionality not working, Sev 2 Major functions are not working Sev 3 mean which has a work around**
      - **Load/endurance/Stress Test Requirements**
- **Functionality Requirements**
  - **Must have Functionalities**
- **Contractual Obligations**

# SLA

- **Testing Requirement**
  - **Load Testing/Performance Testing**
    - **Example: Simulate number of users for the Normal Load**
  - **Endurance Testing**
    - **Example: How many days it can run**
  - **Stress Testing**
    - **Example: Whether it can happen the peak load**

# Use Case Documentation

- **Steps**
  - Understand the flow of application
  - Understand what user will do with application
- **Sample Use Case**
  - **Add Student**
    - Open Student registration
    - Add a new student
    - Enter the student information
    - Save student information
    - Receive Confirmation of Student information saved.
  - **Fully Dressed, Partially dressed, Happy Path.**

# Design Documentation

- **Steps**
  - **Software design flow charts**
  - **State diagrams**
  - **Sequence Diagrams**
  - **Interface point to external systems**
    - **Receiving and Sending**
  - **Application user interface documentation**
  - **Software architecture document**

# Informal interview

- **Steps**

- **Developers and Project managers**

- Explain the application design
    - Explain the application requirements

- **Business analyst/Product Manager/Customer**

- Application requirements
    - Deployment requirements
    - Peak usage
    - User experience requirements

# Navigate the application

- **Steps**
  - **Navigate through application**
  - **Understand the difference between the documentation and application**
  - **Understand the difference between the business requirements and Actual Application**
  - **Any major navigation defects**

# Common Sense

- **Steps**

- Doing the right thing
- Doing it the right way
- Doing it right the first time
- Look for Test cases every discussion

# Questionnaire Preparation(Testers)

- **Based steps above**
  - **Develop a questionnaire to ask different people**
    - **Developers**
    - **Business analysts**
    - **Product/project managers**
    - **Customer**
  - **Tabulate the answers**
  - **Discover any other test cases hidden in the questionnaire**

# Deriving Test Cases

- **Steps**
  - **Tabulate the test cases**
  - **Group the test cases using relevance in the area**
  - **Create order for execution of test cases (TEST PLAN –Order in which the test cases are executed)**
  - **Get with Senior QA engineer to review the test cases**
  - **Review test cases with Developers**
  - **Get the feedback**
  - **Incorporate the feedback in the test cases**

# Executing Test Cases

- **Steps**
  - Understand the execution order of the test cases
  - Execute the test cases
  - Tabulate the results
  - Enter defects into defect management system
  - Inform status during the meetings/responsible application owners
    - understand existing organization structure
  - Quality Center (defect tracking tools)

# Test Deliverable

## **Test Deliverables (What are you responsible)**

### **– Test Metrics**

- **Number of Test Pass/Fail**
- **Current Status of Testing**
- **Type of Test Completed**
  - **Functional/Regression/Integration**

# Deliverables

## **Test Deliverables**

### **– Testing**

- **Functional Testing**

### **– Quality Assurance**

- **Ensuring the Service Level agreements are met**
- **SLA( Service Level Agreement)**

# Deliverables

## Test Deliverables

### – Configuration Management

- **Version Control (Testing based on the version of software. Several iteration of software will be delivered.)**
- **Source control**
  - **Changes Implemented in the new release**
  - **Known Defects**
  - **Release Documentation**
  - **Revisions**
  - **Test results are saved into source control**
- **TOOLS( VSS, MKS etc)**

# Deliverables

## Test Deliverables

### – Change Control Board

- Compliance can be a part of Business requirement document. BA is responsible for verification of the same
- *SOX(Sarbanes-Oxley* Compliance)
- Financial systems and Auditing
- Change Prioritization

### – Defect Classification and Reporting

# Deliverables

## Test Deliverables

- **Defect Classification and Reporting**
  - Different Defect Level
    - » Level 1,2,3,4,5
    - » Severity 1,2,3,4,5
- **Reporting Metrics**
  - Used as project Management Tools
  - Understand progress in testing

# Defect Management

## Executed after defect(s) fixed

### – Steps

- **Verify the defect is fixed**
  - Make sure any adjoining functions are working
- **Regression testing (conducted after all verification of Defects)**
  - Make sure any adjoining functions of defects fixed, are working
  - Make sure major functionalities are still working
  - Regression are often conducted using automated tools

# Release Process

**Executed after defect(s) fixed and regression testing completed**

**– Steps**

- Make performance/endurance/stress testing completed
- Make sure User Acceptance Testing completed
- Make sure SLA are met
- Make release documentation completed
- Make sure all the test deliverable presented and accepted
- Production Move process will be initiated.(Change Control. Run book( Operation manual), Reviews etc.)

# Static Testing

## – Walkthroughs, Reviews, and Inspections

- Test Case Doc Walk through
- Joint Discovery of Test cases
- Test case Inspections after test is completed

# **Static Testing - Review**

- Requirements Reviews**
- Design Reviews**
- Code Reviews**
- Toll Gating Process(RUP)**

- **Static Testing**

- **Control Flow Analysis**

- Understanding implementation flows
    - Understanding the System Flow Charts
    - Designing the test cases based on the flow

- **Software Complexity**

# Static Testing

- **Software Complexity**
  - **Complexity analysis**
  - **Creating the Flow charts based on the interfaces software provides**
    - **User interface designer.**
      - » **If BA is responsible for creating UI requirements,BA will be creating the UI flow charts**
  - **Creating Internal implementation flow charts**

# **Dynamic Testing**

- **Equivalence Class Partitioning**
  - **method for deriving test cases**
  - **classes of input conditions called equivalence classes are identified such that each member of the class causes the same kind of processing and output to occur**
  - **Inputs produce Outputs**
    - **Vary input to get different outputs**

# Dynamic Testing

- **Boundary Value Analysis**
  - **Understand the capacity requirements**
    - Data requirement
    - Environment requirement
    - Load/performance requirement
    - Lower and Higher ends

# Dynamic Testing

- **Condition Driven Testing**
  - Functional Testing
  - To address a specific Issue
  - Test is conducted to make broken functionality is fixed
  - Driven By business rules

# Dynamic Testing

- **White Box Testing**

  - **Glass Box Testing**

    - **Code Coverage**
    - **Control Flows in the implementation**
    - **Control Flows in the system Requirement Document**

# Dynamic Testing

## – Non-Structured Testing

- Ad-hoc testing
- Based on Dead line effort
- Software Sanity check
- Sampling of test cases( you use random sample of entire test case family)

# **Software Risk Analysis Techniques**

- **Impacted Systems**
- **Importance of Impacted systems**
- **Prevention methods**
- **Warning methods in place**
- **Workarounds if any.**
- **Disaster recovery Methods in Place**

# The Limits of Testing

- How much testing
- Test Case Designs and Documentation
- Prioritizing Tests
  - Importance every functionality
  - Understanding the current need
  - Time available to execute testing

# The Limits of Testing

- **Test Completion Criteria**
  - **Deadlines (release deadlines, testing deadlines...)**
  - **Test cases completed with certain percentage passed**
    - **95% of test case passed with complete agreement of SLA.**
    - **5% defects will tracked using a defect management system and will be fixed later**
  - **Coverage of code/functionality/requirements reaches a specified point**
  - **Bug rate falls below a certain level**
  - **Beta or alpha testing period ends**

# Quality Statistics

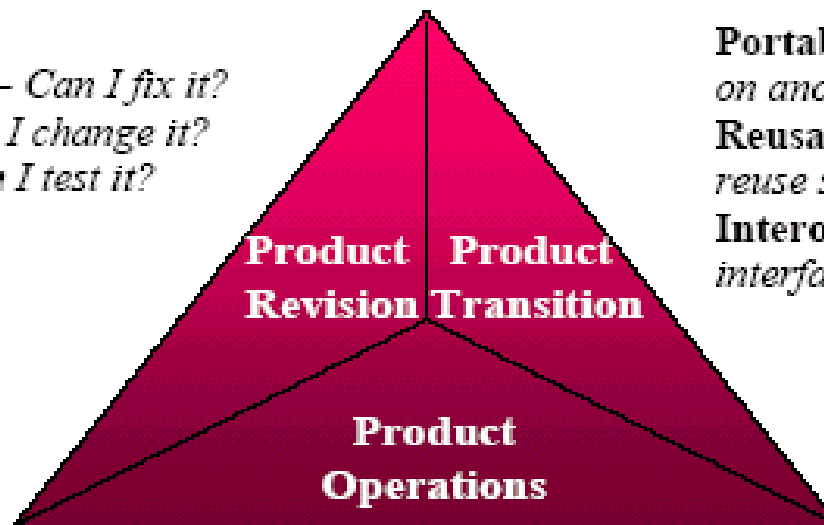
- Low quality software costs U.S. businesses \$60 billion per year.
- 80% of unplanned downtime is caused by software application and production errors, costing as much as \$100,000 per hour.

# Non-Functional Requirements

**Maintainability** - *Can I fix it?*

**Flexibility** - *Can I change it?*

**Testability** - *Can I test it?*



**Portability** - *Will I be able to use on another machine?*

**Reusability** - *Will I be able to reuse some of the software?*

**Interoperability** - *Will I be able to interface it with another machine?*

**Correctness** - *Does it do what I want?*

**Reliability** - *Does it do it accurately all the time?*

**Efficiency** - *Will it run on my machine as well as it can?*

**Integrity** - *Is it secure?*

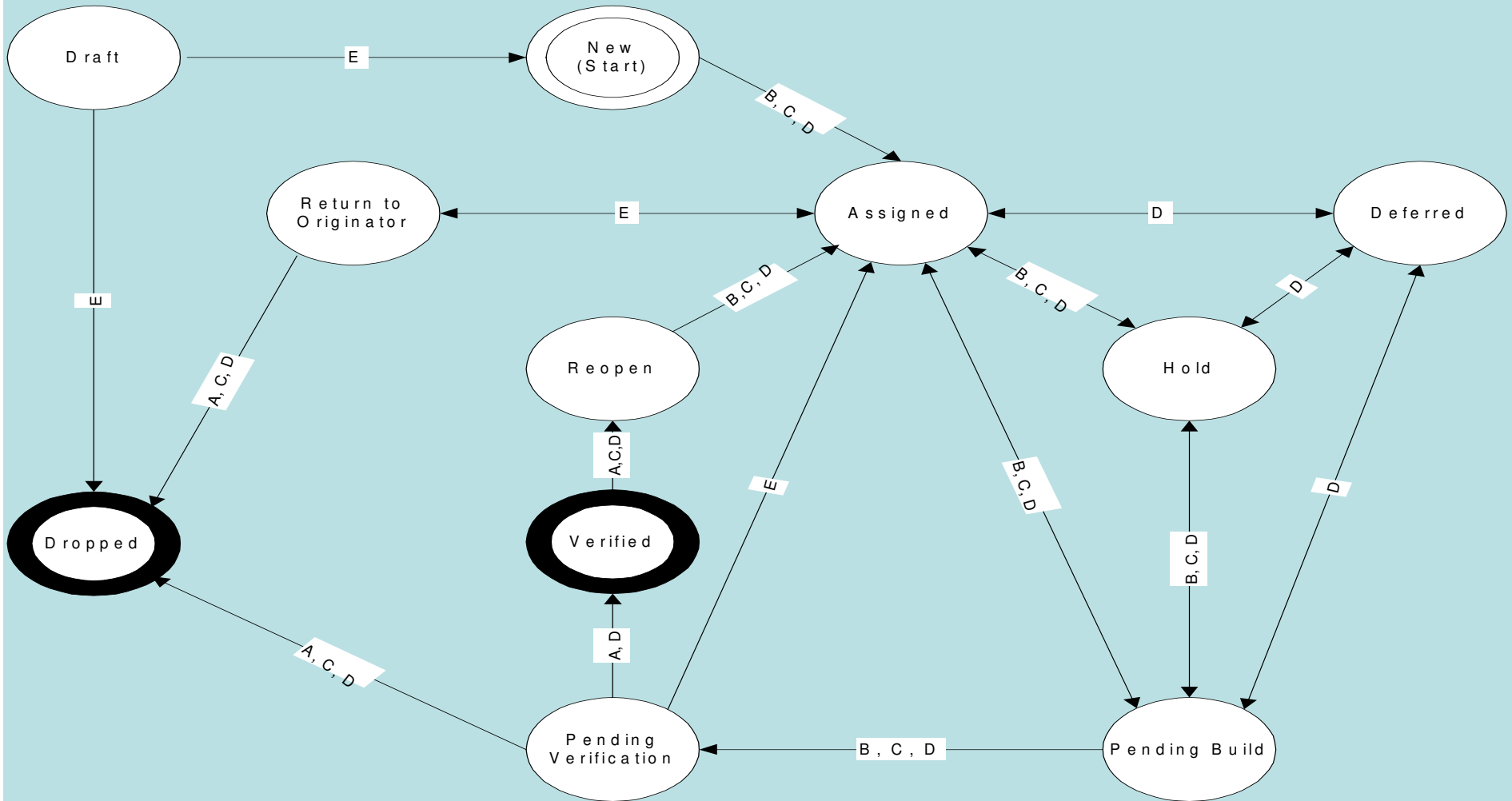
**Usability** - *Can I run it?*

# Basics of Software Testing

## – BOOKS

- **The art of software testing - Glenford J Myers**
- **The Complete Guide to Software Testing- Bill Hetzel**

# DEFECT MANAGEMENT



**Key:**

- A - QA Engineer/QA Lead/Requirement Management
- B - Developer
- C - Developer Lead
- D - Project Manager/Director\*
- E - All Users Groups

\* - Members of this group may set an issue to any status regardless of enforced transition rules.

# This Session is about executing a project

- Understanding Requirement
- Creating Test Plan ( order in which you execute)
- Creating Test Case (Every functionality)
- Creating Test Script (Actual values for test case.  
Employee: John Smith, 32, 1234, Abcd Line Atlanta GA)
- Executing Test Case ( Manual as well as automatic)
- Reporting Issue ( any thing Failed)
- Escalation Process (Assigning to PM, Developer based on process in the (organization)
- Software Release Matrix(Are we meeting SLA)

# Sample Project

- The Program reads three integer numbers from Input dialog.
- The values represent length of 3 sides of the triangle
- Program displays following message
  - Isosceles(Two sides are equal)
  - Equilateral (All sides are equal)
  - Scalene (No sides are equal)

# Identifying the Test Cases

- Every line in the document counts.
- Positive and negative approach
- Conditional flow(Triangle can be made)

# Test Plan

- Reverse route
  - Create the test case first
  - Group the test cases with heading
  - Then create the order of execution
  - Become Test plan
- What is Test Plan
- Why is that important ( order of execution)
- How often it is created. ( in most cases once)

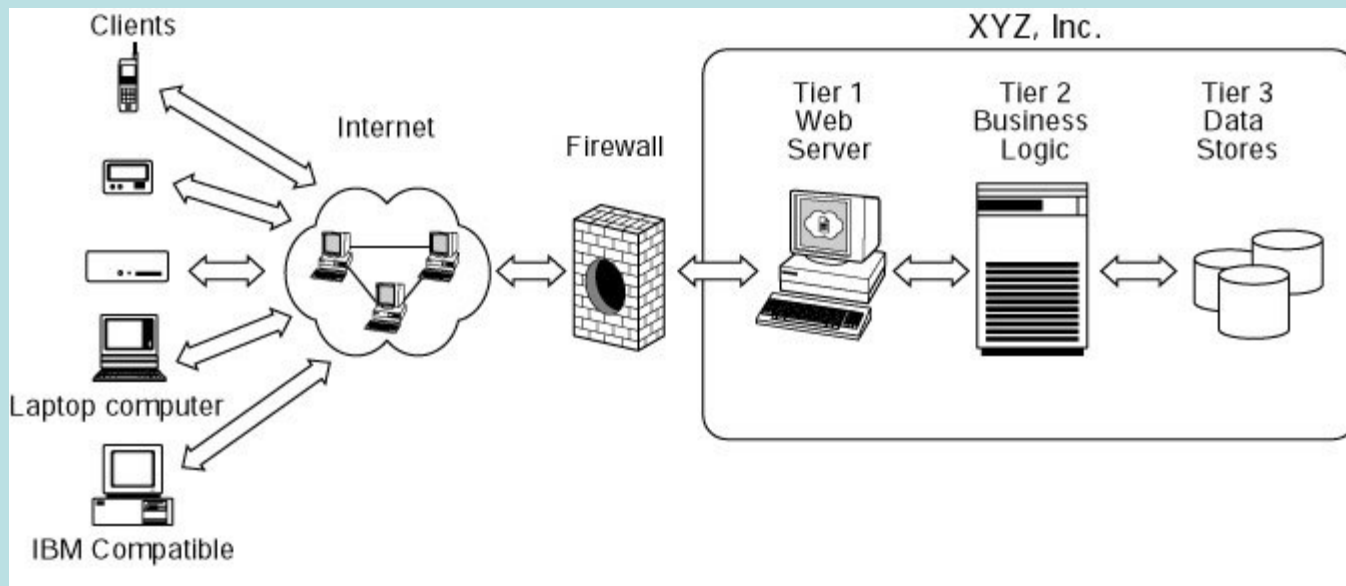
# Test Case

- What is Test Case
  - Every step expanded and may not have actual values. But the steps and processes are explained
- Actual Values are used in Test Script

| <u>Steps</u>   | <u>Data</u>                           | <u>Expected Results</u>   | <u>Actual Results</u> |
|--|---------------------------------------|---|-----------------------|
| Enter User Name and press LOGIN Button               | User Name= COES                       | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter Passw ord and press LOGIN Button               | Passw ord= COES                       | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter user Nameand Passw ord and press LOGIN Button  | USER = COES AND<br>Passw ord = XYZ    | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter user Name and Passw ord and press LOGIN Button | USER = XYX AND<br>Passw ord = COES    | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter user Name and Passw ord and press LOGIN Button | USER = XYZ AND<br>Passw ord = XYZ     | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter user Name and Passw ord and press LOGIN Button | USER = " " AND<br>Passw ord = " "     | Should Display Warning Message Box "Please Enter User name and Passw ord" |                       |
| Enter User Name and Passw ord and press LOGIN Button | USER = COES AND<br>Passw ord = COES   | Should navigate to CoesCategoryList.asp page.                             |                       |
| Enter User Name and Passw ord and press LOGIN Button | USER = ADMIN AND<br>Passw ord = ADMIN | Should navigate to Maintenance page page.                                 |                       |

# Internet Application Testing

# Internet



- Transaction processing
- User authentication
- Data validation
- Application logging

# Challenges

- *Large and varied user base.* The users of your Website possess different skill sets, employ a variety of browsers, and use different operating systems or devices. You can also expect your customers to access your Website using a wide range of connection speeds. Not everyone has T1 or broadband Internet access.
- *Business environment.* If you operate an e-commerce site, then you must consider issues such as calculating taxes, determining shipping costs, completing financial transactions, and tracking customer profiles.
- *Locales.* Users may reside in other countries, in which case you will have internationalization issues such as language translation, time zone considerations, and currency conversion.
- *Testing environments.* To properly test your application, you will need to duplicate the production environment. This means you should use Web servers, application servers, and database servers that are identical to the production equipment. For the most accurate testing results, the network infrastructure will have to be duplicated as well. This includes routers, switches, and firewalls.
- *Security.* Because your site is open to the world, you must protect it from hackers. They can bring your Website to a grinding halt with denial-of-service (DoS) attacks or rip off your customers' credit card information.