# ZenQ Quality Assurance (QA) Process

A FUNCTIONAL TESTING WHITE PAPER

# Contents:

# Executive Summary

In the current day scenario, there are numerous options available to the clients when it comes to outsourced Testing services-making it a difficult choice for finding the right IT partner for their business. With over 10+ years of industry experience successfully executing multiple Testing engagements , our technical & domain expertise, structured project management methodology makes us a reliable partner of choice for your business - for delivering cost effective high quality Testing solutions to speed up time to market, increase revenue and drive business growth

This white paper discusses the strategy and process that we adopt at ZenQ for functional testing enabling clients to meet the set quality targets – regardless of platforms, technology, project complexity or organization size.

# Testing Services

ZenQ offers the following testing services to improve the quality of the product –

## Functional Testing

The general purpose of software functionality testing is to verify if the product performs as expected and documented. ZenQ specializes in testing new and regression functionality of software products through Alpha, Beta and FCR release phases. Whether the product development is done in-house or outsourced or whether agile processes are implemented or otherwise – we are familiar with the challenges in identifying all the important defects.

## Test Automation

ZenQ's Test Automation services are aimed at making repeat regression testing of products fast, accurate and seamless for our clients via automation of regression test suite – reducing overall testing cycle time, enhancing testing productivity and as a result, improving product quality. Equipped with deep understanding of automation best practices, experience in the use of several industry standard commercial and open source test tools, we can recommend right test tool to automate testing for your product.

## Performance Testing

ZenQ offers Load, Stress and Capacity Testing services to performance test Web, Application and Database servers. We create realistic models simulating different business environments and provide a repository of information to easily analyze and report results throughout application development process.

## Security testing

ZenQ offers Web Application Penetration Test service – a set of comprehensive tests to identify application vulnerabilities. We use a risk-based approach, grounded in both the application's architectural reality and the attacker's mind set, to gauge the application security adequately. By identifying risks in the system and creating tests driven by those risks, we focus on areas of code in which an attack is likely to succeed.

# Manual Testing Services

ZenQ performs the following types of testing manually at different phases of testing cycle to improve the quality of the product

## Functional testing

ZenQ performs Functional testing of the application to find if each function in the application is performing what it is expected to do and is not doing what it is not expected to do.

## Smoke testing

ZenQ performs testing of main flows of the application for every new build released. This helps us to determine if the build is ready for detailed level testing. A build failed in Smoke test is rejected and requested for another build.

## Regression Testing

ZenQ performs full cycle of regression when a new feature is added or any bug is fixed to make sure that no new bugs were introduced with the new code/bug fix. ZenQ performs regression on multiple environments to make sure that no environment specific bug is introduced with the code change.

## Usability testing

It is very important to have user friendly GUI for smooth interaction of users with the application. The main objective of usability testing is to ensure that an application is easy to understand and navigate. ZenQ verifies usability by considering the following parameters –

★ Performance –– How much time, and how many steps, are required for users to complete basic tasks? (For example, find something to buy, create a new account, and order the item.)

★ Accuracy -- How many mistakes did Users make? And were they fatal or recoverable with the right information? – Proper error messages are displayed or not?

★ Recall -- How much does the user remember afterwards or after periods of non-use?

★ Emotional response -- How does the user feel about the tasks completed? Is the person confident, stressed? Would the user recommend this system to a friend?

## Exploratory testing

Once we are familiar with the application we perform exploratory testing considering positive and negative scenarios which are not covered in test suite created. In general we do not have a test suite to carry out exploratory testing. During the execution we do find scenarios and add them to existing test suite. This is an experience based testing technique.

The main advantage of exploratory testing is that less preparation is needed, important bugs are found quickly, and at execution time, the approach tends to be more intellectually stimulating than execution of scripted tests.

# Test Life Cycle

The Table 1.0 below lists the project phases in the chronological order along with the owner responsible for the task/action. Each of the phases are discussed in detail in the subsequent sections.

| Phase | Action | Responsibility |
|---|---|---|
| Test Plan creation | Provide Product documentation (User Manual and/or Functional Specifications), a stable copy of the product and/or URLs, system usernames/passwords | Client |
| | Provide Known bugs/issues document* | Client |
| | Provide a Single point of contact/emergency contact | Client/ZenQ |
| | Submit initial version of the Test Plan | ZenQ |
| | Discuss issues related to the Test Plan | Client/ZenQ |
| | Review and Sign-off the Test Plan | Client |
| Test Design | Develop Manual test cases/Automated Test scripts | ZenQ |
| | Review draft version of test suite and provide feedback | Client |
| | Update the test suite to incorporate the client feedback | |
| | Create traceability matrix to map the requirements to the test cases. Submit traceability matrix and final version of test suite for client sign off | ZenQ |
| | Create Users representing Client and ZenQ on web based bug tracking system | ZenQ/Client |
| Test Execution | Setup test environment(s) per test plan | ZenQ |
| | Submit a specific build of the product for testing | Client |
| | Execute the Test cases | ZenQ |

| | Report bugs with detailed information in to the web based bug tracking system | ZenQ |
|---|---|---|
| | Update each test case with test result | ZenQ |
| Reporting | Prepare and Submit Test Results Report, Bug Reports and Observations made during the testing | ZenQ |
| | Provide clarifications/reproduce bugs on demand* | ZenQ |
| Test suite maintenance | Any bug found not related to the test case being executed, Update the test case document with the test that yielded new bug | ZenQ |
| Bug fix Validation | Provide a list of fixed bugs | Client |
| | Retest the fixed bugs | ZenQ |
| | Update the status of bugs in the bug tracking system's database | ZenQ |
| | Track the bugs until all bugs have been closed, rejected or postponed | ZenQ |
| Test closure | Send a quality report with ZenQ recommendation on the build. Send summary reports | ZenQ |

*Optional – may or may not occur

# Test Plan

During this phase, planning on how to approach testing an application/module /release is carried out. Test plan document is created and shared with the client. The Test Plan typically includes the following sections. Each of these sections and activities are described in the Appendix A.

★ Test requirements
★ Scope
★ Assumptions & Constraints
★ Test Strategy
★ Test environments
★ Schedule
★ Resource requirements
★ Test Execution & Reporting
★ Test Closure
★ Effort estimates and costs associated

# Test Design

After creating the test plan, we generate test ideas for functionalities within the scope using mind maps and other tools. We generate test ideas applying different techniques like Functions, Users, flows, Scenarios, Risks, and Data and considering different criteria like capability, reliability, usability, installability, compatibility etc. as applicable. Different test case design techniques like Boundary value analysis and equivalence class partitioning are applied on any fields under test.

These test ideas are then converted into test cases, and a comprehensive test suite is created. We map test cases created to the requirements via traceability matrix to make sure that every requirement has been associated with one or multiple test cases. We always ensure that the test suite is live and we'll keep adding any scenarios as they are discovered during the test execution. The Figure 1.0 below depicts the Test Design Process Flow
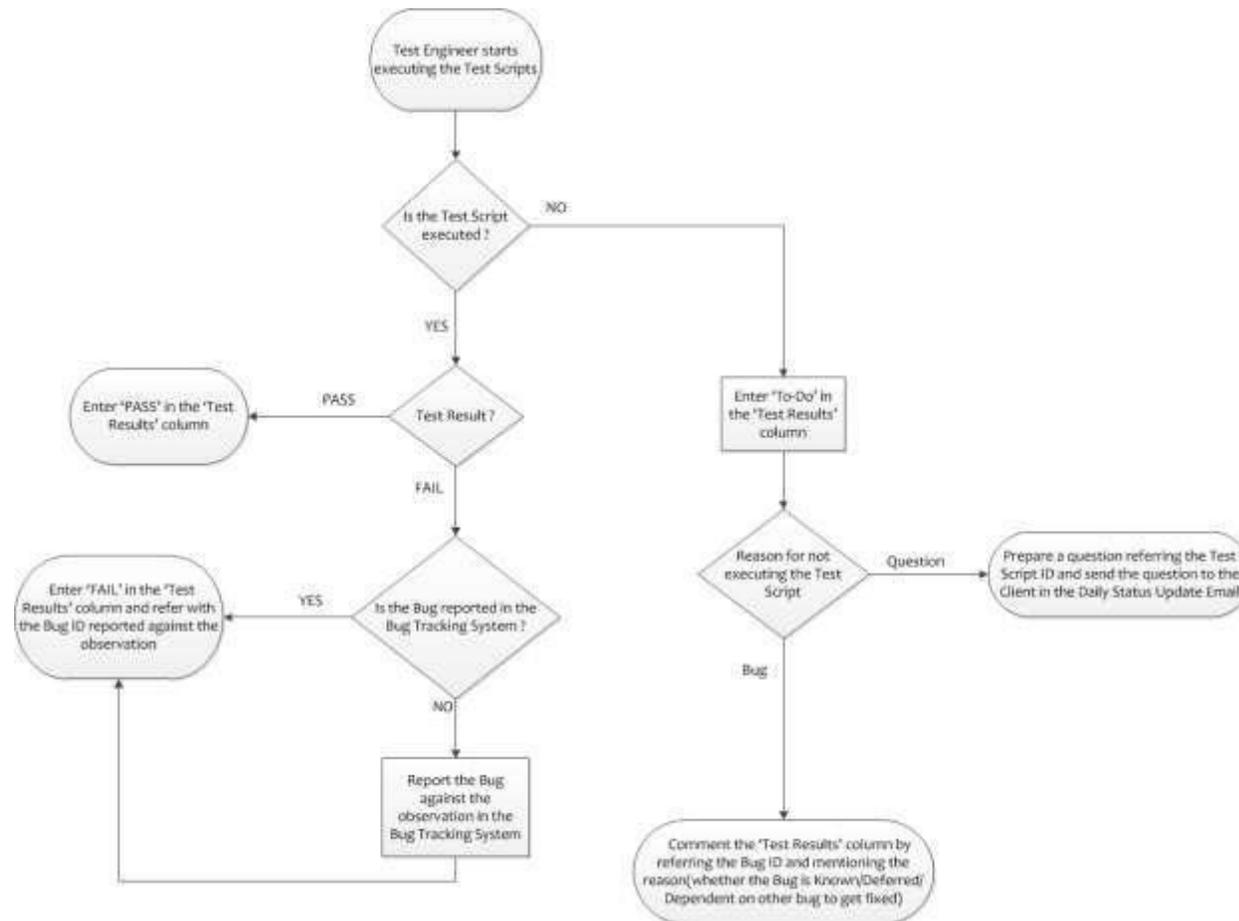
Fig 1.0: Test Design Process Flow
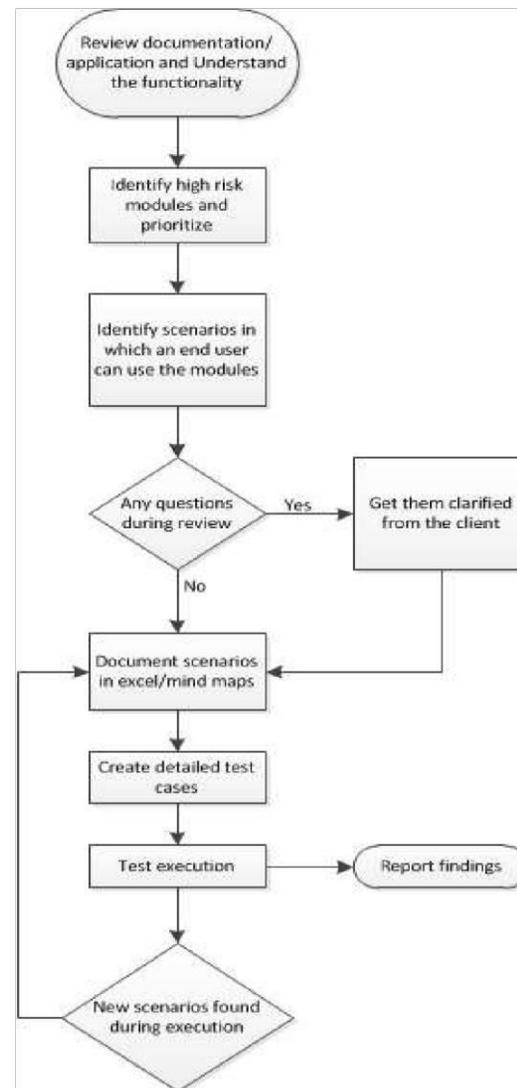
# Test Execution

Test Engineers at ZenQ follow the process shown in the below Flowchart for the manual execution of the Test Scripts -

Each Test case executed is updated with either of the test results -Pass or Fail in the 'Test Results' column.

If a test case fails, the 'Test Result' column is updated with the text 'FAIL'. A new Bug is reported (if not reported already) and corresponding Bug ID is included in the 'Comments' column against the test case.

If a Test case cannot be executed, 'Test Results' column is updated with the text 'To-Do'. If an existing bug is preventing to execute a test case, 'Comments' column will be updated with the Bug ID and appropriate reason.

*If any clarifications are required before the Test case can be executed then the 'Test Results' column is updated with 'To-D0' and the*

*Question is sent to the Client separately*

# Bug Reporting

Test Engineers at ZenQ provide the following information for each bug reported in the Bug Tracking System:

★ Status of each new bug reported is set to 'Open'

★ Severity level – Critical, High, Medium, Low (The description of each of the Severity Levels is described in the Appendix B)

★ Test Case Number

★ Test Environment

★ Functional area

★ Reproducibility – Always, occurs sometimes, occurs randomly, occurred only once

★ Steps to follow in order to reproduce the bug (including Login details, Prerequisites(if any), Link/Selection Navigation)

★ Screenshots captured depicting the bug discovery process with the following details –
  - Bug Title
  - Summary of the Bug
  - Actual Result
  - Expected Result
  - Areas affected

Workarounds, if any

# Bug Life Cycle

The general lifecycle of a bug or defect followed at ZenQ is depicted in the below Flowchart. If a Client has a different set of states to track the bugs, we will follow the client's bug tracking workflow.
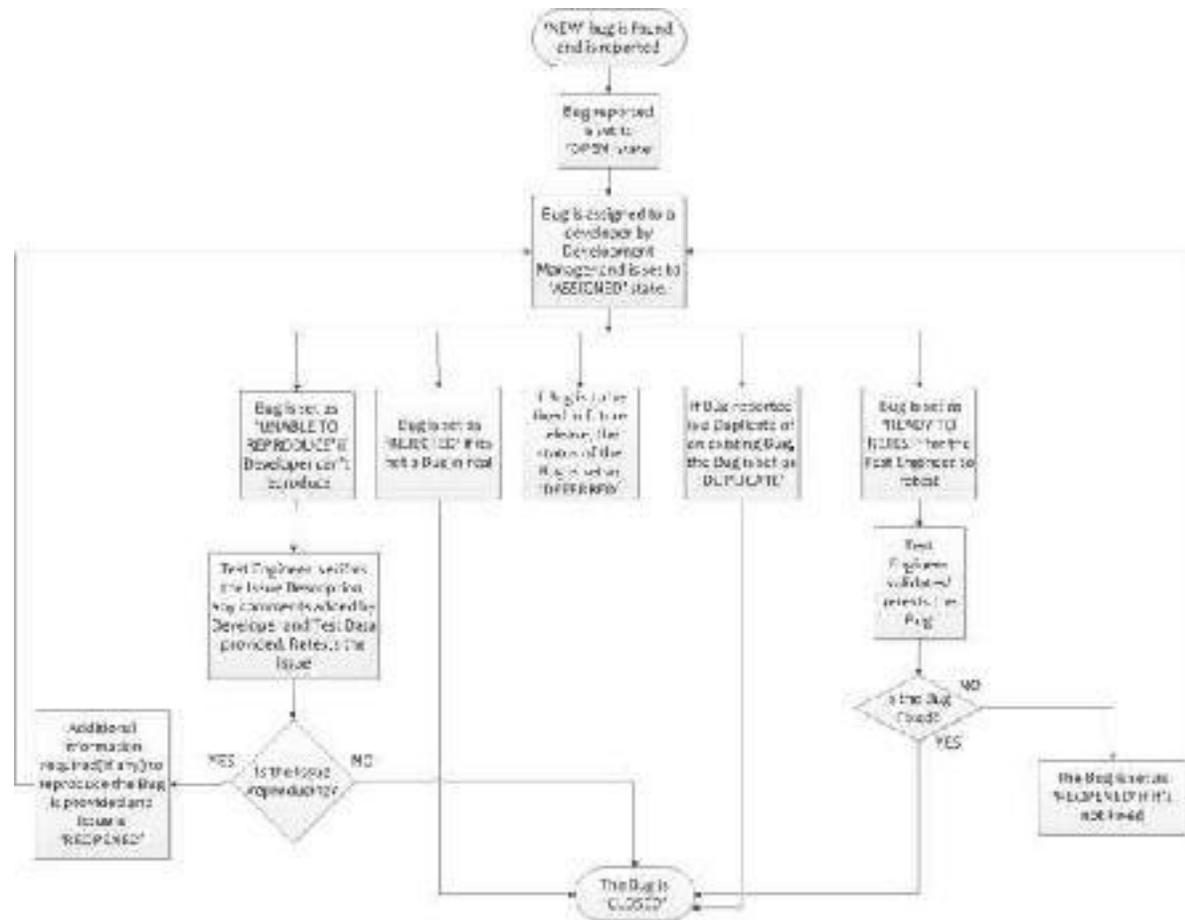
# Bug discovery process over multi-cycle testing

Execution of each test script during each test cycle starts with an empty 'Test Results' column. While executing each test script, if one or more bugs are discovered, Test Engineer verifies if the bug is already reported in the bug tracking system. If the Bug is not found in the tracking system, a new bug will be reported against the observation.

If the bug is found reported, the below outlined process is followed:
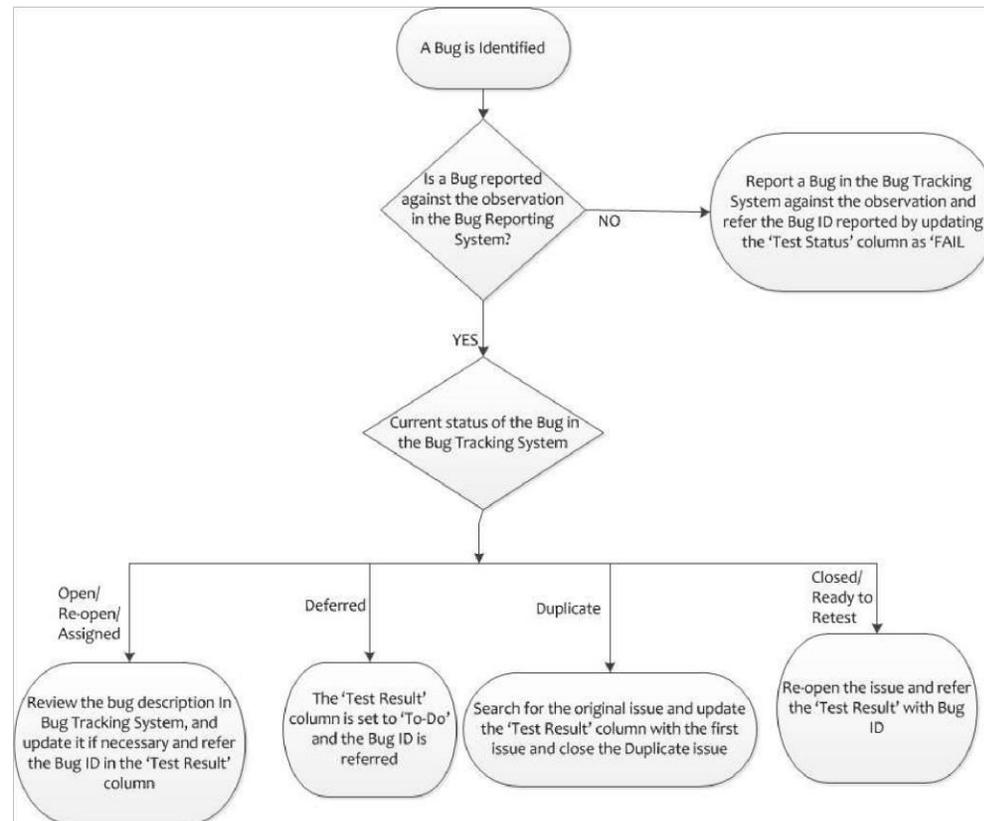
## Fix Validation

ZenQ has a process (depicted in the diagram below) in place for the testers to follow during the retest and validation of the bugs, which have been fixed by the Client's development team and flagged with 'Ready for Retest' status in the Bug tracking system. The Status of each Retested bug is set to either 'Closed' or 'Reopen' – reflecting the outcome of the validation with appropriate comments in the Bug tracking system. All bug fix validation testing results are collated at the end of testing every day and the summary is conveyed to the client as part of the 'Daily Status Report'.
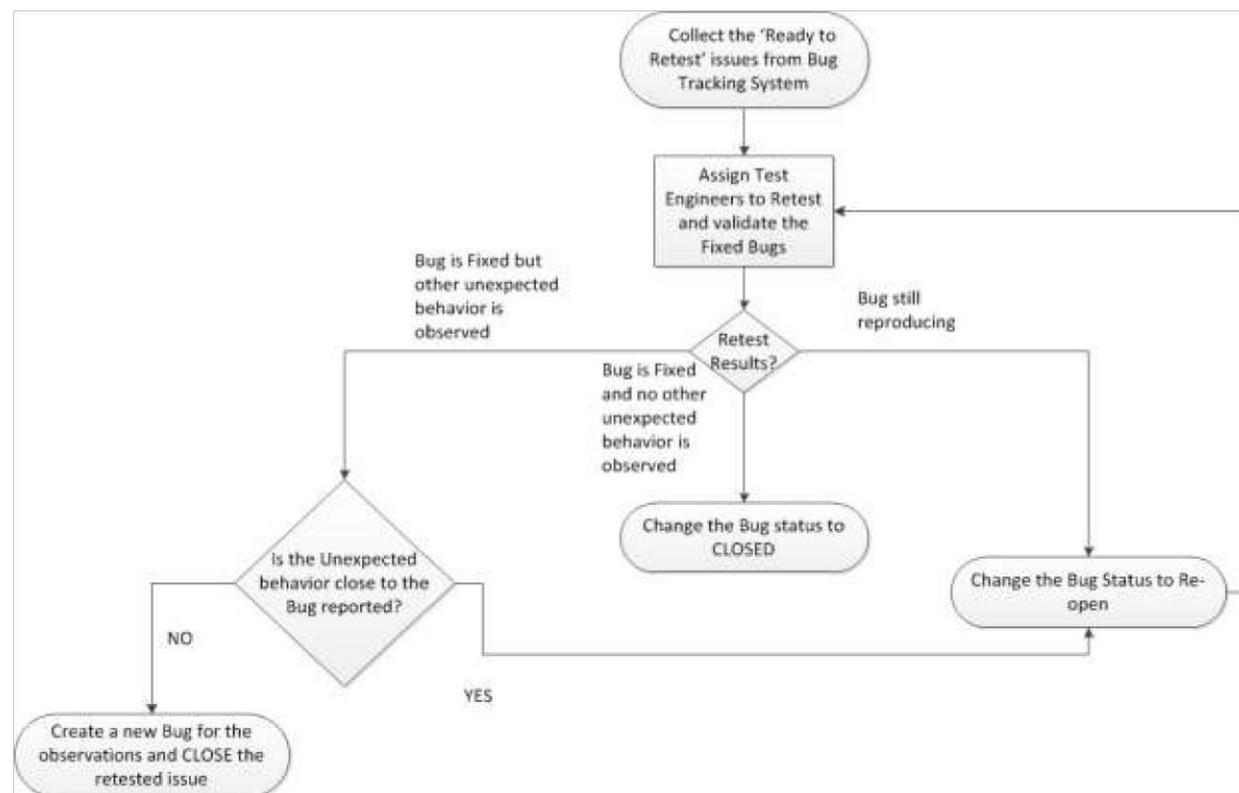
## Bug Tracking System

We enter new bugs; update the status and description of the existing bugs according to the retest results in the bug tracking system during the test cycle. Clients will be able to run queries and generate custom online reports to get information real-time.

## Tracking Tools

Our Test Engineers employ various software products for tracking the bugs through the phases of open, retest, and close. These include web based tools such as TFS, version1, Quality center, JIRA, BugZilla, Mantis, TeamTrack and Rally. If the client has already been using a certain bug tracking tool, we will track bugs using the same tool.

# Reporting

ZenQ provides Real-Time, Daily, Project Status and End-of-Cycle reports to the client throughout the QA cycle.

## Daily Status Reports

Daily status report help us keep apprise the client of the summary of the work done during the day as well as current status of the project and to help resolve issues requiring client attention. At the end of the testing day, a daily status report is prepared with the following information:

- New Bugs reported
- Retest Results
- Task performed by each resource and status of Tasks
- Observations/Questions (product behaviour related) during the day
- Bugs with 'Rejected', 'Deferred' and 'Duplicate' status that may be impeding further testing or require to be reclassified upon client approval
- Project current status document with information on the total number of test cases, number and percentage of test cases executed until today and the Test schedule for the following day

## Project Status Dashboard

Project Status Dashboard is sent to the client to inform them about the Status of the tasks carried out during the week. It will contain the following information:

- Tasks performed in the current week
- Status of the Tasks performed
- Test schedule for the following week

## End-of-cycle Reports

At the end of the release cycle, we provide clients a complete set of test results for the cycle. It will contain the following information:

- Number of test cases executed
- Number of test cases passed/failed
- Number of bugs discovered by functional area, severity level and bug status
- Quality Report with the Recommendation for that release

# Automated Testing

As software projects become bigger, the cost and effort for manual testing will increase phenomenally. Functional Test automation is one powerful solution, which has been widely accepted all over the world, to reduce the effort and cost of software testing to a reasonable level. Once tests have been automated, they can be run quickly and repeatedly. This is often the most cost effective method for software products that have a long maintenance life, because even minor patches over the lifetime of the application can cause features to break which were working at an earlier point in time.

ZenQ recommends automated testing when –

★ The test case should be executed on each new build

★ The test case workflow and GUI would be changed very slowly over time

★ The test case validates a workflow or business process rather than GUI (Look and feel, font, layout etc)

★ When a test case has to be repeated with multiple sets of Data

★ When a test case has to be executed on multiple test environments

Once the scripts are developed, ZenQ maintains them with respect to any minor changes in the application.

We develop automation test scripts for the test cases using the following test automation tools - HP Quick Test Professional, IBM Rational Robot, Test Complete, Selenium, Microsoft Coded UI & Frog Logic Squish.

# Communication

We maintain communication channels between ZenQ and the client team to ensure that we provide quick response and always accessible information to the client. We provide single point of contact and assign a dedicated test team. The communication mechanisms employed by us are:

- ★ Daily Emails
- ★ Daily, Weekly, and End-of-Cycle Reports
- ★ Daily and Weekly Teleconference calls
- ★ Online chats via Skype, MSN, Windows Live Messenger, Lync etc
- ★ Product Demos or Product trainings via WebEx, GoToMeeting
- ★ FTP downloadable files

# Client Engagement

## Request for Pilot:

For those potential clients who are new to outsourcing their QA efforts and/or are interested in evaluating our processes and abilities in test preparation, test execution, bug discovery, documentation and communication, we offer to carry out pilot for testing a specific function of one of their products. Pilot is recommended by many industry experts as one of the first steps in choosing a right QA partner for your organization. It offers the following benefits:

- ¤ Enables the client to evaluate ZenQ at no cost and no risk
- ¤ Enables the client to evaluate the quality of the deliverables
- ¤ Enables the client and ZenQ to calibrate the expectations
- ¤ Tests the communication processes and channels between the client and ZenQ

Pilot testing is done with two test engineers for two weeks. In response to the client's request for Pilot test, after the initial assessment, we submit a test plan and test cases to the client. The test plan will have information on the pilot test process, scope, schedule, deliverables and the strategy to successfully initiate and execute the pilot test. Upon review and signoff of the test plan by the client, we execute the tests. At the end of test execution - test results, bug reports and observations are documented, collated and submitted as test results report to the client. At the end of pilot project, we submit summary reports with graphs.

## Pilot steps:

The table 1.1 below summarizes the tasks involved in the Pilot

| Task | Responsibility |
|------|----------------|
| Sign Client NDA | ZenQ |
| Define Scope & Expectations | Client |
| Provide access to Product & documentation | Client |
| Review functionality in scope | ZenQ |
| If required 30 min demo on Product functionality | Client |
| Create Test Suite | ZenQ |
| Review & approve Test Suite | Client |
| Execute User case and Test cases | ZenQ |
| Submit daily Reports (status, defects, execution) | ZenQ |
| Review reports | Client |
| Post Pilot feedback meeting | Client & ZenQ |

## Engagement model:

The below Fig 1.4 depicts the engagement model that we follow at ZenQ for Testing Projects.

```
┌─────────────────┐        ┌─────────────────┐
│     Initial     │  ──▶   │  Pilot Project  │
│  Teleconference │        │                 │
└─────────────────┘        └─────────────────┘
                                     │
                                     ▼
┌─────────────────┐        ┌─────────────────┐
│    Services     │  ◀──   │   Post -Pilot   │
│    Agreement    │        │    Feedback     │
│     Signed      │        │     Meeting     │
└─────────────────┘        └─────────────────┘
         │
         ▼
┌─────────────────┐        ┌─────────────────┐
│     Assign      │  ──▶   │    On-Going     │
│ Dedicated team  │        │    Project      │
│   & Kick-off    │        │                 │
│     Meeting     │        │                 │
└─────────────────┘        └─────────────────┘
```
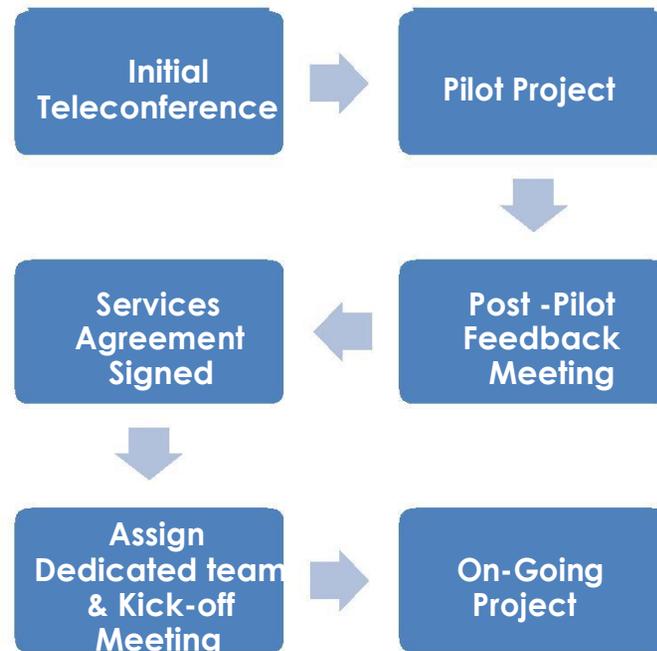
# Appendix A – Test Plan Document

This section describes the sections covered in the Test Plan Document created by ZenQ during the Test Planning Phase:

## Test requirements

Test requirements are collected from the client to understand the expectations out of the current testing effort. All requirements will be listed under this section.

## Scope

Scope of the effort will be defined with modules and sub-modules of the product to be tested. This section also specifies the inclusions and the exclusions as part of the current testing effort.

## Assumptions and Constraints

Any assumptions and constraints will be listed under this section.

## Test Strategy

We follow risk based testing approach in testing a product. We prioritize different modules of the product based on the risk; the impact it would have if failed. We spend majority of our time testing on the high priority modules. In this section, we mention the details of prioritized modules and different testing techniques that we apply in testing them.

# Test environments

We list all test environments that we are expected to test the product. Below are different test environments we work with.

## Desktop Operating systems

| Operating system | Version |
|---|---|
| Microsoft Windows | Windows 7 (32bit & 64 bit) |
| | Windows Vista |
| | Windows XP |
| Apple Macintosh | 10.4 Tiger |
| | 10.5 Leopard |
| | 10.6 Snow Leopard |
| | |
| | 10.7 Lion |
| Linux/Unix | Latest versions |

## Smartphone operating systems

| Operating system | Version |
|---|---|
| Android | 2.1 |
| | 2.2 |
| iOS | 4 |
| | 5 |
| Blackberry | 5 |
| | 6 |
| | 7 |

## Smart phone devices

| Device type | Version |
|---|---|
| iOS devices | iPhone 4 |
| | iPod touch |
| | iPad 2 |
| Android devices | HTC Desire |
| | Samsung tablet |

Note: We can procure any device which is not listed above on need basis

## .NET Frameworks

| .Net Framework | 2.0 |
|---|---|
| | 3.0 |
| | 3.5 |
| | 4.0 |

## Microsoft Office

| Microsoft Office | 2003 |
|---|---|
| | 2007 |
| | 2010 |
| | Office 365 |

## Virtual Environments

| Virtualization | Citrix |
| --- | --- |
| | Hyper view |
| | VMware |
| | Esx Server |

## Schedule

A detailed schedule on the tasks planned to execute will be listed in this section.

- Resource requirements

- All hardware/software and Test engineer requirements will be listed here.

## Test execution & Reporting

This section is discussed in detail through Test Execution and Reporting sections above in the document

## Test Closure

After the execution of Test Suite, ZenQ performs Test Closure activity which involves the following -

- Creating Quality report with recommendation on the readiness of the build to get released

- Archiving results, logs, reports, and other documents and work products(Builds)

- Meet, discuss and analyze testing artefacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle.

- Submit Summary Defect Reports with relevant graphs and charts

## Effort estimates and Costs associated

Effort estimates will be calculated and provided for the current testing effort. Depending on the tasks, resource assignment will be recommended. All costs associated to the resources, licenses will be included in this section.

# Appendix B - Severity Levels

Each Bug reported is assigned with one of the following Severity levels –

## Severity Level 1: Critical

A Bug in the application which leads to a Crash/Hang or unrecovered data loss. In general, these are the bugs that can prevent the product from being released.

## Severity Level 2: High

A major functional area of the application (significant to business process) is affected and testing can be continued on the scenario using difficult workarounds, and/or significantly impacts the business' ability to use the application or its ability to operate the service.

## Severity Level 3: Medium

Incident affects an area of functionality, but there is a workaround which negates impact to business process. This is a problem that:

★ affects a more isolated area of functionality

★ occurs only at certain boundary conditions

★ has a workaround (or)

★ occurs intermittently

## Severity Level 4: Low

Problem that is unlikely to occur in normal use: such as failures at extreme boundary conditions or minor error in layout/formatting. The issue does not limit the use of the product in any substantive way. Incidents those are cosmetic in nature, and of zero or very low impact to business process.

## About ZenQ

ZenQ is a global provider of high quality Software Development & Testing Services, offering cost effective value-add outsourcing solutions to our clients. Our highly competent IT Professionals, Domain experts, combined with industry best practices & our investments in state-of-the-art technologies makes us a dependable and long-term IT service partner to all our clients

For more information, email us at : **sales@zenq.com** **OR** Visit us at **www.zenq.com**