QA Process Whitepaper

ZENQA

TABLE OF CONTENTS

1.0 Introduction 3
1.1 Overview 3
1.2 Types of Testing 3
1.2.1 Functional/Regression Testing 3
1.2.2 Compatibility Testing 3
1.3 Automated Test Tools 3
1.4 Bug Tracking Tools 3
2.0 Initial Contract 4
2.1 Request for Information 4
2.2 Request for Pilot Test 5
2.3 Request for Proposal 5
3.0 Test Life Cycle 6
3.1 Formal Entry and Exit Criteria 6
4.0 Test Plan 7
4.1 Process 7
4.2 Structure 8
4.2.1 Scope 8
4.2.2 Test Requirements 8
4.2.3 Test Strategy 8
4.2.4 Functional Areas Tested 9
4.2.5 Test Case Id Index 9
4.2.6 Traceability Matrix 9
4.2.7 Detailed Test Cases 10
4.2.8 Test Environments 10
5.0 Test Execution 11
5.1 Process 11
5.2 Bug Reporting 12
5.3 Severity Levels 12
5.4 Bug Life Cycle 13
5.5 Bug discovery process over multi-cycle testing 14
5.6 Bug Fix Validation 15
6.0 Reporting 16
6.1 Bug Tracking System 16
6.2 Daily Status Reports 16
6.3 Weekly Status Reports 16
6.4 End-of-cycle Reports 16
7.0 Automated Testing 17
8.0 Compatibility Testing 18
8.1 Operating Systems 18
8.2 Productivity Software 18
8.3 Browsers 18
9.0 Communication 19

1.0 Introduction

1.1 Overview

ZenQA provides outsourced software QA services. This document outlines the processes we implement during various phases of the testing – Project inception, Test plan creation, Test execution, Test Results documentation & reporting, and Bug fix validation.

1.2 Types of Testing

Procedural overview for the following types of Black box testing:

1.2.1 Functional/Regression Testing: We develop test cases based on specific functional test requirements, client provided documentation and review of the existing product. We create manual and automated test scripts based on the test cases. We allocate dedicated test teams to execute these test scripts.

1.2.2 Compatibility Testing: ZenQA test lab has the infrastructure that allows testing products in a mix of operating systems, browsers, and productivity software of different versions (with or without service packs). Test environments are setup and verified on systems connected via independent LANs. Specific combinations of software products making up the test environments are determined by the product compatibility test requirements. Test scripts used for the functional testing are used to perform compatibility testing.

1.3 Automated Test Tools

ZenQA has prior expertise in the development and maintenance of automated test scripts using test tools from Mercury (Winrunner, QuickTest Professional, TestDirector), Rational (Robot, Test Manager) and Segue (SilkTest). Typically, mature and stable functions of the product are considered for automated testing during regression testing of the product.

1.4 Bug 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 JIRA, BugZilla, Mantis, TeamTrack and PR-Tracker. If the client has already been using a certain bug tracking tool, we will track bugs using the same tool.

2.0 Initial Contact

Clients initiate search for a QA partner by sending a 'Request for Information' (RFI) and/or a 'Request for Proposal' (RFP). We respond with relevant and specific information to every RFI received by us in a timely manner. In response to every RFP received, we submit a comprehensive proposal. If a client is new to outsourcing their QA efforts, and/or interested to test drive and gauge our abilities in test planning, test execution and bug discovery at no cost and no risk, we offer to pilot test a specific function of one of their products.

Request	Actions	Responsibility
RFI*	Solicit response with RFI document	Client
	Respond with specific information requested	ZenQA
Request for Pilot*	Select a specific functionality of the product to be tested and outline the test requirements	Client
	Sign NDA*	ZenQA
	Provide Product documentation (User Manual and/or Functional Specifications), a stable copy of the product and/or URLs, system usernames/passwords	Client
	Submit Pilot test plan document	ZenQA
	Review and Sign-off the Pilot test plan document	Client
	Create Users representing Client and ZenQA on web based bug tracking system	ZenQA
	Provide a Single point of contact	Client/ZenQA
	Pilot Test Execution	ZenQA
	Enter discovered bugs with description, severity level and reproduction steps in to the web based bug tracking system's database	ZenQA
	Submit Test Results Report with bugs/observations	ZenQA
	Provide clarifications/reproduce bugs on demand	ZenQA
	Provide feedback on the Pilot Test	Client
	Invite ZenQA to submit proposal against an RFP*	Client
RFP	Sign NDA*	ZenQA
	Provide specific test requirements	Client
	Submit Comprehensive Proposal	ZenQA
	Clarify questions/Resolve issues	Client/ZenQA
	Invite ZenQA to start on the project*	Client

*Optional – may or may not occur

2.1 Request for Information (RFI)

Some clients plan on short listing outsourced QA vendors based on a predetermined selection criteria. We often receive RFI from potential clients soliciting specific information about ZenQA. We respond with relevant information in a timely manner.

2.2 Request for Pilot Test

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 testing a specific function of one of their products. Pilot testing 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 ZenQA at no cost and no risk.
- Enables the client to evaluate the quality of the deliverables
- Enables the client and ZenQA to calibrate the expectations.
- Tests the communication processes and channels between the client and ZenQA

In response to the client's request for Pilot test, after the initial assessment, we submit a test plan with test cases to the client. The test plan will have information on the pilot test process, scope, schedule, deliverables and exactly what needs to be done at the client end 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.

2.3 Request for Proposal (RFP)

Proposals received in response to RFP will help the client compare the QA vendors against each other - in terms of testing abilities, test tools expertise, price and experience in executing in similar projects. In response to every RFP received, we first evaluate if we have the expertise and resources to successfully prepare and execute to meet the specific testing requirements. If so, we submit a comprehensive proposal, wherein the following information is detailed: scope, deliverables, assumptions & constraints, resource requirements, test strategy, pricing, roles&responsibilities, communication process and our past experience in executing similar projects.

3.0 Test Life Cycle

Chronological order of the project phases and listed responsibitlies of the client and ZenQA:

Phase	Actions	Responsibility
Test Plan	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/ZenQA
	Submit initial version of the Test Plan with test case index	ZenQA
	Discuss issues related to the Test Plan	Client/ZenQA
	Develop Manual/Automated Test Scripts (Detailed test cases)	ZenQA
	Verify the correctness of the Test Scripts	ZenQA
	Submit final version of the Test Plan with detailed test cases (test scripts)	ZenQA
	Review and Sign-off the Test Plan	Client
	Create Users representing Client and ZenQA on web based bug tracking system	ZenQA
Test Execution	Submit a specific build of the product for testing	Client
	Setup test environment(s) per test plan	ZenQA
	Execute the Test Scripts	ZenQA
	Enter information about the discovered bugs (with description, severity level and reproduction steps) in to the web based bug tracking system's database	ZenQA
	Update each test case with test result	ZenQA
Test Documentation	Prepare and Submit Test Results Report, Bug Reports and Observations made during the testing	ZenQA
	Provide clarifications/reproduce bugs on demand	ZenQA
Bug fix Validation	Provide a list of fixed bugs	Client
	Retest the fixed bugs	ZenQA
	Update the status of bugs in the bug tracking system's database	ZenQA
	Track the bugs until all bugs have been closed, rejected or postponed	ZenQA

*Optional – may or may not occur

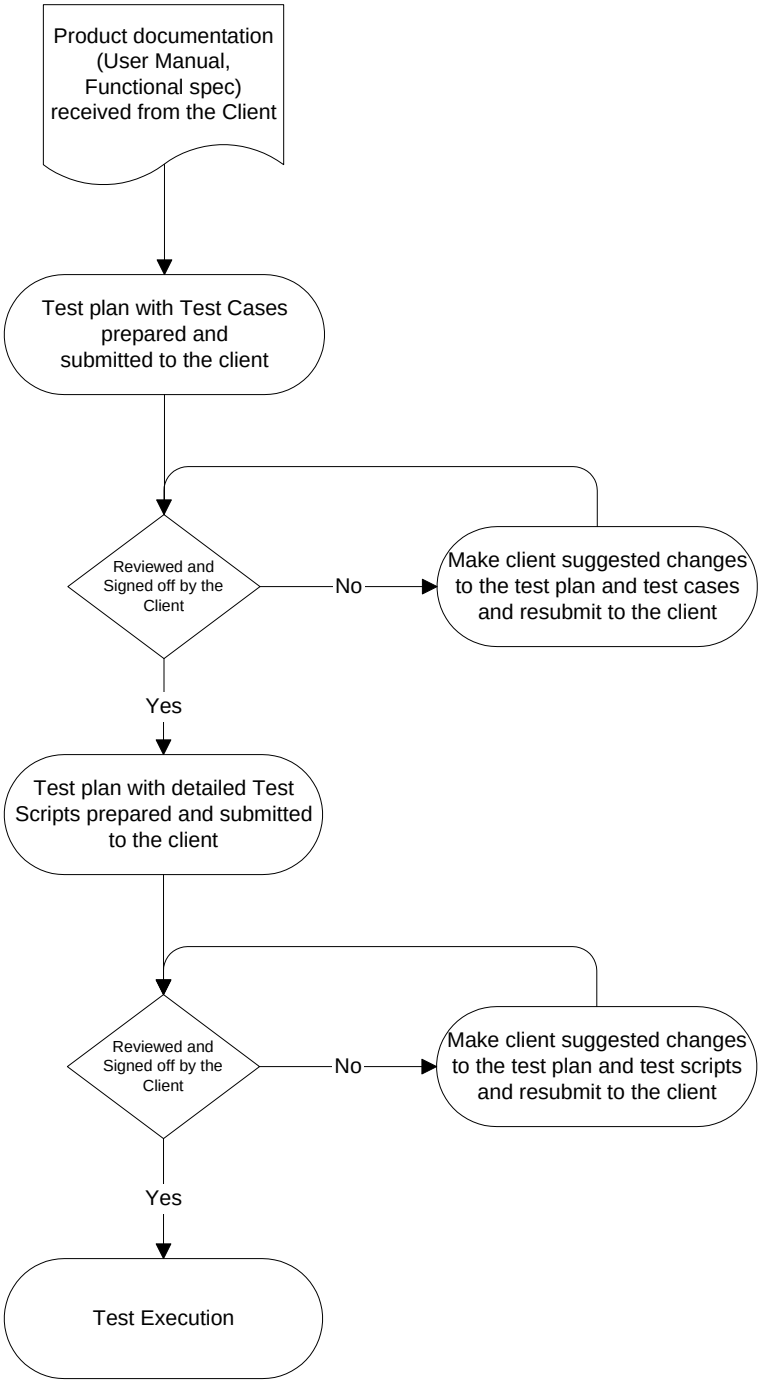
3.1 Formal Entry and Exit Criteria

We carry out 'smoke tests' (series of test cases that are run prior to commencing full-scale testing of an application) to test the high-level features of the application to ensure that the essential features work. For every product, we work with the development team and develop a set of test cases that constitute smoke tests. We accept the build of the product upon the product successfully passed all the smoke tests.

Similar to smoke tests, we run a pre-defined test cases (subset of detailed test cases) to determine if the application has passed the QA exit criteria and is ready to be deployed to a staging or production environment.

4.0 Test Plan

4.1 Process



4.2 Structure

We have a well defined structure for creating test plan that is designed to provide quick cross reference between functional areas tested and test cases:

1.0	Scope
2.0	Assumptions & Constraints
3.0	Test Requirements
4.0	Test Strategy
5.0	Functions Areas tested
6.0	Test Case Index
7.0	Traceability Matrix
8.0	Detailed Test Cases
9.0	Test Environment
10.0	Test Execution
11.0	Test Recording Procedures
12.0	Resources Requirement

4.2.1 Scope

Specific functional areas to be tested are listed defining the scope of the project. Functional areas that are not to be tested are mentioned as falling outside the scope. Test environments in which the product is to be tested are also listed.

4.2.2 Test Requirements

Use cases are created from the functional specifications document and the user manual. Use cases help in defining the test requirements. Test requirements with unique id (to trace from the traceability matrix) are listed in this section.

4.2.3 Test Strategy

Highest-risk functional areas are identified, and the test effort will focus on this functionality. Where testing can't be performed by using automated test tool (i.e if the automated test tool is not compatible with some of the functionality and no feasible automation work-around solutions can be found), such testing is performed manually.

4.2.4 Function Areas Tested

Function Area	Function Area Reference Id	Description
Import a document	FA1	Verify the functionality of locating and importing a document
Edit a document	FA2	Verify that the document attributes can be edited
Delete a document	FA3	Verify that a document can be deleted from the document list
View a document	FA4	Verify that a document can be viewed
Print a document	FA5	Verify that a document can be printed
Scan a document	FA6	Verify that a document can be scanned into the Document Manager
Sort Documents	FA7	Verify that the listed documents can be sorted by document type, receipt date, creation date, source, description, state, signoff and filename
Send for review	FA8	Send the document for review

4.2.5 Test Case Id Index

Function Area Reference Id	**Test Case Id**	**Test Case Description**
FA8	TC8.1	Send the document for review with no comparison and without cleaning metadata
	TC8.2	Send the document for review with comparison and without cleaning metadata
	TC8.3	Send the document for review with no comparison and with cleaning metadata
	TC8.4	Send the document for review with comparison and with cleaning metadata

4.2.6 Traceability Matrix

Test Requirement	**Functional Area**	**Test Case Id**
EDIT.TR1	FA2	TC2.1, TC2.2, TC 2.3, TC2.4
SFR.TR1	FA8	TC8.1, TC8.2, TC8.3, TC8.4

4.2.7 Detailed Test Cases

Test Case Id	Test Script	Test Result Status	Bug#
TC8.1	Send the document for review: 1. From the menu select send for review as a document with no comparison and not cleaning metadata (default settings), close doc. 2. Save sent document to desktop. 3. **Verify: that no metadata has been cleaned.** 4. Make changes and Return to Sender. 5. With the original document closed, import the response. **6. Verify: that the responder's name is populated in intelligent import dialog.** **7. Verify: that "Save email with response" is checked by default.** 8. From the File menu (using Send) send the original document for Review, browse the DMS/LFS for a comparison as PDF, clean metadata and enable full PDF security. 9. Save the sent word document to the desktop. **10. Verify: that metadata is cleaned from the document.** 11. Make changes, save and close. 12. Attach the word document saved to the desktop in step 9 to an email and send. 13. With the original document open, import as response. 14. Select send for Review, browse the DMS/LFS for a comparison as DOC and clean metadata. Add an additional document as DOC, include a comparison as PDF, clean metadata and enable full PDF security. **15. Verify: that metadata was cleaned from the original and additional document** 16. Make changes to both documents and Return to Sender. 17. With the original document closed, import the response, Change responder's name and uncheck save email with response. **18. Verify: that email is not attached to response and the email cannot be viewed from the action menu.** 19. Repeat step 17, for the additional document but import with "Save email with response" checked. **20. Verify: that email can be viewed from the "Action Menu" after the response is imported.**		

4.2.8 Test Environments

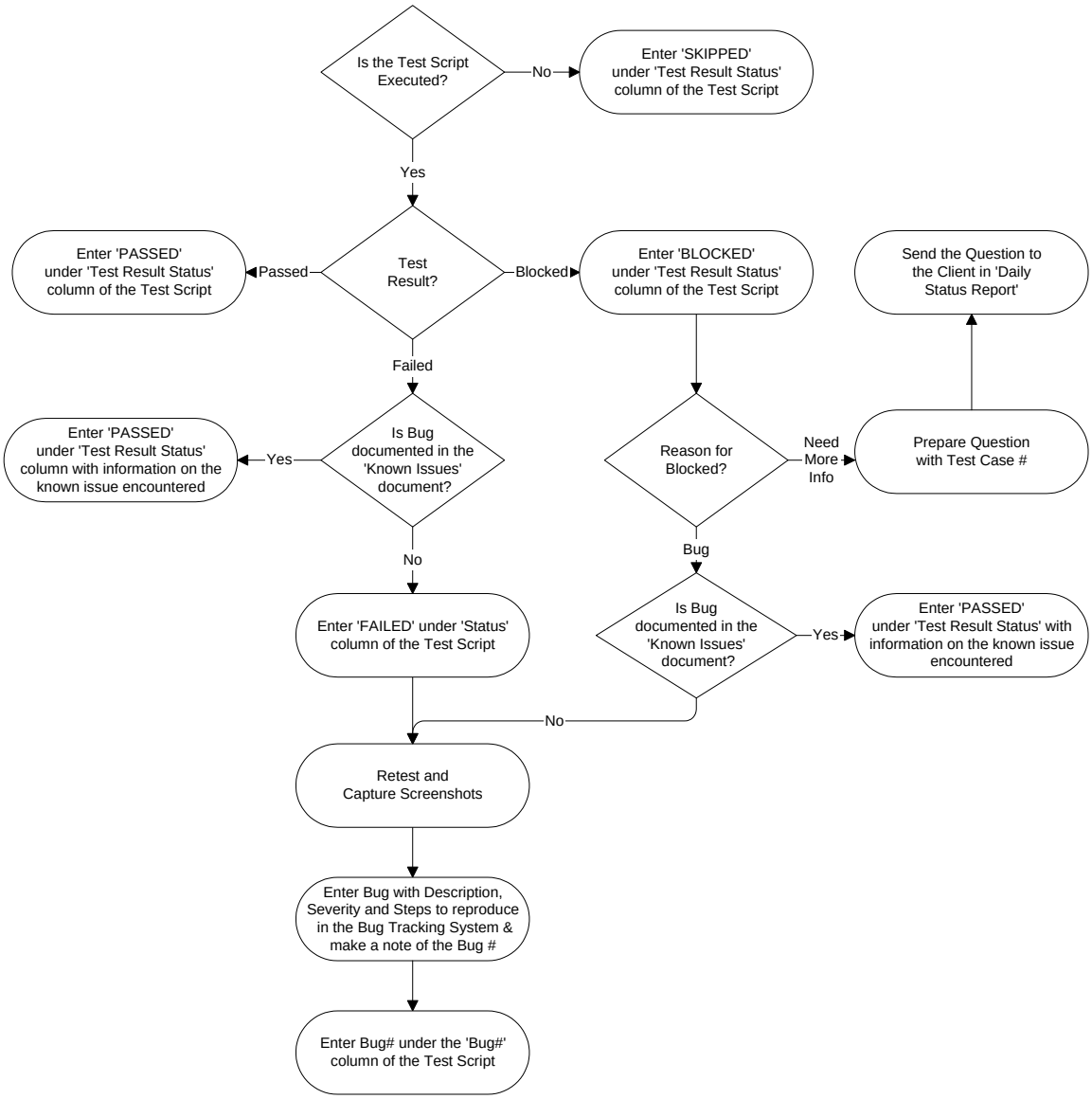
All the test environments in which the functional testing is to be performed are listed under this section.

DMS	Windows	Office	Outlook
Hummingbird DOCSOpen 3.9.6	2000 SP4 (2K)	2002 SP3 (XP)	2002 SP3 (XP)
Interwoven Desksite 6.5HF	2000 SP4 (2K)	2000 SP3 (2K)	2000 SP3 (2K)
Interwoven Mailsite 8.0 HF	2000 SP4 (2K)	2000 SP3 (2K)	2000 SP3 (2K)

5.0 Test Execution

5.1 Process

ZenQA Test engineers follow the process shown below during manual execution of the test scripts from top to bottom in the assigned test environment:



Each test script is updated with the test result (Passed, Failed, Blocked or Skipped) entered under ‘Test Result Status’ column. If a bug is discovered, it is entered in to the ‘Bug Tracking System’. The resulting bug number is entered under ‘Bug#’ column of the test script. A ‘Daily status report’ sent to the client that includes the following information: the number of tests executed, passed, failed, blocked, skipped and ‘clarification requests’ with test case numbers (for each test script that was blocked due to the tester requiring additional information from the client).

5.2 Bug reporting

ZenQA Test engineers follow a strict reporting process, providing the following information for each bug discovered in the Bug Tracking System:

- Status set to 'Open'
- Severity level
- Test Case Number
- Tester Name with contact info
- Test Environment (Browsers, OS, other software)
- Is it reproducible? – Always, occurs sometimes, occurs randomly, occurred only once, Not reproducible
- Steps to follow in order to reproduce the bug (including Login details, Landing page, Link/Selection Navigation, Issue identification)
- Screenshots captured depicting the bug discovery process
- Description with the following details -
    - Specific definition of the issue
    - Expected Result
    - Actual Result
    - How the issue impacts other screens and databases
    - Workarounds, if any

5.3 Severity Levels

One of the following four severity levels is specified for each bug reported -

Severity Level 1: Critical
Application is aborting.

Severity Level 2: High
A major area of the application is affected by the incident and it is significant to business process.

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. Problem does not impact use of product in any substantive way. Incidents those are cosmetic in nature, and of zero or very low impact to business process.

5.4 Bug Life Cycle

We implement the following bug tracking workflow, i.e the lifecycle of a bug or defect. If a Client has a different set of states to track the bugs, we will follow the client's bug tracking workflow.

Status	What it means
Open	When a bug is newly created, it's state is set to 'Open' by the Tester
Assigned	Bug has been assigned by the Client Development/Product Manager to a developer to fix the problem
Ready for Retest	Client Assigned Developer has fixed the problem. Wants to have the bug retested and closed
Closed	Tester has retested, found that the bug has been fixed and changed the status to 'Closed'
Rejected	Client Assigned Developer could not recreate the bug or found that the product behavior is as expected
Duplicate	Client Assigned Developer found that this bug is similar to another bug in the Bug Tracking System (which may or may not have been fixed)
Deferred	The Client Development team has decided to postpone fixing this bug to a future release or build
Reopened	After consulting the Client and Client approval, Tester has changed the bug status to 'Reopened' (from 'Ready for Retest', 'Closed', 'Rejected', 'Duplicate', or 'Deferred' state) due to a need to have this bug fixed now

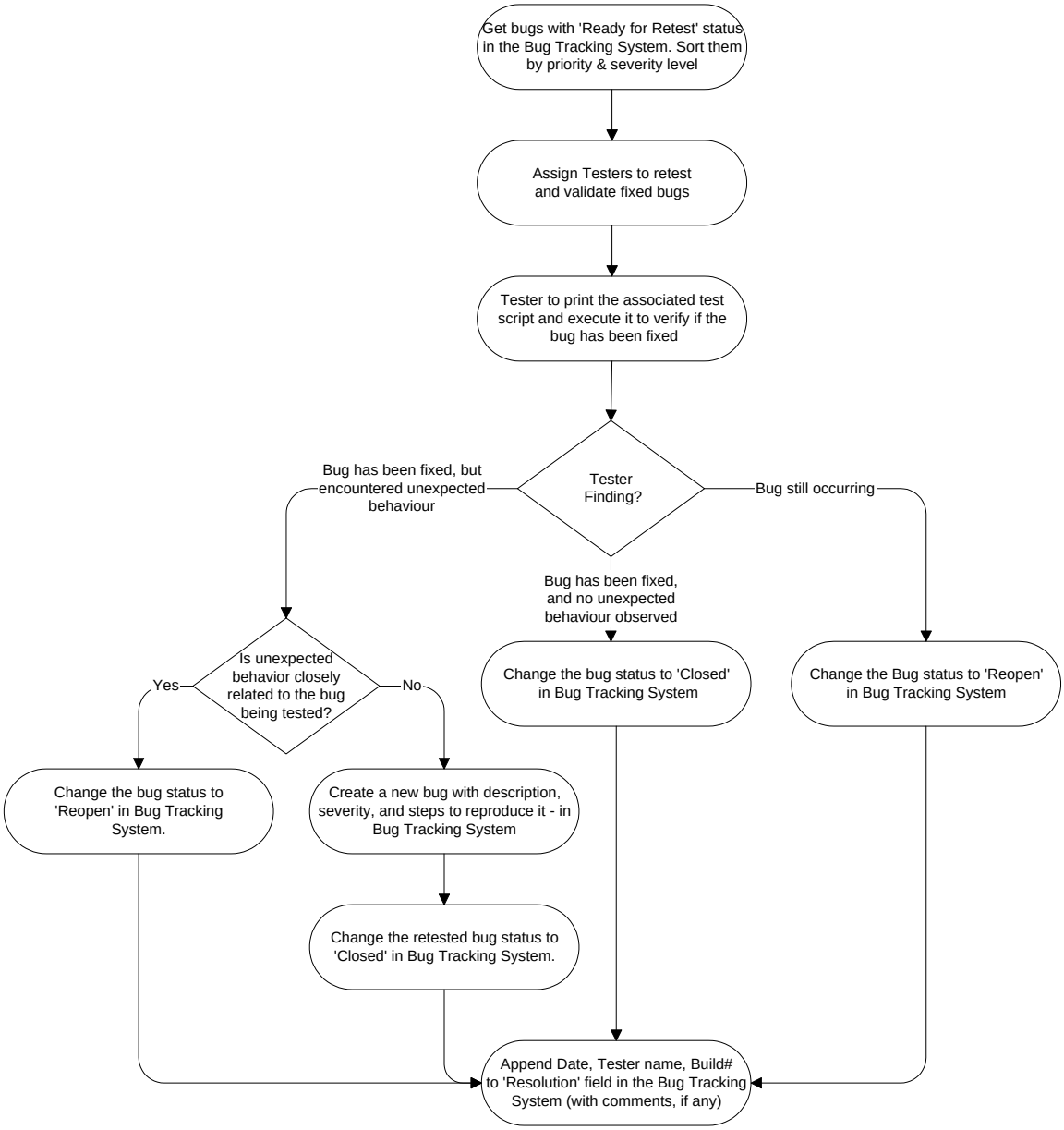
5.5 Bug discovery process over multi-cycle testing

While executing each test script, if one or more bugs are discovered, bug numbers (obtained from entering the bug information into the Bug Tracking System) are entered under the 'Bug#' column of the test script. Execution of each test script during each test cycle starts with empty 'Test Result Status' column and zero or more bug numbers under the 'Bug#' column, if any were found during previous test cycles.

At the end of the test script execution, 'Test Result Status' column is filled with the result outcome as outlined in section 5.1. If the execution of a test script results in discovery of a bug in the current test cycle, that bug is handled following the below outlined process:

5.6 Bug Fix Validation

ZenQA 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. It provides for updating the status of the bugs to '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 everyday and the summary is conveyed to the client as part of the 'Daily Status Report'.



6.0 Reporting

ZenQA provides real-time, daily, weekly and end-of-cycle test results reports to the client throughout the entire QA cycle.

6.1 Bug Tracking System

We enter new bugs, update the status and description of the existing bugs and close the bugs (found to have been fixed) in the web-based bug tracking system during the test cycle. Clients will be able to run queries and generate custom online reports to get information real-time.

6.2 Daily Status Reports

Daily status reports 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:

- Number of test cases executed
- Number of test cases passed, failed, blocked and skipped
- Number of bugs discovered sorted by severity level
- Observations made (product behavior 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
- New issues/questions
- Project current status document with information on the total number of test cases, number and percentage of test cases executed until today

And

- Test schedule for the following day

6.3 Weekly Status Reports

Weekly status reports are provided to the client to inform them of the testing carried during the week. It will contain the following information:

- Beginning of the week - Summary of bugs by functional area, bug status and severity level
- End of the week – Summary of bugs by functional area, severity level and bug status
- Observations made (product behavior related) during the week

And

- Test schedule for the following week

6.4 End-of-cycle Reports

At the end of the testing for a particular build, 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, blocked and skipped
- Number of bugs discovered by functional area, severity level and bug status
- Observations made (product behavior related) during the testing

7.0 Automated Testing

Creating automated tests is more time consuming and is more expensive than doing the test manually. ZenQA recommends automated testing when - testing mature and stable functional areas of the product, regression testing, compatibility testing and load testing. Automated tests produce their value after the code changes. When deciding whether to automate a test, we examine approximately how many code changes it will survive. Before recommending automating testing, we ask the following questions:

- Automating this test and running it only once will cost more than simply running it manually once. How much more?
- Automated test has a finite lifetime, during which it must recoup that additional cost. Is this test likely to die sooner or later due to code changes?
- During its lifetime, how likely is this test to find additional bugs (beyond whatever bugs it found during the first run)? How does this uncertain benefit balance against the cost of automation?

We convert existing manual test scripts, develop and update automated test scripts using Mercury's Winrunner, QuickTest Professional tools, Rational's Robot and Segue's Silktest. Based on the client's test requirements, the depth and breadth of utilizing automated testing will be set at the beginning of the project. As the QA project progresses and the product become more stable, there are additional areas where automated testing may prove beneficial. To facilitate unattended automated testing, we develop shell scripts. We use Mercury's TestDirector and Rational's TestManager to plan, run and manage the execution of automated tests.

8.0 Compatibility Testing

ZenQA can perform functional/regression testing on a variety of platforms, productivity software browsers and specific software combinations. Number of test environments, and the combination of software products in different test environments are dependent upon the client requirements.

8.1 Operating Systems

ZenQA maintains many operating systems, their versions and service packs to verify if a software product can work as expected on selected set of operating systems in combination with browsers and applications. Machine configurations will be adjusted in our QA lab to meet the system requirements of these operating systems:

1	Client OS: Windows 95, 98, ME, XP Home, XP Professional Server OS: Windows NT 3.51, 4.0, 2000, Windows 2003 server
2	Redhat Linux 8.0, 9.0, 10.0, Advance Server, Enterprise Linux
3	Solaris 2.7, 7, 8

8.2 Productivity Software

ZenQA can create test environments consisting of the following software products along with OS and browsers:

1	MS Office 97, 2000, 2003
2	Adobe Acrobat 4.0, 5.0, 6.0
3	Macromedia Flash 4.0, 5.0, MX 2004

8.3 Browsers

ZenQA can perform cross-browser compatibility tests to ensure that the website or web-based interface of the product will appear and behave as expected on these different browsers:

1	IE 5.0, 5.5, 6.0
2	Netscape 6.2, 7.0, 7.1, 7.2
3	Mozilla Firefox 1.0

9.0 Communication

We maintain communication channels between ZenQA 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 Yahoo! Messenger and MSN
- FTP downloadable files