System Test Plan for Vivo Meeting

Document History

Version Number	Author	Date	Description
0.1	Manish Khanna	15-09-2011	Initial version
0.2	Manish Khanna	19-09-2011	Second draft

Contents

Introduction

Purpose of Document

The document has been created to define the testing strategy and approach for Vivo product testing prior to release.

Product Overview

The product involves design and development of meeting management, thus, the scope of work would be:

- Understand requirements
- Design and develop the requirements
- Testing of developed requirements
- Provide maintenance and support services as per defined service level agreement

Product Scope

The scope of the product is not only limited to online meetings and desktop sharing. They provide all the features required for web conferencing, online meeting and desktop sharing on multiple platforms. While designing the Vivo Meeting product there are lots of features which have been kept in mind like compatibility on different platforms, sound quality and different devices used for testing. This makes Vivo a unique product into the market.

The product features include

- **Live or Streaming Video** – where full motion webcam, digital video camera or multi-media files are pushed to the audience.
- **VoIP** (Real time audio communication through the computer via use of headphones and speakers)
- **Meting Recording** – where presentation activity is recorded on the server side for later viewing and/or distribution.
- **Whiteboard with annotation** (allowing the presenter and/or attendees to highlight or mark items on the slide presentation. Or, simply make notes on a blank whiteboard.)
- **Text chats** – For live question and answer sessions, limited to the people connected to the meeting. Text chat may be public (echoed to all participants) or private (between 2 participants).
- **Polls and surveys** (allows the presenter to conduct questions with multiple choice answers directed to the audience)
- **Screen sharing / desktop sharing/application sharing** (where participants can view anything the presenter currently has shown on their screen. Screen sharing applications allow for remote desktop control, allowing participants to manipulate the presenters screen.)
- **File Sharing** – It is the practice of distributing or providing access to digitally stored information, such as computer programs, multimedia (audio, presentations, and video), documents, or electronic books.

- **Telephony** – This encompasses the general use of equipment to provide voice communication over distances, specifically by connecting to each other through telephones as well as over the internet through TCP/IP by transmitting digitized voice data.

  The product team is working on multiple platforms like Windows, MAC and Android to provide this solution.

Roles & responsibilities

The Roles & responsibilities have been mentioned here under:

Quality Assurance Lead

- Creating Test Plan
- QA Resource Allocation and Management
- Supervising Development of Test Cases and Scenarios
- Responsible for reviewing test cases
- Responsible for finding out the new methodologies of testing the product
- Responsible for conducting usability test.
- Establishing Bug Reporting Mechanism
- Status Reporting to PM and CSA

Test Engineer/Senior Test Engineer

- Responsible for creation of Test Scenarios
- Creating Test Cases
- Executing Test Cases
- Bug Reporting
- Status Reporting to QA Lead
- Preparation of test result

Test Environment

Client Configuration	
Software	**Hardware**
Microsoft Windows XP Professional SP 2 and above	Intel Pentium 4 Processor 3.06 GHz, 1 GB RAM or above, 80 GB HDD or above

Development Environment
Tools • Development Tool**: Visual Studio 2008 SP 1.** • Configuration Management Tool**: Tortoise SVN** • Project management Tool**: Microsoft Project Plan 2003** • Defect Tracking Tool**: Oracle Test Manager**
Others (Please Specify)

Platform	CPU	RAM	Hard disk	Testing tool	Any Other Tool
Windows XP Professional	Pentium IV	1 GB or above	80 GB or above	Oracle Test Manager (OTM)	Collabtive Project management tool
Windows Vista	Pentium IV	1 GB or above	80 GB or above	Oracle Test Manager (OTM)	Collabtive Project management tool
Windows 7	Pentium IV	1 GB or above	80 GB or above	Oracle Test Manager (OTM)	Collabtive Project management tool
Windows 2008	Pentium IV	1 GB or above	80 GB or above	Oracle Test Manager (OTM)	Collabtive Project management tool
MAC	Pentium IV	1 GB or above	80 GB or above	Oracle Test Manager (OTM)	Collabtive Project management tool
Android	-	-	-	-	-

Windows compatibility

The application is compatible with all windows versions that are equivalent or later than Windows XP.

MAC compatibility

The application is compatible with MAC version 10.3 or above.

Android compatibility

The application is compatible with Android.

Work Products

Documents for Review

Documents	Reviewed by
Product Schedule	Chief Software Architect Review
Functional Specifications	BD/Analysis team
Technical Specifications	Project/Technical Lead
Product Plan	Chief Software Architect Review
Test Strategy	PM review
Test Plan	PM review
Test Cases	QA Lead Review
Bug Report	QA Lead Review

Builds for Testing

Types of deployment	Reviewed by
Alpha-Deployment (internal release for testing purpose) on the test server	QA Lead review
Beta Deployment (external release for user suggestions) on the live server	QA Lead review

Gold Deployment (external release for actual usage of product by users) on live server	QA Lead review

Testing strategy

Functional Testing

Objective

- Test against system requirements.
- To confirm all the requirements are covered.

Approach

- Equivalence Class Partitioning
- Boundary Value Analysis
- Error Guessing

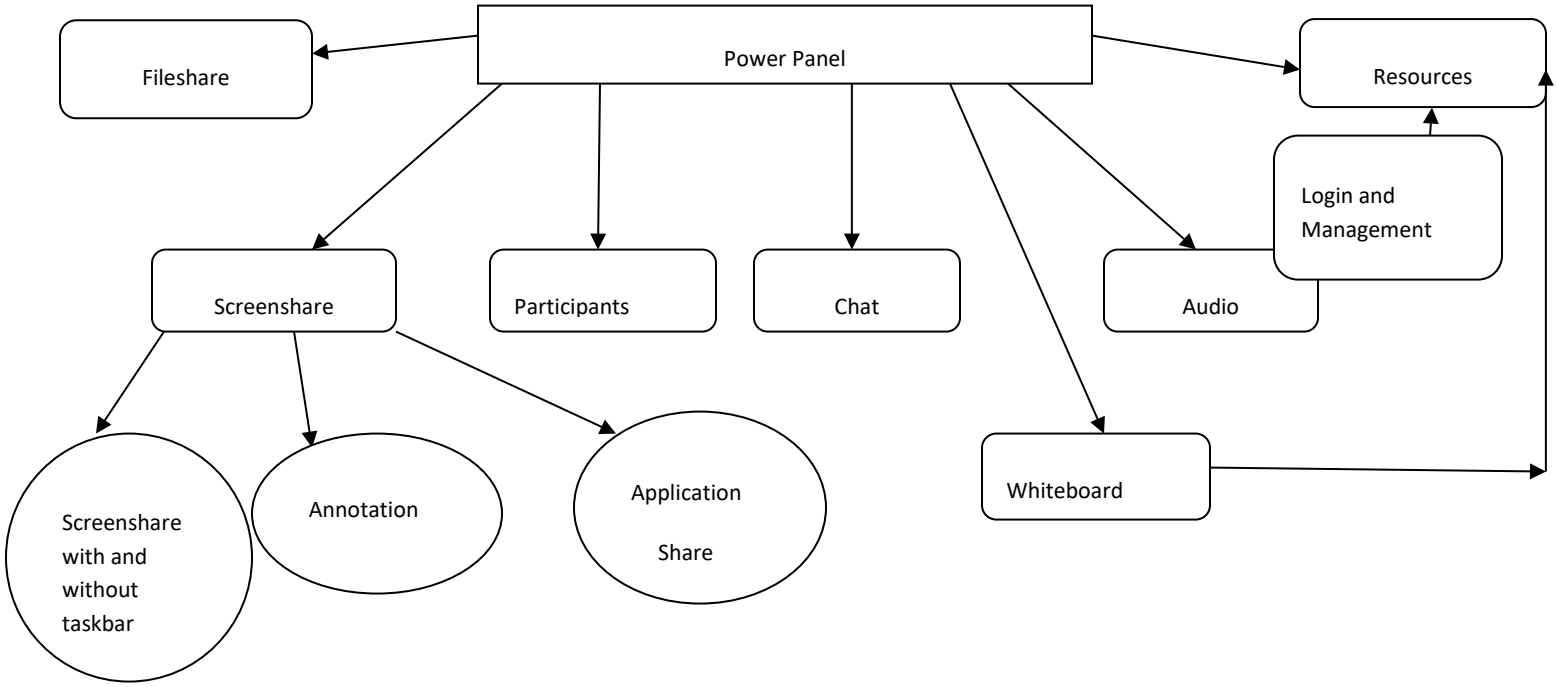
Entry Criteria for Functional Testing

Functional Entry Criteria	**Owner**
Unit tested Packages along with defects fixed	Development
Software release Note	Project Lead
Build Release Note	QA
Approved Test Plan	QA
Approved Functional Test Cases	QA
Test Environments established and built to specifications with all appropriate access	QA

Exit Criteria for Functional Testing

Functional Exit Criteria	**Owner**
Tested Software Build	QA
Test Report	QA
Updated Traceability Matrix	QA
Closure of reported defects	Project Lead

Integration Testing



Objective

- Test all integration points in the application to verify that data properly flows from one module to another

Approach

- Understand all integration points of the application
- Prepare integration test cases
- Execute integration test cases
- Report defects

Entry Criteria for Integration Testing

Integration Entry Criteria	**Owner**
Integrated tested Packages along with defects fixed	Development
Software release Note	Project Lead
Build Release Note	QA
Approved Test Plan	QA
Approved Integration Test Cases	QA
Test Environments established and built to specifications with all appropriate access	QA

Exit Criteria for Integration Testing

Integration Exit Criteria	Owner
Tested Software Build	QA
Test Report	QA
Updated Traceability Matrix	QA
Closure of reported defects	Project Lead

Regression Testing

Test Objective	To validate all functionalities are correctly working even after a major/minor enhancement has been introduced or a bug has been fixed
Technique	Functional/Regression/System Testing: 1) Retest the bug fixes 2) Test the functionality of any new enhancement that has been introduced 3) Execute the regression test cases for that module in which bug fixes have been made or enhancement has been introduced 4) Execute the regression test suite to test the stability of the application
Completion Criteria	1) All bug fixes have been done correctly 2) All open bugs have been closed with special focus on critical and major severity bugs 3) All regression test cases specific to a particular module have been executed and no new bugs are left open with special focus on critical and major severity bugs 4) Regression test suite has been executed and no new bugs are left open with special focus on critical and major severity bugs All planned tests have been executed. All identified defects have been addressed.
Special Considerations	None

Approach

- Understand the functionality
- Prepare test cases for that functionality
- Execute the test cases
- Report defects
- Retest fixed defects and close them accordingly

Entry Criteria for Regression Testing

Regression Entry Criteria	Owner
Functional tested Packages along with defects fixed	Development
Software release Note	Project Lead
Build Release Note	QA
Approved Test Plan	QA
Approved Regression Test Cases	QA
Test Environments established and built to specifications with all appropriate access	QA

Exit Criteria for Regression Testing

Regression Exit Criteria	Owner
Tested Software Build	QA
Test Report	QA
Updated Traceability Matrix	QA
Closure of reported defects	Project Lead

System Testing

Objective

- Test the whole application along with its integration with other applications like Arkadin's audio over telephone

Approach

- Understand the end to end functioning of Vivo application
- Understand the integration points of Vivo application with other applications
- Prepare system test cases and get those approved
- Execute system test cases
- Report defects

Entry Criteria for System Testing

System Test Entry Criteria	Owner
Integration tested Packages along with defects fixed	Development
Software release Note	Project Lead
Build Release Note	QA
Approved Test Plan	QA
Approved System Test Cases	QA
Test Environments established and built to specifications with all appropriate access	QA

Exit Criteria for System Testing

System Test Exit Criteria	Owner
Tested Software Build	QA
Test Report	QA
Updated Traceability Matrix	QA
Closure of reported defects	Project Lead

Testing Approach

Build Release

Build release would be shared with QA team along with all the details of functionalities released. This helps the QA team to focus on which all features need to be tested.

Testing

Testing of all the features listed in the build release would be tested as per the test cases created. All the bugs would be logged in the bug tracker with appropriate details so that they can be fixed.

Fixing

The development team will fix all the bugs logged in the bug tracker and change their status to retest, so that testing of the fixes can be done.

Regression

Regression testing of all the fixed bugs would be done to verify that all the features are functioning as per functional specification document.

Build Declaration

Declaration of all the known issues along with their bug list and stable modules would be specified in the build declaration.

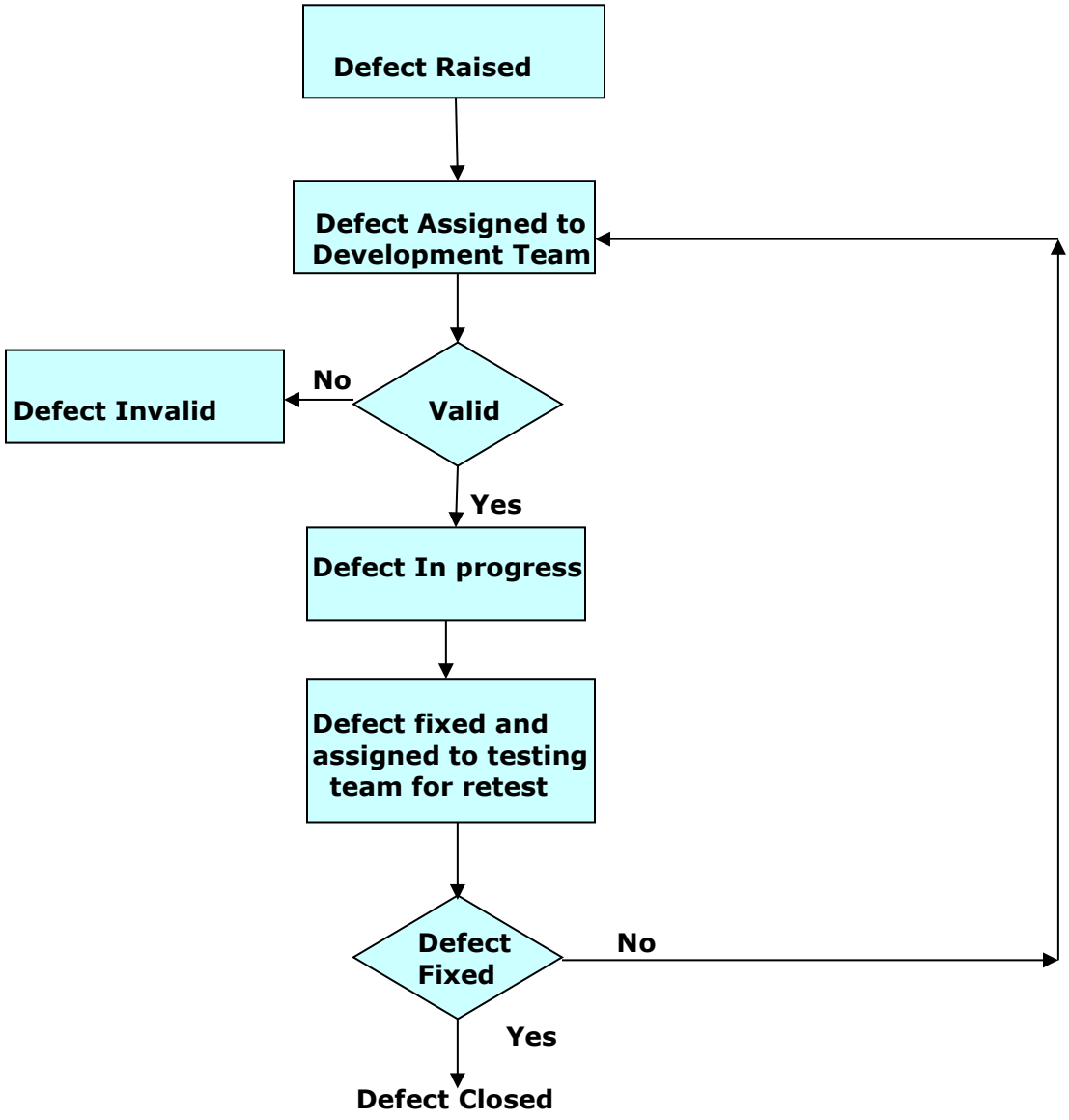
Bug Criteria/Definition

- Ensuring that the bugs that have been pointed out, are written in a simple language and are easily understandable by the developers.
- Before a module is tested, the QA member should have a clear understanding of what is to be tested. The functionality of the module and its relation with the other modules should be known, otherwise there would be cases when all the issues are not being logged or there could be situations where illogical errors are reported.

- Mentioning scenarios in case an error is reported so that the developer knows in what conditions the mentioned error occurs. This would also be helpful in saving the developer's time in finding the bug.
- Ensuring that the solution setup is done properly. All the necessary steps have been followed while setting up the project on the systems.

Defect Life Cycle

Defect Life Cycle helps in handling defects efficiently. This DLC will help the users to know the status of the defect.



Release Criteria

Test Pass/Fail Criteria

If the feature being tested works in accordance with the functionality described in the functional specification document then the feature would be deemed as passed the test criteria otherwise it would come under failed criteria.

Suspension Criteria

If the major functionality of the system which hinders the proper functioning of the system is encountered the testing process would be suspended.

Resumption Criteria

When the criteria due to which the system testing was suspended has been overcome.

Bug Analysis

Bug analysis would be done on the basis of the bug statistics.

Acceptance Criteria

The acceptance criteria of the Vivo product are as follows:

- Product must include all the features and customizations those are within scope.
- Product must be compliant with the performance criteria.
- Product must cater unlimited users with a concurrency of 1000 users.
- All the existing/requested features should function as said and demonstrated by G-Cube.
- Any deviation from the approved feature / functionality / customizations / performance will be taken care by G-Cube within a shortest time span.
- Any change in approved feature / functionality / customizations will be treated as change request, and should be catered through Change Management process.