COMPANY

QE Project Leader Guide
Quality Engineering Projects

SIGN-OFF SHEET

The following approvals are required to initiate the COMPANY NAME Quality Engineering Project Leader Guide document for all future quality engineering projects:

Approval Name:	Department	Approval Signature:	Date Approved:

Other Project Participants for Distribution Only:

Development Team
Quality Assurance Team
Project Managers
Program Managers

PREFACE

This is the Quality Engineering Project Leader Guide document.

Purpose

The purpose of this document is to provide a detailed guide to a Quality Lead Engineer on a project in which to plan, organize and execute testing to best ensure a quality/successful product launch.

Audience

This plan is intended for use by the Engineering- Quality Assurance and Release Engineering project teams to manage QE projects and to ensure that a quality product is developed before installing in a production client environment.

Revision History

Version	Date	Who	Revision
V1.0	1/03/01	FAL	
V1.1	3/21/01	FAL	Updated and added some things based on review. Also added more graphs and A Project Estimation Worksheet.

1.0 DESCRIPTION

This document is to be used by QA Project Leaders assigned to lead a project. It is a guideline for managing all QA projects. You may add to this document, as you lead this testing effort to its completion and it will become more useful for future QA Project Leaders.

Related Documents:

Statement of Work
Engineering Requirements Document
Functional Specifications
Page Schematics
Site Architecture
Creative Page Comps
Feature/Function Matrix
Technical Specifications
Quality Plan
Performance Plan
Test Cases
Test Results/Reporting
QA Summary

2.0 ACQUIRING PROJECT DOCUMENTATION AND TEST PLANNING

Every project assigned should have the proper documentation and project schedule in place. The Project Engineer is responsible for the distribution of project documentation and a preliminary schedule for the project. Estimating QA test time for the entire project should be a joint effort between the PE, PM and QA Project Lead. Refer to Appendix I: QA Project Estimation Worksheet.

The Lead QE should participate in the review of the initial Engineering requirements document, during the discovery phase. This happens prior to any functional specs being authored. The purpose of this review is to determine if the requirements are testable and clear enough to proceed to the writing of the functional specification and other project documentation.

When the functional specification is received from the Project Engineer, the QA person(s) will review the specification. This document is reviewed to determine, if the document has the information required to do test planning and writing other test documentation. Read the document thoroughly and provide your comments to the Project Engineer immediately for clarifications and missing information. Review other project documentation, such as, schematics, feature/function matrix, page comp, use cases, etc., to acquire additional knowledge for planning the QA project.

For writing the Quality Plan and other test documentation and creating graphs for Test Results refer to the Quality Engineering standards and templates stored in the SQA directory Methodology and Standards on Folsom. Also refer to the QE Workflow and Methodology document for additional information on the testing process.

Always review each other's work for standards and clarity. If someone else does not understand your work, it will be difficult for him or her to execute your test cases. Set a time for doing a walkthrough of test

documentation and make sure that the assigned Project Engineer and other interested project team members have reviewed your test documentation. There should be an on going peer review of all QA documentation, for consistency and adherence to QA standards, including QA Test Plans, QA Test Cases, Matrices, and other QA project documentation.

3.0 TEST STRATEGY OR TEST APPROACH

This is the most important piece of the Quality Plan. Check out other Quality Plan templates on folsom. This section is an overall approach to the strategy that will be used in testing the product. It will contain strategic things that need to happen during the testing effort (stress test, timing of downloading and uploading files, special data considerations, and specific test configurations). Discuss overall methodologies to be used, like black box, white box, manual vs. automated testing.

Don't forget issues that must be considered like resource planning. When the QE lead gets more information on a project, will resources need to be added or removed.

This section should include the approach for smoke testing builds and regression testing. Definitions for this terminology are described in the QE Workflow and Methodology document.

Make sure that you get a real understanding of the functionality of the product and its testing challenges and include it in this section.

Other Things To Consider:

1. Functional changes to areas that could impact the product across the board: Full or partial regression test may be required. . Also consider Engineering or SE/CE changes.
2. Scope "creep". The addition of new features to the project late in the development cycle.
3. QA schedule slips based upon Engineering schedule slips. Is the QA time being compressed, do you need more resources.
4. Performance: Some changes may impact performance. Pay attention to the timing of a transaction (Log Pages too slow, file downloading and/or uploading very slow). Try to do a small manual stress test (enlist your fellow QA members to assist with this).

Always save test results for each round of testing. This proves you tested what you said was tested. This is what the Test Case/Feature Test Log sheets are all about (% complete Test Cases/Features): Appendix B and C. Also document and save performance/timing logs, so that if a performance issue comes out, we are covered and can prove that we communicated this information to the right people. We do not just test and expect people to believe it.

4.0 TEST DOCUMENTATION STORAGE AND NUMBERING

Test documentation should be updated as changes occur in the project. This is a very important process and will allow the team to document areas that require additional testing, for example, client requirement changes.

The numbering system for QA documentation is very important as the QA test documentation grows. The numbering system is explained in Appendix F.

5.0 TEST RESULTS AND REPORTING

Test results consists of logging the execution of test cases, reporting defects and reporting the percent complete of the test cases/features exercised (Refer to Test Log and Feature Logs Appendix B and C and % Complete Test Cases or Features: Appendix D).

When the QA Team is unable to execute a full set of test cases, then adjustments must be made to our reporting process. In this case, report test results based on functional areas or specific features (See Feature Log: Appendix C). Unofficial high-level test scenarios may be executed (testing the flow or path of functionality, but not necessarily executing test cases). In this case report pass/fail based on features exercised.

A daily, weekly or by test cycle summary of defects and the trend of defects is important to show your team. The trend chart will show defect progress and show when the product is ready to release, once it has reached its peak. See Appendix E – Sample Defect Trend Graph. It should not matter which defect tracking system is being used, as long as the information is there for reporting results.

These samples are available on Folsom – SQA – Methodology and Standards – Defect Analysis and Statistics. They can be cut and pasted and text updated for modifications to suit your project.

These reports are essential tools for a QA Project Lead to communicate with other groups on the progress of the project. Use them at your discretion. It only makes us look better.

6.0 QUALITY ASSURANCE SUMMARY

Refer to Appendix H: QA Summary. This QA Summary is done for all projects. At the end of a cycle of testing a QA Summary should be written. A test cycle is a complete cycle of executing all the test cases or features of a project. If a complete cycle can not be executed, then only report on what can be executed. Supporting documentation should be referenced (Quality Plan, Test Cases) and supporting statistics attached (Defect Statistics, Current outstanding defects). See examples under Folsom: SQA/Methodology and Standards/QA Summary. It should be just one page.

7.0 TEST ASSIGNMENTS

Always develop a Feature Test Log or Test Case Log (Appendix B and C) to plan and test the product, with a Pass/Fail, defect and comments column. Then distribute this list to your team members. At the end of the day ask for % complete by test cases/features actually exercised. If you are the only team member and you need help, ask other QA members to assist you.

Coordinate the testing effort based on your Feature or Test Log. Have one member of the team do an Ad Hoc test of the new functionality. Always determine if the new changes will have an impact on current functionality. Make sure that you cover various operating systems, browsers and dial-up testing. Do a limited manual stress test with multiple

concurrent uses and record timing results. If required, do timing and metrics of various file types and the time it takes to do a transaction or manipulation of files. Make sure that you record this on some sort of logging sheet. This is to prove our testing efforts. Always store away this documentation. Keep your team, your manager, the Project Engineer informed of the results. Leave no surprises or open issues at the end of the testing effort. At every key issue/development/problem, the QE lead needs to constantly assess the impact to QA, overall, including planning, resource allocation, schedule impact, and other related issues.

8.0 DEFECT CONTROL PROCESS

Make sure that you have a clearly defined defect control process, in place, before the Production phase of the project. . Critical defects and showstoppers that are keeping you from making progress should be verbally communicated to the appropriate person immediately, then put it into the assigned tracking tool. Do not wait form someone to fix something. Go out and make it happen. In addition, send out a broadcast email to the project team, as part of the showstopper bug escalation process.

9.0 QA PROJECT MANAGEMENT AND PROJECT LEADER MEETINGS

The important thing on managing your QA project is to take charge of the quality process and keeping team members informed of critical areas and time to complete the project. If you are not going to make the date or the development effort starts to eat up QA time, then inform your manager immediately. Be firm, but somewhat flexible and creative in trying to get the job done.

Acquire project documentation as early as you can, so that you may start the planning process as soon as possible. If documentation is not available and the project schedule is slipping, then inform your manager. You need time to plan and make sure you demand this time.

When scheduling QA project time, you will usually get a target date, then work backward. There is nothing to keep you from testing earlier, if requested, especially when the code may not be 100% code complete. This type of testing allows for incremental builds and incremental testing of the client site. This will help the Developers identify problems early on. Normally QA starts when 100% code completion is achieved. Sometimes this is not always possible, but shoot for what you can get.

The project schedule should contain specific QA tasks (See Appendix A for an example):

For critical projects, Issue Council meetings should occur daily, during the execution phase of QA. For non-critical projects, meet 2 or 3 times weekly. I leave this to the discretion of the QA Project Lead. Like I said before, take charge and make these meetings happen. If the Project Engineer does not call a Issue Council meeting to review defects, then you should arrange it. If it is primarily a defect meeting, then you should lead it and point out critical areas. If the Project Engineer wants to lead that's okay, but make sure you get an opportunity to speak and cover QA areas. Just report the facts.

Always have an agenda ready. Use Meeting Maker to schedule a meeting and of course try to get a conference room or some meeting room to host the event. Bring Defect Trend Chart and outstanding defect list. The agenda should look like the following:

Agenda – Hosted by QA Project Lead or Project Engineer

Overview of defect meetings

a. Overview of Defect Trend Graph.
b. Provide statistics on % code complete and % test cases and/or features exercised.
c. Review new P1 and P2 defects.
d. Review the Project Schedule and note any date changes for QA.
e. Open Discussion on project issues.

10.0 PROJECT LAUNCH CHECK LIST SIGN-OFFS

The following document is probably the most important document a QA Project Leader will be responsible for. This is the document that will be used to get approvals for launching a build into a client production site. Read the checklist thoroughly and make sure everything is covered.

This checklist contains an Open Deferred List that must be reviewed. Selected defects will be identified for fixing in the Maintenance and Warranty phases of the project. This is the opportunity for the QA Team to submit what they think should be fixed and it must be signed.

Sample QA SOP - ENGINEERING PROJECT LAUNCH CHECKLIST
(Production Release - Launching Project Use Only)

Release Form

Quality Engineer Project Lead: _____

Build ID: _____ **Date:** _____

QA Launch Recommendation: [GO] [NO GO]
(QA Launch Recommendation will be based on the following Launch Criteria.)

✔/✗/NA	QA Launch Criteria	Comments
	Deliverables met Requirements/Specifications-Feature Matrix	
	Completed Test Set	
	All outstanding issues/defects are reviewed and found acceptable by the Project Team and client.	
	Open Deferred Defect List reviewed and selected defects approved for fixing during Maintenance/Warranty Phases	
	Received Creative Approval (see below) All Deliverables are in place.	
	Received Engineering Approvals (see below) All Deliverables are in place.	
	Received Client Acceptance Approval	

Engineering Sign-Off: *(Please indicates Title)*

Lead Interface Engineer: _____ Signature: _____ Dated: _____

Lead Software Engineer: _____ Signature: _____ Dated: _____

Lead System Engineer: _____ Signature: _____ Dated: _____

Lead Project Engineer: _____ Signature: _____ Dated: _____

Creative Sign-Off: *(Please indicates Title)*

Name: _____ Signature: _____ Dated: _____

Project Management Sign-Off: *(Please indicates Title)*

Name: _____ Signature: _____ Dated: _____

Outstanding Issues: *(Refer to Defect/Issues Open List)*

Appendix A – QA Project Scheduled Tasks

Some of these tasks should be considered in the overall project schedule created by the PE:

Parade

Requirements Review
QA Estimates, Schedule and Resource Planning
Project Documentation Review
Quality Plan
Automation Plan: Performance and Functional
Build Plan and Schedule
Write Test Cases
Performance Test Scripts Design
Client System Architecture Diagram Presentation to QA
Data Model Presentation to QA (SQL)
Create Test Environment and Data
First Code Delivery
RE Build – First Project Build
QA Smoke Test
Test Case Execution
Test Results and Reporting
Defect Review
Fix, Build and Release Notes
QA Summary
Execute Performance Test Scripts
Report Results
Review Performance Problems
Final Build and Release Notes
Final Test Execution and Regression Test
Final QA Summary and Reports
Ship To Client Site
Warranty: Fix, Build, Test and Report
Maintenance: Fix, Build, Test and Report
Project Postmortem

NOTE: Tasks in RED are an iterative process and test cases will be recycled through this entire process for each iteration of testing. One iteration of testing could contain many builds. One full cycle of testing is a complete execution of all the possible test cases of features for the current available functionality.

Appendix B – Test Case Log

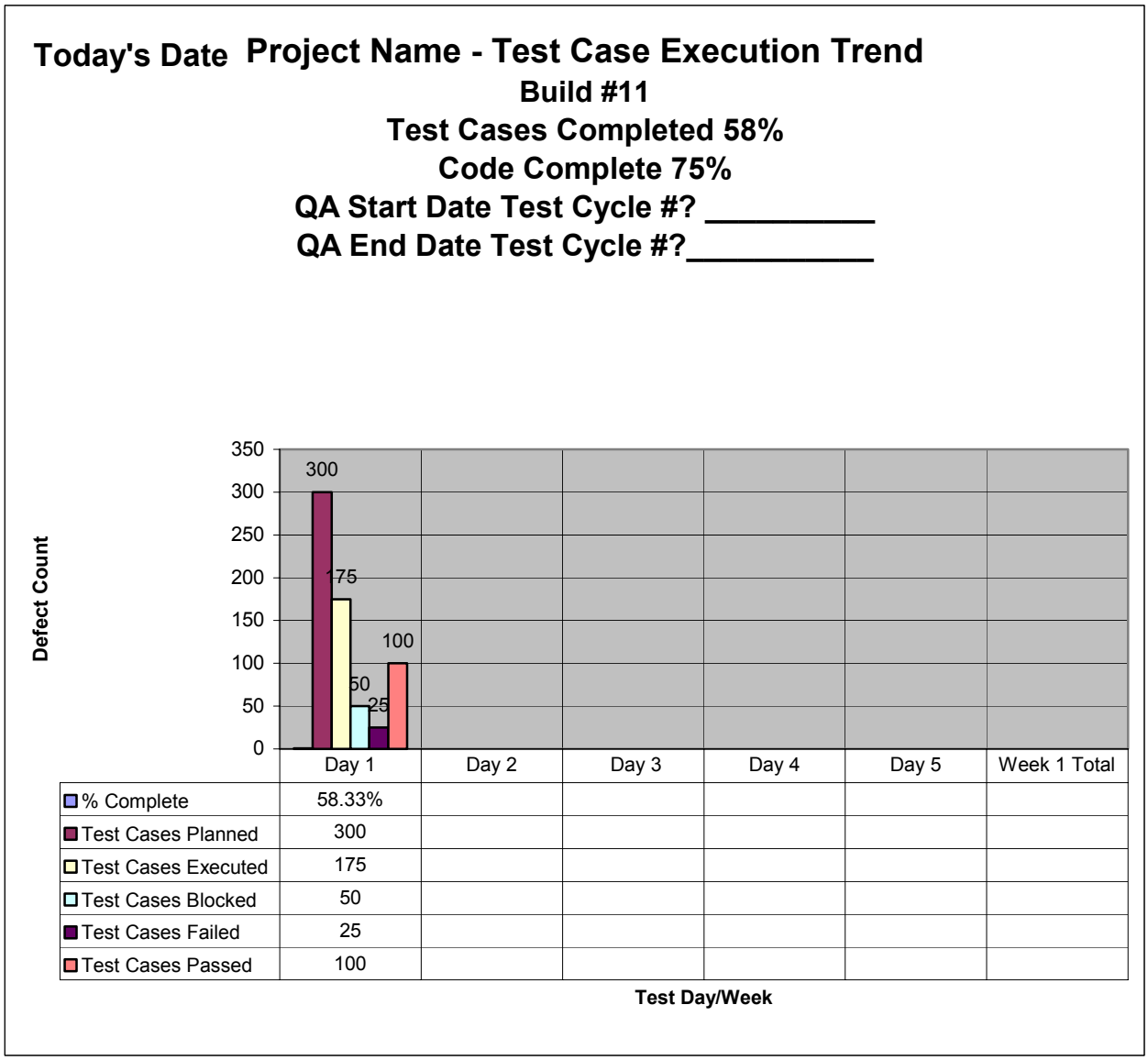
Test Engineer(s)	Start/Finish Date	Test Round	% Complete	Project
			Total TC: Number Passed: Number Failed: Number Stopped: % Complete:	
OS:	**BROWSER and VERSION:**	**Build Number:**	**Defects Found:** **P1:** **P2:** **P3:** **P4:** **Total Open:** **Total Closed:** **Grand Total:**	**Test Type:** Functional Regression Performance/Stress Metrics/Timing **NOTE**: Electronically File test results
Test Case ID	**Functional Area**	**Pass/Fail**	**Defect Number**	**Comments**

Appendix C – Feature Test Log

ID	FEATURE	PASS	FAIL	% FEATURES COMPLETE	Total Number of Test Cases	Number of Test Cases Executed	% TC Complete	RESOURCE	CONFIGURATION	COMMENTS
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										

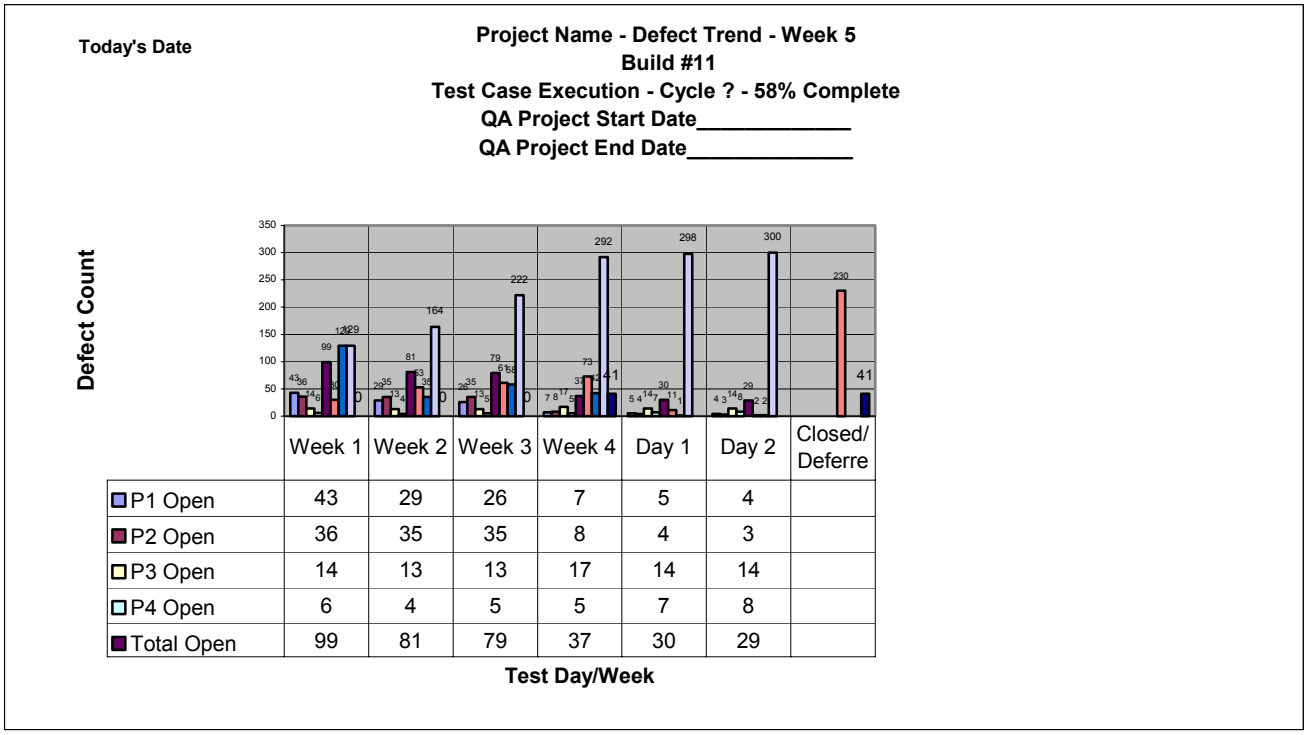
Appendix D – % Complete Test Cases

	Day 1	Day 2	Day 3	Day 4	Day 5	Week 1 Total
% Complete	58.33%					
Test Cases Planned	300					
Test Cases Executed	175					
Test Cases Blocked	50					
Test Cases Failed	25					
Test Cases Passed	100					



Appendix E – Sample Defect Trend Graph

	Week 1	Week 2	Week 3	Week 4	Day 1	Day 2	Closed/Deferred
P1 Open	43	29	26	7	5	4	
P2 Open	36	35	35	8	4	3	
P3 Open	14	13	13	17	14	14	
P4 Open	6	4	5	5	7	8	
Total Open	99	81	79	37	30	29	
Closed Today	30	53	61	73	11		230
Opened Today	129	35	58	42	1		
Total Found	129	164	222	292	298		
Deferred	0	0	0	41			41



Appendix F – Documentation Numbering System

The purpose of this numbering standard is to prepare for the growth for the number of test cases being created for the COMPANY NAME products quality assurance program. Sometime, in the future, QA will be acquiring a data base test case management tool, such as Test Diector. QA will use this tool to manage the growing number of test cases QA will write.

In addition, numbering QA test cases will facilitate managing results of executing these test cases. A test log will be created and the numbering system will be used to manage % complete and pass/fail results. If you have a problem coming up with a number, see your Manager, also make sure you do not duplicate the numbering system. Make adjustments where necessary.

Quality Plan Numbering System

1. PQP001.00 – P = Parade Project, QP = Quality Plan, 001 = Quality Plan number, .00 = versioning number when making updates to the Quality Plan. If project names are duplicates, then add another letter

Test Case Numbering System

A variation is using the same set of steps in a Test Case, but the set-up may be different. Sometimes it does not make sense to create an entire new Test Case, just to number a new variation.Use your best judgment and keep it simple, but follow this standard. If you are going to deviate from the standard, then you need to see your Manager.

1. PTC001X.00 – P = Parade Project, TC = Test Case, 001 = Test Case number, X=variation of the same Test Case being executed, but just a different variation (lower case a-z), .00= versioning number when making updates to Test Cases. If you do not have a variation for the Test Case do not use X.

Other Examples with P = Parade project. Change project name.

Automated Test Scripts Numbering System: PATS001X.00

Automation Plan Numbering System: AP001.00

Performance Plan Numbering System: PPP001.00

Project Leader Guide Numbering System: PLG001.00

Client Architecture Diagram Numbering System: PCAD001.00

QA Summary Numbering System: PQAS001.00

Test Case Logs Numbering System: PTCL001.00

NOTE: You may also want to consider numbering your test cases based on the functional specification and create a Functional/Test Case Matrix (mapping test cases to the functional specification). Mapping this will give the client a comfort feeling that what they requested (based on requirements in the functional specification) was in fact delivered. This avoids having to go back and re-test everything to verify that all the functions defined in the Functional Specification were executed and delivered. The important thing is to number your documentation, so that when you use the Test or Feature Logs, some level of organization is apparent and this will facilitate communication of problem areas.

Appendix G – Defect Definitions – Priority and Severity

PRIORITY DEFINTIONS: This is the priority in which defects will get fixed.

Priority 1 (P1): This is an **Emergency** Priority defect. This problem should be fixed IMMEDIATELY. Criteria should be:

No Workaround
Complete stoppage of workflow
Project is down
Loss of an account
Production is down

Priority 2 (P2): This is a **High** Priority defect. This defect is blocking other testing (Project related). This is a functional failure (Production or Project related) too severe that most or all users or other components of the product are seriously affected. There may be a workaround, but it is unacceptable for this release or the customer.
Priority 3 (P3): This is a **Medium** Priority defect. This defect is not serious enough to hold up a release (Project related) or to be fixed immediately in Production. This problem can be fixed after all P1 and P2 fixes are addressed. There is an acceptable workaround for this problem. This defect may be scheduled to be fixed, as part of a future major release.
Priority 4 (P4): This is a **Low** Priority defect. This defect is a problem, but it is low in priority for getting fixed and will only get fixed when other serious defects are fixed first or will be scheduled as part of a future major release.

NOTE: All Priority 1 and 2 defects, regardless of Severity, must be fixed prior to shipping a release into production (Project related).

SEVERITY DEFINITION: Severity defines the seriousness of the problem and not when it will get fixed. Severity also provides us a means of categorizing defects and determining where our application problems reside. A defect can have a Severity 6 (S6), but because the cosmetic defect is serious enough to the functionality of the product or it's aesthetic value, it could have a Priority 1 (P1) assigned to ensure that it gets fixed immediately.

Severity 1 (S1) Crash/Connectivity: The severity of this defect is a "Mission Critical" defect. It means to stop everything and work on this defect. This is always a P1 Priority. Criteria should be:

1. Client Site Down
2. Feature Down
3. Hardware Down
4. Application is Down
5. System Crash

Severity 2 (S2) Performance: This is a defect related to the functional, application and architectural performance of the product. Currently performance is measured based on what would be acceptable to the customer. For example, performance should be measured on how long it takes to complete a single functional task. Performance problems could also be the result of exercising the function or the application with unusual heavy loads of data and with or without simultaneous users on the system.
Severity 3 (S3) Data Loss: This defect is seriously affecting customer data and the customer is unable to complete the work started.
Severity 4 (S4) Incorrect Behavior: This defect is primarily a functional problem. The function does not work as specified. This could also be a design flaw. Meaning that the function as specified was misinterpreted or that it was specified incorrectly.
Severity 5 (S5) Security Flaw: This defect is a breach of functional, application or system security. An application breach could be that a user is able to access the client application without going through the login process. An example of a functional breach is the user is able to by-pass an information lockout, meant strictly for a particular user. An example of a system breach of security is that a user is able to get to the client domain and by-pass the firewall.
Severity 6 (S6) Cosmetic Problem: This is a defect that is not a functional problem, but it is a cosmetic fix that could or could not have serious repercussions, depending on how it looks. This problem could be a P1 for serious cosmetic defects that are unacceptable to customer or to product aesthetics.

Appendix H – QA Summary

Date:	**QA Engineer:**	**Release:**	**Project and Build Number:**
Start Date: **Author:**	**End Date:**	**Total Time:**	**Type of Test:**

Test Strategy: See Quality Plan and QE Workflow and Methodology Documents

Test Summary:

Outstanding Defects/Issues: See Attached – Open Defects/Issues Report

Release Recommendation:

Yes	**No**	**With Reservation**

Comments:

Appendix I – QA Estimation Worksheet

Task ID	INXS TASKS	Estimated Hours	Allocated Hours	Actual Hours	Done (X)	Start Date	End Date	Resource(s)	Comment
	Discovery								
1.0	Review Documents	20	15					QE(20) QE(10)	
2.0	Attend Meetings	10	5					QE (10)	
	Planning								
	Review Final Documentation	**40**	30					QE (20) QE (20)	
3.0	Client Requirements								
4.0	Project Plan/Schedule								
5.0	Functional Specificiation								
6.0	Feature/Function Matrix								
7.0	Architectural Page Flow								
8.0	Page Comp								
9.0	Schematics								
10.0	Use Cases/UML								
11.0	Client System Architecture								
12.0	**Write QA Test Plan**	40	30					QE(40)	
13.0	**Integrated QA Schedule Input**	**10**	8					QE (5) Director (5)	
14.0	**Write Feature/Function Test Log (QA Report Log)**	**10**	8					QE (10)	This is used for Test Log % Complete and QA assignments. Who is doing what?
15.0	**Prepare Reporting Tools**	**20**	15					QE (20)	
16.0	Defect Tracking System								
17.0	Graphs/Reports								
18.0	**Request QA Box**	**1**	1					QE (1)	
19.0	**Write Test Cases/Scenarios/Matrices**	**80**	75					QE (40) QE (40)	
20.0	**Review QA Project Documentation**	**8**	6					QE(4) QE (4)	
21.0	**Prepare Test Data for QA Box**	**8**	6					QE (4) QE (4)	
	Production								
	Design Test Cases								
22.0	Canada Order (6.1)	24	18					QE (24)	
23.0	Discounts (6.2)	24	18					QE (24)	
24.0	DataBase Fields (6.7)	16	10					QE (16)	
25.0	Code Archive (6.3)	24	15					QE (12 QE (12)	
26.0	Sorting (6.4)	24	15					QE (12) QE (12)	
27.0	News Feed (6.6)	8	4					QE (8)	
28.0	User Password Reset after CSR (6.5)	8	4					QE (8)	
29.0	Inventory Synchs (6.8)	8	4					QE (8)	
30.0	Additional Content (6.9)	24	18					QE (24)	
31.0	Redesign Registration (6.10)	16	8					QE (16)	
32.0	Additional Subcats (6.11)	16	8					QE (16)	
33.0	Left Navigation Subcategory Listing (6.12)	8	4					QE (8)	
34.0	Test Case Review	8	6					QE (4) QE (4)	
	Execute Test Cases								
35.0	Canada Order (6.1)	8	8					QE (8)	
36.0	Discounts (6.2)	8	6					QE (8)	
37.0	DataBase Fields (6.7)	4	4					QE (4)	
38.0	Code Archive (6.3)	4	4					QE (2) QE (2)	
39.0	Sorting (6.4)	4	4					QE (2) QE (2)	
40.0	News Feed (6.6)	2	2					QE (2)	
41.0	User Password Reset after CSR (6.5)	2	2					QE (2)	
42.0	Inventory Synchs (6.8)	2	2					QE (2)	
43.0	Additional Content (6.9)	8	8					QE (8)	
44.0	Redesign Registration (6.10)	4	4					QE (4)	
45.0	Additional Subcats (6.11)	2	2					QE 2)	
46.0	Left Navigation Subcategory Listing (6.12)	2	2					QE (2)	
47.0	Test Case Update/Review	8	8					QE (4) QE (4)	
48.0	**Reporting/ Defect Tracking**	**50**	20					QE (25) QE(25)	We need to track and report pwc bugs as well.
49.0	**Meetings**	30	?						
	Launch								
50.0	Alpha Test Cycle	110	96					QE(55) QE (55)	
51.0	Beta Test Cycle	110	96					QE(55) QE (55)	
47.0	**Post Launch**								
48.0	Test Case or Scenarios Execution/Reporting	55	55					QE (27) QE (27)	
49.0	Cycle 1/QA Summaries and Graphs								
50.0	Cycle 2/QA Summaries and Graphs								
51.0	Cycle 3/QA Summaries and Graphs								
52.0	Test Case Update/Review	16	16					QE (8) QE (8)	
53.0	**Warranty**								
54.0	Test Case or Scenarios Execution/Reporting	55	55					QE (27) QE (27)	
55.0	Cycle 1/QA Summaries and Graphs								
56.0	Cycle 2/QA Summaries and Graphs								
57.0	Cycle 3/QA Summaries and Graphs								
58.0	Test Case Update/Review	16	16					QE (8) QE (8)	
59.0	**Maintenance**								
60.0	Test Case or Scenarios Execution/Reporting	25						QE(24)	
61.0	Cycle 1/QA Summaries and Graphs								
62.0	Cycle 2/QA Summaries and Graphs								
63.0	Cycle 3/QA Summaries and Graphs								
64.0	Test Case Update/Review	4						QE (4)	
	TOTAL Hours Estimated/Allocated	984	741						