|  |  |
| --- | --- |
| Product Name | IRIS Exchequer |
| Proposed Version | v6.3 |

25/03/2009

**High Level Design**

**Mark Higginson**

**Version**

# Revision and Signoff Sheet

## Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Position | Version Approved | Date |
| Kevin Horlock | Sat Down |  |  |
| David Rustell | QA Manager |  |  |
| James Waygood | QA Executive / WebRel Administrator |  |  |
| Training / Tech Support????? |  |  |  |

Contents

[Revision and Signoff Sheet 2](#_Toc234917468)

[Revision 2](#_Toc234917469)

[Reviewers 2](#_Toc234917470)

[Purpose of Document 5](#_Toc234917471)

[SDLC Checkpoints That Must Be Covered 5](#_Toc234917472)

[SDLC Checkpoints Currently Covered 5](#_Toc234917473)

[HLD Instructions 6](#_Toc234917474)

[Marketing Requirements Scope 7](#_Toc234917475)

[User Profiles 7](#_Toc234917476)

[Architecture 7](#_Toc234917477)

[Deployment Analysis 7](#_Toc234917478)

[System / Platform Analysis 8](#_Toc234917479)

[Coding Standards 9](#_Toc234917480)

[Source Control Procedures 9](#_Toc234917481)

[Requirements 10](#_Toc234917482)

[Requirement < IBIF\_EXL\_RQ01> – Requirement 1 10](#_Toc234917483)

[Estimates 10](#_Toc234917484)

[Requirement < IBIF\_EXL\_RQ02> – Requirement 2 11](#_Toc234917485)

[Estimates 11](#_Toc234917486)

[Iteration Design 12](#_Toc234917487)

[Proposed Development Machine Specification 13](#_Toc234917488)

[Proposed Build Machine Specification 13](#_Toc234917489)

[Proposed Test Machine Environments 14](#_Toc234917490)

[Revised Gate Dates 14](#_Toc234917491)

[Appendix 1 – Technical Design 15](#_Toc234917492)

[Requirement < IBIF\_EXL\_RQ01> – Technical design 15](#_Toc234917493)

[Technical High Level Design 15](#_Toc234917494)

[Requirement < IBIF\_EXL\_RQ02> – Technical design 16](#_Toc234917495)

[Technical High Level Design 16](#_Toc234917496)

[Appendix 2 - Test Approach 17](#_Toc234917497)

[Test Plan Outline 17](#_Toc234917498)

[References 17](#_Toc234917499)

[Test Items (Functions) 17](#_Toc234917500)

[Features to be Tested 17](#_Toc234917501)

[Features not to be Tested 18](#_Toc234917502)

[Approach (Strategy) 19](#_Toc234917503)

[Pass Strategy 19](#_Toc234917504)

[Testing Tools 19](#_Toc234917505)

[Test Development Tools 19](#_Toc234917506)

[Test Execution Tools 19](#_Toc234917507)

[Testing Metrics 19](#_Toc234917508)

[Metrics Collected During Test Development 19](#_Toc234917509)

[Metrics Collected During Test Execution 20](#_Toc234917510)

[Configuration Management 21](#_Toc234917511)

[Test Configurations 21](#_Toc234917512)

[Regression Testing 22](#_Toc234917513)

[Automated Regression Testing 22](#_Toc234917514)

[Manual Regression Testing 22](#_Toc234917515)

[Status Reporting 23](#_Toc234917516)

[Item Pass/Fail Criteria 23](#_Toc234917517)

[Test Deliverables 23](#_Toc234917518)

[Test Environment 24](#_Toc234917519)

[Staffing and Training Needs 24](#_Toc234917520)

[Roles and Responsibilities 24](#_Toc234917521)

[Schedule 26](#_Toc234917522)

[Planning Risks, Contingencies and Assumptions 27](#_Toc234917523)

# Purpose of Document

The purpose of the High level Design (HLD)

1. To design each requirement to the extent that
   1. We are confident with the estimates contained in the document
   2. We provide enough information for testers to produce test scripts
   3. All requirements are fully understood by development and test teams
2. To provide an agreed level of testing in advance of building the product
3. That requirements are designed to such a level that the target completion date can be achieved.

# SDLC Checkpoints That Must Be Covered

The High Level Design (HLD) document is part of the planning stage of the project, the planning stage of the project seeks to pass the following check points. The checkpoints that should be contained in this document are indicated in the first part of the table.

|  |  |  |
| --- | --- | --- |
| Id | Description | In Document |
| P1 | Have High Level Design Documents been completed for all requirements? | High Level Design |
| P2 | Have detailed work estimates been produced? | High Level Design |
| P3 | Has an approved test plan been produced for the release? | High Level Design |
| P4 | Has build frequency been defined and communicated? | High Level Design |
| P5 | Have coding standards been reviewed and published? | High Level Design |
| P6 | Have backup and recovery processes been reviewed and published? | High Level Design |
| P7 | Have source control procedures been reviewed and published? | High Level Design |
| P8 | Are separate build and test environments in place? | Detailed in High Level Design |
| P9 | Has the process for transferring builds to test team/test environment been reviewed and published? | High Level Design |
| P10 | Is a base lined, detailed project plan in place covering all Development Team release activity? | Project Plan |
| P11 | Has a revised target completion date been approved by Product Management and published? | Project Plan |

# SDLC Checkpoints Currently Covered

This section should include the checkpoints from the previous table that are currently satisfied in the current version of this document.

# HLD Instructions

*This document should use the Product Requirements Document (PRD) as its input. Each requirement from the PRD should*

* *Have a technical Analysis Performed on it*
* *Have a test plan (per requirement) completed for it.*
* *Refine the estimates when the high level design is completed (which will be used to create a base lined project plan)*

*The Architect should define the overall architecture, and perform a high level analysis of the deployment considerations*

*The development team should produce a set of agreed coding standard (or point to a set of already published ones) and indicate which platforms will be supported for the release.*

*The test should define the test environments*

*All the development team should define the iteration plan (i.e. which features will be delivered in which iteration)*

# Marketing Requirements Scope

*This section should give a brief overview of the requirements that the HLD will address, it should also point to published requirements list e.g. a SharePoint list or reproduce the agreed requirements from the PRD (In MoSCoW format)*

*Link to copy of PRD*

# User Profiles

*Profiles and personas of typical users can be found in the Marketing Requirements Document for this project.*

*Link to copy of MRD*

# Architecture

IRIS Exchequer is a 32-bit windows client application using either the Pervasive.SQL database engine or Microsoft SQL Server to store the data. IRIS Exchequer is typically installed into a central network folder and client workstations are configured to run it from there.



The dual database support is largely implemented in a layer below the Exchequer components to maximize commonality of code, however to improve performance under the Microsoft SQL Edition direct calls to the SQL Emulator and direct calls to SQL Server using ADO are performed where necessary.



## Changes for this release

??????

# Deployment Analysis

IRIS Exchequer is installed and upgraded using the IRIS Exchequer setup program in conjunction with a site specific licence file. The IRIS Exchequer setup program covers the core of the system together with the standard modules, separate installers are available for plug-ins that are purchased separately.

The setup programs are generated using Wise Installation System 9.01 which pre-dates MSI technology.

## Changes for this release

New Prospects tables/data files in all companies??????? New Forms/Form Definition Sets???

RSS Feeds?

Importer Templates?

Bank Rec mods for 8 char sort codes

# System / Platform Analysis

The system requirements and platforms supported by IRIS Exchequer are controlled by Product Management, the official technical specifications applying to this release can be found in the ?????????? folder.

Additions to the system requirements and platforms supported by IRIS Exchequer must be detailed as individual requirements in the requirements section.

# Coding Standards

The standard Exchequer Coding Standards will be used (see ????????).

# Source Control Procedures

IRIS Exchequer uses Subversion for source code control, the following repositories are installed on BMTDEV1:-

svn://bmtdev1/exchequer/trunk – IRIS Exchequer Application Code

svn://bmtdev1/exchequerCD/trunk – IRIS Exchequer Setup program

svn://bmtdev1/thirdparty/trunk – Delphi 6.01 Components used for IRIS Exchequer

## Backup Procedure

The development team checkout files from the repositories to their working folders on the BmtDev1 server, these folders are backed up nightly by IS.

The repositories themselves are written to ‘dump’ files each evening which are then backed up by IS that night, it is expected that no more than one days work will be lost in the event of a complete systems failure.

## Project Hierarchy

The IRIS Exchequer project hierarchy has evolved since 1995 and contains over 100 directories, the diagram below shows the main directories and selected subdirectories which are especially important:-



## Check-out / Check-in Policy

Check-outs are performed without locking, programmers are responsible for ensuring they are always working on the latest version and for merging changes done by other programmers to the same files.

Files should not be checked in until they build and, wherever possible, the built component has been tested for correct functionality.

The Check-In comments should contain a summary of the changes and include fault numbers where relevant.

## Security

Access to the Exchequer and ExchequerCD repositories is limited to the programmers working on Exchequer only, each programmer has full rights:-

|  |  |  |
| --- | --- | --- |
| mhigginson | Mark Higginson | Exchequer Team Leader |
| csandow | Chris Sandow | Exchequer Analyst / Programmer |
| prutherford | Paul Rutherford | Exchequer Analyst / Programmer |
| nfrewer | Neil Frewer | Part Time Exchequer Analyst / Programmer |
| bmtdevbld |  | Exchequer Build User |

The ThirdParty repository is additionally open to programmers working on software that links to Exchequer:-

|  |  |  |
| --- | --- | --- |
| asweetman | Andrew Sweetman | Part Time Bespoke Analyst / Programmer |
| bharkcom | Barry Harkcom | Part Time Bespoke Analyst / Programmer |
| bbuck | Ben Buck | Bespoke Analyst / Programmer |

## Builds

The final builds of the core components are performed automatically using the FinalBuilder service installed in the vm-exch601 VMWare virtual machine running on the BMTDev1 server.

Components whose build is not yet automated within FinalBuilder are built manually on workstations by the programmer.

The FinalBuilder service is accessed either via a Tray Icon utility which allows builds to be started and monitored or via a web page which allows builds to be started, monitored and maintained.

Access to the tray component and web page is limited at this time to:-

|  |  |  |
| --- | --- | --- |
| Administrator | Mark Higginson | Exchequer Team Leader |
| Start/Monitor Builds | Chris Sandow | Exchequer Analyst / Programmer |
| Start/Monitor Builds | Paul Rutherford | Exchequer Analyst / Programmer |

*Note: Build scripts are stored in svn://bmtdev1/exchequer/branches/Finalbuilder Scripts*

# Requirements

*This section should list all requirements for the project; these requirements can be deployment, system, platform, or functional types. The table below represents the details needed for one requirement. One table must be completed for every requirement.*

## Requirement < 6.3.WIN.32 > – Requirement 1

| Requirement Name and Number | | Windows 7 Support | | 6.3.WIN.32 |
| --- | --- | --- | --- | --- |
| Requirement Type | Platform | | | |
| Business Analysis | | | 6.3.WIN.32 – Refer to PRD | |
| Technical Design | | | *N/A – This is a QA testing task that may result in either faults or further requirements being raised against v6.3.* | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.COL.33> – Requirement 2

| Requirement Name and Number | | Column Sorting – Sales & Purchase Daybook | | 6.3.COL.33 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.COL.33 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.COL.33 > – Technical design*](#_Requirement_<_6.3.COL.33) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.APP.34> – Requirement 3

| Requirement Name and Number | | Acorn – Apps & Vals Mod | | 6.3.APP.34 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.APP.34 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.APP.34 > – Technical design*](#_Requirement_<_6.3.APP.34) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.RSS.35> – Requirement 4

| Requirement Name and Number | | RSS Feeds | | 6.3.RSS.35 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.RSS.35 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.RSS.35 > – Technical design*](#_Requirement_<_6.3.RSS.35) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.REA.36> – Requirement 5

| Requirement Name and Number | | Read-Only Exchequer | | 6.3.REA.36 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.REA.36 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.REA.36 > – Technical design*](#_Requirement_<_6.3.REA.36) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.FOC.37> – Requirement 6

| Requirement Name and Number | | Customisation Focus | | 6.3.FOC.37 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.FOC.37 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.FOC.37 > – Technical design*](#_Requirement_<_6.3.FOC.37) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.FIL.38> – Requirement 7

| Requirement Name and Number | | Column Sorting – Trader List Filters | | 6.3.FIL.38 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.FIL.38 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.FIL.38 > – Technical design*](#_Requirement_<_6.3.FIL.38) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.FIL.39> – Requirement 8

| Requirement Name and Number | | Column Sorting – Customer/Supplier Ledger Filters | | 6.3.FIL.39 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.FIL.39 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.FIL.39 > – Technical design*](#_Requirement_<_6.3.FIL.39) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.FIL.40> – Requirement 9

| Requirement Name and Number | | Column Sorting – Stock List Filters | | 6.3.FIL.40 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.FIL.40 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.FIL.40 > – Technical design*](#_Requirement_<_6.3.FIL.40) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.FIL.41> – Requirement 10

| Requirement Name and Number | | Column Sorting – Job Ledger Filters | | 6.3.FIL.41 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.FIL.41 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.FIL.41 > – Technical design*](#_Requirement_<_6.3.FIL.41) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.8 > – Requirement 11

| Requirement Name and Number | | Importer Templates | | 6.3.PQIP.8 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.8 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.8 > – Technical design*](#_Requirement_<_6.3.PQIP.8) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.9 > – Requirement 12

| Requirement Name and Number | | Importer Templates | | 6.3.PQIP.9 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.9 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.9 > – Technical design*](#_Requirement_<_6.3.PQIP.9) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.10 > – Requirement 13

| Requirement Name and Number | | Protect BoM Stock Kitting Options | | 6.3.PQIP.10 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.10 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.10 > – Technical design*](#_Requirement_<_6.3.PQIP.10) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.14 > – Requirement 14

| Requirement Name and Number | | Currency Revaluation Audit PDF | | 6.3.PQIP.14 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.14 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.14 > – Technical design*](#_Requirement_<_6.3.PQIP.14) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.20 > – Requirement 15

| Requirement Name and Number | | Debtors & Creditors Report Breakdown | | 6.3.PQIP.20 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.20 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.20 > – Technical design*](#_Requirement_<_6.3.PQIP.20) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PQIP.24 > – Requirement 16

| Requirement Name and Number | | Standard Reports – Report Parameters Cover Sheet | | 6.3.PQIP.24 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PQIP.24 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PQIP.24 > – Technical design*](#_Requirement_<_6.3.PQIP.24) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.42 > – Requirement 17

| Requirement Name and Number | | Trader List – Prospects Tab | | 6.3.PRO.42 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.42 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.42 > – Technical design*](#_Requirement_<_6.3.PRO.42) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.43 > – Requirement 18

| Requirement Name and Number | | Toolkits – Prospects Support | | 6.3.PRO.43 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.43 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.43 > – Technical design*](#_Requirement_<_6.3.PRO.43) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.44 > – Requirement 19

| Requirement Name and Number | | Importer – Prospects Support | | 6.3.PRO.44 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.44 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.44 > – Technical design*](#_Requirement_<_6.3.PRO.44) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.45 > – Requirement 20

| Requirement Name and Number | | Data Dictionary – Prospects Support | | 6.3.PRO.45 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.45 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.45 > – Technical design*](#_Requirement_<_6.3.PRO.45) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.46 > – Requirement 21

| Requirement Name and Number | | Prospects – Form Designer Support | | 6.3.PRO.46 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.46 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.46 > – Technical design*](#_Requirement_<_6.3.PRO.46) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.47 > – Requirement 22

| Requirement Name and Number | | Prospects – Quotes | | 6.3.PRO.47 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.47 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.47 > – Technical design*](#_Requirement_<_6.3.PRO.47) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

# Iteration Design

*This section should design the iterations that will be present in the release, the number of iterations should be decided, and every requirement should be assigned to one of the iterations, it is advised that*

* *The first iterations should concentrate on the high risk / complex requirements so as to minimize the risk towards the end of the release.*
* *The iterations are managed in JIRA using the versions feature in JIRA*

**

**

# Proposed Development Machine Specification

The following table should be completed for the “reference” development machine that will be used to develop the release.

|  |  |
| --- | --- |
| Machine Specification |  |
| CPU | 2.0 GHz Minimum |
| Memory | 2 GB Minimum |
| Development Tools | Borland Delphi 6.01  Pervasive.SQL v10 WGE  SQL Server 2005 |
| Design Tools | Microsoft Visio |
| Documentation Tools | Microsoft Word |
| Issue Reporting Tools | JIRA  (Account Also Needed) |
| Project Tools | MS Excel 2008 |
| Source Control Tools | Subversion  (Account Also Needed) |

# Proposed Build Machine Specification

The following table should be completed for the “reference” build server that will be used to build the release.

|  |  |
| --- | --- |
| Machine Specification |  |
| OS | Windows XP SP2 Pro – VMWare Virtual Machine |
| CPU | N/A – Virtual Machine |
| Memory | 512 MB Minimum |
| Build Tools | FinalBuilder 6  FinalBuilder Server 6  Borland Delphi v6.01 |
| Redistributables Needed | MDAC  Windows Installer 3.1  Ifx  Etc. |
| Installer Tools | Wise Installation System 9.01 |
| Source Control Tools | Subversion |

# Proposed Test Machine Environments

Here is where the test manager will define a list of test environments what will be needed to ensure testing is performed correctly. These environments will more than likely take their initial state from the system / platform analysis which defined what environments should run on.

It would be advantageous at this stage to define which the main test environments are, and which will be used to smoke test the application.

These environments will be used during development testing and the full testing at the verify stage.

|  |  |
| --- | --- |
| Machine Specification | Test Machine Specification ID |
| CPU | 2.0 GHz Minimum |
| Memory | 2 GB Minimum |
| Virtualised PC OK? | Yes No |
| Operating System |  |
| Service Pack Level for OS |  |
| Applications | *Application list*  *Including list of service pack levels for the application* |
| Settings | *Detail any special settings that are required on the machine, network, internet access, firewall etc.* |
| Client / Server / Or Both? | *Client, Server, Both* |

# Revised Gate Dates

This section should provide a revision of the gate milestone now that the high level design is complete.

**

## Appendix 1 – Technical Design

### Requirement < 6.3.COL.33 > – Technical design AWAITING PRD

| Requirement Name and Number | | Column Sorting – Sales & Purchase Daybook | 6.3.COL.33 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.APP.34 > – Technical design CS

| Requirement Name and Number | | Acorn – Apps & Vals Mod | 6.3.APP.34 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.RSS.35 > – Technical design AWAITING PRD

| Requirement Name and Number | | RSS Feeds | 6.3.RSS.35 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.REA.36 > – Technical design Awaiting BA work on interdepartmental interfaces

| Requirement Name and Number | | Read-Only Exchequer | 6.3.REA.36 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level DesignOverview *Due to changes in the way IRIS Exchequer is being sold it is required that it support Annual Licensing by moving into a 1 month Grace Period on expiry of the Annual Licence followed by going into a Read-Only mode to allow customers to access their data, but not allowing them to continue using IRIS Exchequer to keep their accounts.*  *The Exchequer code base has inherited limited support for the Grace Period and Read-Only mode from IRIS Accounts Office, however this needs to be significantly extended to cover the IRIS Exchequer Licensing system and to support Perpetual Licensing.* ImplementationDesign *SecRel*   1. *In order to allow the Annual Licensing to work securely a new format Release Code will be developed in the following format:-*   *XXXXXXXXXX-YYYYYY-ZZZZ*  *Where XXXXXXXXXX is a standard IRIS Exchequer System Release Code, YYYYYY is an encoded Annual Licence Expiry Date and ZZZZ is a checksum to prevent tampering.*   1. *Extend the ‘Exch Security’ tab in SecRel with a Licence Type field and extend the Release Code field to correctly calculate the code based on the Licence Type.*   *CD / Exchequer Licence File Changes*   1. *Two new fields will be required in the Licence File to store the licence type and annual licence expiry date:-*   *LicenceType : Byte; // 0=Perpetual, 1=Annual*  *LicenceExpiry : TDateTime; // Annual Licence Expiry Date*   1. *Update the licCopyLicence routine in sbslib\win\excommon\LicFuncU.Pas to copy the new fields. NOTE: This routine is used by the Installer when creating new licences during installs.* 2. *Add a new function into sbslib\win\excommon\LicFuncU.Pas to return the string description of the Licence Type.*   *CD Licence Generator*   1. *Move the Exchequer Version Number field from the ‘Exchequer Version’ dialog to the ‘Installation Type’ dialog.* 2. *Extend the Exchequer Version Number field to distinguish v6.3 from previous versions.* 3. *Extend the ‘Customer Details’ dialog to include new Licence Type and Annual Licence Expiry Date fields if the Exchequer Version is v6.3 or later, the Annual Licence Expiry Date field will only be available if the Licence Type is set to Annual. The Licence Type and Annual Licence Expiry Date fields should not be defaulted to force the user to explicitly set them.* 4. *The new Licence Type and Annual Licence Expiry Date will be written to the licence file including the human readable text header within the licence file.* 5. *The CSV log file will be extended to include the Licence Type and Annual Licence Expiry Date fields.* 6. *The notification link to WebRel using WRTransmit.Exe will be extended to include the Exchequer Version and the new Licence Type and Annual Licence Expiry Date fields. (see WebRel point 1).* 7. *Update the Licence Viewer to correctly display the Licence Type.*   *Installer*   1. *Modify the Read Exchequer Licence dialog in the CD Auto-Run to display the Licence Type.* 2. *Extend the UpdateEntLic routine in Entrprse\MultComp\ModRels.Pas to copy the Licence Type across, this routine is used during upgrades to update the Exchequer Licence in the existing installation.* 3. *Modify UpdateEntLic in Entrprse\MultComp\ModRels.Pas to detect Annual License installations in Grace Period or Read-Only mode so that it doesn’t incorrectly reset the Expiry Date to today+7 days during upgrades.* 4. *Modify the validation routine in the Directory Dialog (Entrprse\Setup\DirDlg4.Pas) to check for changes in the Licence Type during Upgrades and set a warning flag if found.* 5. *Modify the ‘Ready To Upgrade’ dialog to warn about the change in Licence Type during upgrades (see point 4 above) and require the user to authorise the change by ticking a check box.*   *WebRel*  *WebRel will need to be modified to support Annual Licensing to allow our Technical Support team and our Dealers/Distributors to supply Release Codes to customers:-*   1. *WRTransmit.Exe will be extended to support the Exchequer Version, Licence Type and Annual Licence Expiry Date fields.* 2. *WRListener.Exe will be extended to support the Exchequer Version, Licence Type and Annual Licence Expiry Date fields and populate the WebRel database (see point 3). The Exchequer Version field will be used to set the existing Version field held against the customer.* 3. *The WebRel database will extended to include Licence Type and Contract Expiry Date fields against the Customer.* 4. *The Version Number held against the Customer ESN will be extended to support ‘v5.xx-v6.2’ and ‘v6.3’.* 5. *The Customer ‘Security Codes’ page will be extended to include a line containing the licence details, e.g. ‘Perpetual Licence’ or ‘Annual Licence expiring 31/06/2010’.* 6. *The System Security section on the Customer ‘Security Codes’ page will extended to support the new format Annual Licence Release Codes (see SecRel section above).* 7. *The Use Dummy ‘Security Codes’ page will be extended in a similar manner to the Customer ‘Security Codes’ page except that the Use Dummy mode will be limited to Perpetual Licences.* 8. *The WebRel Administration Customer Record page will be extended to include the Licence Type and Annual Licence Expiry Date fields.*   *Multi-Company Manager*  *Blah, blah blah, etc…*  *Core Enter1.Exe*  *Blah, blah blah, etc…*  *Importer*  *The Importer will be extended to implement read-only mode preventing the user from importing data.*  *Toolkits*  *The Toolkits will be extended to implement read-only mode preventing users from changing data, this includes Adding new data, updating or deleting existing data and running any processes that change data, e.g. Delivering Orders. Each function/method affected should return an error code, preferably common across all routines, indicating that the system is in read-only mode.*  *Additionally, Ex\_GetSysData in the Toolkit DLL and the SystemSetup object in the COM Toolkit will be extended with a new ReadOnly : Boolean property to allow bespoke applications to detect Read-Only mode and handle it gracefully.*  *OLE Server*  *The OLE Server will be extended to implement read-only mode preventing the users from changing data, this includes Adding new data, updating or deleting existing data and running any processes that change data, e.g. Delivering Orders.*  *E-Business*  *The E-Business module will be extended to implement read-only mode preventing the user from importing new transactions or editing or posting existing E-Business Daybook transactions into Exchequer.*  *Outlook Dynamic Dashboard*  *The ODD controls will be extended to implement read-only mode preventing the user from adding, editing or deleting data in Exchequer, this covers the Authoris-e, Add Timesheet and Accounts on Hold controls. The user should still be able to change the ODD configuration itself.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.FOC.37 > – Technical design

| Requirement Name and Number | | Customisation Focus | 6.3.FOC.37 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level DesignOverview *This Customisation Focus issue affects certain hook points in Exchequer where a plug-in displays a popup message or window, when the message/window is closed by the user the focused control in Exchequer does not display correctly so the user is uncertain which control is active. Some controls also fail to process changes by the user in this condition.* ImplementationDesign *Detailed investigation by Paul Rutherford has shown that modifying the basic customisation event class to post a Windows WM\_SETFOCUS message to the active control on completion of the hook point resolves the issues.* | | | |

### Requirement < 6.3.FIL.38 > – Technical design CS

| Requirement Name and Number | | Column Sorting – Trader List Filters | 6.3.FIL.38 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.FIL.39 > – Technical design CS

| Requirement Name and Number | | Column Sorting – Customer/Supplier Ledger Filters | 6.3.FIL.39 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.FIL.40 > – Technical design CS

| Requirement Name and Number | | Column Sorting – Stock List Filters | 6.3.FIL.40 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.FIL.41 > – Technical design CS

| Requirement Name and Number | | Column Sorting – Job Ledger Filters | 6.3.FIL.41 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PQIP.8 > – Technical design PR / MH

| Requirement Name and Number | | Importer Templates | 6.3.PQIP.8 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PQIP.9 > – Technical design Awaiting PRD

| Requirement Name and Number | | Protect System Setup Switches | 6.3.PQIP.9 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PQIP.10 > – Technical design

| Requirement Name and Number | | Protect BoM Stock Kitting Options | 6.3.PQIP.10 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level DesignOverview *Technical Support have identified that changing the Bill of Materials Kitting Options after the Stock item has been used is a cause of severe stock reconciliation problems requiring a lot of effort by technical support to resolve.* ImplementationDesign *When a user tries to change the Kitting Options on the Build tab of the Stock Record window the system will check for the presence of transaction lines for that stock item which cause either stock movements or allocations – the Stock Code index should be used to minimise the time taken.*  *Each transaction line will be evaluated based on its transaction type and how the system is configured to determine whether it qualifies to block the change:-*   |  |  |  |  | | --- | --- | --- | --- | | Transaction Type | Quotes to allocate stock | Orders to allocate stock when picked | Qualifies | | SIN, SRI, SJI, SJC, SCR, SRF, PIN, PPI, PJI, PRF, PJC,PCR, ADJ, SRN, PRN, SDN, PDN, WOR | n/a | n/a | Yes | | SQU, PQU | True | False | Yes | | SQU, PQU | False | False | No | | SQU, PQU | True/False | True | No | | SOR, POR | n/a | False | Yes | | SOR, POR | n/a | True | Yes if Picked Qty is non-zero, otherwise No |   *If blocking transaction lines are found then a dialog should be shown informing the user that the Kitting Options cannot be changed, but which allows an override using the Daily Password. If the override is performed then the change will be applied otherwise the change will be ignored.* | | | |

### Requirement < 6.3.PQIP.20 > – Technical design Awaiting PRD

| Requirement Name and Number | | Debtors & Creditors Report Breakdown | 6.3.PQIP.20 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PQIP.24 > – Technical design PR

| Requirement Name and Number | | Standard Reports – Report Parameters Cover Sheet | 6.3.PQIP.24 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PQIP.14 > – Technical design

| Requirement Name and Number | | Currency Revaluation Audit PDF | 6.3.PQIP.14 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level DesignOverview *The Currency Revaluation Audit Trail report is a key report when auditing the system, to ensure that the report is available when required it should be automatically saved as a .PDF file to remove the current reliance on users keeping printed copies.* ImplementationDesign *When the Currency Revaluation report is printed a second copy will automatically be printed to a .PDF file in the Reports directory in the current company dataset.*  *The file should be named “Currency Revaluation YYYYMMDD-N.Pdf” when YYYYMMDD is the currency system date and N is a sequential number to ensure uniqueness and allow for multiple revaluations on the same day.* | | | |

### Requirement < 6.3.PRO.42 > – Technical design Awaiting PRD

| Requirement Name and Number | | Trader List - Prospects Tab | 6.3.PRO.42 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PRO.43 > – Technical design Awaiting PRD

| Requirement Name and Number | | Toolkits – Prospect Support | 6.3.PRO.43 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PRO.44 > – Technical design Awaiting PRD

| Requirement Name and Number | | Importer – Prospects Support | 6.3.PRO.44 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PRO.45 > – Technical design Awaiting PRD

| Requirement Name and Number | | Data Dictionary - Prospect Support | 6.3.PRO.45 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PRO.46 > – Technical design Awaiting PRD

| Requirement Name and Number | | Prospects – Form Designer Support | 6.3.PRO.46 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

### Requirement < 6.3.PRO.47 > – Technical design Awaiting PRD

| Requirement Name and Number | | Prospects - Quotes | 6.3.PRO.47 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Technical High Level Design *Armed with the use cases and the high level architecture diagram each functional requirement should be assessed on the following areas,*   * Complexity * Skills Needed * Data Changes * Business Logic Changes   *Every requirement that has been agreed in the MRD should be listed here in their own section. If it makes sense then multiple functional requirements can be merged in to a single section. This would normally be the case where two functional requirements interact.* Overview *A brief description of how the technical high level design is being approached.* ImplementationTechnologies *Provide a brief description of the technologies that will be used to implement the requirement, and whether those skills are available in the team.* Interactions *Provide details of the knock on effects of implementing this requirement, such as changes to other areas of the application.* Data changes *Provide a list of the changes in data that are needed to implement the requirement, making sure to signal where data is coming from, new fields etc.* Business Logic Changes *Provide a description (and preferably some high level diagrams) of the new and / or changed business logic, this should also show any interactions and changes with existing business logic.* Open Issues Standard section to record any unknown elements or assumptions made during the design of the requirement | | | |

# Appendix 2 - Test Approach

# Test Plan Outline

This document outlines the testing strategy and approach to be taken regarding the functionality of PROJECT A Release n.nn developed by the Iris Software and Services development team. Add any relevant information as an introduction.

# References

The following list reference documents that support this section of the HLD.

|  |  |  |
| --- | --- | --- |
| Ref | Document(s) | Version |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

# Test Items (Functions)

The following test items are planned to be tested in PROJECT NAME: [delete or add to the list as appropriate]

* Functional testing of new features – new features will be tested to validate that they have been implemented successfully and meet their requirements
* Installation and upgrade testing – testing of the product installation/un-installation and the ability to upgrade from previous versions. The direct upgrade ability of the two previous versions of ILB (i.e. release 1.9 and 1.10) is supported and will be tested.
* Defect Fixes – Validation of the implementation of defect fixes. Defect fixes are performed by the development team on a priority basis.
* Regression testing – perform regression testing to validate that existing functions are not adversely affected by changes made. Regression testing will primarily be focused at areas that have undergone change, either by new functionality.

Automated regression testing will also be performed during PROJECT NAME.

# Features to be Tested

The following features are planned to be delivered in PROJECT NAME and will be tested. Please refer to the relevant design/requirement specifications for further details of the feature.

Each feature/requirement has been designated a requirement id as this is used in various testing tools.

|  |  |
| --- | --- |
| Req Id | Feature |
| 0001 |  |
| 0002 |  |
| 0003 |  |
| 0004 |  |
| 0005 |  |
| 0006 |  |
| 0007 |  |
| 0008 |  |
| 0009 |  |

Additional testing will be performed in the following areas: : [delete or add to the list as appropriate]

* Installation Testing – validation that the software components install successfully on supported systems and configurations
* Uninstall Testing – validation that un-installation cleanly removes installed files, system settings, registry settings etc
* Upgrade Testing – validation that upgrades from supported existing versions are performed correctly
* Interoperability Testing – Verifying that the product interacts correctly with other products (MS Office, SQL server etc)

# Features not to be Tested

For example:

Testing of the functionality of all new features is planned.

Non-functional testing of the new functionality is not planned to be performed by the ….

Testing via RDP or Citrix connections is not planned for this release.

# Approach (Strategy)

## Pass Strategy

[delete or add to the list as appropriate]

PROJECT NAME is being delivered in n iterations. Each iteration will include a sub-set of the new functionality to be developed

Testing will be split into n cycles to reflect the n iterations into test. Each test cycle will contain a single testing pass (the set of tests assigned to that cycle is planned to be executed once).

Some tests that are executed in cycles one and two will be repeated in subsequent cycles to verify that functionality has not been adversely affected by subsequent functionality additions or defect fixes.

Testing within the A4-A6 gates (known as the ‘verify’ stage within the PROJECT NAME project plan) will consist primarily of defect fix verification, new functionality testing re-tests, installation/upgrade testing and system regression testing. This period will also allow some contingency to perform new functionality testing if the feature could not be fully tested during the earlier cycles.

## Testing Tools

### Test Development Tools

It is planned that all new functionality testing will utilise manual testing techniques.

Regression testing will utilise both manual and automated techniques.

Test scenarios and cases will be developed initially using Excel spreadsheets.

### Test Execution Tools

Test execution recording will initially be performed using Excel spreadsheets

Other tools will be used as required. These may include a screen capture tool to record tester transactions (to assist with defect reproduction).

## Testing Metrics

Testing metrics will be collected during test development and test execution. These metrics will be used to measure the effectiveness of the testing effort and to help improve the testing process.

The ease and effectiveness of metrics collation will be improved by the adoption of a test management and defect tracking tool (Test Link and Jira respectively).

The collection of other metrics may be included as required.

### Metrics Collected During Test Development

The following metrics will be collected during test development:

|  |  |  |
| --- | --- | --- |
| **Metric** | **Description** | **Formula** |
| Test Specification Progress | Comparison of the actual time spent preparing test scenarios, conditions and scripts against the estimated time | Actual v estimated |
| Scripts Passed Review First Time Rate | The relative proportion of test scripts that are successfully peer reviewed at the first attempt.  The rate of first time review passes is an indicator of the quality of the test preparation activity and associated deliverables.  A lower than expected pass rate could be due to poor test preparation quality, and is likely to lead to increased costs and extended timescales. | Scripts Passed Review First Time / Scripts Reviewed First Time |

### Metrics Collected During Test Execution

The following metrics will be collected during test execution:

|  |  |  |
| --- | --- | --- |
| **Metric** | **Description** | **Formula** |
| Test Execution Progress | Comparison of the actual tests executed compared to the plan.  Provides an indication of capability to meet deadlines  Significant variance (i.e. +/- 5%) should be accompanied by an explanatory comment. | Actuals v Plan |
| Test Environment Availability | Availability of test environments expressed as a percentage of test execution time.  Provides an indication of the ability to progress testing.  Significant outages should be accompanied by an explanatory comment. | Percentage of environment availability during working hours |
| Defect Progress | Comparison of the rate at which defects are being raised compared to the rate at which defects are being closed.  If defects are being raised at a rate significantly greater than the rate at which they are being closed, it is an indicator that testing schedules may have to be extended in order to achieve test exit criteria | Defects raised v Defects closed |
| Defect Age | Analysis of the time that outstanding defects have been open, by severity.  Defects outstanding for long periods are an indicator that testing schedules may have to be extended in order to achieve test exit criteria (e.g. no outstanding Severity 1 defects). | Time outstanding, by defect severity |
| Tests Passed First Time Rate | The relative proportion of tests that are successfully executed at the first attempt.  The rate of first time test passes is an indicator of product quality.  A lower than expected pass rate could be due to poor product quality, and is likely to lead to increased costs and extended timescales. | Tests Passed First Time / Tests Executed First Time |

## Configuration Management

Configuration Management is necessary at different areas of the testing project. The correct versions of software, tests, data and code must be available throughout the testing life cycle to ensure the validity of the testing effort.

|  |  |
| --- | --- |
| **Test Area** | **Configuration requirements** |
| Test Preparation | Test cases will be versioned with data relating to the source documentation used to create the test case being recorded. Test author, date and associated details will be recorded. |
| Test Environment | The setup of the test environment will be recorded during test execution. This will include things such as the installed operating system, database, applications and configuration as well as the version of the software under test. |
| Test Execution | Each execution of a test case will record the necessary details such as:- test case version, environment setup and configuration, tester + date/time, defects raised etc. |

## Test Configurations

PROJECT NAME is required to work on a number of different operating systems and setups.

Testing will be performed on a number of these configurations to verify correct operation.

The test environment is not currently able to support x64 windows versions.

Therefore, the majority of functional testing during iterations 1-3 will be performed on the following configuration, with a single server machine been accessed by XP and Vista clients:

|  |  |  |
| --- | --- | --- |
| **Server** | Win 2K3 x86 | SQL 2005 |
| **Client** | Windows XP x86 | Office 2003 |
| **Client** | Windows Vista x86 | Office 2007 |

During the verification testing phase, further test configurations will be utilised and testing will be spread across each configuration. A new test server capable of hosting 64-bit windows versions should have been introduced and testing to a remote SQL instance will also be included (ie SQL installed on a different server to the ILB instance).

The tables below show the server and client configurations that will be utilised during PROJECT NAME verification testing:

|  |  |
| --- | --- |
| **Server** | |
| Windows 2K3 x86 | SQL 2005 |
| Windows 2K3 x64 | SQL 2008 |
| Windows 2K8 x86 | SQL 2005 |
| Windows 2K8 x64 | SQL 2008 |

|  |  |
| --- | --- |
| **Client** | |
| Windows XP x86 | Office 2K3 |
| Windows XP x64 | Office 2K7 |
| Windows Vista x86 | Office 2K3 |
| Windows Vista x64 | Office 2K7 |

## 

## Regression Testing

### Automated Regression Testing

Automated testing is planned to be included during PROJECT NAME.

The automated solution has been planned to be setup and executing the automated tests in Knutsford by mid-July. From this point, the automated tests will be run whenever a new build is taken by the test team (i.e. when the iteration releases are taken and when fix releases are taken during verification).

### Manual Regression Testing

Regression testing will be performed on multiple configurations of the supported platforms and applications (please refer to section 8.5).

Regression testing will be performed by using a mixture of pre-planned test cases and exploratory testing techniques. The exploratory testing will be documented so that a record of the testing completed is maintained and to assist in the creation of future test cases.

## Status Reporting

A status report will be made available to project stakeholders following each cycle of testing. This report will include details of the relevant metrics mentioned above as well as comment on the progress of the testing effort.

Daily reports will be made during test execution to the Test Manager detailing daily execution and defect status.

# Item Pass/Fail Criteria

Testing will have been considered as completed successfully when the following criteria have all been met:

* All planned test cases have been executed and all associated details (application version, test environment, execution date, tester name, result) have been recorded against them
  + Any tests that have not been executed will require dispensation from the appropriate parties
* High severity defects, if any, are all closed, with documentation of the steps taken to correct them and of the steps taken to test the correction.
* Lower severity defects should also be closed, as above. However, any that do not present an unacceptable risk may remain outstanding if dispensation has been agreed by the relevant parties.

# Test Deliverables

The following table lists the test items that will be delivered as part of this plan.

|  |  |
| --- | --- |
| **Deliverable** | **Details** |
| Test Approach | In this document |
| Testing Schedule | MS Project document detailing the tasks, assignees and timescales of the testing effort for PROJECT NAME. |
| Test Cases | Excel spreadsheets containing the test cases produced for PROJECT NAME (test cases may also be produced within Test Link if it is available during PROJECT NAME) |
| Test Results | Excel spreadsheets containing the test execution results for the PROJECT NAME testing (results may also be produced within Test Link if it is available during PROJECT NAME) |
| Status Reports | A status report will be generated at the end of each of the three test cycles |
| Test Completion Report | A test completion report will be generated at the end of the testing project that will summarise the testing performed, the results, the closed and outstanding defects etc. This report should also include a lessons learnt section to help improve the quality of future projects. |

# Test Environment

PROJECT NAME supports a number of operating systems and applications (please refer to section 8.5 for the platforms and applications that will be tested against).

Testing will be performed such that a proportional amount of testing is performed on each operating system / Office version to match the usage within the customer base.

Testing will utilise a Virtual Server environment. Testing may also be executed on local and virtual machines if appropriate. The Virtual Server environment allows greater flexibility in testing over different operating systems and configurations as well as getting more benefit from minimal hardware resources.

# Staffing and Training Needs

The PROJECT NAME test team received product training prior to the start of the project and no specific additional training has been identified as being required. Domain experts may be required to be consulted as needs arise. Additional training may be required for the ILS if testing for that product is required in Knutsford (this will be covered in a separate test plan).

No specific training on testing tools has been identified. The test automation tool ‘Test Complete’ is planned to be used for the first time during PROJECT NAME and the required knowledge of the tool will be sought via online resources and consultation with the RAVE team if required. Knowledge transfer within the team and wider community will be used on additional testing tools used by the team.

# Roles and Responsibilities

The following have been identified as the main roles associated with the PROJECT NAME testing project:

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| ?? | Quality Engineering Manager | * Stakeholder management * Assist test lead * Review and Sign-off testing deliverables * Resource Management * Partake in Go/No-Go decisions |
| ?? | Test Team Lead | * Test planning and management * Produce the testing strategy/approach * Ensure all required elements are in place * Review design specifications * Review testing deliverables * Create and maintain testing schedule * Produce status/progress reports * Produce test closure report * Produce test scenarios/conditions/cases and data * Perform test execution * Define and monitor the testing environment * Partake in Go/No-Go decisions |
| ?? | Tester | * Review design specifications * Peer review testing deliverables * Produce test scenarios/conditions/cases and test data * Perform test execution * Assist in the maintenance of the testing environment |
| ?? | Tester | * Review design specifications * Peer review testing deliverables * Produce test scenarios/conditions/cases and test data * Perform test execution * Assist in the maintenance of the testing environment |
| ?? | Development Manager | * Review testing deliverables * Respond to testing queries * Partake in Go/No-Go decisions |
| ?? | Development Lead | * Review testing deliverables * Respond to testing queries |
| ?? | Product Manager | * Review testing deliverables * Respond to testing queries |
| ?? | Project Manager | * Review testing deliverables * Maintain the overall project plan * Inform test team of project changes/delays that may affect the testing schedule |

# Schedule

The major testing milestones are shown below:

|  |  |  |
| --- | --- | --- |
| Milestone | Comment | Date |
| Start of Test Preparation |  | Date |
| End of Test Preparation | Test preparation will continue during iteration 1 and iteration 2 test execution | Date |
| Start of Iteration 1 Execution | Iteration 1 functionality testing | Date |
| End of Iteration 1 Execution |  | Date |
| Start of Iteration 2 Execution | Iteration 2 functionality testing | Date |
| End of Iteration 2 Execution |  | Date |
| Start of Iteration 3 Execution | Iteration 3 functionality testing | Date |
| End of Iteration 3 Execution |  | Date |
| Start of post-functionality testing | Installation/upgrade testing  Release testing etc | Date |
| End of post-functionality testing |  | Date |
| PROJECT NAME Testing Complete | Test Closure Document completed | Date |
| Test maintenance operations completed | Date |

Test duration estimates have been based on estimates provided by the test team for the duration of preparation and execution activities for each of the testing requirements. In addition ‘time buckets’ have been estimated for the testing of defect fixes, execution of regression tests and performing installation testing.

# Planning Risks, Contingencies and Assumptions

The following risks and related contingencies have been recognised for the PROJECT NAME testing project.

|  |  |  |
| --- | --- | --- |
| **#** | **Risk** | **Contingency** |
| 1 |  |  |
| 2 |  |  |

The following assumptions have been made regarding the testing of PROJECT NAME.

|  |  |
| --- | --- |
| **#** | **Assumption** |
| 1 | Development, training and product management resources will be available to support queries that the test team may have |
| 2 | Development will deliver software iterations to the timescales detailed in the project plan |
| 3 | The defect management system utilised for PROJECT NAME will be available to both testing and development teams |
| 4 | No further requirements will be included in the PROJECT NAME release |