Tessella Project Number: 7567

**Tessella Ltd** 26 The Quadrant, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YS, UK

T: +44 (0)1235 555511 | E: info@tessella.com | www.tessella.com

Client: RAL & ORNL

Mantid Ticket System

User Requirements Document



Document Control

This document is under document control. All members of the distribution list will receive updated copies whenever alterations are made. All other copies are uncontrolled.

Document Owner

|  |  |  |
| --- | --- | --- |
| Name | Role | Company |
| Nicholas Draper | Owner | Tessella |

Document Approval

The following are required to approve this document. Their approval signifies that the document is fit for purpose and approved for use.

|  |  |  |
| --- | --- | --- |
| Name | Approval Role | Company |
| Stuart Campbell |  | Tessella |
|  |  |  |

Distribution

|  |  |  |
| --- | --- | --- |
| Name | Role | Company |
| Mantid Documents repository | Filing | GitHub |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Issue | Date | Description |
| Nicholas Draper | V1.R1.M0 | 28-May-2013 | 1st version written |
|  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc357526332)

[1.1 Purpose of this Document 4](#_Toc357526333)

[1.2 Scope of this Document 4](#_Toc357526334)

[1.3 Context of this Issue 4](#_Toc357526335)

[1.4 Definition of Terms 4](#_Toc357526336)

[2 Functional Requirements 5](#_Toc357526337)

[2.1 Requirements for Tickets 5](#_Toc357526338)

[2.2 Requirements for Ticket creation 5](#_Toc357526339)

[2.3 Requirements for developers 6](#_Toc357526340)

[2.4 Requirements for testers 6](#_Toc357526341)

[2.5 Requirements for Project Management 7](#_Toc357526342)

[2.6 Requirements for ad-hoc queries 8](#_Toc357526343)

[3 Non-functional Requirements 8](#_Toc357526344)

[3.1 Speed and Time Requirements 8](#_Toc357526345)

[3.2 Capacity Requirements 9](#_Toc357526346)

[3.3 Reliability Requirements 9](#_Toc357526347)

[4 Design and Implementation Constraints and Standards 9](#_Toc357526348)

[4.1 Safety 10](#_Toc357526349)

[4.2 Security 10](#_Toc357526350)

[4.3 Target Platforms 10](#_Toc357526351)

[4.4 Development Tools 10](#_Toc357526352)

[4.5 Project Requirements 10](#_Toc357526353)

[4.6 Statutory and Regulatory Requirements 10](#_Toc357526354)

[5 Delivery Requirements 11](#_Toc357526355)

[5.1 Delivery 11](#_Toc357526356)

[5.2 Post-Delivery 11](#_Toc357526357)

# Introduction

## Purpose of this Document

Defines the purpose of the URD, and specifies the intended readership of this version of the URD.

This document describes the user requirements for the Mantid Ticketing system, used to support the development of the Mantid project.

It is intended to be used to select the most suitable way forward from our current implementation of TRAC.

## Scope of this Document

This document describes the requirements for any system that would potentially replace our current TRAC implementation.

## Context of this Issue

This is the first issue of the document. It is intended for informal review by Stuart Campbell before being used to evaluate potential candidates for replacement.

## Definition of Terms

Carefully defines, or provides references to definitions of, terms, acronyms, and abbreviations used in the document.

Term definition.

SRD The Software Requirements Document, specifies the behaviour of the software system.

URD The User Requirements Document, catalogues the users’ requirements for the system.

# Functional Requirements

This section contains all the users functional requirements. Each requirement is prioritised as follows:

M Mandatory requirement. This feature must be built into the final product.

D Desirable requirement. This feature should be built into the final product unless its cost is too high.

O Optional requirement. This feature can be built into the final product at the Project Manager's discretion.

## Requirements for Tickets

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.1.1n | Tickets must be able to store and display the following:   1. A unique identifier 2. A short title 3. A description supporting markup (bold, lists, code blocks) 4. Who created the ticket 5. When the ticket was created 6. The release planned for this ticket 7. The priority of the ticket 8. The developer that is responsible for fixing the ticket 9. The Tester who has passed testing of the ticket   Source: Nick Draper  Issue: | M |
| U2.1.2 | Tickets should also be able to store and display the following:   1. Keywords or labels associated with the ticket 2. The component of Mantid that is affected by this ticket 3. The type of Ticket (enhancement, bug, task, help)   Source: Nick Draper  Issue: | D |
| U2.1.3 | Tickets would ideally also be able to store and display whether this ticket is blocked or is blocking another ticket.  Source: Nick Draper  Issue: | O |
| U2.1.4 | Tickets must be able to store a list of comments using the same markup as the description.  Source: Nick Draper  Issue: | M |
| U2.1.5 | It must be possible to record a number of different resolutions of the ticket such as fixed, duplicate, invalid, works for me.  Source: Nick Draper  Issue: | D |
| U2.1.6 | It should be possible o attach images and other files to tickets  Source: Nick Draper  Issue: | D |

## Requirements for Ticket creation

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.2.1 | New tickets should have sensible default values for the following:   1. Milestone (M) 2. Priority(M) 3. Component (O)   Source: Nick Draper  Issue: | See items |
| U2.2.2 | Users should be able to view a list of tickets created by them.  Source: Nick Draper  Issue: | M |
| U2.2.3 | You should be able to create tickets using script based tools.  Source: Nick Draper  Issue: | D |
| U2.2.4 | You should be able to create scripts via email. Certain values would need defaults.  The first example of this would be to automatically create “Help” tickets from any emails to [mantid-help@mantidproject.org](mailto:mantid-help@mantidproject.org)  Source: Nick Draper  Issue: | O |

## Requirements for developers

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.3.1 | Developers should be able to list tickets by the following:   1. Open tickets assigned to them (M) 2. Open tickets assigned to them , for the current release (M) 3. Openl tickets for the current release (D) 4. Tickets completed by them (D)   Source: Nick Draper  Issue: | See items |
| U2.3.2 | Developers must be able to add comments to the ticket via commit comments to the source code control system.  Source: Nick Draper  Issue: | M |
| U2.3.3 | The source code control system should link commit references to the ticket used.  Source: Nick Draper  Issue: | M |
| U2.3.4 | It must be possible to assign a ticket to another developer.  Source: Nick Draper  Issue: | M |
| U2.3.5 | It must be possible to assign a ticket as fixed, duplicate, invalid, won’t fix or works for me.  Source: Nick Draper  Issue: | M |

## Requirements for testers

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.4.1 | Testers should be able to list tickets by the following:   1. Tickets awaiting testing (M) 2. Tickets undergoing testing, this release (M) 3. Tickets undergoing testing, all releases (D) 4. Testing tickets assigned to them (D) 5. Tickets completed by them (D)   Source: Nick Draper  Issue: | See items |
| U2.4.2 | Testers must be able to select tickets to test in a way that prevents possible duplication of effort by other testers working at the same time.  This approach must not require repeated inspection of tickets to find one that is not already under test.  Source: Nick Draper  Issue: | M |
| U2.4.3 | Testers need to be able to pass the ticket, reopen the ticket, or abandon the testing of the ticket.  Source: Nick Draper  Issue: | M |

## Requirements for Project Management

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.5.1 | Users should be able to list tickets by the following:   1. Open Tickets arranged by developer (M) 2. All tickets for the current release (open, testing and closed) (M) 3. All tickets (D)   Source: Nick Draper  Issue: | See items |
| U2.5.2 | It must also be possible to create the following reports:   1. Ticket Burn Rate – tickets grouped by release and priority (M) 2. Tickets fixed by developer per release (D) 3. Tickets tested by developer per release (D) 4. Tickets by status (open, testing, closed) per release (M)   Source: Nick Draper  Issue: | See items |
| U2.5.3 | Testers need to be able to pass the ticket, reopen the ticket, or abandon the testing of the ticket.  Source: Nick Draper  Issue: | M |
| U2.5.4 | It should be possible to edit tickets in bulk.  For example to move a whole developers worklist to someone else, to move a collection of tickets to another milestone, to change the priority of a collection of tickets.  Source: Nick Draper  Issue: | D |
| U2.5.5 | It should be possible to list tickets within a release that have passed testing between two dates.  This is used to see what needs to be added to release notes after a further period of testing.  Source: Nick Draper  Issue: | D |

## Requirements for ad-hoc queries

| Label | Requirement | Necessity |
| --- | --- | --- |
| U2.6.1 | Users should be able to search tickets using simple search terms.  Source: Nick Draper  Issue: | M |
| U2.6.2 | The search should allow the state of tickets to be limited or grouped in the results, e.g. open tickets vs closed tickets.  Source: Nick Draper  Issue: | D |
| U2.6.3 | The system should support ad hoc queries, of numerous search terms: For example:  Tickets by developer A, for release 2.2 in a closed state and were fixed.  Source: Nick Draper  Issue: | M |
| U2.6.4 | The mechanism for ad hoc querying should not require specialized knowledge such as scripting.  Source: Nick Draper  Issue: | D |

# Non-functional Requirements

## Speed and Time Requirements

| Label | Requirement | Necessity |
| --- | --- | --- |
| U3.1.1 | Viewing a single ticket by identifier in a system containing 5000 tickets should take no longer than:   * 10 seconds (M) * 4 seconds (D) * 2 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.2 | Listing open tickets assigned to a developer in a system containing 5000 tickets should take no longer than:   * 10 seconds (M) * 4 seconds (D) * 2 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.3 | Listing all tickets in a release, in a system containing 5000 tickets should take no longer than:   * 10 seconds (M) * 4 seconds (D) * 2 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.4 | A search on a common word in a title, in a system containing 5000 tickets should take no longer than:   * 15 seconds (M) * 6 seconds (D) * 3 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.5 | A search on a common word in a description, in a system containing 5000 tickets should take no longer than:   * 20 seconds (M) * 7 seconds (D) * 4 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.6 | A new ticket should take no longer than:   * 5 seconds (M) * 3 seconds (D) * 2 seconds (O)   Source: Nick Draper  Issue: | See items |
| U3.1.6 | A change to an existing ticket (marking fixed, or passing testing):   * 10 seconds (M) * 4 seconds (D) * 2 seconds (O)   Source: Nick Draper  Issue: | See items |

## Capacity Requirements

| Label | Requirement | Necessity |
| --- | --- | --- |
| U3.2.1 | The system must support up to 20000 tickets, and at least 50 releases.  Source: Nick Draper  Issue: | M |
| U3.2.2 | The system must support 50 concurrent users  Source: Nick Draper  Issue: | M |

## Reliability Requirements

Here days are considered to be normal working days (not weekends), these outages include loss of service due to network outages.

| Label | Requirement | Necessity |
| --- | --- | --- |
| U3.3.1 | Unplanned outages per year   * Total 5 days, longest 1 day (M) * Total 3 days, longest 12 hours (D) * Total 1 day, longest 4 hours (O)   Source: Nick Draper  Issue: | See items |
| U3.3.2 | Planned outages per year   * Total 10 days, longest 3 day (M) * Total 5 days, longest 2 days (D) * Total 2 days, longest 1 day (O)   Source: Nick Draper  Issue: | See items |

# Design and Implementation Constraints and Standards

This section must contain all the following sub-sections, if there are no relevant requirements there must be a comment to that effect.

## Safety

Safety is not a concern for this system.

## Security

Lists any general security requirements for the system.

| Label | Requirement | Necessity |
| --- | --- | --- |
| U4.2.1 | Any identified user should be able to create a ticket  Source: Nick Draper  Issue: | M |
| U4.2.2 | Any user even anonymous should be able to view (but not edit tickets)  Source: Nick Draper  Issue: | M |
| U4.2.3 | Users identified as developers should be able to edit tickets  Source: Nick Draper  Issue: | M |
| U4.2.4 | Users identified as administrators should have full control and be able to define new releases etc.  Source: Nick Draper  Issue: | D |

## Target Platforms

| Label | Requirement | Necessity |
| --- | --- | --- |
| U4.3.1 | The system and interfaces (including ad hoc querying) should be usable from windows (XP+), Mac OSX and linux platforms.  Source: Nick Draper  Issue: | M |

## Development Tools

Describes any design tools, test tools, programming languages, packages etc. that the client requires to be used to develop the system.

| Label | Requirement | Necessity |
| --- | --- | --- |
| U4.4.1 | Any custom development should be done in C++ or Python  Source: Nick Draper  Issue: | D |

## Project Requirements

No specific requirements

## Statutory and Regulatory Requirements

There are no regulatory and statutory requirements relating to this system.

# Delivery Requirements

This section must contain all the following sub-sections, if there are no relevant requirements there must be a comment to that effect.

## Delivery

| Label | Requirement | Necessity |
| --- | --- | --- |
| U5.1.1 | Upon implementation all open tickets must be migrated from the previous system.  Source: Nick Draper  Issue: | M |
| U5.1.2 | Upon implementation all closed tickets should be migrated from the previous system.  Source: Nick Draper  Issue: | O |

## Post-Delivery

Describe any constraints or requirements the client has specified relating to activities which are to take place following successful delivery of the system

| Label | Requirement | Necessity |
| --- | --- | --- |
| U5.2.1 | Following implementation, all wiki references to the previous system must be updated, and workflows updated to reflect the new system  Source: Nick Draper  Issue: | M |
| U5.2.2 | The development team must be trained in how to work with any new system.  Source: Nick Draper  Issue: | M |