

Quality Assurance Guide

**For Internal Use Only**

Version 0.1.1

Last Updated: 16 June 2015

Table of Contents:

i. [Introduction](#id.brx417crvd70)

ii. [Glossary of Terms](#id.blmxs8cycf1l)

1. [Updating the QA Ticket](#id.5aab9zkmo42p)
2. [Preparing the QA Report Sheet](#id.gvzufboe0nas)
3. [The Design Brief](#id.xcjrna9bve1u)
4. [The Project Scope](#id.tqf3zxdn9rxg)
5. [Backend Testing](#id.jy7cc0zedkc6)
6. [Design Testing](#id.1f2y43yhuwwr)
7. [Cross-Browsing](#id.odvmwmqmivho)
8. [Mobile Testing](#id.rfkhk4ri3m0r)
9. [Launchlist](#id.8yjogyh9pm4p)
10. [Fix Verification](#id.ejmzac93nd87)
11. [Cleanup and Completion](#id.ovzy3s3epi91)
12. [Summary](#id.s7b38tjaszyi)

Introduction:

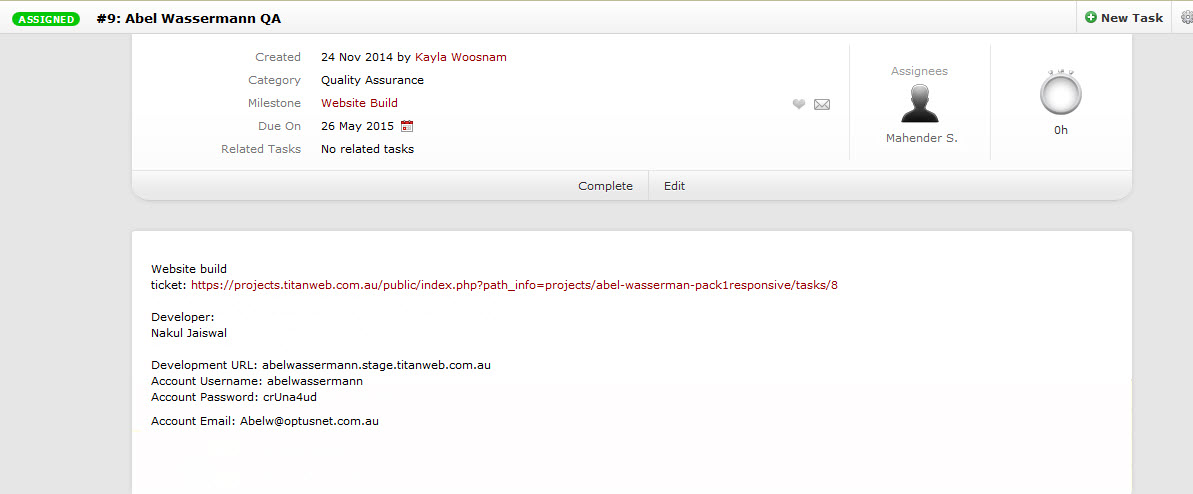
To be updated in version 0.5 of the document

Glossary of Terms:

To be updated in version 0.4 of the document

I. Updating the QA Ticket

The first order of business when a project arrives is to update the QA ticket which you have been assigned to. The label should be changed to INPROGRESS



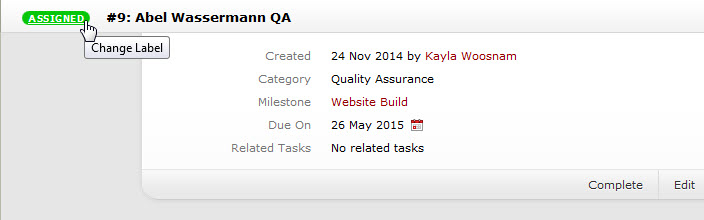
*When the ticket has just recently arrived, it should look like this*

\*\*\*

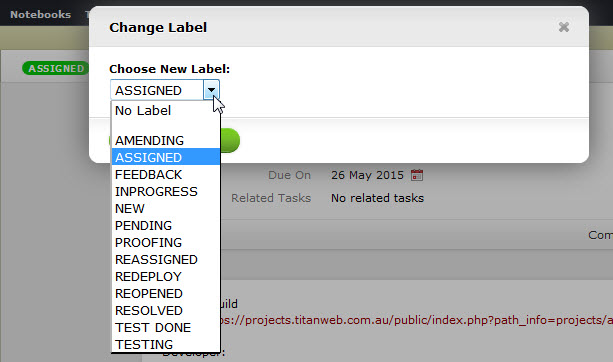
Steps

\*\*\*

1. Click the green assigned button in the top left



1. Expand the drop-down menu that appears



1. Select INPROGRESS and click on the **Update Label** button

A short overview of the labels:

For the QA process, only four labels are ever used.

* Assigned - This label is used to assign the task back to the developer
* Feedback - This label is used by the developer when returning the project - The QA should never use this label
* InProgress - This label is used to indicate that the project is being worked on
* Resolved - This label is used at the end, when all the bugs have been fixed

These labels will be revisited and explained in more detail in the sections where they are relevant.

II. Preparing the QA Report Sheet:

To begin with, it is important download a new copy of the QA Report Sheet template and the testing files in the following link:

<https://knowledgebase.titanweb.com.au/documentations/web-qa-report-sheet-templates>

This should contain:

* The Pre-Live Testing Sheet
* The standard text content for testing
* The testing PDF file
* The testing image file

Pre-Live Testing Sheet

This is the main sheet where most of the project’s development details are collated, including but not limited to the Project Information, the Project Scope, the Launchlist and the bug reports. These files should not be reused for new projects, another sheet should be downloaded.

Placeholder Data

Not just any data should be inputted and uploaded to the site over the course of development. Indecent images and foul language should be avoided at all times, and if the developer has left these on the site himself, it should be reported to management immediately. As the staging sites are already online, the client is capable of checking it at any time and it is best to err on the side of caution.

**Sample Text**

The attached sample text is notable for two reasons. Firstly it covers all types of text input a client is likely to enter in a WYSIWYG (what you see is what you get) field, namely:

* Paragraph text
* Numbered/ordered lists
* Bulleted/unordered lists
* Nested lists

Secondly, it is labelled properly, clearly showing that the text between the labels were added by the QA. This is important as if there is any lorem ipsum or otherwise misplaced text left after project completion, we can easily argue that all our test data is labelled and the misplaced text cannot possibly be ours.

**Sample PDF**

This PDF is a simple placeholder that should be used whenever testing sites that have PDF downloads. The reason behind this is simple, it is devoid of content and is clearly a placeholder. Thus when test data is being removed, it’s simple to spot.

Even if it’s missed, it’s harmless enough not to cause offense.

**Sample Image**

This image is ideal for testing for two reasons, its large image size and its relatively small file size. This means that if the developer isn’t anticipating user input properly, this image will break the layout quickly. Because of the file size, it isn’t resource hungry and will upload and download quickly, reducing our downtime.

Additionally, its dimensions are the very maximum that one of our more commonly used plugins, Slick, is capable of supporting.

\*\*\*

Steps

\*\*\*

1. The first step after downloading the sheet is to create a new folder with the same name as the project.

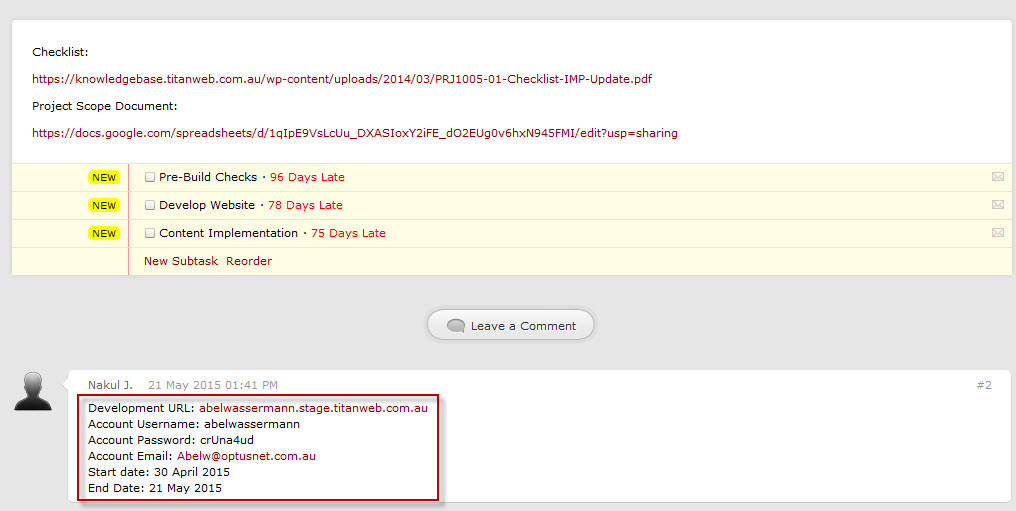
Project Sheet - 03.jpg

1. Move the the pre-live testing document into the newly created folder. Rename the document. The proper format should be “**Project Name - Pre Live Testing Document**”

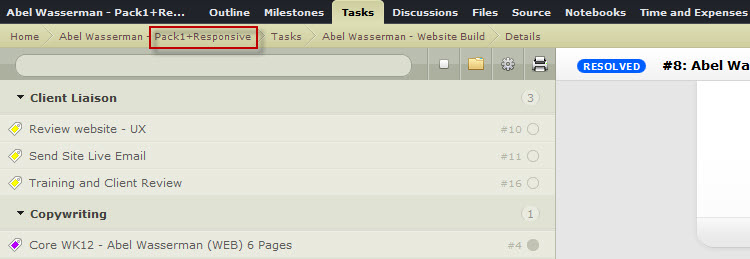
Project Sheet - 01.jpg

1. A new folder should be created and renamed “Screenshots” this is where all the screenshots for the project should go
2. Open the pre-live testing document
3. Open the **Project Info** tab and fill in all the fields, the required data for the first table should be located in the Website Build ticket which is linked to in the QA ticket.

For the second table, enter your full name in the **QA Tester** field and enter the current date in the **Start Date** field.



*The Project Details are usually located in a comment in the Website Build ticket*



*For the Package field, check the top left of the ActiveCollab screen*



*If there are details missing, like the Implementer’s Due Date, just leave them blank as shown above*

Guidelines on the fields:

**Project Name**  : Ensure proper spelling and capitalization

**Development URL** : Ensure that the staging link is used

If there is no ‘http://’ at the beginning of the URL, add it.

**Package**  : All projects are now Responsive even if not indicated

**Account Username** : This refers to the client editor account

The KB standard regarding it is [here](https://knowledgebase.titanweb.com.au/implementation-practices/admin/client-user-account)

**Account Email** : The email used should always be admin@

The KB standard regarding it is [here](https://knowledgebase.titanweb.com.au/implementation-practices/admin/client-user-account)

If another email is used, report it as a bug

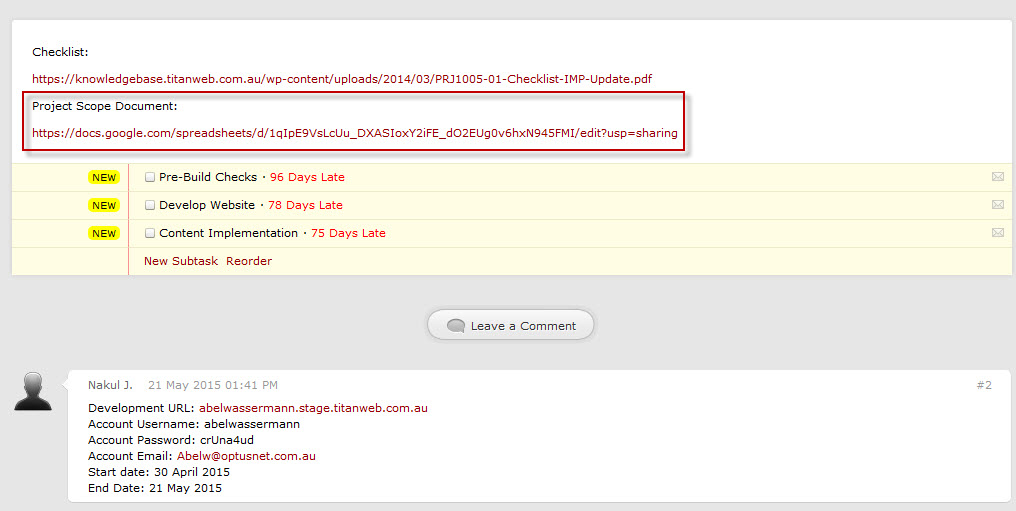
**Implementer** : Use the full name of the Implementer

**Due Date** : This is the Due Date for the Implementer

**Start Date** : This is the Start Date for the Implementer

**End Date** : This is the End Date for the Implementer

1. On the **Website Build** ticket, check if there is a **Project Scope Document** link included



If yes:

* 1. Open the **Project Scope** tab in both Google Sheet and the Pre-Live Testing Document
  2. Copy the contents of the Project Scope tab of the Google Sheet to the Project Scope tab of the Pre-Live Testing Document
  3. You may close the Google Sheet after this

The reason behind copying the scope is so that we have an independent copy of the Project Scope. Thus, if the original scope is altered in any way after being given to the QA, we can refer to our copy in case of any disagreement with the developers.

III. The Design Brief

The design brief is a short document created early in the project’s life cycle. It is usually worth looking through the design brief to get a quick look at what the client was initially looking for when they commissioned the site.

Beyond that, there are several sections of the design brief that are of particular interest to the Quality Assurance process.

**Company Details:**

One of the more common mistakes done by a new developer is the use of the Company Details section of the design brief for data used in various sections of the website. This shouldn’t be allowed as the contents of the Company Details section is created for the use of the Sales team to coordinate with the client’s representative.

Thus, the details contained within might not refer to the person who will use any part of the website at all. This is especially true for the contact details, as these may be the personal email and phone numbers of the person negotiating sales.

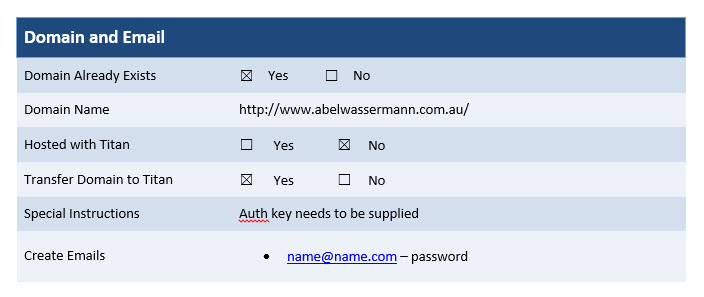


**Domain and Email:**

The Domain and Email section of the design is notable for two features only. The **Domain Name** and the **Create Emails**.

The **Domain Name** is the eventual URL of the site once the site is launched. Thus all emails, like info@, will use this domain name. So in this example, every instance of **info@** in the design brief actually mean **info@abelwassermann.com.au**.

The **Create Emails** field isn’t always filled in, but when it is, it lists all emails that should be created for the client.

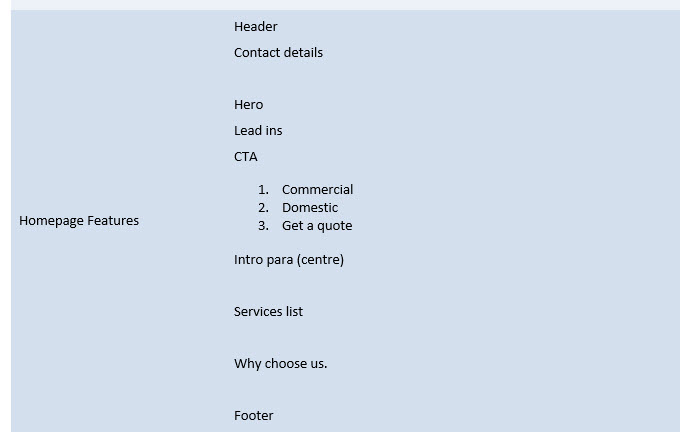


**Design Information & Feature/Page:**

The Design Information section also has two points of interest, the first is the **Homepage Features** section and the **Internal Page Hero banner size** section.

It should first be noted that all information shown in these sections **shouldn’t** be adhered to strictly. These sections are created to give the designer a general idea of what the client is looking for. The designer may opt to add or remove page features as he sees fit, and that’s fine. In most cases, if there is any disagreement between the Design Brief and the actual design, the actual design takes precedence over the Design Brief.

The point of looking at these, instead, is that in case there is a particular page element or functionality that you’re unsure of, you may look at these sections of the design brief in hopes that it is explained more thoroughly there.



*Design Information Sample*

The **Feature/Page** section is generally a bit more important than the Design Information section.

IV. The Project Scope

V. Backend Testing

To be updated in version 0.2 of the document

VI. Design Testing

To be updated in version 0.2 of the document

VII. Cross-Browsing

To be updated in version 0.2 of the document

VIII. Mobile Testing

To be updated in version 0.2 of the document

IX. Launchlist

To be updated in version 0.2 of the document

X. Fix Verification

To be updated in version 0.3 of the document

XI. Cleanup and Completion

To be updated in version 0.3 of the document

XII. Summary

To be updated in version 0.3 of the document