Software Engineering Group Projects – User Interface Specification Standards

Author: C. J. Price
Config Ref: SE.QA.04
Date: 22nd December 2016
Version: 1.0
Status: Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1 INTRODUCTION

1.1 Purpose of this Document

The purpose of this document is to describe the format of, and information which must be supplied in, user interface specifications produced in software engineering group projects.

1.2 Scope

This document specifies the standards for writing user interface specifications. It describes the components that make up a user interface specification, and how they are specified and delivered.

This document should be read by all project members. It is assumed that the reader is already familiar with the QA Plan [1].

1.3 Objectives

The main objective is to aid the production of a user interface specification which covers all of the requirements of the system being built, and which makes clear what the user can see and do at any point in their interaction with the system.

2 STRUCTURE OF THE SPECIFICATION

The User Interface Specification has two parts:

1/ A document which describes the main use cases of the system. The basic layout and information content of this document must conform to the general documentation standards [2], which, amongst other items, specifies that there must be an introductory section and a references section.

2/ An interactive online presentation which shows the user experience for each of the use cases. The presentation should describe what is happening at each point in the interaction, and show its results on the screen. It would not be sensible for this presentation to try to meet the general documentation standards [2], but it should be given a version number and be stored in the project repository [3].

A more detailed description of each part of the user interface specification is given in the following sections.

3 Use Case Document

This document will have the following specific sections:

- Typical Users. Many applications have more than one kind of user. For example, the university web site has to provide services for staff, for present students, for potential future students, and several other kinds of users. Each kind will have different needs from the site. You should think about each kind of user, and describe their needs.

- Use cases. For each kind of user, there will be a set of tasks that they wish to carry out using the software. You should list all of the tasks, and for each of the tasks, write a description of how it is carried out - what the user has to do to initiate the task, and how they then interact with it.

- Error conditions. Showing all of the error screens in the online presentation would get tedious, but you do need to think about them. Once you have the online presentation, you will have thought how the user interface should look to the user. You should then go through each use case, and think what could go wrong and list what is done about it here.

3.1 Typical Users

You should identify each possible type of user. Write about both the general user class, and specific types of user that cover the class. For example, a web site for parents of new babies might have two general classes of users. The first would be people who contribute articles to the site. They have a different set of needs from the second general class - people who read news on the site. Within those general classes, there might be sub-classes - so some contributors of articles might have extra responsibilities, perhaps they are able to delete articles or rearrange what is emphasised on the site.

When you have these general classes of user, you can then describe what different typical users want to do. It is helpful if you give the typical user some background detail so that you get a fuller picture of what is important to them.

For the above example, one typical contributor might be depicted this way:

> Jennie is a mother of two who writes about her experiences as the mother of a Downs syndrome child. She prepares her articles in Word and submits them to the web site by cutting and pasting them into an online form. This preserves the structure of her article, but the formatting is lost, and she then has to add any formatting in the online html editor. Having done this, she checks that the spelling and grammar are still good, and that the article fits within the word limit before submitting it for publication on the site.

For some applications, e.g. many games, there is only one real type of user. Think of Pokemon Go for example. In that case, this section is likely to be very short, but for other applications it is very important to identify the needs of all of the types of users.

3.2 Use cases

Having talked about typical users, you now need to decide and to document the tasks that each type of user will complete with the application. For an application like the the new baby web site mentioned above, this might be something like the following:

Use cases for contributors:

- Logging in
- Submitting an article
- Editing an article
- Deleting an article
- Changing my profile

Use cases for editors:

- Logging in
- Adding new contributors
- Deleting contributors
- Featuring articles on the website
- plus all of the contributor use cases

Use cases for readers

- Selecting and reading an article
- Marking articles as favourites
- Finding a favourite article to read again

Each of the use cases should be given a reference number, so that they can be unambiguously referred to in the online presentation.

There is a diagrammatic format for UML Use Cases, but it is optional in this document. Much more important is that there is a clear description of what happens with each use case - what does the user do to initiate the use case? what do they see then? What do they have to do to complete it?

For example, in the submitting an article example discussed under typical users, the following description might be given.

> Use Case 1.2 Submitting an article
>
> When a contributor wishes to submit an article, they must first log on with their contributor ID (see use case 1.1 for how to log on). They will then see a Submit button, and should press it. This will bring up a form prompting them for article title, and giving them a choice of subjects. They fill in a title and subject, and the system will also record the author name and the date. A HTML editor will then appear and they can fill in the contents of the article - either by typing, or by pasting from another application. Formatting options, previewing and spell checking are available as part of the editor. When the contributor has completed the article, they can press the Publish button to make the article available. The web site will give a confirmation, and return to the main contributor screen.

3.3 Error conditions

A more detailed description of each part of the user interface specification is given in the following sections.

4 Online presentation

The online presentation will be a mock-up of your application that will give visual representation of each of the significant steps in each use case, along with brief notations of what is happening. This will give a good idea of how the final product will look and feel without the expense of having built the product at this point.

The default way of doing this would be to use PowerPoint. The lectures on the course will give examples of how this might be done. It is permissible to use other more specialised tools to build your user interface prototype, e.g. there is a Chrome plug-in called Moqups which assists in building mock-ups of web sites. However any such tool needs to be able to deliver the mock-ups independently of having to install the tool (so if you don't use PowerPoint, you need to use a tool that gives stand-alone deliverables of the output).

It is important that your mock-ups look as believable as possible, so if the ideal size for your site was 1024 by 720, then your buttons etc. should show at the size they would be on such a site. It is also important that your content looks as much like the real application as possible. So if your application would have a list of articles plus more detail on one or two articles, then you put the detail of what that would look like in the mock-up.

Examples of online presentations will be given in the lectures.

REFERENCES

[1] QA Document SE.QA.01 - Quality Assurance Plan.

[2] QA Document SE.QA.02 - General Documentation Standards.

[3] QA Document SE.QA.08 - Operating Procedures and Configuration Management Standards.

DOCUMENT HISTORY

Version	*CCF No.*	*Date*	*Changes made to document*	*Changed by*
1.0	N/A	22/12/16	New document	CJP