E-vote 2011

SSA – U Appendix 5

Testing and Approval

Project: E-vote 2011

Change log

Version	Date	Author	Description/changes
1.0			

CONTENT

1. Introduction

In this document we'll first elaborate on the requirements as specified in Appendix 2B, i.e. ST4.4 and M1 – M5. In the last chapter we make the necessary specifications as requested in the general software development agreement.

2. Elaboration on requirement ST4.4: The Tenderer’s Test Strategy

Continuous Testing Throughout the Project!

2.1. Introduction

The purpose of all testing is to verify that the Customer receives the deliveries as requested without errors. The Tenderer’s test methodology encompasses all work related to the whole test process: the planning, the documentation, the actual testing, the registration of deviations and the final reporting.

The Tenderer recommends that the involved parties agree upon a test strategy during the initial phase of the project. A test strategy should include a list of tests and at what stage of the project they will be applied. The test strategy should also clarify the division of roles and responsibilities for the testing.

The Tenderer recommends the Customer to take greater responsibility for the testing compared to what has traditionally been seen as normal. No one is better suited for writing a test manuscript or verify delivered functionality that a business users or an end users, or their representatives. When the Customer participates in the project, we get, at a very early stage, a sense of whether the project is going the right direction or not. At the same time, the Customer gets a sense of how the final system will look like, and they can do corrections as we go. This kind of tight cooperation is of vital essence in order to achieve the proper quality at the right time.

2.2. Types of Tests and Levels of Tests

The Tenderer divides tests into the following types and levels of tests. (Hence, we use the same types and levels of tests as defined in IEEE 610-1991):

Unit Test – a detailed test on the lowest level used to verify that the different units behave in accordance to the specification.

Integration Test – a test of the functionality that the units can execute together

System Test – when the whole system is integrated in a test environment, one may perform a System Test. The test environment is a simulation of the runtime environment.

Installation Test – the purpose of this test is to find deviations between the real procedure for installation and the documentation that describes the installation procedure. This applies to both the initial installation as well as the installation of updates.

Acceptance Test – the Customer’s test of the final system. The main purpose of the test is for the Customer to verify that the terms of the contract is fulfilled, and thereof accept the deliveries.

2.2.1. Unit Test

The purpose of unit testing is to focus on all the identified units of the system, and search for malfunctions in each one of these individually. This testing on the lowest possible level verifies that each unit behaves as expected. It is the developer’s responsibility to write and execute unit tests.

The Tenderer's developers use Test Driven Development (TDD). TDD was first introduced in the agile development method known as Extreme Programming (XP). This method also highlights the importance of refactoring the code in order to continuously improve the quality, reduce the number of duplications, increase maintainability etc.

The basic principle of TDD is to write the test first, even before the code itself. The routine is to write a test, then write the code that the test is testing, make sure that the test runs without errors, write a new test, write the code that the test is testing etc.

The main advantages of this method are:

- The Unit Tests are actually written – it is all too easy to skip writing Unit Tests when the code is running
- Enforcing the usage of the interfaces and services helps the developer make better interfaces and services.
- The test serves as a documentation on how to use the interfaces.
- The Units Tests provides a repeatable, automatic and tangible verification that the system works.
- The set of Unit Tests provides the developers with the self confidence that is necessary in order to dare to do changes in the code. If a developer needs to do some changes to the existing code and the changes make the unit tests fail, (s)he knows immediately that the change cannot go through.

2.2.2. Integration Test

The purpose of an Integration Test is to focus on the connections and interfaces between several individually tested units. The attention is on the cooperation of the units, and it is the functionality that the units only can perform together that is the scope of the test.

During Integration Testing more and more components become a part of the test, and hence, one may eventually run non functional tests like performance tests, robustness tests, stability tests, etc.

The Tenderer use continuous integration and automatic testing in all projects.

It is the developer's responsibility to write and execute Integration Tests.

2.2.3. System Test

The System Test is carried through by the Tenderer in order to verify that the system contains all the functionality that was agreed upon at the proper quality. The System Test contains both functional and non functional tests.

The Tenderer is responsible for planning and executing all the tests that make up the System Test. The Customer is responsible for making available business users or end users that can test the functionality from a business perspective. If suitable for the purpose, the Customer may participate in some technical tests, e.g. tests where knowledge of another IT system on the Customer site is vital.

During testing the deviations are to be registered without further delay. The deviations could be comments on the test manuscripts or issues registered in the issue tracking system.

In the period of the System Test, the Customer and the Tenderer meet regularly to look through all the issues that have been registered as well as their status. In these meetings the parties discuss and prioritize the issues, and agree whether the issue at hand is a deviation or a change.

The System Test is a kind of a “dress rehearsal” for the Acceptance Test. It is executed by the Tenderer in order to verify that the deliveries are mature enough for the Acceptance Test.

The System Test is legally bound by the test plan, which is the Customer and Tenderers common document which contains, among other things, the test manuscripts and the criteria for approval of acceptance test.

The System Test is the responsibility of the Tenderer. However, customer participation is crucial for a successful result.

2.2.4. Acceptance Test

The main purpose of the Acceptance Test is for the Customer to verify for himself that the terms and conditions of the delivery are fulfilled. The Acceptance Test is the Customers responsibility.

The Tenderer will evaluate reported issues during the Acceptance Test and correct and retest issues in the same manner as in the System Test.

The system is formally delivered to the Customer when the Customer has finished the acceptance test, and accepted the system.

2.3. The Test Process

2.3.1. Planning the Test

Planning the test is a vital phase of the test process. A widely used method for the planning and running of tests is the “V-model”. We would like to start with a few comments to this traditional V-model which is often mentioned in many beginners books about testing. The nice part of this model is that it proposes that general criteria for acceptance can be clarified at the same time as the specification of the system (the test first principle). However, the V-model has the same weakness as the traditional waterfall method. It assumes that the conditions at the end of the project are the same as in the beginning of the project. The awareness of the learning process constantly taking place in the project is ignored here, as in the waterfall method. This applies if one considers the V-model as valid for the whole project.

However, the V-model could be simplified and used as a “Recurring-V-Model”, having one V per iteration. Using the V-model this way, it gives a picture of the validation and verification which takes place within each iteration. No matter which development methodology one may adhere to, it is always recommended to focus on finding the deviations in the stage where they are created. A focus on testing on a low level will be profitable for the project. The “Recurring-V-model” looks more like a W and the Tenderer uses a W-model as shown in the figure below:

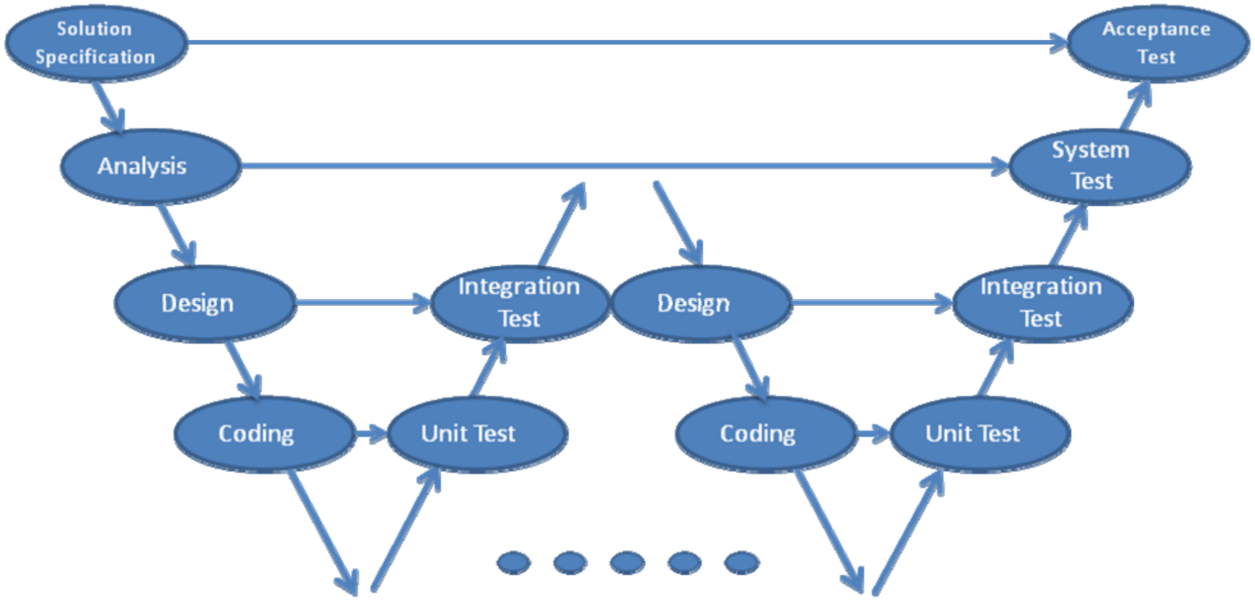

Figure 1: The W-model

The plan for the Acceptance Test is usually elaborated during the solution specification phase; the plan for the System Test is created in the early stages of the development phase, while Integration Tests and Unit Tests are planned in each iteration.

In addition to planning the different types of tests, one must also make a selection of tools and methods, consider training, documentation and the allocation of test resources. The Tenderer has developed a template tool for the test plan in order to ease the creation of a test plan and ensure a good result by the end of this vital phase.

2.3.2. Test Specification

Writing documents specifying rules and regulations for testing on all levels, the creation of a document structure, templates for specification of different kinds of tests, e.g. functional tests, performance tests, installations tests etc. are the main tasks of the Test Specification.

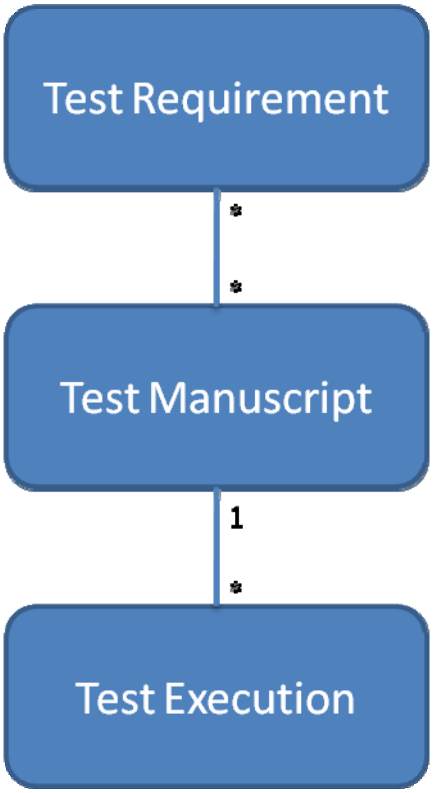

Figure 2: The relationship between Test Requirements, Test Manuscripts and Test Execution

Usually, functional tests are based on spesific test requirements. One or more test requirements are covered by one or more test manuscripts. The test manuscripts are executed, and the result is registered in a "Test Execution".

Test Requirement

A Test Requirement describes a requirement that the system must fulfill. This may be functional or more technical or other non functional requirements.

Test Manuscript

A Test Manuscript contains a step-by-step description on how to test the different Test Requirements in the system. A Test Manuscript should contain the following:

- Pre conditions, e.g. what kind of data is needed before you start the test
- How the functionality should be tested, step-by-step
- Post conditions, e.g. what results are to be expected.

Test Execution

A Test Execution is used to register that a Test Manuscript has been executed, and if there are any issues they are registered and linked to the Test Execution. Normally one would repeat running Test Manuscripts until all of them can be run without discovering any new discrepancies.

2.3.3. The System Test

One of the features of agile methodology like Scrum is that testing can be done as soon as a demo of the system is available. This underpins the principle of testing as early as possible in order to reveal deviations early. Our experience is that this reduces the overall cost of the project. By the end of each Sprint, and usually even more often, new parts of the system are finished and ready for production. Tasks that traditionally have been thought of as parts of the System Test can now be done continuously through the development project. This means that, when using an agile method, one may rightfully expect that the final System Test will be less extensive than usual.

There will still be a need ahead of a larger delivery to consolidate the complete system, and perform a more comprehensive test. This test will address both the functional and the non functional aspects of the solution. Life cycle testing and value chain testing will, among others, be the kind of tests that are the most suitable to run as part of the System Test.

The System Test may itself be arranged as one or more Sprints.

The Test Manuscripts should be written as a part of the development sprints. When the System Test is about the begin, the Test Manuscripts and other non functional parts that need testing should be put in a separate backlog called “Test Backlog”. All the elements in this backlog will be attached to, and underpin, the testing of elements in the Product Backlog.

The elements in the Test Backlog should, as the elements in the Product Backlog, be arranged in a prioritized order. We recommend letting the priority reflect the risk connected to the particular element. This means that each test object is evaluated according to the probability for an error as well as the consequences of an error in this test object while in production. The priority is set according to number you get when multiplying the probability with the consequent.

It is possible to save quite a lot of time if the documentation of the project, e.g. use cases, user stories, or business rules, are written in such a way that it is possible to use this documentation almost like Test Manuscripts.

The System Test may be arranged as one of more System Test sprints, where the backlog for the System Test Sprint consists of the top priorities from the Test Backlog. It should be a separate System Test sprint dedicated to non functional testing. Quite early in the System Test some parts of the development team should be dedicated to correct the errors that have been discovered earlier on. Doing it this way, we will get the test teams discovering errors and issues going into the issue registration system, which will in turn be dealt with by the correction team in prioritized order.

Ahead of a successful System Test the following needs to be in place:

1. A freeze of the functionality. This basically means that no new elements from the Product Backlog will be developed during the System Test.
2. A backlog with Test Manuscripts and test cases should be established, the Test Backlog. This backlog is the list that provides the System Test with test related tasks. The Product Owner is responsible for the Test Backlog.
3. A backlog for issues like deviations, discrepancies and errors should be established, The Error Backlog. This backlog provides the correcting development team with tasks.
4. Establish an organization consisting of several teams with complementary areas of focus.

5. A test plan for the complete System Test. The responsibility for the creation of the test plan and the System Test is the Test Leader of the project. The Test Leader has the role of a Scrum Master during the System Test.

Freezing functionality

The purpose of the System Test is to prove that the solution is ready for the Acceptance Test, find discrepancies, deviations, and bugs and correct these. During development sprints the main focus is the development of functionality specified by elements in the Product Backlog, whereas in the System Test the focus is verification, testing, and fixing bugs. The sprint backlog which earlier on was populated with items from the Product Backlog, is now populated with items from the Test Backlog and the Error Backlog. Please note that elements in the Product Backlog no longer will find its way into a Sprint Backlog and hence, the functionality has been freezed.

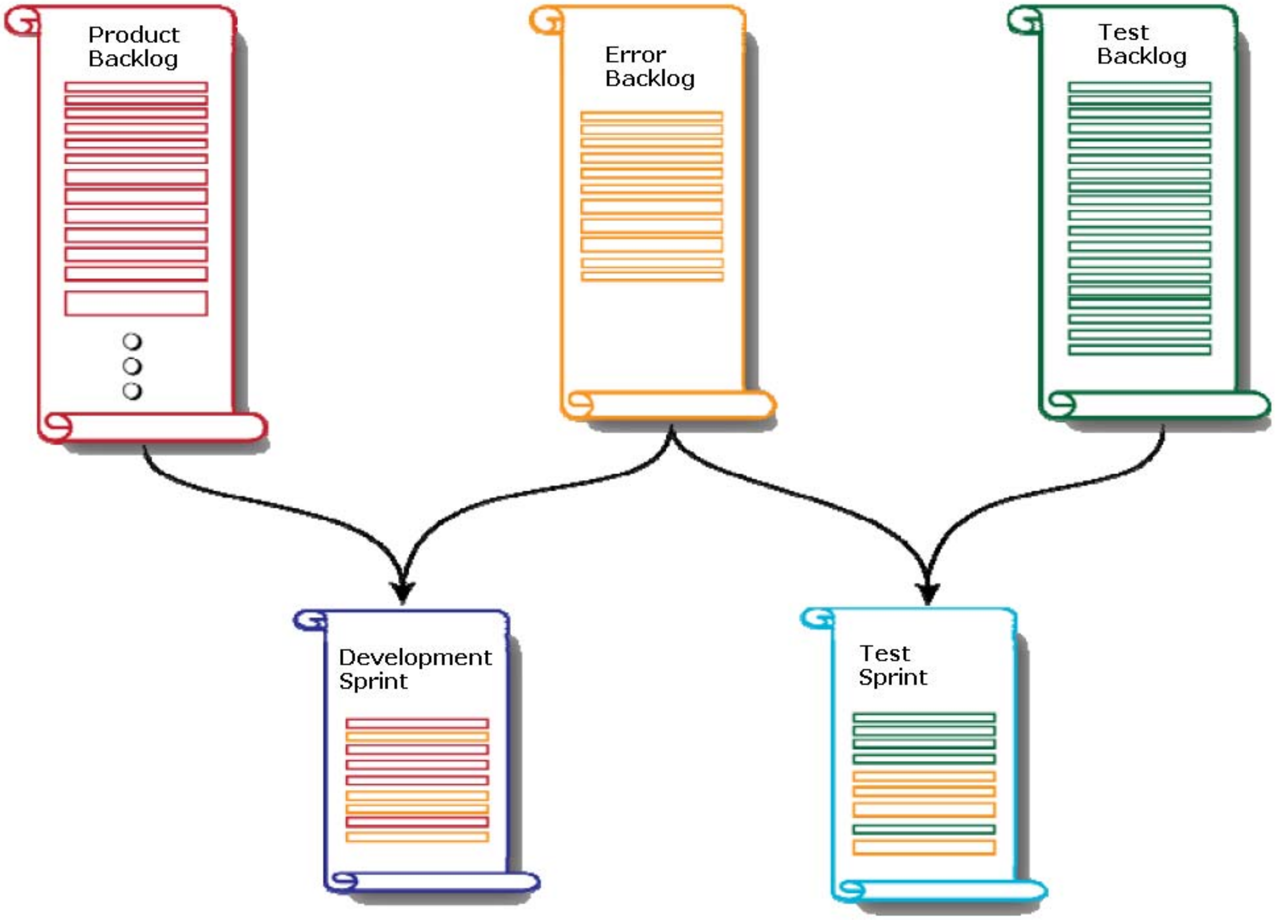

Figure 3: Development Sprint versus Test Sprint

Under certain ad hoc circumstances one may consider t to move elements from the Product Backlog into the Test Sprint Backlog. However, this is not recommended.

What Is Done During the System Test

The System Test is the comprehensive test testing all aspects of the solutions. It will not be enough just to run the Test Manuscripts again, something which has probably been done many times during the sprints anyway. Looking more closely at the Test Manuscripts one has to search for the limits of the solutions. What has not been covered by the narrow Test Manuscripts? The list below includes different kinds of tests that look beyond a single Test Manuscript.

- **Lifecycle Test.** A lifecycle test is an extensive test which deals with an object's lifecycle in the system and the events that may occur. One example could be a lifelong customer relationship in a bank where all kind of bank services are used in different combinations. A lifecycle test represents and tests organic complexity and unpredictability.
- **Value Chain Testing:** Another extensive test which searches to verify the steps of a predefined process or a pattern of behavior and reveal accidental deviations or errors in this respect. Typically several sub systems take part in this kind of testing in order to verify that they works as expected as a whole. This could for example be a loan application process that starts in a portal, continues to a business application, uses some external interfaces etc. In addition to a test of the standard process, alternatives should be testes as well
- **Random Testing.** Random testing means using the system in a creative way in order to provoke errors, or simulate users who believe themselves that they are doing the right thing. This is done in order to find new errors, and to test the general robustness of the functionality of the system.
- **Usability Testing.** This test is used to check the user's ability and efficiency in respect to understanding and using the system as well as understanding the documentation, without further assistance. This test focuses on human factors like esthetics, consistency, online assistance, tutors, user documentation and training material. The actual test could be performed as an expert evaluation or a user test or a combination of these. User tests could be observations or a combination of observations and interviews.
- **Large Scale User Testing.** This test's purpose is to identify bottlenecks in the system as a consequent of several concurrent users. There could be functional challenges related to deadlocks, but also non functional problems like response time. These kind of tests are often done with tools, i.e. software, which have the ability to simulate a real traffic of more than a 1000 users.
- **Regression Test.** A regression test tests previously corrected errors.

The System Test also includes some non functional tests:

- **Performance Test.** In this test it is the response time and efficiency of a machine, system or network that is tested in order to verify that it works according to the requirements. This test is not very different from Large Scale User Testing, but one is more free in the simulation of work, and it is usually an easier test to do.
- **Volume Test.** The Volume Test looks at how the system deals with heavy load, what limits the system has, which components reaches their limits first and what will happens if new components are added.
- **Security Test:** A Security Test tries to ensure that the system adheres to the data security in a sufficiently good way and in accordance with the requirements.
- **Technical Robustness Test**. What happens if the network is lost? What happens if a machine goes down etc.

2.4. Test Tools

The Tenderer uses different test tools for different purposes.

JUnit - http://www.junit.org/
JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development.

TestNG - http://testng.org/doc/index.html
TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use. TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc...

Selenium - http://seleniumhq.org/
Selenium is a suite of tools to automate web app testing across many platforms.

HtmlUnit - http://htmlunit.sourceforge.net/
HtmlUnit is a "GUI-Less browser for Java programs". It models HTML documents and provides an API that allows you to invoke pages, fill out forms, click links, etc... just like you do in your "normal" browser.

It has fairly good JavaScript support (which is constantly improving) and is able to work even with quite complex AJAX libraries, simulating either Firefox or Internet Explorer depending on the configuration you want to use.

It is typically used for testing purposes or to retrieve information from web sites.

HtmlUnit is not a generic unit testing framework. It is specifically a way to simulate a browser for testing purposes and is intended to be used within another testing framework such as JUnit or TestNG

jWebUnit - http://jwebunit.sourceforge.net/
JWebUnit is a Java-based testing framework for web applications. It wraps existing testing frameworks such as HtmlUnit and Selenium with a unified, simple testing interface to allow you to quickly test the correctness of your web applications

Apache JMeter - http://jakarta.apache.org/jmeter/
Apache JMeter is open source software, a 100% pure Java desktop application designed to load test functional behavior and measure performance. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load

FitNesse - http://fitnesse.org/
FitNesse is a software development collaboration tool, a software testing tool, a wiki and a web server. FitNesse enables customers, testers, and programmers to **learn what their software should do**, and to automatically compare that to **what it actually does do**. It compares customers' expectations to actual results. From another perspective, FitNesse is a lightweight, open-source framework that makes it easy for software teams to: 1) Collaboratively define Acceptance Tests -- web pages containing simple tables of inputs and expected outputs. 2) Run those tests and see the results. It requires no configuration or setup.

Techsmith Morae - http://www.techsmith.com/morae.asp
Morae is a usability testing and market research software. It is the premier software for deeply understanding customer experiences…and sharing those insights clearly and powerfully. From usability testing to focus groups and beyond, Morae helps you transform designs and marketing to make things people love. Morae makes it easy to pinpoint usability problems in software applications, websites, prototypes, or mobile devices.

2.5. Registration

The Tenderer suggests that JIRA[1] is used for the registration of issues discovered while testing.

The registration of issues serves mainly two purposes:

- A tool made specifically for issue registration is suitable for reports on progress and status, and the organization of the work related to deciding whether an issue is a change or an error.
- The registrations are a documentation of the quality of the testing. The registrations can be measured on different dimensions and this knowledge could be used to improve the overall testing, the test methodology, ways of testing etc.

The Tenderer has experience with other tools as well, and the decision as to which tool to choose in this project should be made in cooperation in the early stages of the project.

2.6. Validation of Quality

An important part of our test methodology is to validate the quality of the testing. This validation is also vital for the quality assurance of the project. Validation of the quality of the testing includes an investigation to check the test results. Is the test coverage and the test efficiency good enough?

The acceptance criteria for the test results are drawn up in the development phase, and in the final stages of the project the parties will validate whether the expected test result was achieved or not.

The test coverage measures to what extent the test cases actually verifies all specifications for a given test object, or to what extent all code has been tested during the test.

Test efficiency is another way of measuring test quality. Test efficiency focuses on a specific test level and measures the efficiency of e.g. the unit tests or the integration tests, and calculates how efficient the tests were by investigating what share of the errors that were covered in percentage of the number of errors that later was discovered in the system later.

$$\text{Test Efficiency} = \frac{\text{Number of Errors in Interation} * 100}{\text{Numer of Errors in Iteration} + \text{Number of Errors discovered later}}$$

[1] http://www.atlassian.com/software/jira/

2.7. Test Reports

The observations and experiences gained in the project regarding the test methodology, the test roles and responsibilities, the test results etc. are summarized in a test report by the end of the project.

This test report serves two purposes. Firstly it’s an evidence of the testing that has been carried out in the project, e.g. the test rapport which is the result of the Acceptance Test. Secondly, the report contains experiences which are used when planning the test process in a new project.

3. Accessibility and Usability

3.1. Elaboration on Requirement M1

The supplier shall document how accessibility and usability ties in with overall the development process/methodology.

As part of the Product Development Process and methodology, extensive consideration is given to Accessibility and Usability requirements.

The Tenderer will focus on usability and accessibility through the whole project. The different parts of the complete E-vote 2011 solution, will follow the same methodology for ensuring a high quality system, also in the field of usability and accessibility. The internet voting part of the solution will, however, require more extensive use of different reference groups than the other parts of the system, as this will be a public service.

The Tenderer understands that planning an accessible and usable experience for all voters is a central objective of the overall solution. The Tenderer will engage with the appropriate groups to attempt to remove all access barriers across all platforms, voter support material and support structures. The Tenderer will carry out user trials with voters with a diverse range of impairments and the results will be used to aid the implementation of the system.

The Tenderer believes that by engaging with the electorate they demonstrate that they have taken into consideration the specific needs of a wide range of impaired groups. During the planning stage, the Tenderer, in conjunction with the customer, will identify which groups may be excluded in order that measures may be put in place by administrators to ensure they are catered for.

The Tenderer will ensure that there is consistency across all platforms, voter support material and support structures for any terms, expressions or graphics. The consistency is applied across a solution to prevent variation occurring in areas such as press releases or training for support staff. We will ensure that consistent terms, expressions and graphics are used. Navigation will be used in a consistent format throughout the site.

The Tenderer will carry out user trials with voters, including voters with a diverse range of impairments and the results will be used to aid the implementation of the system.

The user interaction designer/responsible will be responsible for design and carrying out the usability and accessibility quality tests that is planned into the project. Usability and accessibility quality tests can be performed in different phases of the project, on prototypes, part deliveries and the final delivery. The result of the usability and accessibility quality testing is input to further design/implementation. The Tenderer is mainly using two well proven test methods: expert evaluation and user testing.

3.2. Elaboration on Requirement M2

The supplier shall document accessibility test strategy, how they intend to implement accessibility routines and what tools they intend to use for accessibility validation during development.

Following the agreement of accessibility requirements with the customer and any public advocacy groups a set of project accessibility standards (test strategies and methodologies) will be produced and disseminated with all project stakeholder groups. The project team will then define a set of development standards based on the defined project accessibility standards which are used in all phases of the development.

Our accessibility test strategy is based on extensive involvement of user groups with different disabilities in the design, construction and testing of the solution. This includes activities such as:

- Identifying and inviting adequate user groups with relevant disabilities.
- Dialog with disabled users to identify special challenges they may have.
- Identifying all relevant tools used by people with disabilities.
- Bringing users with disabilities into design meetings
- Implementing proper routines for developers to follow during the whole project to ensure that no code is marked as finished before accessibility is tested and verified. (checklist)
- Establishing an accessibility test plan
- Frequent user testing of accessibility in new running versions of the software after each development-iteration.
- Use of manual and automated tools for identifying possible accessibility issues
- Involvement from reference groups in the system and acceptance testing.

The overall strategy is based on identifying and handling all possible accessibility issues as early as possible in the project.

Some of the tools that we use for identifying possible accessibility issues are:
The WCAG conformance checker at: http://www.achecker.ca/checker/index.php
The Windows Vista built-in Ease of Access Center which includes Narrator (Screen Reader), Magnifier (Zoom in/out) and High Contrast tools

At key project phases, end user testing with users with a range of impairments and users experienced with different assistive technologies will be carried out.

3.3. Elaboration on Requirement M3

The supplier shall document usability test strategy, including methodology and how they intend to implement usability routines within the development team and with regards to the project phases.

Usability and accessibility are tightly coupled so the test approach and strategy is similar. Following the agreement of Usability requirements with the customer and any public advocacy groups a set of project Usability standards (test strategies and methodologies) will be produced and disseminated with all project stakeholder groups. The project team will then define a set of development standards based on the defined project Usability standards which are used in all phases of the development.

The key to accomplish a user friendly solution is to involve the different user groups in all project phases. Our experience is also that taking usability into account as early as possible in the project, such as in the design phase is by far the most cost effective way to secure a user friendly solution, and a successful project.

Our usability methodology covers activities such as:

- Establishing user groups as a part of project startup
- Workshops with the user groups to identify their priorities
- Developing sketches and mock ups
- Establishing a design guide for developers
- Early focus on interaction design principles in the solution rather than only graphical design
- Establishing a usability test plan
- Frequent user testing
- Heuristic testing by usability experts
- Close collaboration with the customer through the whole project

Usability (and accessibility) will be attended to, as described here for the three phases:

Solution design phase

Identify reference groups and user groups that will take part in the design and testing of the solution. This must be a variety of different users, with and without disabilities, covering a large age span. The Tenderer will arrange a kick off session with these users to give them an introduction to the project scope, plans and our expectations to their contribution. This will be important to ensure that all involved have ownership to the process.

Start developing and evaluating concept sketches and mock-ups through workshops with selected users and the customer. These mock-ups will then serve as a starting point for development in the construction phase. In this phase a design guide for developers and usability test plans are established.

Construction phase

In this project phase, the technical and functional solution is implemented according to the customer’s priority through a sequence of development sprints.

Each sprint is completed with demonstration and testing of what is implemented in the iteration. The reference/user groups and specialists from the customer will give valuable feedback to the development team after this testing and evaluation. In this way the usability and accessibility of the solution can be monitored and adjusted through the whole development process and issues found can be corrected easily at a small cost.

The e-voting part of the system will require a more frequent use of reference groups for testing and evaluation of the solution. An appealing and adequate graphical design will also be implemented in this project phase.

After the last development sprint the total solution will be subject to system testing where the whole solution is thoroughly tested also for potential usability and accessibility issues. It is our experience that it is a great advantage if the customer also takes part in this system testing so that as many open issues and errors as possible are uncovered and handled prior to acceptance testing.

Acceptance and completion phase
Acceptance testing by the customer is performed and the Tenderer carries out error correction. High level of involvement from the customer through the previous project phases will ensure that no major usability or accessibility issues are found in this final testing.

3.4. Elaboration on Requirement M4

> The supplier shall provide support to external consultants, independent people that will be brought in to test the solution

The Tenderer will welcome all test resources that may contribute in the testing of the solution during the whole project. The success of the project will depend heavily on involvement from users and specialists safeguarding the needs of all stakeholders in the project. The Tenderer has engaged with independent external consultants not only to test the solution but to help work with the customer to design the solution. The Tenderer confirms the importance of working with Usability and Accessibility Advocacy groups and other public organizations for which Usability and Accessibility are potential concerns. From our experience, this is a critical stage of ensuring the success of the project.

3.5. Elaboration on Requirement M5

> The supplier shall document how the design process ties in with the overall development process.

The Tenderer’s Quality Management System contains specific procedures and processes which define how the design, development and customization of the election suite of products and services is to be carried out. It includes all elements of planning, design and development inputs, outputs, reviews, verification, validation and changes.

The activities associated with design and development planning are carried out in accordance with the Tenderer’s agreed documented Software Development processes. The latter includes elements such as requirements specification, review of requirements specification, development task analysis and estimate, design specification, test specification, code reviews, audit specification and reviews, phase testing, stress testing, user acceptance testing, user guides and system sign off.

In order to ensure that all elements of the process are adhered to and reviewed for adequacy and completeness, software development project review meetings take place on a regular basis. At this stage progress is reviewed and the project plan and schedules updated accordingly by the Development Manager.

4. Additions to the general agreement

4.1. Additions to 2.4.1

Our test strategy and methodology is described in the previous chapters, as is the installation test.

4.2. Additions to 2.4.2

Not relevant as no conversion is so far identified in the project.

4.3. Additions to 2.5.3

<To be completed by the Customer>

4.4. Additions to 2.5.4

<To be completed by the Customer>

4.5. Additions to 2.6.1

No comments to the general agreement

4.6. Additions to 2.6.2

<To be completed by the Customer>

4.7. Additions to 2.6.3

The Tenderer suggest that the errors in the approval period are reported in the same manner as in the System Test and the Acceptance Test, i.e. as a bug in the issue registration tool selected for the project.