Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 06/09/2018

A System Testing Plan for A Cloud Based On-line Election Software

Nowadays, with the advance of technology, the voices for applying cloud based on-line election system is getting louder. The on-line election software should be complex. Consequently, a system testing plan for this system is required before it is released. In system testing, the first test that I would want to conduct is security testing. In my opinion, security concern is the most important point. This software must secure the selections made by voters. Basically, the security protection needs to be implemented into the election system. No one can query and falsify others' selections and the election results in any way. Accordingly, there are many relevant tests needed to be carried out. For example, when voters' computers are invaded by viruses or other external attacks, this system must be able to ensure the data integrity and avoid exposing voters' privacy information. Also, encryption of the voter's privacy data is necessary. Security tests have to verify it. The concrete approaches are to proceed with functional security testing, adversarial security testing, and security exploratory testing. Functional security testing focuses on encryption and authentication, and adversarial security testing aims at external attacks. In addition, security exploratory testing is to explore the holes within the system according to security concerns.

The following type of system testing is usability testing. In general, there are some software requirement documents that describe functional qualities, such as voting features, UI design, system instruction, etc. for this system. First, the test team will focus on checking whether this system is thoroughly consistent with the specification documents. Sequentially, other subjective criteria, such as understandability, learning curve, simplicity, etc., also need to be tested. During this stage, real users must be involved. Due to the generality of this system, a representative sample of user groups needs to be selected and invited to test. After their examination, collecting their comments and suggestions is crucial for the improvement of this system. In addition, inviting experienced experts to join the test is also a great approach to make this system better.

Accessibility testing must be introduced because this system will be used for all the people including people with disabilities. Blind people are the main targets, and this system should provide the voice auxiliary for them. In addition, people with physical disorders, especially hands, must also be taken into consideration. The voice control function of voting and determining the identity should be included in this online election system. Moreover, checking whether the UI design is appropriate for people is also important, such as the color of contrast and the sizes of font. In the beginning of this test, the test team can conduct this test. After the preliminary test passes, it is necessary to invite people who have disabilities to test this system. Their feedback can be more intuitive and direct.

Because the potential marketing can be located all over the world, internationalization testing is needed. Many localized issues should be considered, such as languages, formats, time zones, etc. Once this software proceeds to system testing, field trial testing is essential. Our team members need to personally travel to the target countries to conduct testing because some localized issues cannot be found remotely.

Online election software will be used by many people simultaneously, and this will cause extreme conditions like race situations and multi-synchronization. These conditions rely on stress testing to determine the capacities of this system. Four types of stress testing can be introduced. The first type is load testing, which can simulate hundreds, even thousands of users using this system simultaneously, and not only functionality but also performance can be verified. The second type is low-resource testing, which can test this system under the particular conditions like low disk space or memory in users' voting environments. The next type is capacity testing that can determine the maximum number of voters. The final type is reliability testing, and this test can measure when the system error will happen after being used a number of times.

Due to several kinds of configurations in every individual voting environment, compatibility testing must be carried out in this cloud based system. This test probably needs to involve different operation systems or browsers on the user side and different databases on the server side. In the meanwhile, using combinational techniques to verify all the possible combination configuration is necessary. In addition, backward compatibility testing has to be performed because the current version must be able to access any files of the previous versions.

Because the online election software is a cloud based system, several servers have to be setup. Consequently, deployment testing needs to be introduced. Simulating the deployment in one server is the first step, followed by setting up the multiple servers. Additionally, the system also needs to be checked whether the default setting is appropriate or not. In my opinion, combining this test with internationalization and compatibility testing might be a more efficient approach.

The final testing is regression testing. When the project enters the system testing phase, adding any new features, changing the configurations, and fixing bugs will be critical. During this, regression testing must be performed after any modification of this system. If the test suite of the regression testing is too large, it means that this test will take a long time to execute, and sometimes this test suite needs to be revised. If this system can pass these tests, this system will be eligible to be released.