KIOSK automation plan document

* Ralf Wang

# Why?

The automation is more and more popular in the testing area. Good automation testing can give the team the benefits as follow:

1. Reduce the cost: The automation not need more human, hardware, software and other support resource. We need less resource to cover more test cases. The professional QA member can only focus on the requirements analyze, case design, process control and other area.
2. Save time: If the automation structure and cases created, the automated cases can execute at any time. The script can execute in workday and weekend, day and night with any time which we need.
3. Enhance the effective: The automation can works with 7\*24. It can help the QA team to enhance the work effective. QA team can focus on the core cases verify and validate works.
4. Consistency and reliability: The execution of automation script can keep the process and result with high consistency. It can also avoid the issue the mistake and different understand during testing from human.
5. CI: Many test tool and structure can integration the published code and software build. It can verify the defect in the latest build quickly and reliability. It will make the DevOps process work smoothly.
6. Performance: The automation can do the professional performance and stress testing with tool. It will be very hard for manual testing. It will enhance the conference for the software quality.

# What`s target

Our product includes the PS, terminal and web site module. We want to cover more test cases for our testing. The automation should can integrated with the build publish and do the BVT for team.

For PS module, QA team want the automation can simulate the patient create, films print, paper print, OCR and other main workflow.

For terminal module, we want the scripts can simulate the films and reports print operations. The script can cover the workflow of terminal.

For the web site module, we want the scripts can do information query, study set, print mode change, configuration setting and other operations.

Follow the requirements, the automation structure should support the DICOM process, file print, web operation, API testing module to meet our target.

# What`s tool

Now, there are many test tool and structure using in current testing area. The popular ones as allow:

* QTP/UFC
* SoapUI:
* Load Runner/JMeter
* C#/Java/Python + Selenium
* Python + Robot Framework
* C#/JAVA + Unit framework

**QTP/UFC**

* Base on the VB script and UI.
* Support the web, win form, WPF control.
* Support log and report publish.
* No strong test framework
* Script execute slow.
* With more maintain works for scripts and library.
* Business software need more cost.

**SoupUI**:

* Base on API testing
* [API Performance Testing](https://www.soapui.org/load-testing/getting-started.html)
* API Security Testing
* Support Test report
* Support the web, web service, web mock, DDT.
* No strong test framework
* Only support API test cannot support application test.
* Business software need more cost.

**Load Runner**

* Performance test pro software
* [Support](https://www.soapui.org/load-testing/getting-started.html) web, web service, database, and socket and etc protocol.
* Support report and analyze module.
* Support windows/Linux/database resource monitor module.
* Only can test for API and performance test
* No Test framework
* Business software, very expensive. About $50K.

**JMeter:**

* Base on Java and for API, performance test.
* Many test framework and plug-in in open source.
* Scripts execute fast.
* Only can test for API and performance test

**C#/Java/Python + Selenium**

* Base on UI.
* Can support multiple develop code: C#, JAVA, Python etc.
* Support good UI verify and identify operations.
* Many test frameworks can support selenium.
* Support multiple browsers.
* Support client/Service works module.
* Only support web UI test.

**Python + Robot Framework**

* Support DDT with good framework.
* Support the report publish and view.
* With good test library like selenium, android, autoit, ios database etc.
* Not support .net and C#
* Need learn the Python
* The test framework is developed by selenium, need python environment.

**C# + Unit framework**

* Support unit, web, and application test.
* Can use the selenium to test web product.
* Design the plug-in or module to test support tool.
* Support design API test with services reference.
* Unit test can test the low level function or code.
* Test execute fast.
* Have strong ability to extend function.
* Good support with CI.
* Can be reused the code from develop `s code.
* QA team has experience for C#.
* Flexibly.
* More develop works for designer.
* No good report publish feature, need design by designer.

As kiosk Shanghai R&D team current status, the team design to choose the C#, MS test platform (unit), autoit, selenium and jerkins for our automation test works.

* Jerkins: Use to integration the daily build to execute automation test.
* AutoIt: Used to design the build install operations.
* C#: Design the function and module for unit test to use.
* MS Test: As a test framework to execute the unit test cases and unit test. The V2 has published with new feature.
* Selenium: Include the test framework to test the web related cases if need.

# Who?

Team will organize an automation group to push the works. The group members should include the cases designer, review and audit, scripts designer, SCM and developer.

**Case designer:**

In change the cases collect, design and review.

**Designer:**

Develop the test framework and automation test cases.

**SCM:**

In change to the build and test CI works.

**Developer:**

Support the QA team for code design and review. Offer the API, function, module design.

# What`s the Process?

First, the SCM will create the build publish and jobs to install the latest one in the automation environment. Notify and execute the automation scripts with Jenkins jobs.

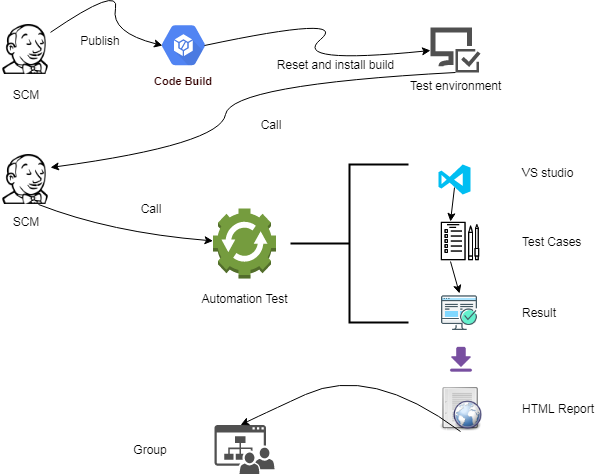
Second, the automation designer will create the test framework and scripts. Ensure the platform can meets the requirements for BVT test.

Then the automation group members will do this work iteratively.

1. Case designer filter or design the automation test cases.
2. Code Designer and developer automated these cases of step1.
3. SCM or code designer build and execute automation cases.
4. All group members review the test result.

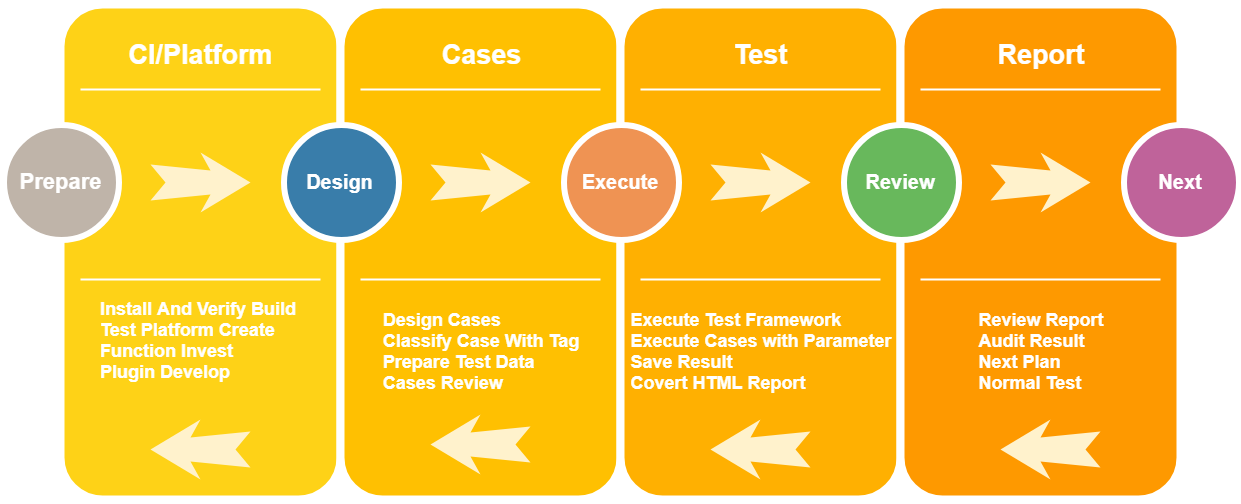
# How is the test platform?

The test platform will integration with Jenkins, Hyper-V, C# and unit test. The structure is follow:



1. SCM publish the latest code with Jenkins.
2. Reset the test environment with Jenkins task.
3. Install the latest build and verify with Jenkins task.
4. If successfully, go to the Jenkins slave start to start the automation task.
5. Jenkins task start the MS Test or VStest to run the test platform.
6. Execute the test cases with parameter with the tasks.
7. Publish the test result.
8. Collect the result and release as HTML style report.
9. Mail the report to automation group members.

# When and plan?



Team will plan to the loop process as upon figure. The first build will finish the BVT automation test full process:

**Phase 1:**

**Prepare:** 2019/07/08 – 2019/07/31

1. Publish, install and verify with Jenkins jobs. – Jasmine
2. Test platform invest and created. – Ralf
3. Function and plug-in design. – Ralf

**Cases:** 2019/08/01 – 2019/08/31

1. Filter and design the BVT test cases. – Cloudy
2. Prepare test data and tool. – Ralf
3. Design the case with code. – Ralf, dev
4. Cases debug and review. – Ralf , dev

**Test:** 2019/09/01 – 2019/09/15

1. Execute the automation test. – Ralf, Jasmine
2. Test platform debug and enhance. – Ralf, dev

**Report:** 2019/09/16 – 2019/09/23

1. Review the test result. – Group
2. Audit the automation process. – Group
3. Next phase test plan discuss. – Group

Add by Ralf Wang 2019-11-11

Until now, QA team have finished the smoke test framework and test case design.

The framework is use the robot framework and develop script is python. The SCM team is try to integration the code with Jenkins.

Now, our automation team get a lot supports and resource from team. The team members as follow:

* Raissa Huang Designer
* Ralf Wang Designer
* Weisong Wang Designer
* Yunfei Zhu Designer
* Jasmine Tang SCM

After the PUMA MR3 project release, we will start the next phase work. Design and organize more test cases about PUMA project. Before we start it, I will organize some trainings for the test framework and base knowledge. The draft plan as follow:

From 2019-11-18 to 2019-11-30:

The training content will include:

1. Python base knowledge:
   1. Object-oriented OOP
   2. Request library
   3. pyodbc library
2. Ride, robot framework:
3. How to self-define robot key words:
   1. Python file
   2. Object
4. PUMA test framework
   1. Library
   2. Python function
   3. Comments.
5. Test Framework Runner

If team members have any other training requirements, please feedback with comments.

Before we start the training, please make sure you have setup the python and robot framework in your laptop.

We also should discuss which module or feature of PUMA should be automation in the future.

After all prepare work finished, we will start to redesign, develop the automation test cases.