Test Automation

Vanich Chaipattanawanich 5415047659 Suphanat Savetapanu 5410547128 Tachin Srisombat 5410546075 Tanachot Techajarupan 5410547624

Testing Method

- Manual
- Test Automation

Manual Testing

The process of manually testing software to find defects. It requires **a tester** to play the role of **an end user**, and use most of all features of the application to ensure correct behavior.

Manual Testing

To ensure completeness of testing, the tester often follows **a written test plan** that leads them through a set of important **test cases**.



What is Test Automation?

In software testing, **test automation** is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes

Why is Test Automation?

Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.



- ●
- Automated Software Testing Saves Time and Money
- Testing Improves Accuracy
- Increase Test Coverage
- Automation Does What Manual Testing Cannot Archive Easily
- Automated QA Testing Helps Developers and Testers
- Team Morale Improves
- TestComplete is a Powerful and Affordable Automated Software Testing Tool

- Automated Software Testing Saves Time and Money
 - Software tests have to be repeated often during development cycles to ensure quality. Every time source code is modified software tests should be repeated. For each release of the software it may be tested on all supported operating systems and hardware configurations.
 Manually repeating these tests is costly and time consuming. Once created, automated tests can be run over and over again at no additional cost and they are much faster than manual tests.

- Testing Improves Accuracy
 - Even the most conscientious tester will make mistakes during monotonous manual testing.
 Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results.
- Increase Test Coverage
 - Automated software tests can easily execute thousands of different complex test cases during every test run providing coverage that is impossible with manual tests. Testers freed from repetitive manual tests have more time to create new automated software tests and deal with complex features.

- Automation Does What Manual Testing Cannot Archive Easily
 - Even the largest software departments cannot perform a controlled web application test with thousands of users. Automated testing can simulate tens, hundreds or thousands of virtual users interacting with network or web software and applications.
- Automated QA Testing Helps Developers and Testers
 - Shared automated tests can be used by developers to catch problems quickly before sending to QA. Tests can run automatically whenever source code changes are checked in and notify the team or the developer if they fail. Features like these save developers time and increase their confidence.

- Team Morale Improves
 - Automating repetitive tasks with automated software testing gives your team time to spend on more challenging and rewarding projects. Team members improve their skill sets and confidence and, in turn, pass those gains on to their organization.
- TestComplete is a Powerful and Affordable Automated Software Testing Tool
 - TestComplete addresses a full range of software testing challenges facing corporate IT departments, product developers, QA engineers, and consultants. TestComplete enhances the software testing process by increasing efficiency, removing complexity and lowering costs.

Cons of Automation Testing

- Proficiency is required to write **the automation test scripts**.
- Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
- **Test maintenance is costly** in case of playback methods. Even though a minor change occurs in the GUI, the test script has to be re-recorded or replaced by a new test script.
- Maintenance of test data files is difficult, if the test script tests more screens.

What/When do we test in Test Automation?

- Unit Test: The Automated Unit Test has a lot of benefits. First, it help us detect defects in an early stage, which will decrease fixing time needed (compare to finding the defect during Integration Test or System Test). Second, we can use this chance to do a Code Review, which will also help detecting defects. Last but not least, we can reuse Automated Unit Test during Regression Test, which help reducing time and manpower needed.
- **Regression Test:** The Regression test is a Test case that have to be run many times before the software is release in order to check that the software still working correctly and ensure that fixed bugs does not affect other part of the software. Isn't this the perfect situation for the Automated testing?
- Random Test: Test case that involve many random data inputs to find defects is also suitable for Automated test
- **Capacity Test:** A System that support a lot of transaction at the same time, for example 50000 transactions at once is definitely unable to be archived manually.
- **Performance and Reliability Test:** Performance and Reliability test for web applications should be done by automated test as well; for example, a test to find responsive time or to check scalability.

Can Test Automation do everything?

- No. . . it can not do . . .
- Unstable GUI test
- Installation & Configuration
- Error handling & Recovery test
- Localization test
- Usability test
- Documentation

Types of Test Automation

- Graphic User Interface Test
- Code Driven Test

Graphic User Interface Test

The GUI Testing can be applied to any application that has a Graphic User Interface. Many test automation tools provide record and playback features that allow users to interactively record user actions and replay them back any number of times, comparing actual results to those expected.

Example of Graphic User Interface Test

Selenium automates browsers

a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language (**Selenium IDE**). It also provides a test domain-specific language to write tests in a number of popular programming languages



Example of Graphic User Interface Test

Selenium IDE (Use with firefox) http://docs.seleniumhq.org/download/

Graphic User Interface Test

Pros

- you don't need to know how the code work to do the test
- can execute some exhaustive test
- can find 90% of the defect that unit test can not found

Cons

- the code coverage percentage are low
- hardly maintainable due the change of program GUI
- the test environment can affect the test

Code Driven Test

Code driven test automation is a key feature of agile software development, where it is known as **test-driven development(TDD)**.

Unit tests are written to define the functionality before the code is written. However, these unit tests evolve and are extended as coding progresses, issues are discovered and the code is subjected to refactoring. Only when all the tests for all the demanded features pass is the code considered complete.

Example of Code Driven Test

JUnit

A unit testing framework for the Java programming language. JUnit has been important in the development of **test-driven development**, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

Example of Code Driven Test

import org.junit.*;

public class TestFoobar {
 @BeforeClass
 public static void setUpClass() throws Exception {
 // Code executed before the first test method
 }
 @Before

```
public void setUp() throws Exception {
    // Code executed before each test
}
```

```
@Test
public void testOneThing() {
    // Code that tests one thing
} }
```

Example of Code Driven Test

JUnit - A programmer-oriented testing framework for Java.

https://github.com/junit-team/junit/wiki

Code Driven Test

Pros

- Code coverage percentage are high
- A failed unit test exactly pinpoints the source of the problem
- can run different test environment
- easily Reusable

Cons

- Coding knowledge is needed.
- Can not work on bad design or complexity code
- Big time investment





1. Why do we need automate test?

Quiz

1. Why do we need automate test?

Answer : More Accurate than manual testing , Save time and money , Increase Coverage



2. What is the most benefit of using automate test?

Quiz

2. What is the most benefit of using automate test?

Answer:

Automation Does What Manual Testing Cannot Archive Easily

Even the largest software departments cannot perform a controlled web application test with thousands of users. Automated testing can simulate tens, hundreds or thousands of virtual users interacting with network or web software and applications.



3. What the worst disadvantage of test automation?

Quiz

3. What the worst disadvantage of test automation?

Answer:

Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to **deadly consequences**.

Reference

- http://en.wikipedia.org/wiki/Test automation
- http://stackoverflow.com/questions/6488075/what-are-the-pros-and-cons-of-unit-test-vs-guiautomated-end-to-end-test
- http://www.eventhelix.com/agile/unit-testing.htm#An_Introduction_to_Unit_Testing
- http://en.wikipedia.org/wiki/Manual_testing
- http://www.softwaretestinggenius.com/pros-and-cons-of-automated-testing
- http://www.xasyst.com/assets/images/MT.gif
- http://stackoverflow.com/questions/64333/disadvantages-of-test-driven-development
- http://support.smartbear.com/articles/testcomplete/manager-overview/



http://www.negisu.com/tachin/***