Software Testing

Goal

- Validate and verify the behavior of software under development
  - Validation ➡️ Errors of designs
  - Verification ➡️ Errors of implementations

Methods

- Static testing: Testing without its execution
  - Code review, Mathematical proof
- Dynamic testing: Testing with its execution
  - The behavior of software satisfies its specifications
  - The software does not cause any error even if its behavior is not defined by specifications

Terminologies

Test case

- A pair of inputs and expected outputs of software under test

Test suite

- A set of test cases

Classification of testing with phase

- Unit testing: Verify the code implementation for each functions
- Integration testing: Verify the system is properly built on the design
- System testing: Verify the system meets speficications
- Acceptance testing: Verify the system meets requirements of users

Automatic Testing

What is?

- A framework to execute software testing
  - Execute testing codes for software under test
  - Execute defined inputs and compare the output with the expected one
- Notice that it is not the automatic test case generation

Why?

- To reduce the workload on executing and checking the implements
- Useful for regression testing and **TDD**

Test for Golang

- Change the file `hello.go`

```
package main

func getHello() string {
    return "Hello world"
}
```

Note

- Remove the main function
- Define the function that returns a string

- Create a file typing test codes; `hello_test.go`

```go
package main

import (
    "testing"
)

func TestGetHello(t *testing.T) {
    expected := "Hello world!"
    result := getHello()
    if expected != result {
        t.Errorf("Test fail expected: %s, result: %s\n", expected, result)
    }
}
```

- Execute the following command in Terminal

```
$ go test -v
=== RUN   TestGetHello
    hello_test.go:11: Test fail expected: Hello world!, result: Hello world
--- FAIL: TestGetHello (0.00s)
FAIL
exit status 1
FAIL    _/home/ec2-user/environment/example     0.002s
```

The standard testing framework

- Test file name: `xxx_test.go`
- In the test file, The signature of test function should be `func TestXxxx(t *testing.T)`
- The package name is usually set as the same as the package in which functions under test belong to.
- Implement the code to judge whether the results of functions under test are right or not. If it finds an error, use `t.Errorf` to display the error message.

Summary

- Software testing is a process to validate and verify the behavior of software under development.
- It can be classified into static testing and dynamic testing.
- Automatic testing is a framework to execute software testing.