IT Quality and Software Test

Lesson 11 Test Tools

> Quiz V1.1

Uwe Gühl



Winter 2013 / 2014

1. Test Tools Overview



A tool that supports traceability, recording of incidents, or scheduling of tests is called:

- a) a dynamic analysis tool
- b) a test execution tool
- c) a debugging tool
- d) a test management tool
- e) a configuration management tool



http://www.istqbsamplepaper.n18.in

1. Test Tools Overview



A tool that supports traceability, recording of incidents, or scheduling of tests is called:

- a) a dynamic analysis tool
- b) a test execution tool
- c) a debugging tool
- d) a test management tool
- e) a configuration management tool



http://www.istqbsamplepaper.n18.in

2. Test Tools Test management tools

Below you find a list of improvement goals a software development and test organisation would like to achieve.

Which of these goals for improving the efficiency of test activities would best be supported by a test management tool?

- a) Improve the efficiency by building traceability between requirements.
- b) Improve the efficiency by optimizing the ability of tests to identify failures.
- c) Improve the efficiency by faster resolving defects.
- d) Improve the efficiency by automating the selection of test cases for execution.

http://www.istqb.org



2. Test Tools Test management tools

Below you find a list of improvement goals a software development and test organisation would like to achieve.

Which of these goals for improving the efficiency of test activities would best be supported by a test management tool?

- a) Improve the efficiency by building traceability between requirements.
- b) Improve the efficiency by optimizing the ability of tests to identify failures.
- c) Improve the efficiency by faster resolving defects.
- d) Improve the efficiency by automating the selection of test cases for execution.

http://www.istqb.org



3. Test Tools Test execution tools



Which one of the following characteristics of test execution tools describes BEST a specific characteristic of a keyword-driven test execution tool?

- a) Actions of testers will be recorded in a script that can be rerun several times.
- Actions of testers will be recorded in a script that is then being generalized to run with several sets of test input data.
- c) The ability to log test results and compare them against the expected results.
- A table containing test input data, action words, and expected results controls the execution of the system under test.

http://www.istqb.org

Winter 2013 / 2014

3. Test Tools Test execution tools



Which one of the following characteristics of test execution tools describes BEST a specific characteristic of a keyword-driven test execution tool?

- a) Actions of testers will be recorded in a script that can be rerun several times.
- Actions of testers will be recorded in a script that is then being generalized to run with several sets of test input data.
- c) The ability to log test results and compare them against the expected results.
- d) A table containing test input data, action words, and expected results controls the execution of the system under test.

http://www.istqb.org

4. Test Tools Introducing a tool



When a new testing tool is purchased, it should be used first by

- a) a small team to establish the best way to use the tool
- b) everyone who may eventually have some use for the tool
- c) the independent testing team
- d) the vendor contractor to write the initial scripts

http://www.ajoysingha.info

4. Test Tools Introducing a tool



When a new testing tool is purchased, it should be used first by

- a) a small team to establish the best way to use the tool
- b) everyone who may eventually have some use for the tool
- c) the independent testing team
- d) the vendor contractor to write the initial scripts

http://www.ajoysingha.info

1 Task Unit Testing (1/2)



- A system was designed to calculate the fine in case of speeding. Speeding 1-15 mph over 65 mph results in a fine of \$146. Additionally:
 - Speeding 16-25 mph over 65 mph \$266 fine.
 - Speeding more than 25 mph over 65 mph

\$380 fine.

- 1. How many test cases should be written? Please explain your answer.
- 2. Two JUnit test cases have been written already (following page), please add another one in detail.
- 3. What could be extracted into an "@Before" Statement?

1 Task Unit Testing (2/2)



package speedingFine; import static org.junit.Assert.*; import org.junit.Test;

```
public class SpeedingFineTest {
```

```
@Test
public void testNoFine() {
   SpeedingFine sf = new SpeedingFine();
   assertTrue("Fine no speeding: ", "$0" == sf.getFine(0));
}
@Test
public void testLowSpeedingLowerBound() {
   SpeedingFine sf = new SpeedingFine();
   assertEquals("Fine low speeding lower bound: ", "$146",
   sf.getFine(1));
   }
}
```

1 Proposal Unit Testing (1/3)



- 4 test cases following equivalence classes.
 6 test cases following Boundary Value analysis.
- 2. Following test cases could be added:

```
@Test
public void testLowSpeedingUpperBound() {
    SpeedingFine sf = new SpeedingFine();
    assertEquals("Fine low speeding upper bound: ", "$146",
sf.getFine(15));
}
@Test
public void testMediumSpeedingLowerBound() {
    SpeedingFine sf = new SpeedingFine();
    assertEquals("Fine medium speeding lower bound: ", "$266",
sf.getFine(16));
}
```

1 Proposal Unit Testing (2/3)



```
@Test
public void testMediumSpeedingUpperBound() {
    SpeedingFine sf = new SpeedingFine();
    assertEquals("Fine medium speeding upper bound: ", "$266",
    sf.getFine(25));
    }
    @Test
    public void testHighSpeeding() {
        SpeedingFine sf = new SpeedingFine();
        assertEquals("Fine medium speeding: ", "$380",
```

```
sf.getFine(26));
```

1 Proposal Unit Testing (3/3)



3. The statement

speedingFine sf = new SpeedingFine(); in each test case could be moved into an @Before statement

```
private SpeedingFine sf;
@Before
public void setup() throws Exception {
    sf = new SpeedingFine();
}
```