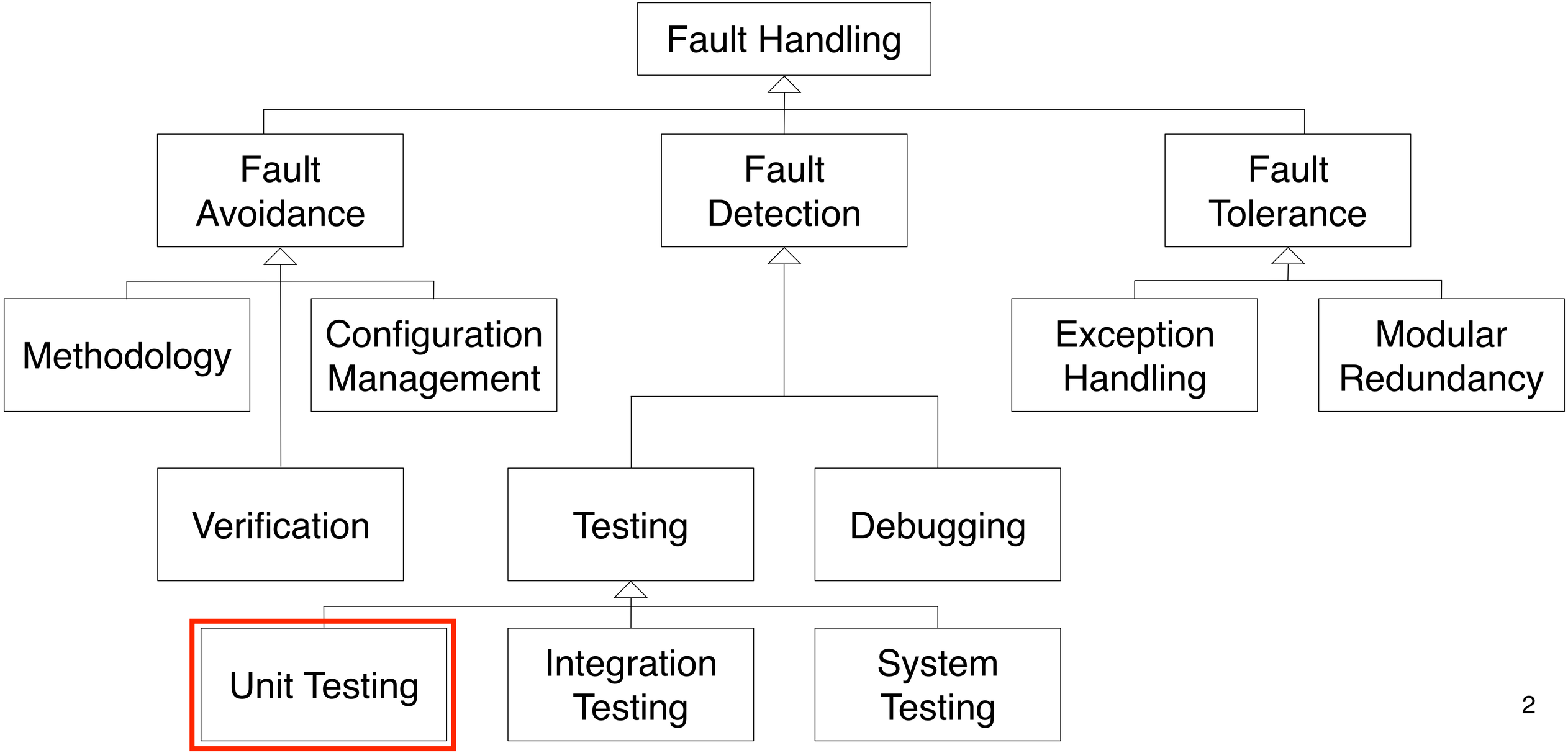
Software Engineering Essentials

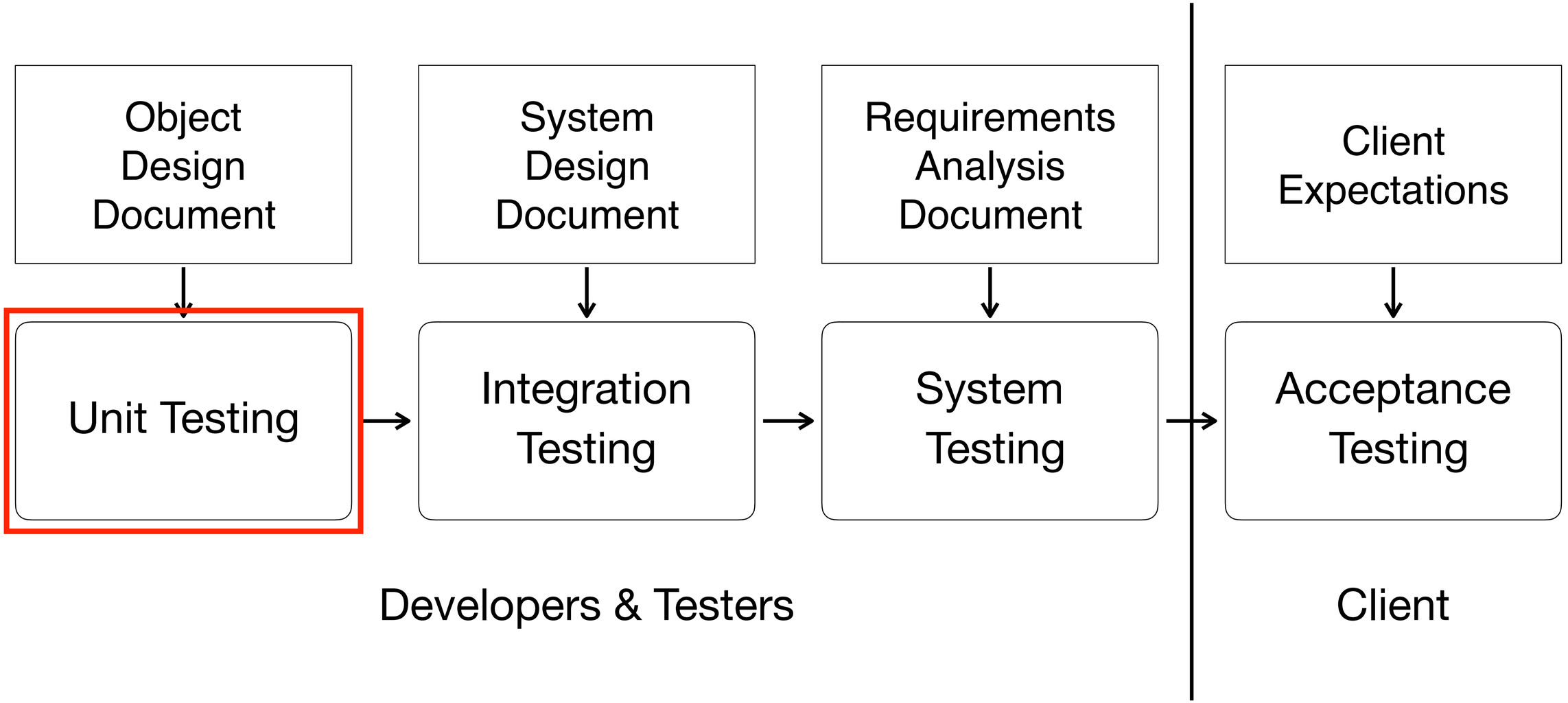
Unit Testing

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

Taxonomy for Fault Handling Techniques



Model-based Testing Activities



Unit Testing with the JUnit Framework

- A Java framework for writing and running unit tests
  - Test cases and test suites
  - Test runner
- Written by Kent Beck and Erich Gamma
- Written with “test-first” and pattern-based development in mind
- JUnit is Open Source
  - http://junit.org/
  - https://github.com/junit-team/junit4
- JUnit 4 uses annotations to control the execution of test cases
- The comparison between expected and observed behavior is validated by assertions

test oracle

Annotations in JUnit 4

Identifies a test method

```
@Test public void exampleTest()
```

Tests if the test method throws the exception

```
@Test(expected=IllegalArgumentException.class)
```

Test fails if it takes longer then 100 milliseconds

```
@Test(timeout=100)
```

Invoked before any test

```
@Before public void setUpTest()
```

Invoked after any test

```
@After public void tearDownTest()
```

Invoked before the execution of all tests

```
@BeforeClass public void beforeClassSetUp()
```

```
@AfterClass public void afterClassTearDown()
```

Invoked after the execution of all tests

Assertions in JUnit 4

optional parameter

```
assertTrue(message, condition);

assertFalse(condition);

fail(message);

assertEquals(message, expected, actual);

assertNull(message, object);

assertNotNull(message, object);
```

Software Engineering Essentials

Unit Testing

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics