

DEVELOPER TESTING

08.09.2016, STEFAN LUDWIG

AGENDA

Testautomation

Lunch Break

Test-Driven Development

Q&A

TESTAUTOMATION

What is the motivation behind
testautomation?

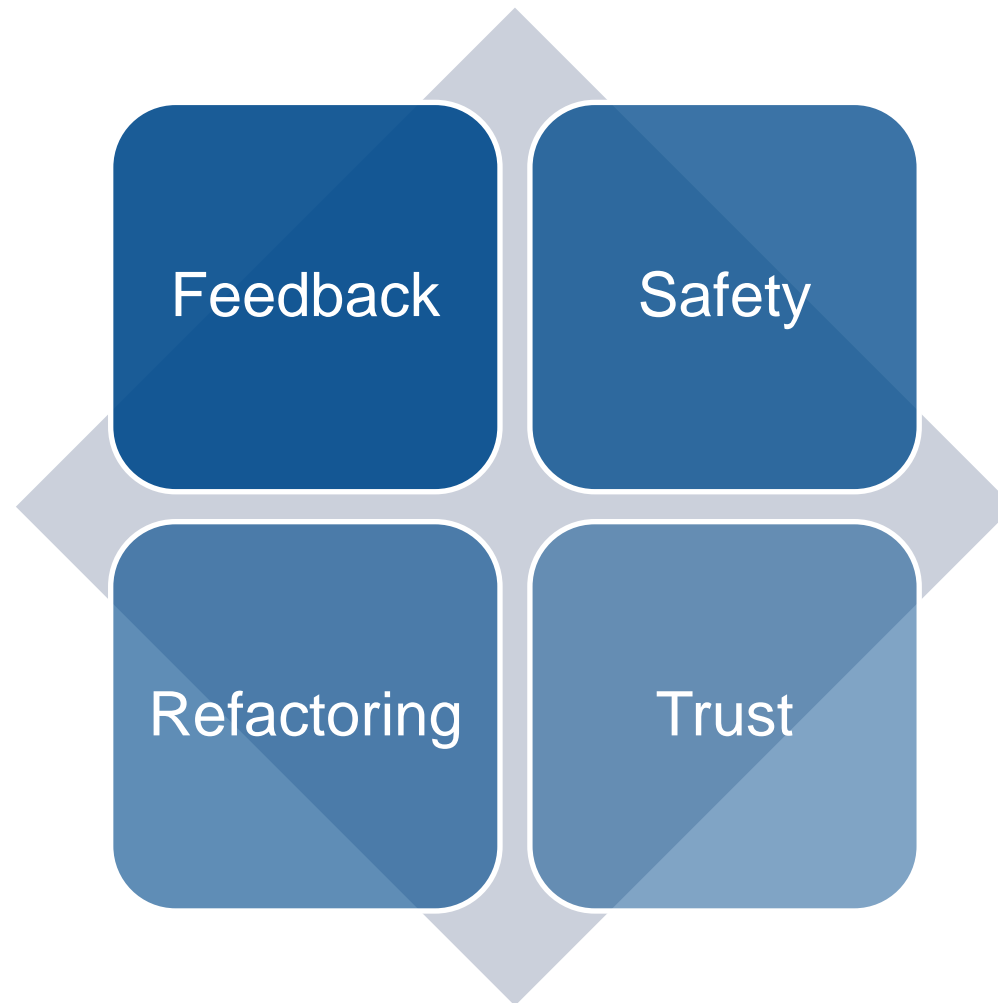
„Software Testing is not about finding bugs.
It's about delivering great software.“

- Harry Robinson -

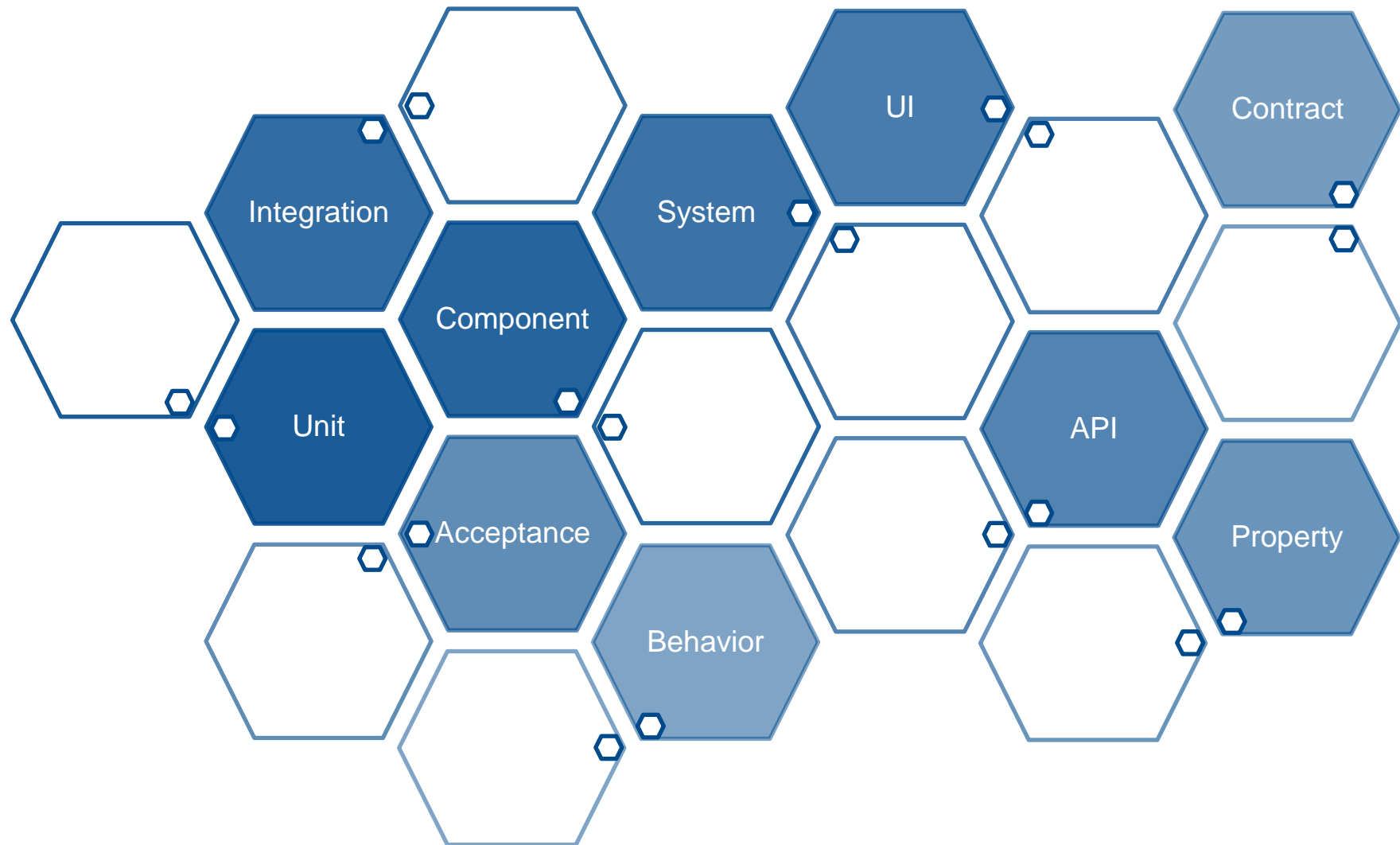
„Instead of looking for bugs, why not focus on preventing them?“

- Jeff Morgan -

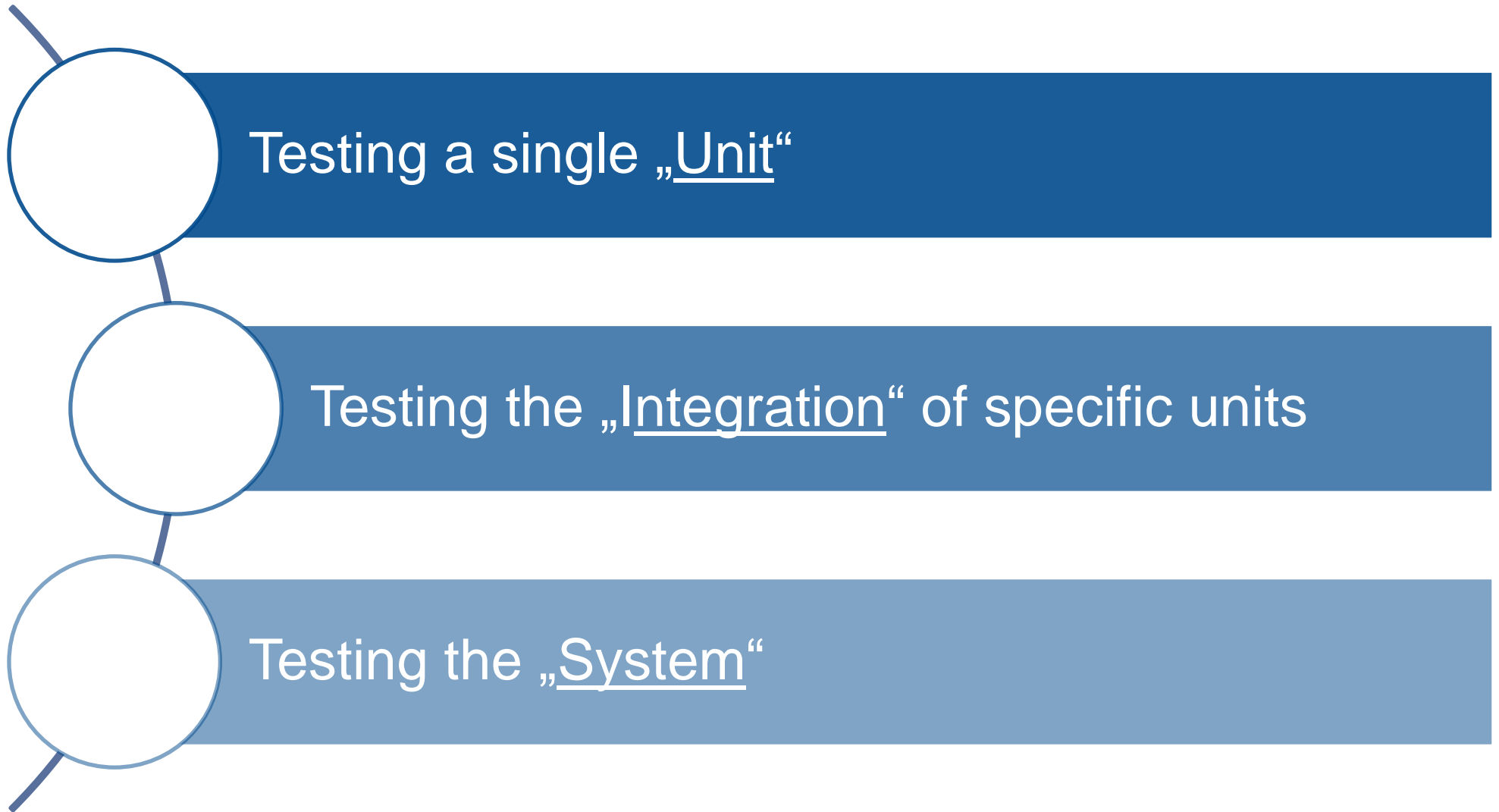
Motivation



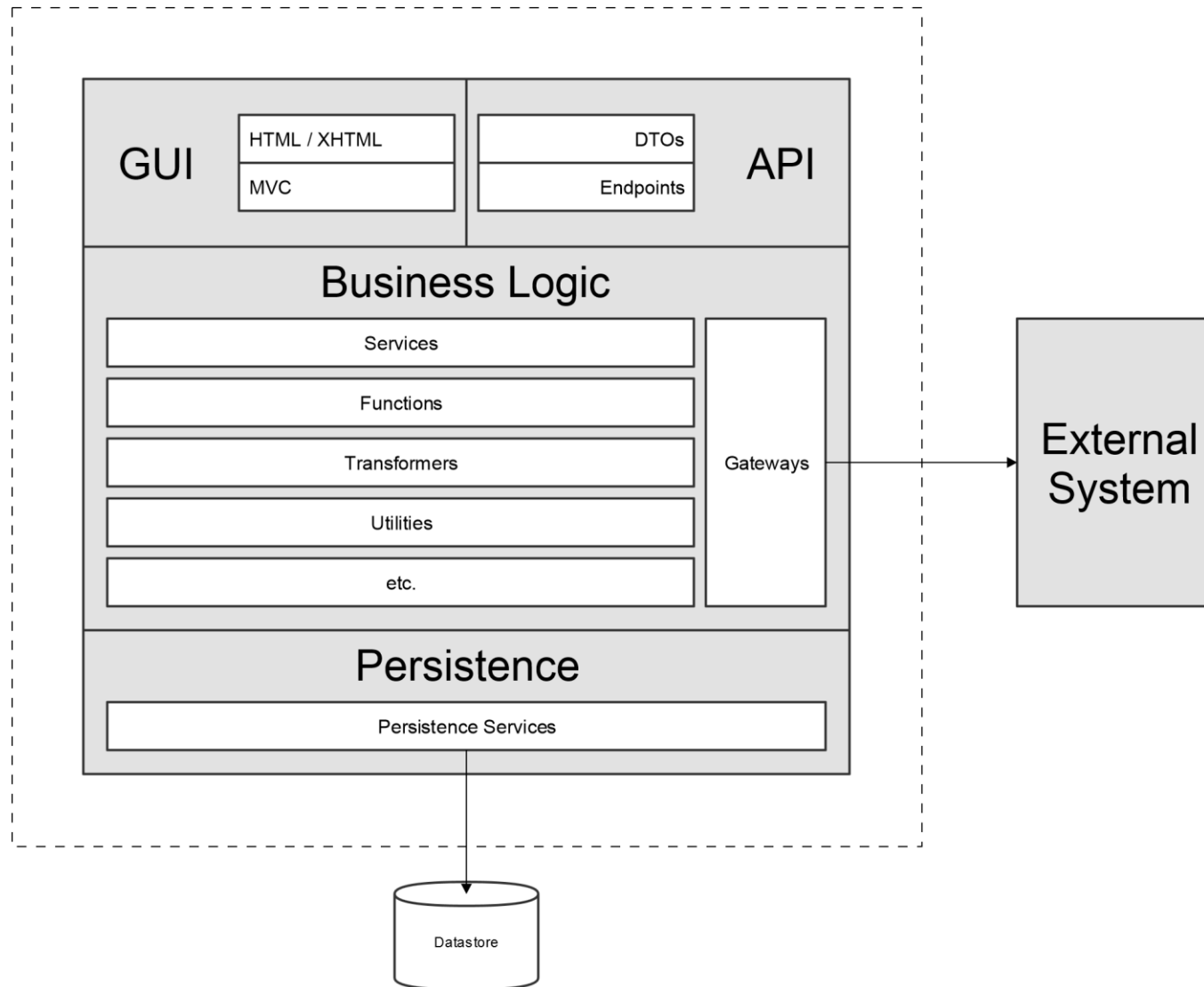
Kinds of Tests



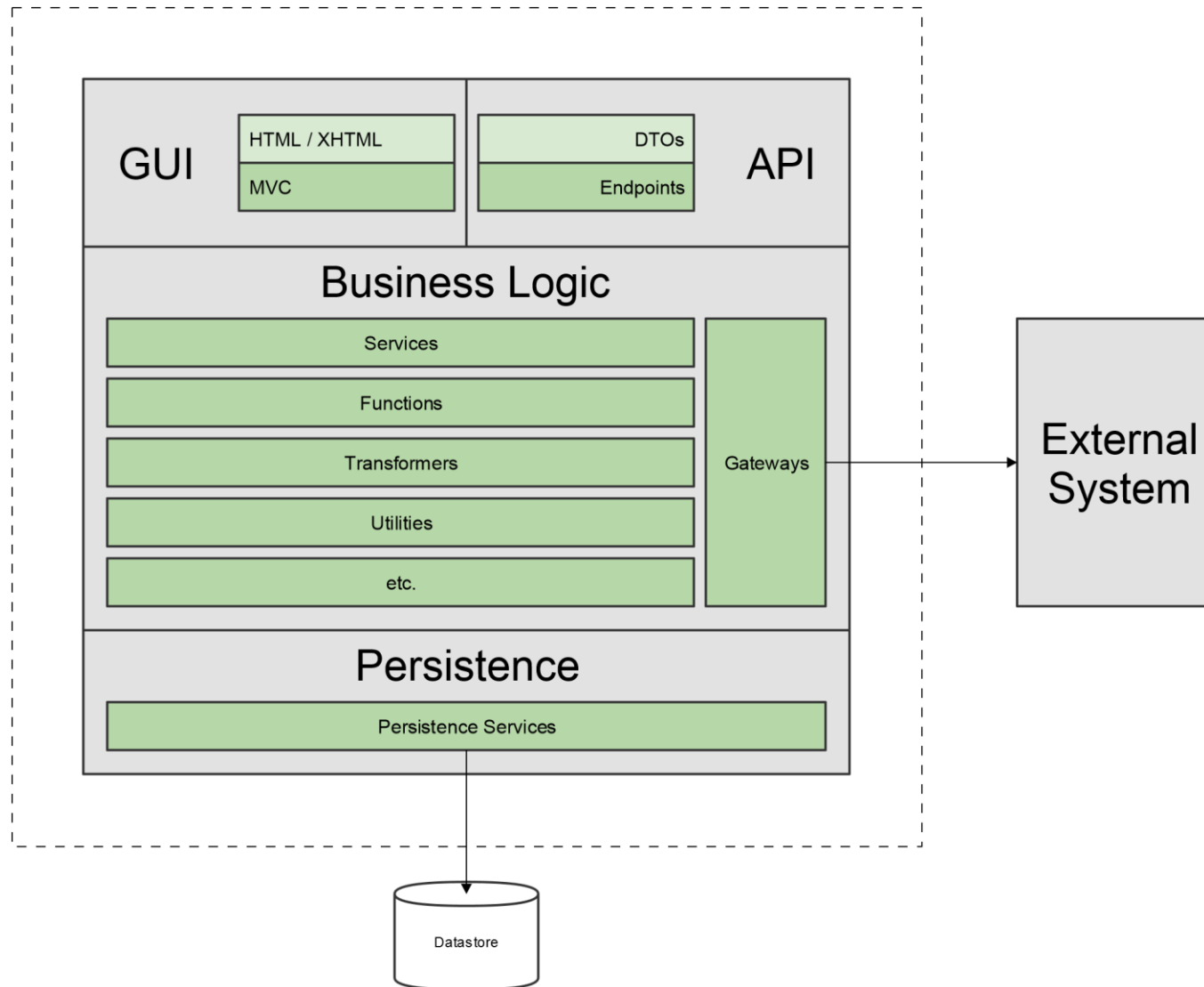
Test Scope



Example Application



Unit Tests



Unit Tests

Runtime:
Milliseconds

Isolated

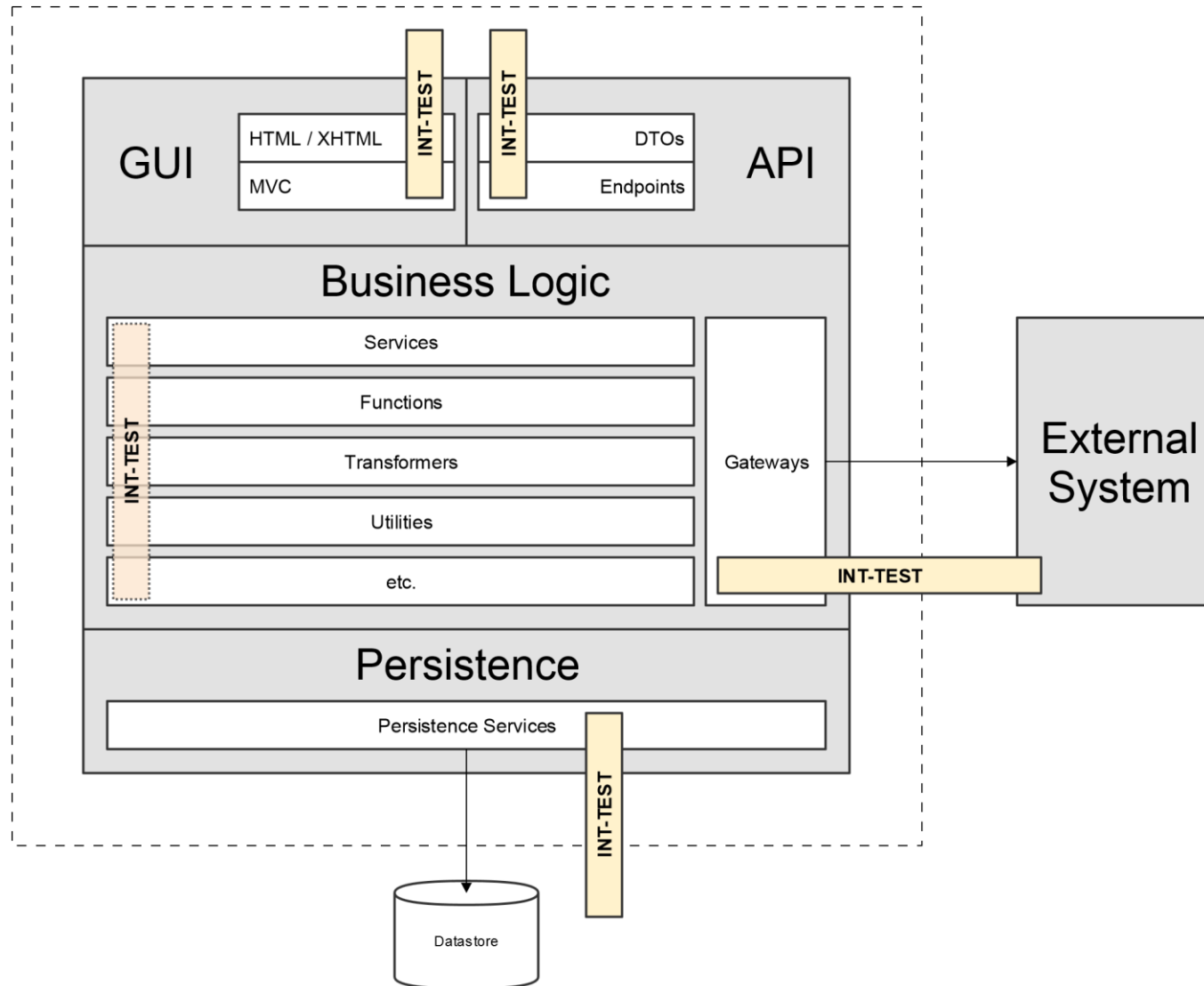
Repeatable

Self-Verifying

Timely

Test
„One Thing“

Integration Tests



Integration Tests

Runtime:
Seconds

Semi-Isolated

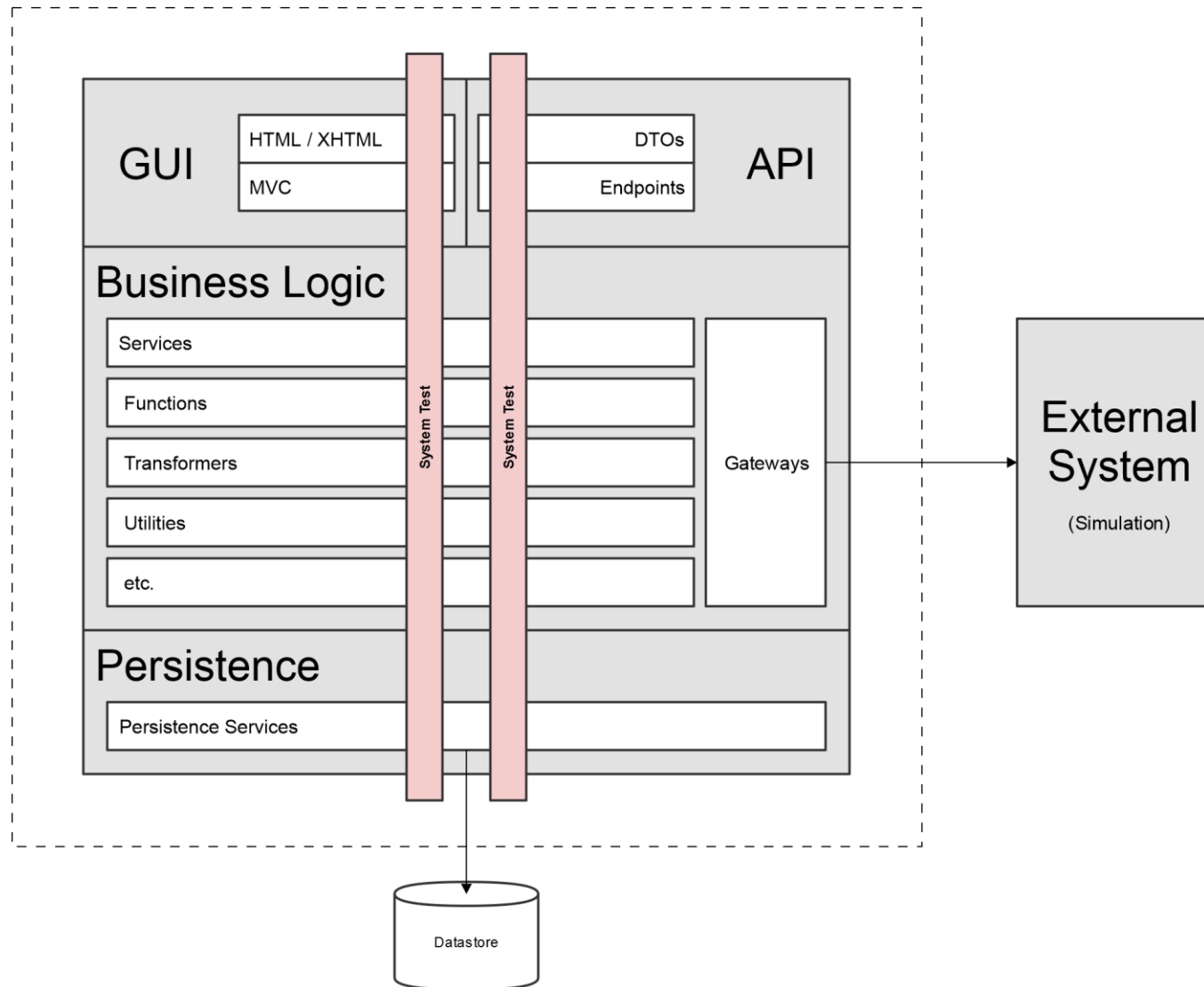
Repeatable

Self-Verifying

Close to
Implementation

Focused on a
Single Aspect

System Tests



System Tests

Runtime:
Minutes

Isolated from
external
Influences

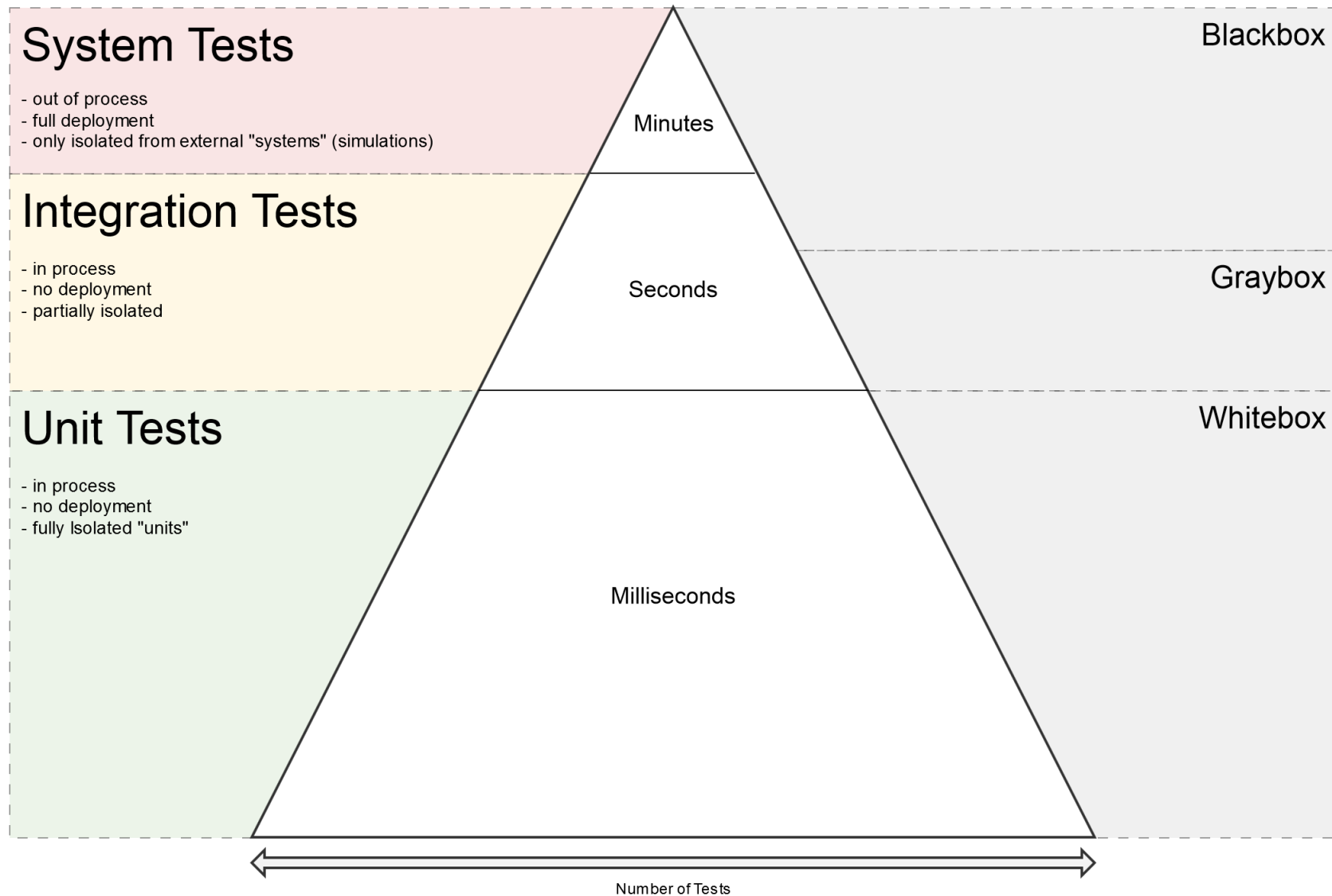
Repeatable

Self-Verifying

After
Implementation

Smoketest

Scope



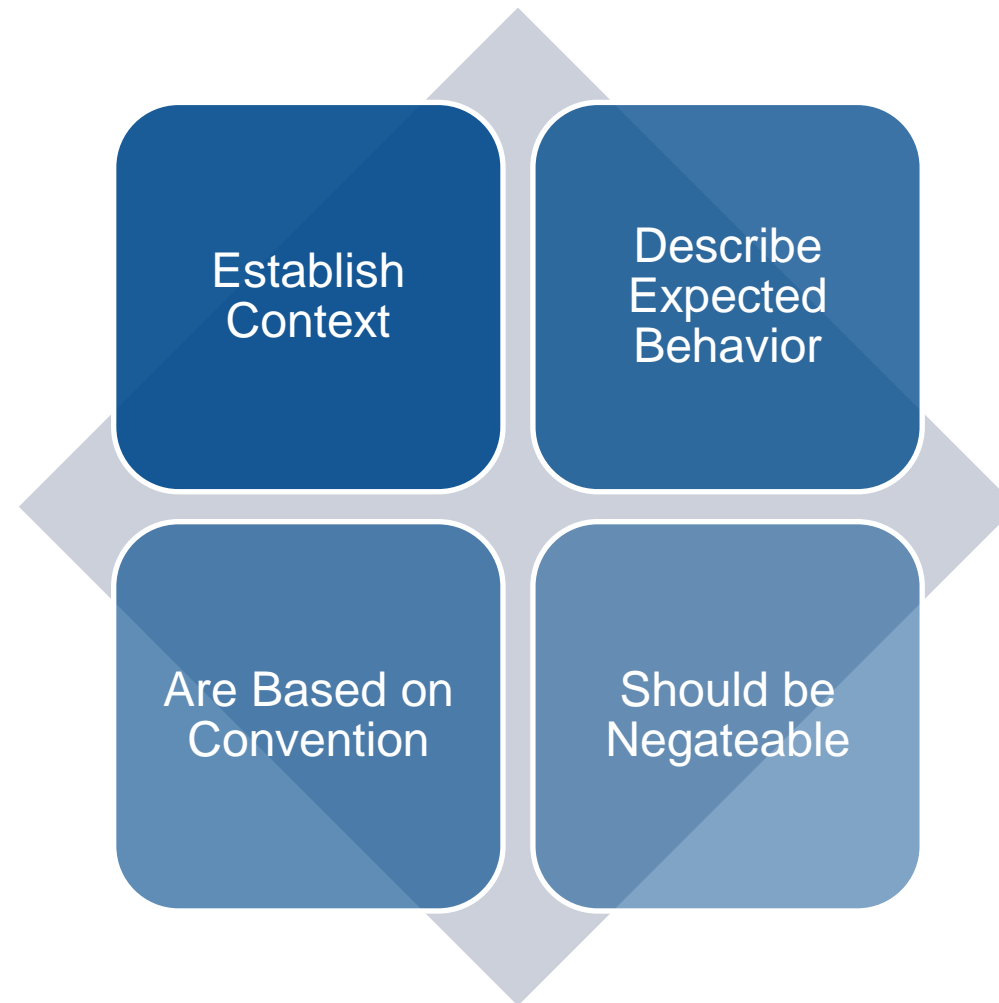
Which of the following should be tested and how?

- public methods
- private methods
- protected methods
- getter / setter
- serializable classes
- GUI controller
- 3rd party code
- „non-critical“ code
- persistence code
- framework features

NAMING

HELPING TO UNDERSTAND THE TEST

Test Names ...



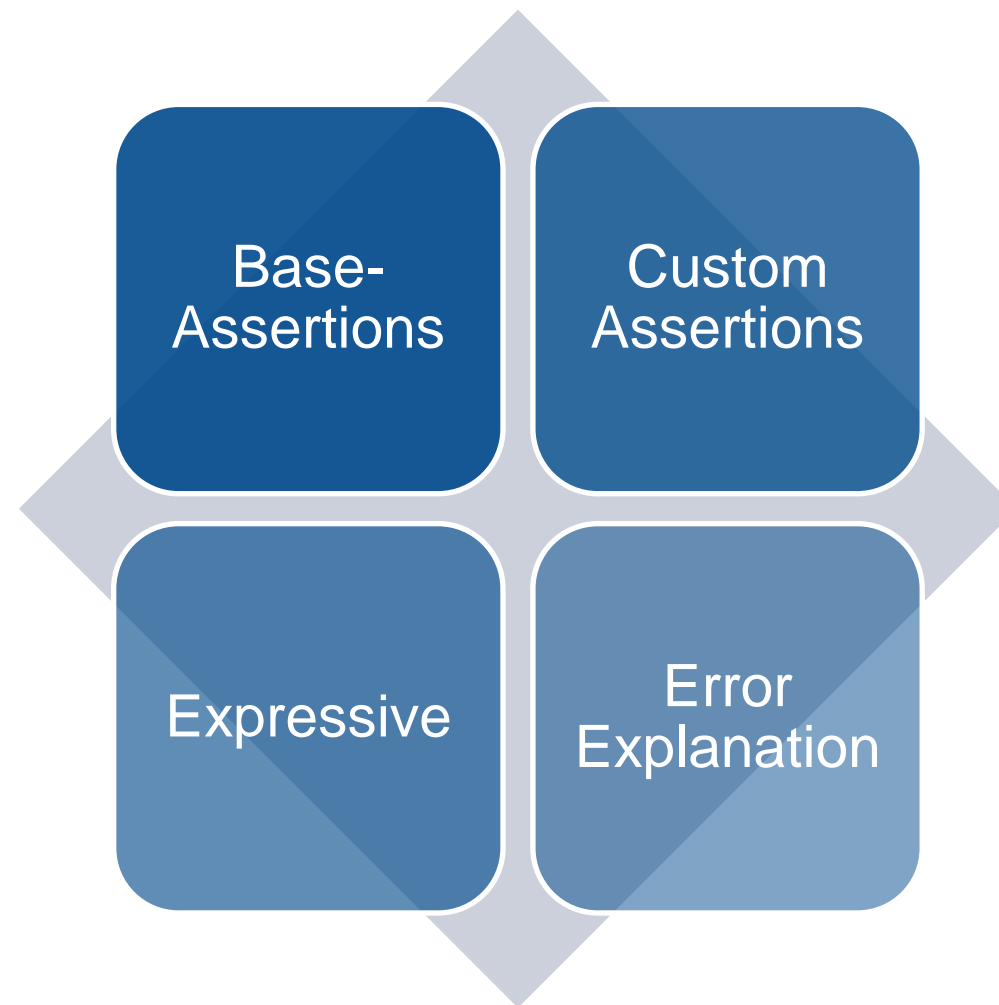
Discussion

- `test1()`, `test2()`,
- `happyPath()`
- `executeSave()`, `executeSaveFails()`
- `nullAsNameArgument()`, `nullAsCountryCode()`
- `addSingleEntry()`, `addMultipleEntries()`, `addNullEntry()`
- `nullIsNotAValidArgumentForName()`, `nullCountryCodeThrowsException()`
- `persistingAnInstanceInvalidatesCache()`
- `mergingDataWithInvalidDateFormatThrowsException()`

ASSERTIONS

VERIFYING RESULTS

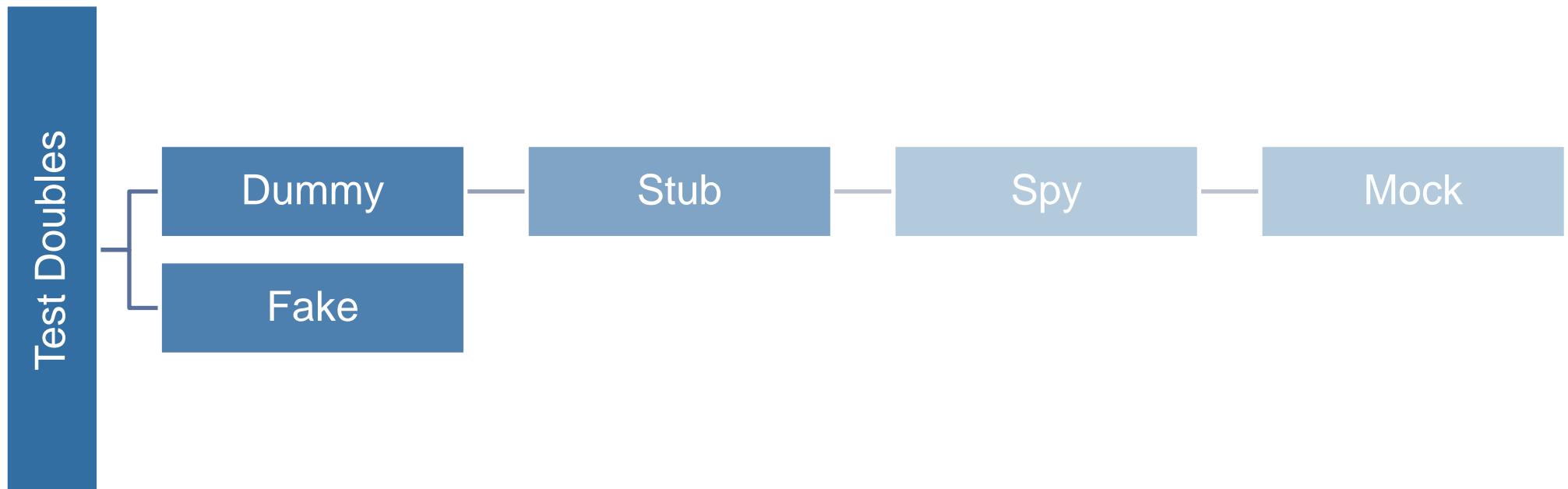
Assertion Frameworks

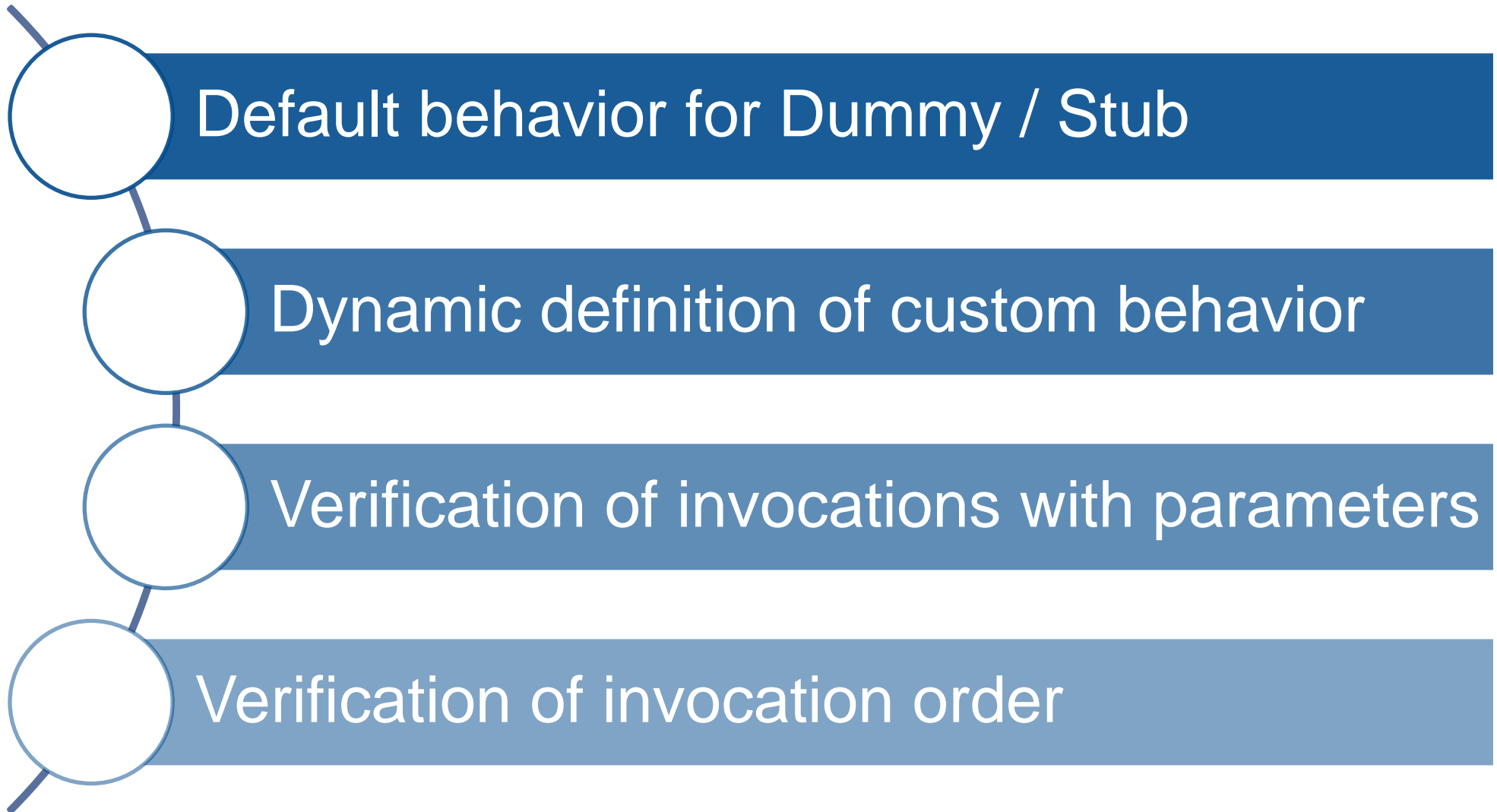


MOCKING

ISOLATING THE UNIT UNDER TEST

Test Doubles („Mocking“)





Testing State vs. Behavior

State Testing

- Focus on result
- How result was achieved is of no concern
- No behavior verification

```
@Test
public void testPow () {
    double power = cut.power(2, 10);

    assertThat ( power, equalTo ( 1024.0 ) );
}
```

Behavior Testing

- Focus on behavior / interactions
- Verification of expected method invocations
- Does not replace state testing!

```
@Test
public void testSafeExecution () {
    Runnable r = mock ( Runnable.class );

    cut.safeExecute ( r );

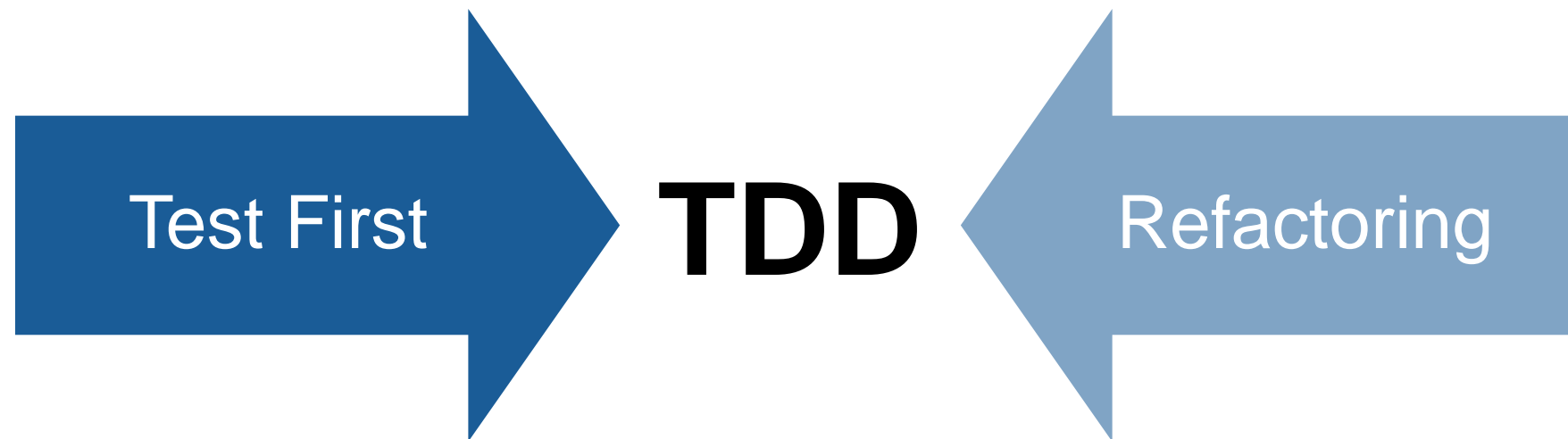
    verify ( r ).run();
}
```

LUNCH BREAK

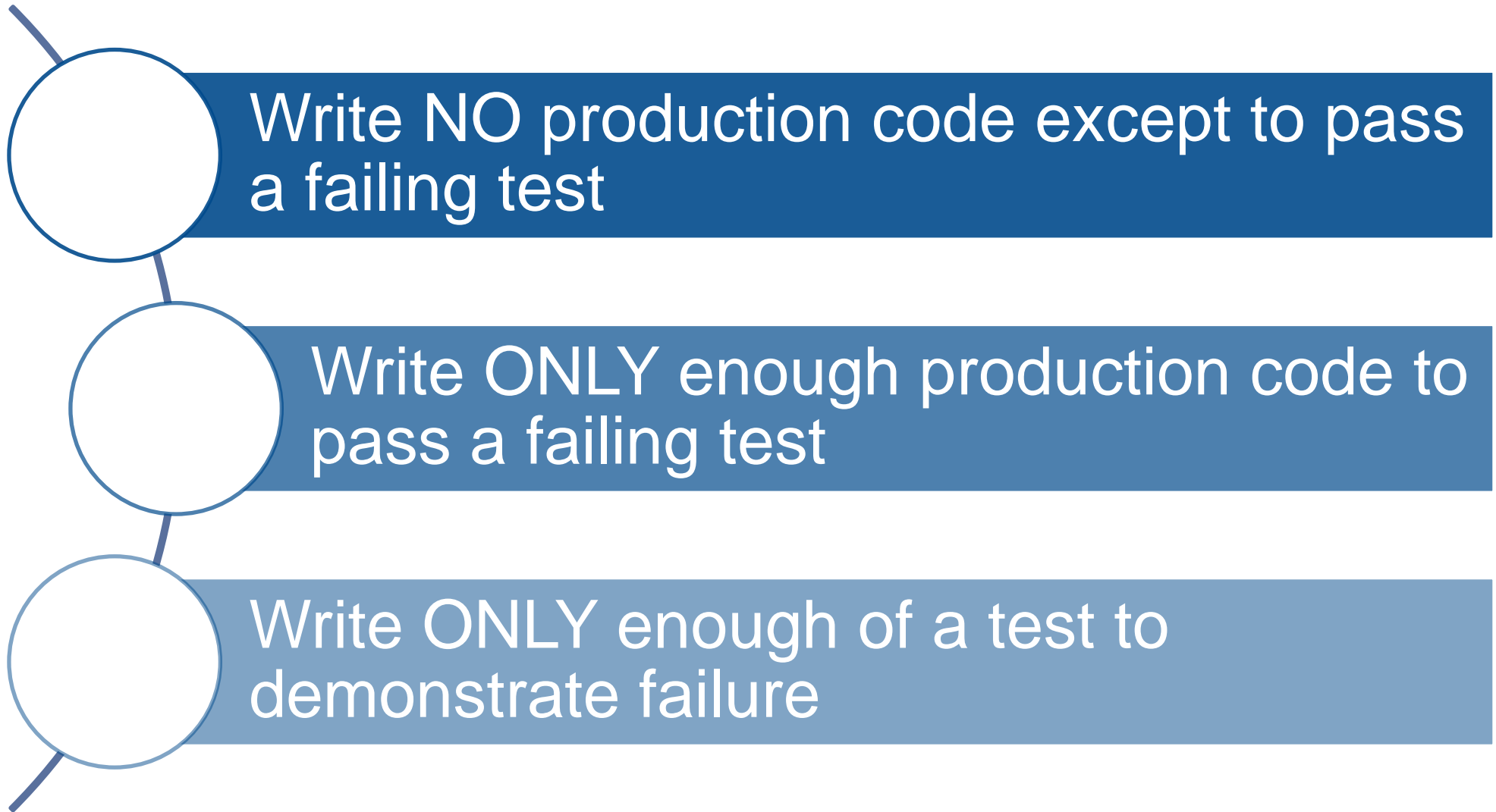
TEST-DRIVEN DEVELOPMENT

What is Test-Driven Development?

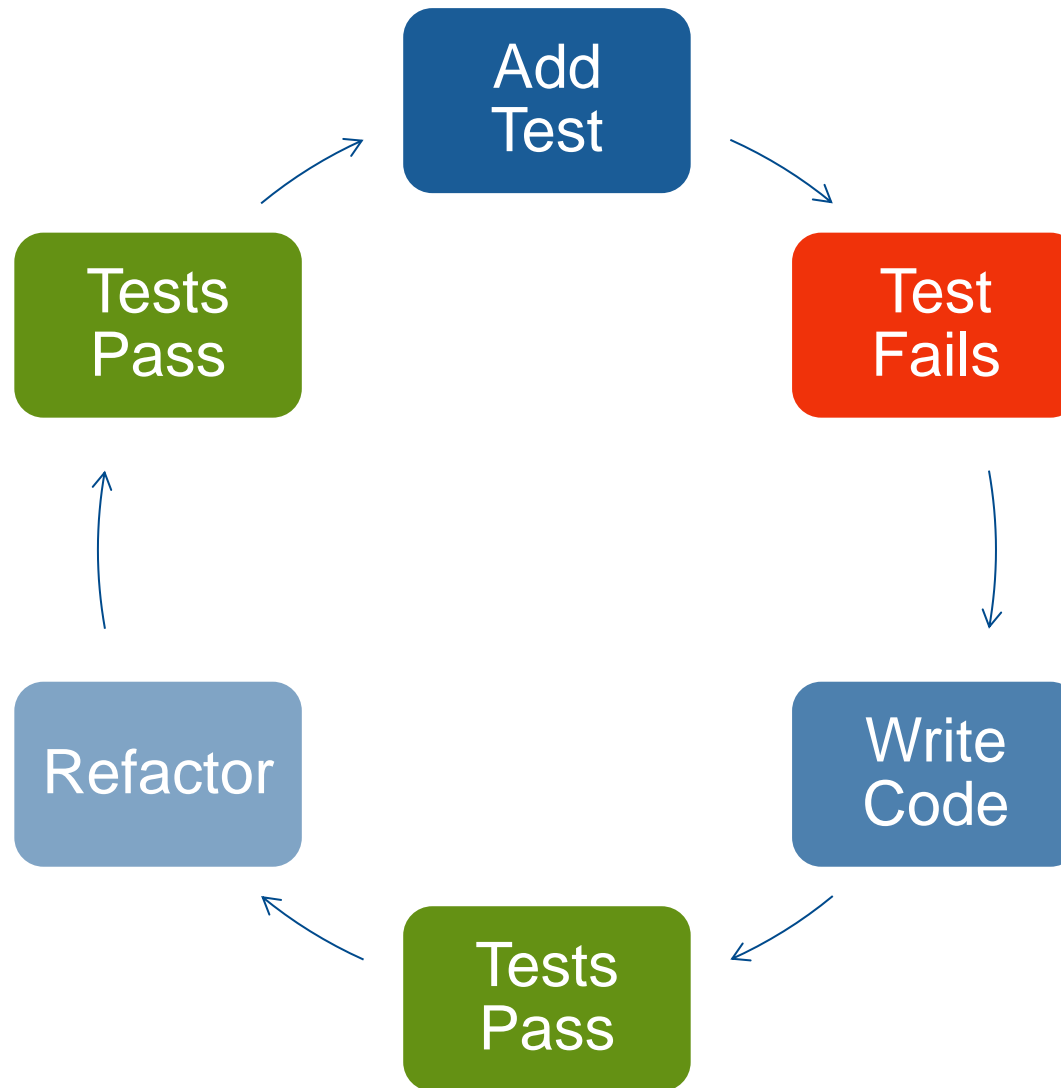




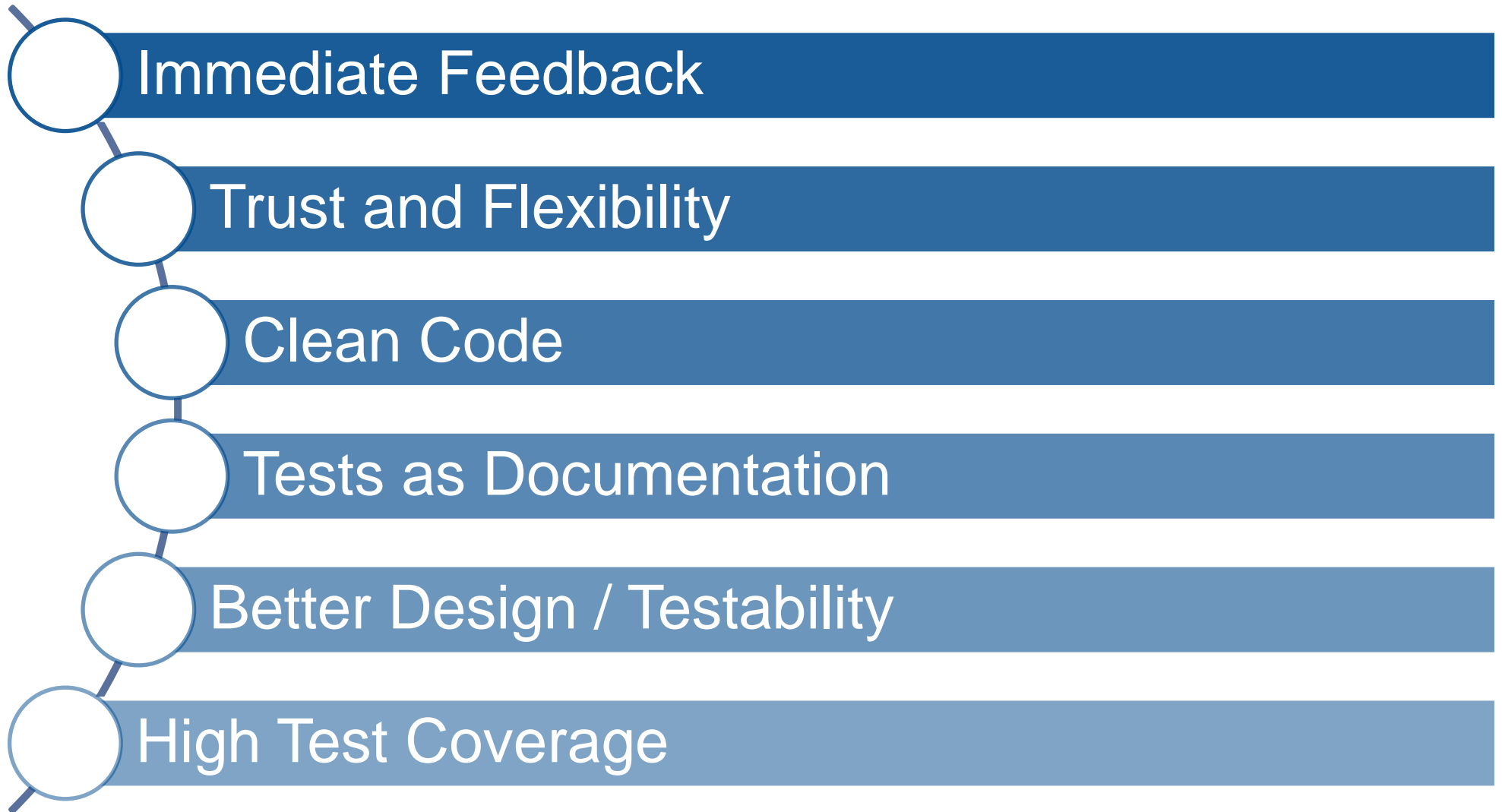
Three Rules of TDD



Life Cycle



(Possible) Benefits of Test-Driven Development



What are your thoughts on TDD?



< DOJOS >

Q & A