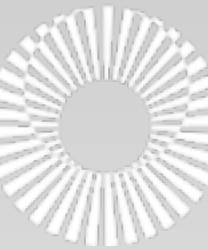


# **From TDD to BDD and back**

## **your guide on building a complete test automation stack**

**Mieke Kemme**  
**Elke Steegmans**  
**University Colleges Leuven Limburg**



# We are...

Elke Steegmans



PhD in Computer Science  
Catholic University Of Leuven

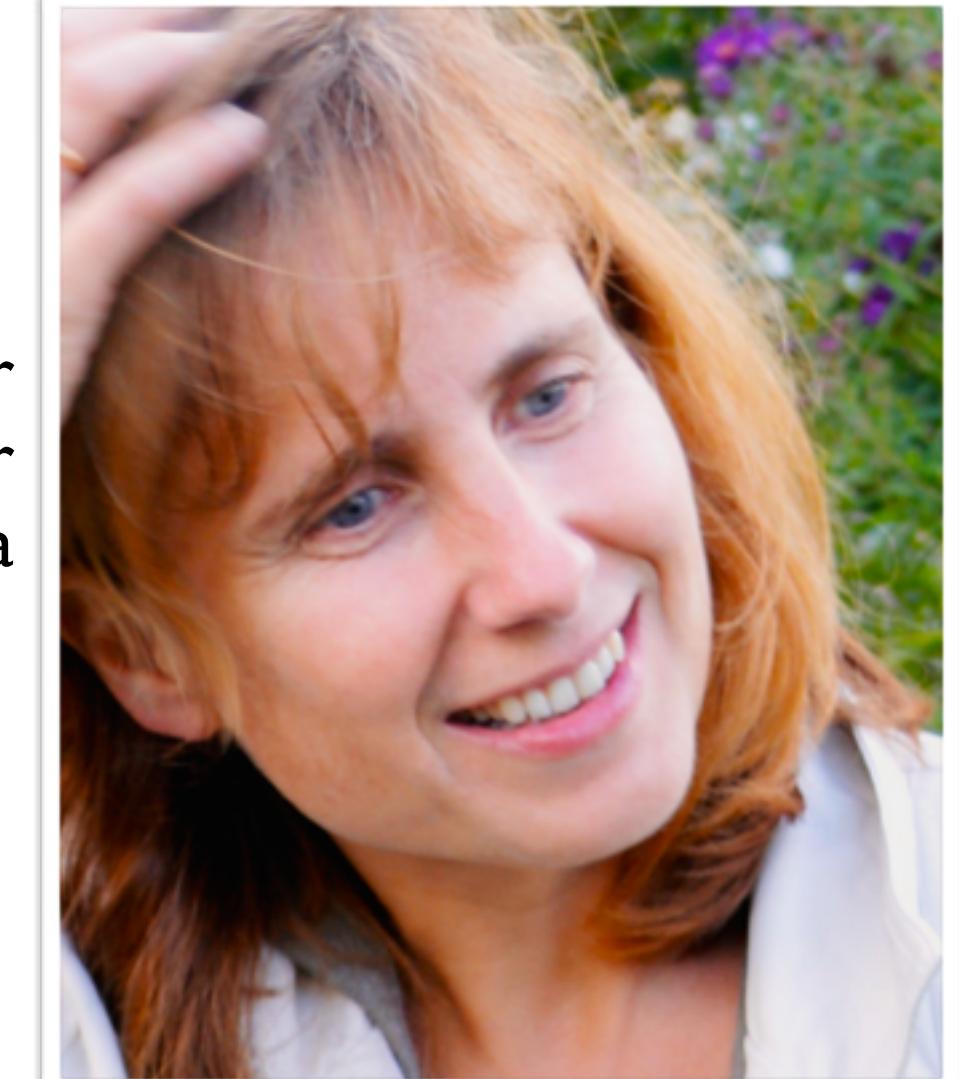


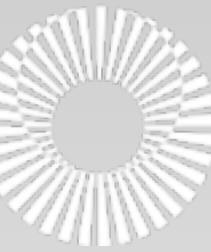
Software tester  
J2EE developer  
Smals, Cegeka



Lector software development  
for professional bachelors  
Applied Information Technology  
UC Leuven Limburg

Mieke Kemme





# Tools

JUnit

jBehave

Selenium

JDave

Serenity

givwenZen  
concordian

Fit

JNario

Spock

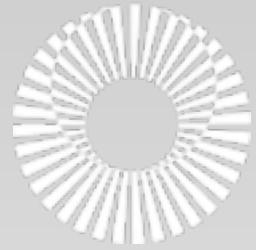
easyB

FitNesse

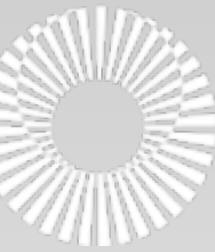
Cucumber

Robot Framework





# Test driven



# Test driven

```
public ExaminationTest() throws Exception {
    dateInPast = DATE_FORMATTER.parse("1996-07-07");
    dateInFuture = DATE_FORMATTER.parse("3996-07-07");
}

@Test
public void Examination_Should_create_an_examination_If_a_valid_length_
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_
        assertEquals(VALID_LENGTH_IN_CM, examination.getLength());
        assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());
}

@Test
public void Examination_Should_create_an_examination_If_a_valid_length_and_weight_
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_
        assertEquals(VALID_LENGTH_IN_CM, examination.getLength());
        assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());
        assertEquals(DATE_CURRENT, examination.getDate());
}

@Test(expected = IllegalArgumentException.class)
public void Examination_Should_thow_an_IllegalArgumentException_If_no_date_is_pro
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, null);
}

@Test(expected = IllegalArgumentException.class)
public void Examination_Should_thow_an_IllegalArgumentException_If_a_date_in_the_f
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, dateInFuture);
}

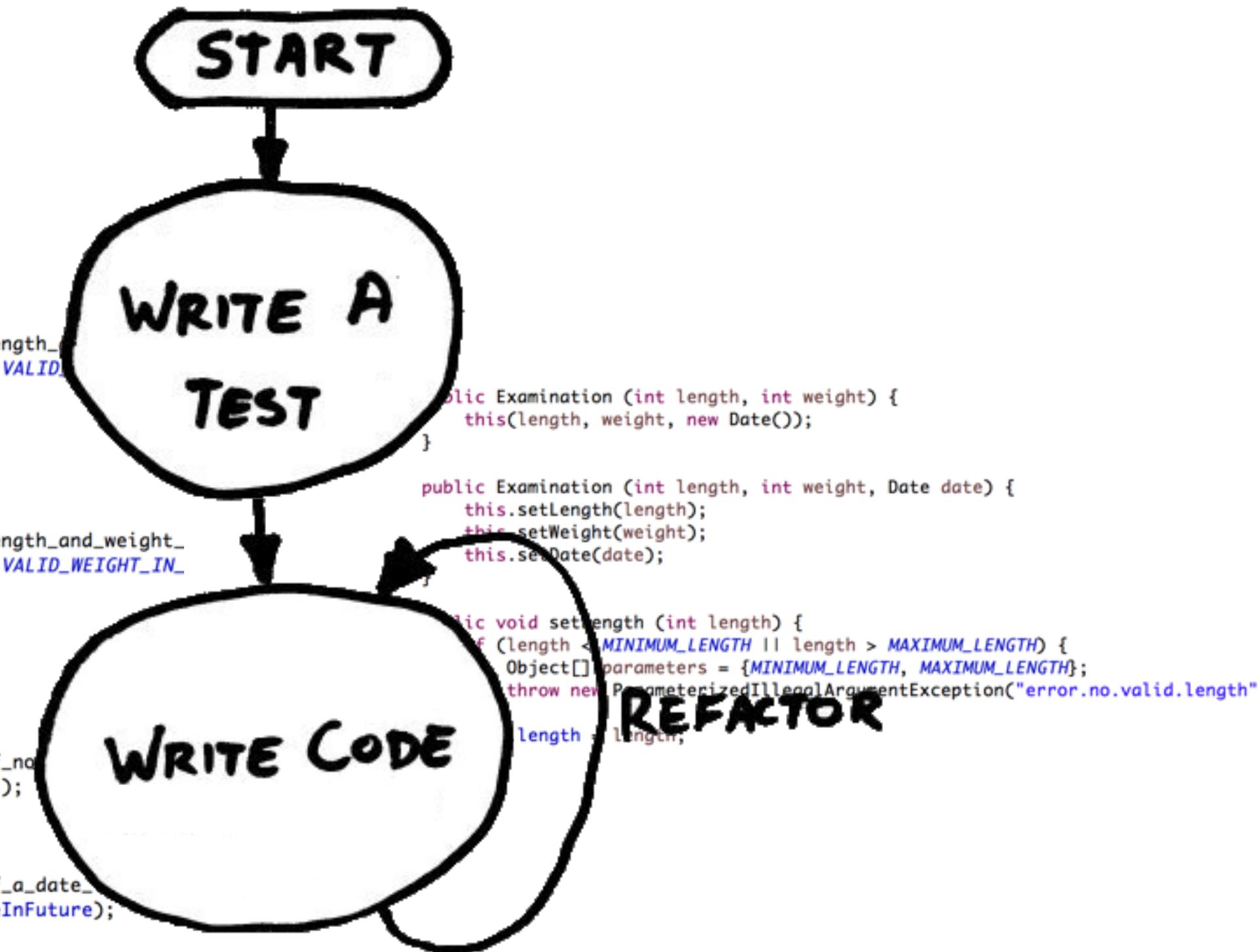
@Test
public void setLength_Should_replace_the_length_If_a_valid_length_is_provided(){
    Examination examination = createExamination();
    examination.setLength(OTHER_VALID_LENGTH_IN_CM);

    assertEquals(OTHER_VALID_LENGTH_IN_CM, examination.getLength());
}
```



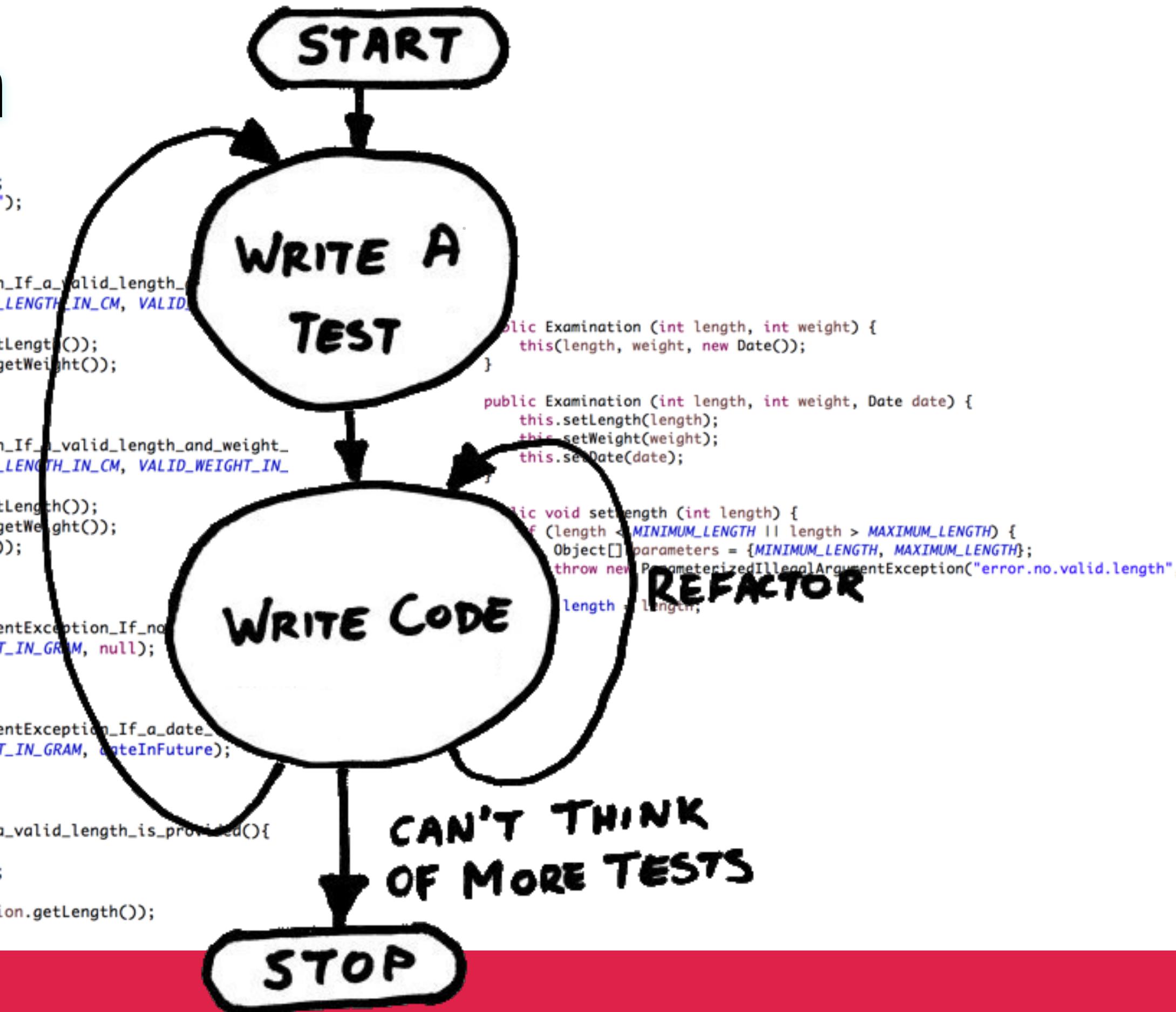
# Test driven

```
public ExaminationTest() throws Exception {  
    dateInPast = DATE_FORMATTER.parse("1996-07-07");  
    dateInFuture = DATE_FORMATTER.parse("3996-07-07");  
}  
  
@Test  
public void Examination_Should_create_an_examination_If_a_valid_length_  
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM);  
  
    assertEquals(VALID_LENGTH_IN_CM, examination.getLength());  
    assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());  
}  
  
@Test  
public void Examination_Should_create_an_examination_If_a_valid_length_and_weight_  
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM);  
  
    assertEquals(VALID_LENGTH_IN_CM, examination.getLength());  
    assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());  
    assertEquals(DATE_CURRENT, examination.getDate());  
}  
  
@Test(expected = IllegalArgumentException.class)  
public void Examination_Should_thow_an_IllegalArgumentException_If_no_date_is_provided()  
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, null);  
}  
  
@Test(expected = IllegalArgumentException.class)  
public void Examination_Should_thow_an_IllegalArgumentException_If_a_date_is_in_the_past()  
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, dateInFuture);  
}  
  
@Test  
public void setLength_Should_replace_the_length_If_a_valid_length_is_provided(){  
    Examination examination = createExamination();  
    examination.setLength(OTHER_VALID_LENGTH_IN_CM);  
  
    assertEquals(OTHER_VALID_LENGTH_IN_CM, examination.getLength());  
}
```



# Test driven

```
public ExaminationTest() throws Exception {  
    dateInPast = DATE_FORMATTER.parse("1996-07-07");  
    dateInFuture = DATE_FORMATTER.parse("3996-07-07");  
}  
  
@Test  
public void Examination_Should_create_an_examination_If_a_valid_length_  
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM);  
  
    assertEquals(VALID_LENGTH_IN_CM, examination.getLength());  
    assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());  
}  
  
@Test  
public void Examination_Should_create_an_examination_If_a_valid_length_and_weight_  
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM);  
  
    assertEquals(VALID_LENGTH_IN_CM, examination.getLength());  
    assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());  
    assertEquals(DATE_CURRENT, examination.getDate());  
}  
  
@Test(expected = IllegalArgumentException.class)  
public void Examination_Should_thow_an_IllegalArgumentException_If_no_date_is_provided()  
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, null);  
}  
  
@Test(expected = IllegalArgumentException.class)  
public void Examination_Should_thow_an_IllegalArgumentException_If_a_date_is_in_the_future()  
    new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, dateInFuture);  
}  
  
@Test  
public void setLength_Should_replace_the_length_If_a_valid_length_is_provided(){  
    Examination examination = createExamination();  
    examination.setLength(OTHER_VALID_LENGTH_IN_CM);  
  
    assertEquals(OTHER_VALID_LENGTH_IN_CM, examination.getLength());  
}
```



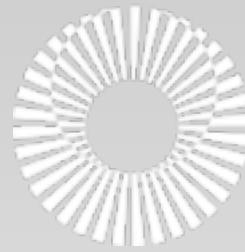
# Example application

- Goal: monitor patients
- General idea:
  - Patients are examined on a regular base
  - Length and weight are recorded
  - Body Mass Index (BMI) is calculated
  - History of examinations is kept

# Example application

- Goal: monitor patients
- General idea:
  - Patients are examined on a regular base
  - Length and weight are recorded
  - Body Mass Index (BMI) is calculated
  - History of examinations is kept

*WEIGHT (IN KG)*  
*HEIGHT2 (IN M)*



# Mock-Up

PERSONS
SOCIAL SECURITY NUMBER
98051822561
87081220062
<input type="button" value="New"/>

# Mock-Up

**PERSONS**

SOCIAL SECURITY NUMBER

98051822561  
87081220062

New

**NEW PERSON**

SOCIAL SECURITY NUMBER

BIRTH DATE

GENDER

MALE ▾

LENGTH (CM)

WEIGHT (KG)

EXAMINED ON

2015-10-05

OK CANCEL



# Mock-Up

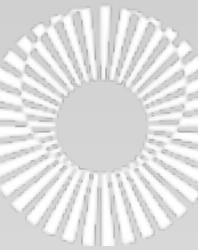
The diagram illustrates a user interface flow for managing person records. It consists of three main windows:

- PERSONS** window (left): Shows a list of Social Security Numbers: 98051822561 and 87081220062. A "New" button is located at the bottom left.
- NEW PERSON** window (bottom center): A form for entering new person information. Fields include:
  - SOCIAL SECURITY NUMBER
  - BIRTH DATE
  - GENDER (with a dropdown menu showing "MALE")
  - LENGTH (CM)
  - WEIGHT (KG)
  - EXAMINED ONButtons at the bottom are "OK" and "CANCEL".
- PERSON 98051822561** window (right): A detailed view of the selected person's record. Fields include:
  - SOCIAL SECURITY NUMBER (98051822561)
  - BIRTH DATE (1998-05-18)
  - GENDER (MALE)
  - LENGTH (CM) (180)
  - WEIGHT (KG) (75000)
  - EXAMINED ON (2015-10-05)
  - BMI (25.15)Buttons at the bottom are "ADD EXAMINATION" and "CANCEL".

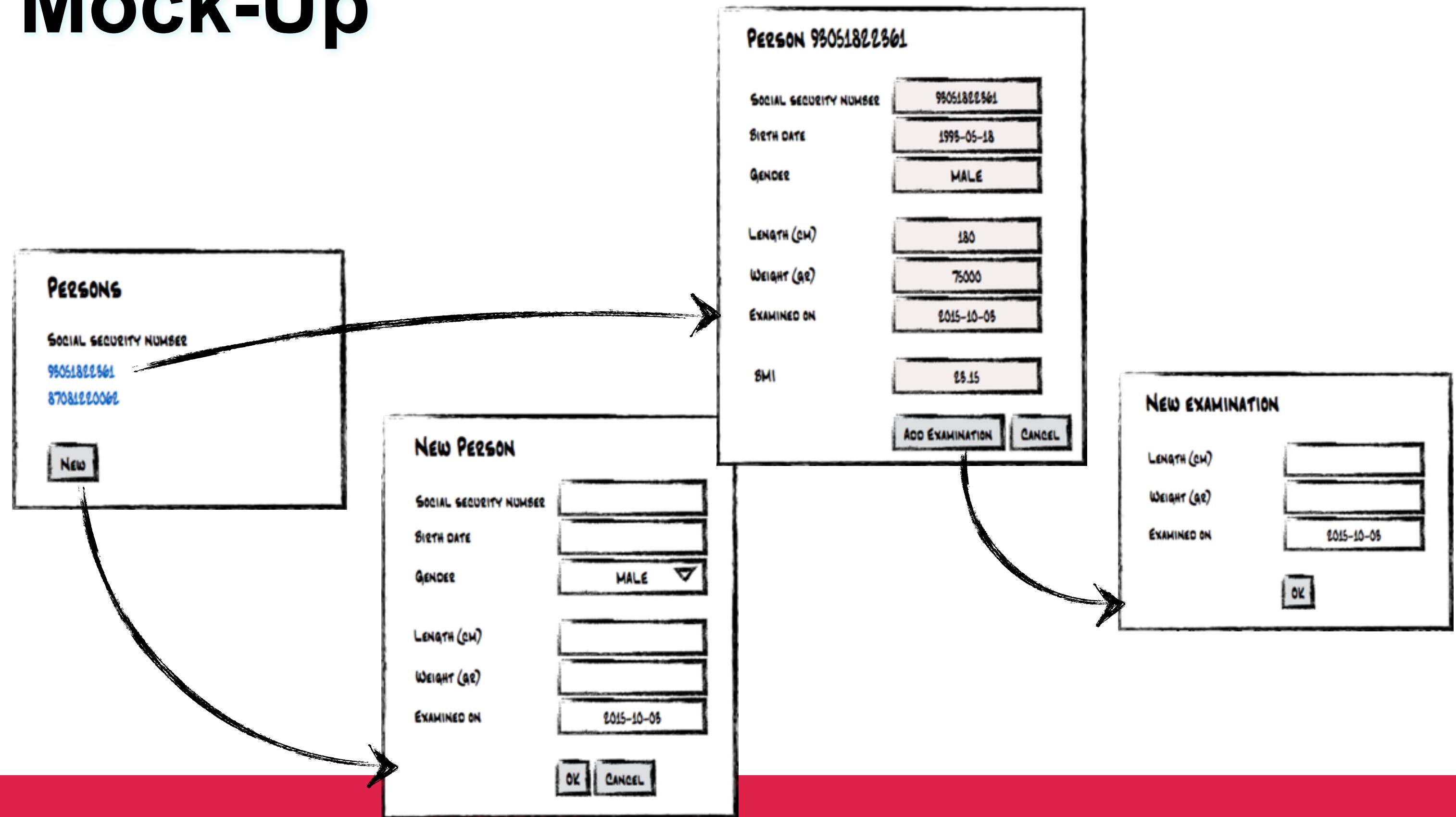
Curved arrows indicate the flow from the "PERSONS" window to the "NEW PERSON" window, and from the "NEW PERSON" window to the "PERSON 98051822561" window.

SOCIAL SECURITY NUMBER	98051822561
BIRTH DATE	1998-05-18
GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2015-10-05
BMI	25.15

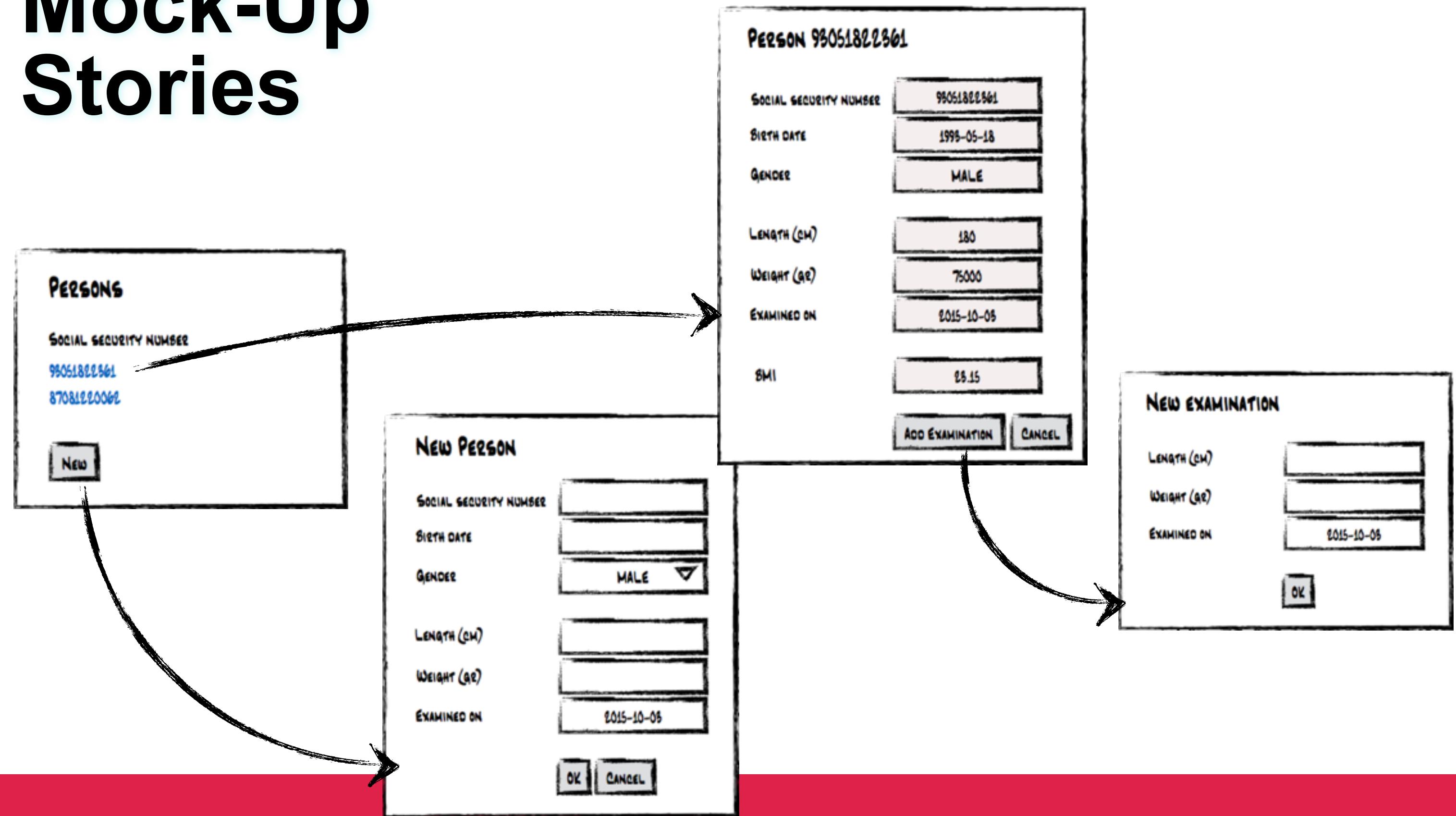
SOCIAL SECURITY NUMBER	
BIRTH DATE	
GENDER	MALE
LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05



# Mock-Up



# Mock-Up Stories



# Mock-Up Stories

Show patient overview

PERSONS

SOCIAL SECURITY NUMBER	
98051822561	
87081220062	

New

Add patient

NEW PERSON

SOCIAL SECURITY NUMBER	
BIRTH DATE	
GENDER	MALE
LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

OK CANCEL

Show patient detail

PERSON 98051822561

SOCIAL SECURITY NUMBER	98051822561
BIRTH DATE	1998-05-18
GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2015-10-05
BMI	25.15

ADD EXAMINATION CANCEL

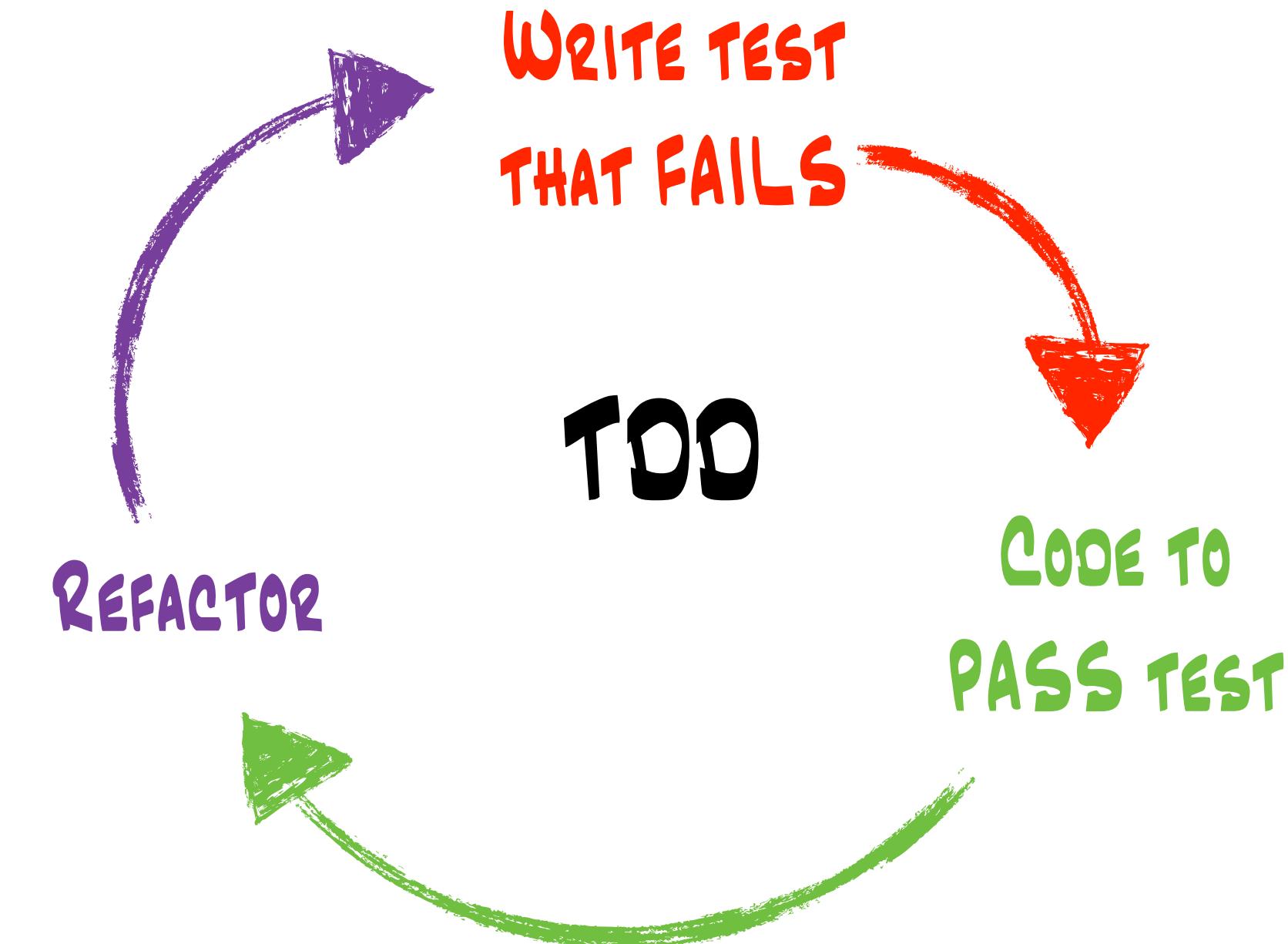
Add examination data

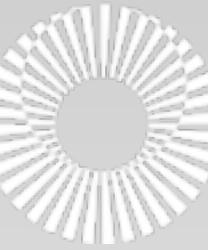
NEW EXAMINATION

LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

OK

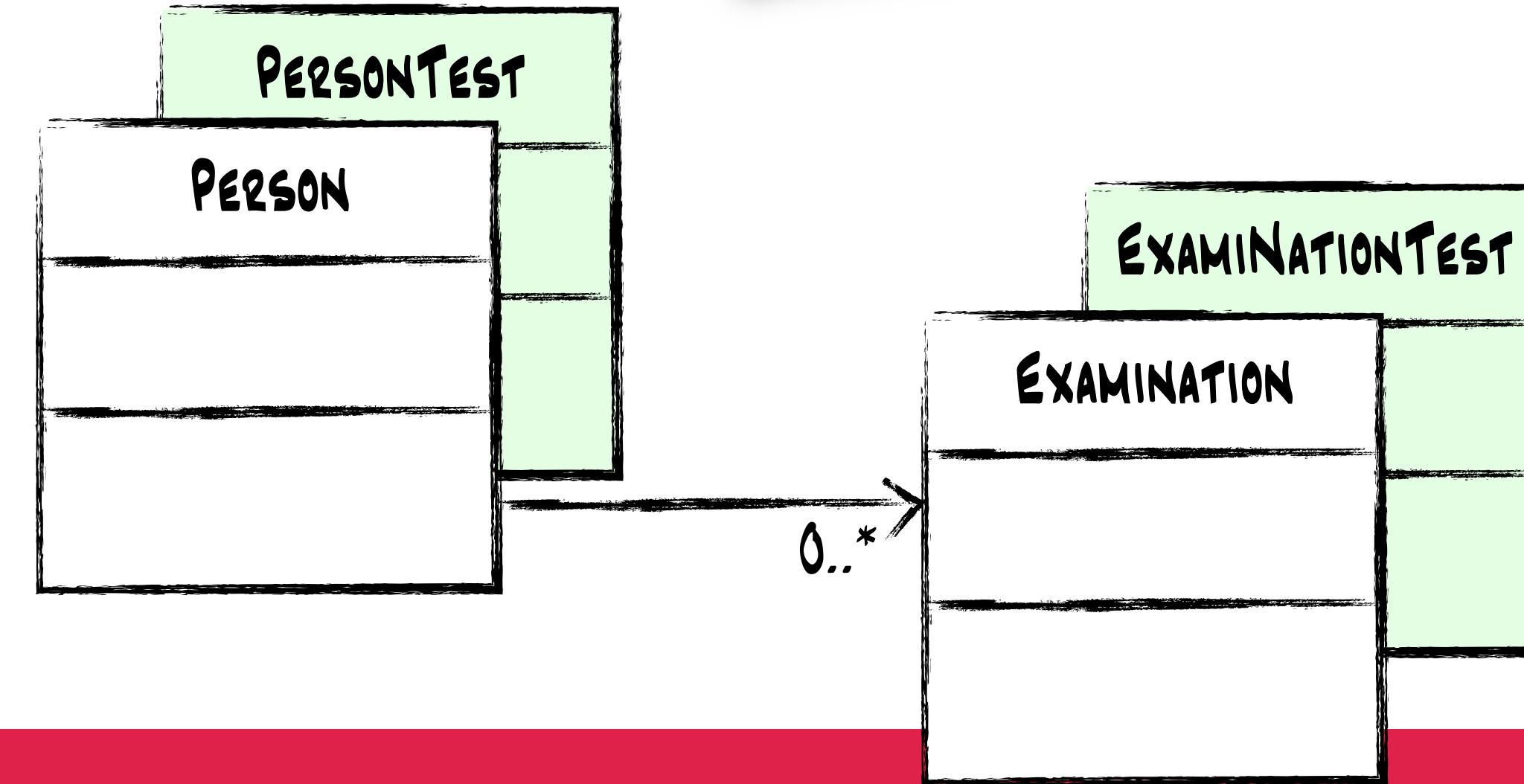
# Test driven...

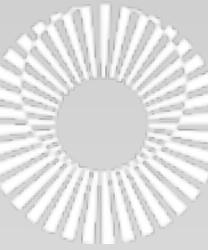




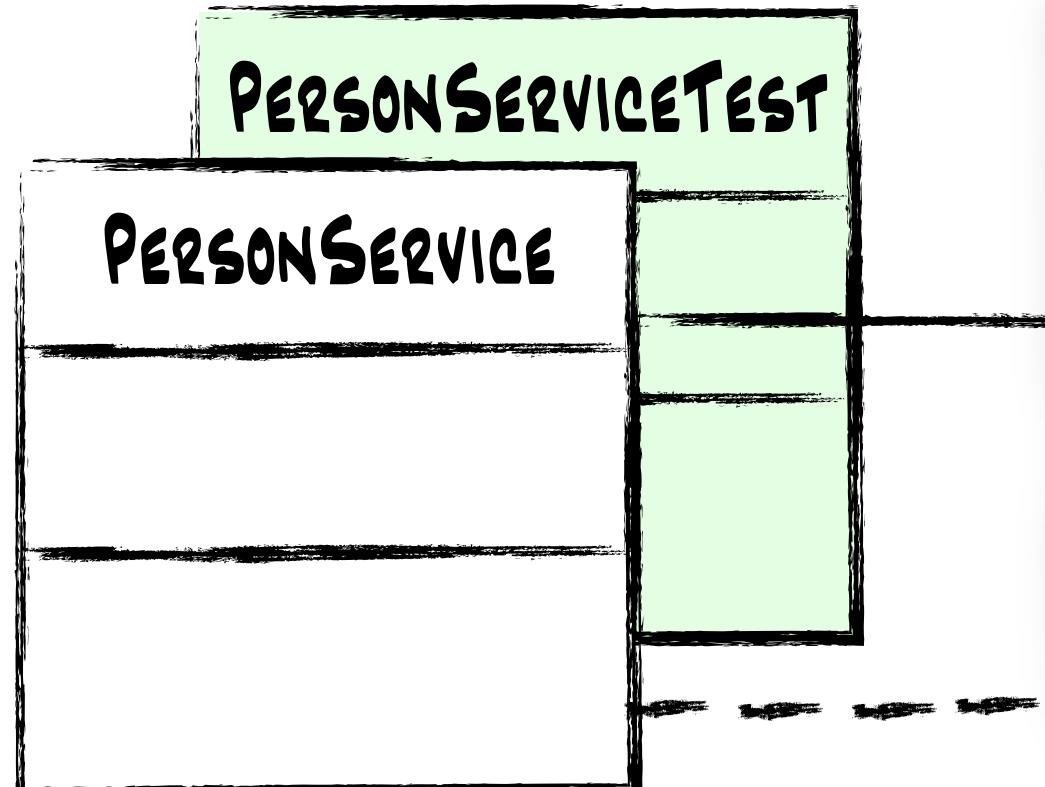
# domain: JUnit

```
@Test  
public void Examination_Should_create_an_examination_If_a_valid_length_and_weight_  
    Examination examination = new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_  
  
        assertEquals(VALID_LENGTH_IN_CM, examination.getLength());  
        assertEquals(VALID_WEIGHT_IN_GRAM, examination.getWeight());  
        assertEquals(DATE_CURRENT, examination.getDate());  
  
    }  
  
    @Test(expected = IllegalArgumentException.class)  
    public void Examination_Should_throw_an_IllegalArgumentException_If_no_date_is_pro  
        new Examination(VALID_LENGTH_IN_CM, VALID_WEIGHT_IN_GRAM, null);  
    }
```



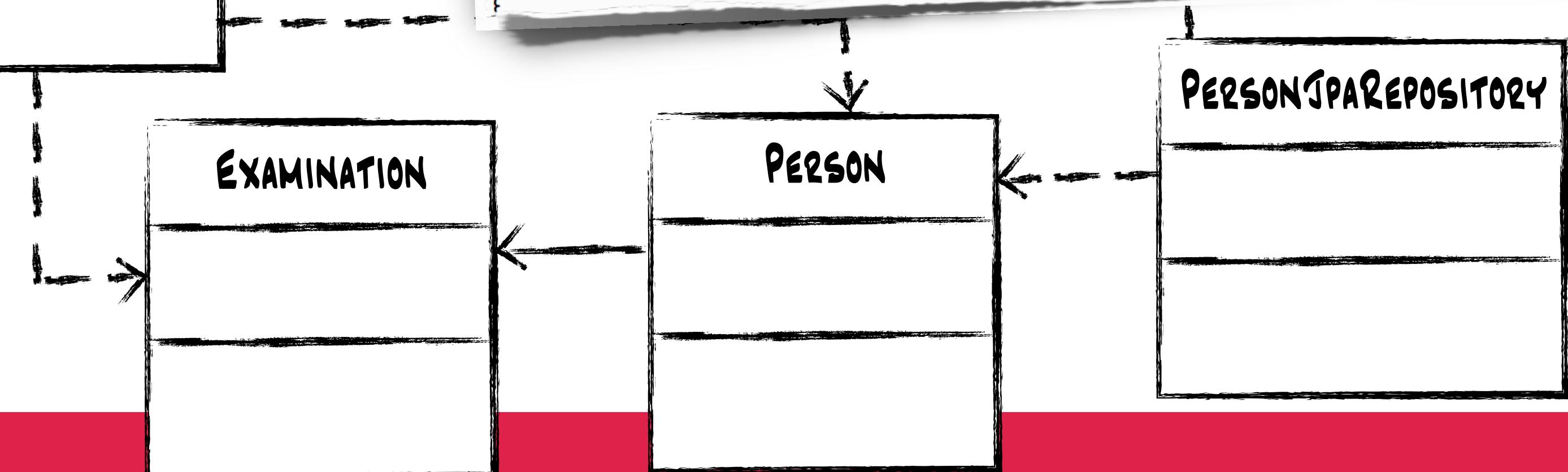


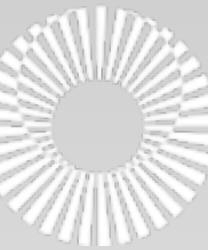
# service unit: Mockito



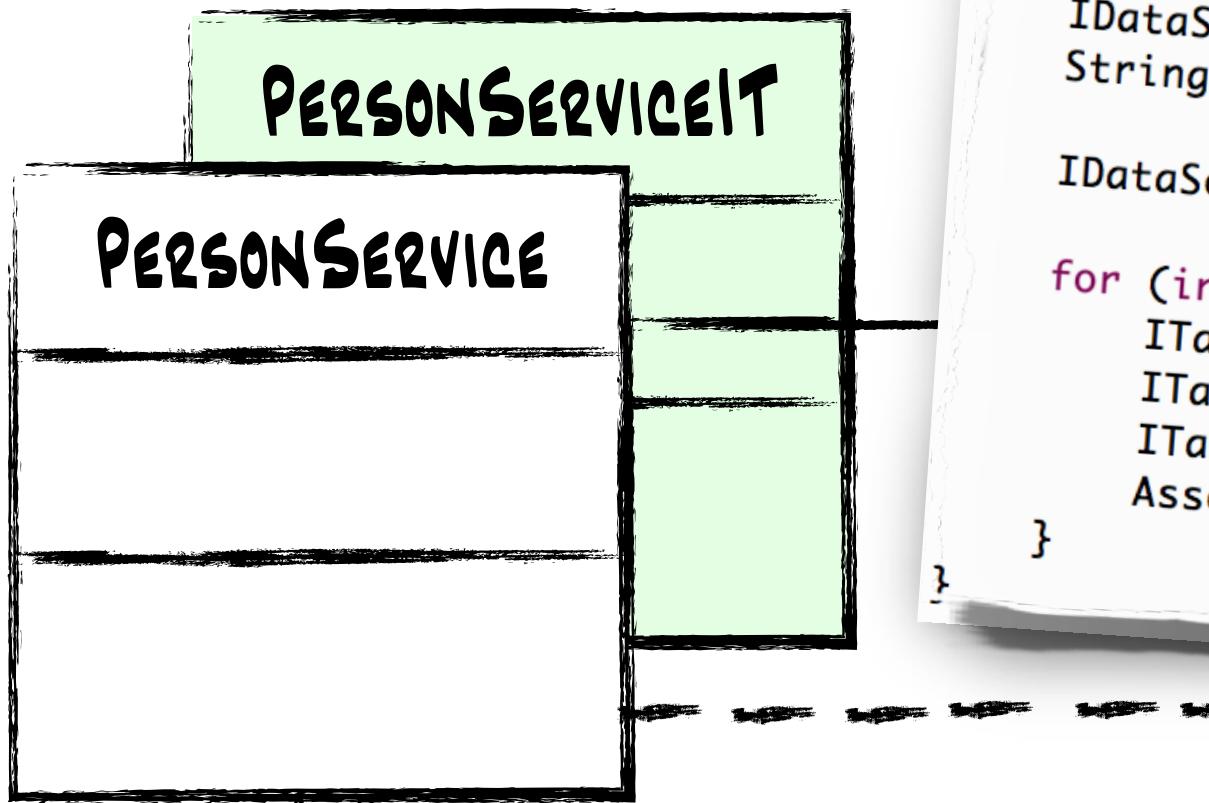
```
    @Test
    public void addExamination_Should_add_the_examination_to_the_person_on_the_repository_If
        given(mockPersonRepository.get(SOCIAL_SECURITY_NUMBER)).willReturn(mockPerson);
        service.addExamination(mockExamination, SOCIAL_SECURITY_NUMBER);
        then(mockPersonRepository).should().get(SOCIAL_SECURITY_NUMBER);
        then(mockPerson).should().addExamination(mockExamination);
        then(mockPersonRepository).should().update(mockPerson);
    }

    @Test(expected = IllegalArgumentException.class)
    public void addExamination_Should_throw_an_exception_If_a_valid_examination_but_no_person
        service.addExamination(mockExamination, null);
    }
```





# service integration: DbUnit, Arquillian



```
private void checkIfDatabaseIsAsExpected(String datasetName) throws Exception {
    IDataSet expectedDataSet = new FlatXmlDataSetBuilder().build(new File(datasetName));
    String[] tableNames = expectedDataSet.getTableNames();

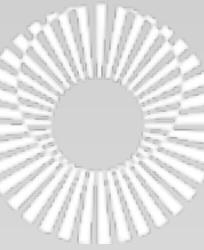
    IDataSet actualDataSet = getConnection().createDataSet(tableNames);

    for (int i = 0; i < tableNames.length; i++) {
        ITable expectedTable = expectedDataSet.getTable(tableNames[i]);
        ITable actualTable = actualDataSet.getTable(tableNames[i]);
        Assert.assertEquals(expectedTable, actualTable);
    }
}

@RunWith(Arquillian.class)
public class PersonServiceIT {

    @ArquillianResource
    private Context context;

    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap
            .create(JavaArchive.class).addPackages(true, "org.ucll.demo")
            .addAsResource("META-INF/persistence.xml", "META-INF/persistence.xml")
            .addAsManifestResource("META-INF/beans.xml", "beans.xml");
    }
}
```



# ui: Selenium

PERSONS

SOCIAL SECURITY NUMBER	95051822361
	87081220062

New

NEW PERSON

SOCIAL SECURITY NUMBER	
BIRTH DATE	
GENDER	
LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

OK CANCEL

SHOWPERSONDETAILTEST

PERSON 95051822361

SOCIAL SECURITY NUMBER	95051822361
BIRTH DATE	1995-05-18
GENDER	MALE

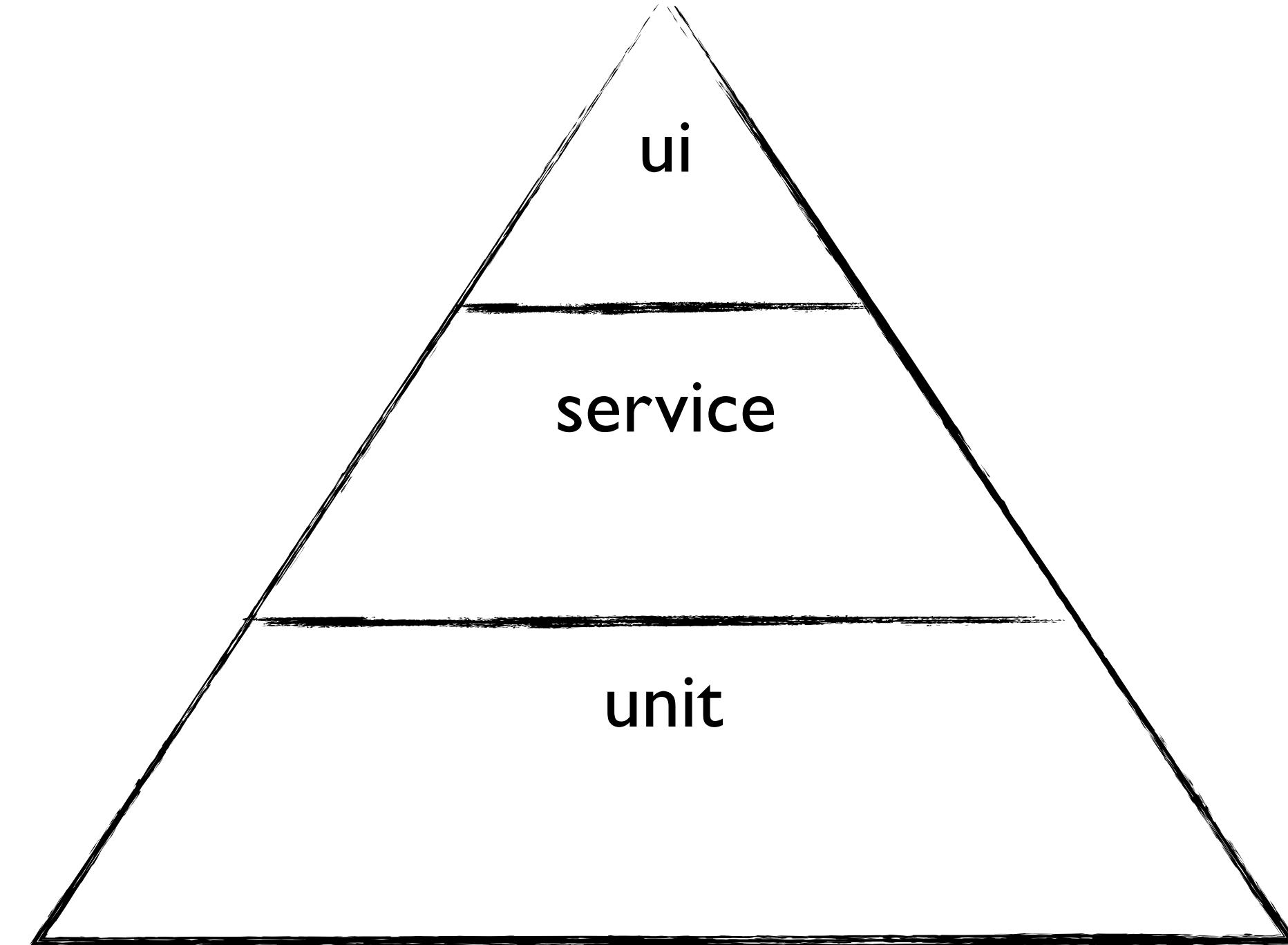
```
public double getBmi() {  
    WebElement bmiField = getDriver().findElement(By.id("personForm:bmi"));  
    String bmiAsString = bmiField.getAttribute("value");  
    return Double.parseDouble(bmiAsString);  
  
}  
  
public ExaminationDetailPage addExamination() {  
    WebElement buttonExamination = getDriver().findElement(By.id("personForm:addEx"));  
    buttonExamination.click();  
    return new ExaminationDetailPage(getDriver());  
}
```

NEW EXAMINATION

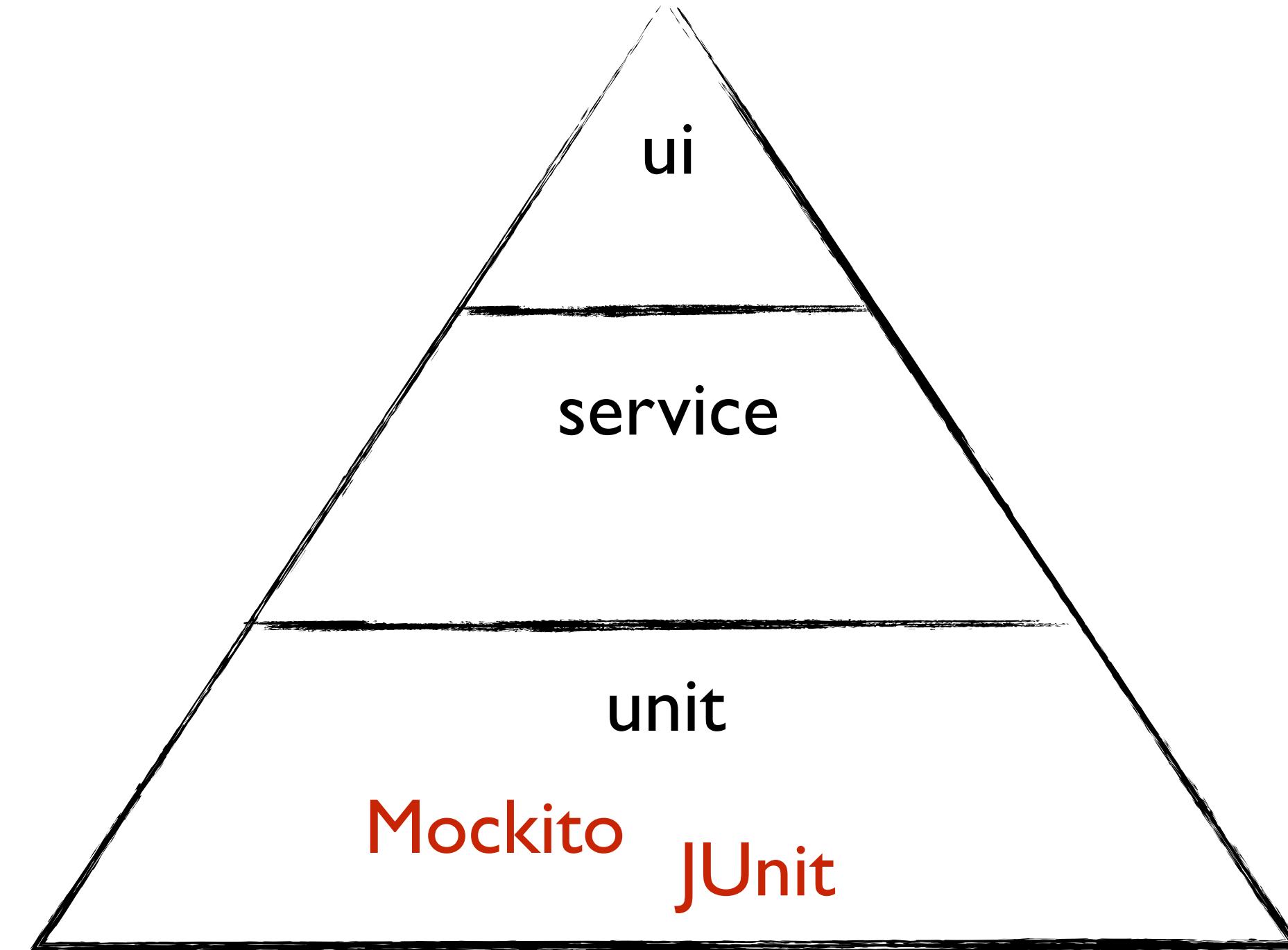
LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

OK

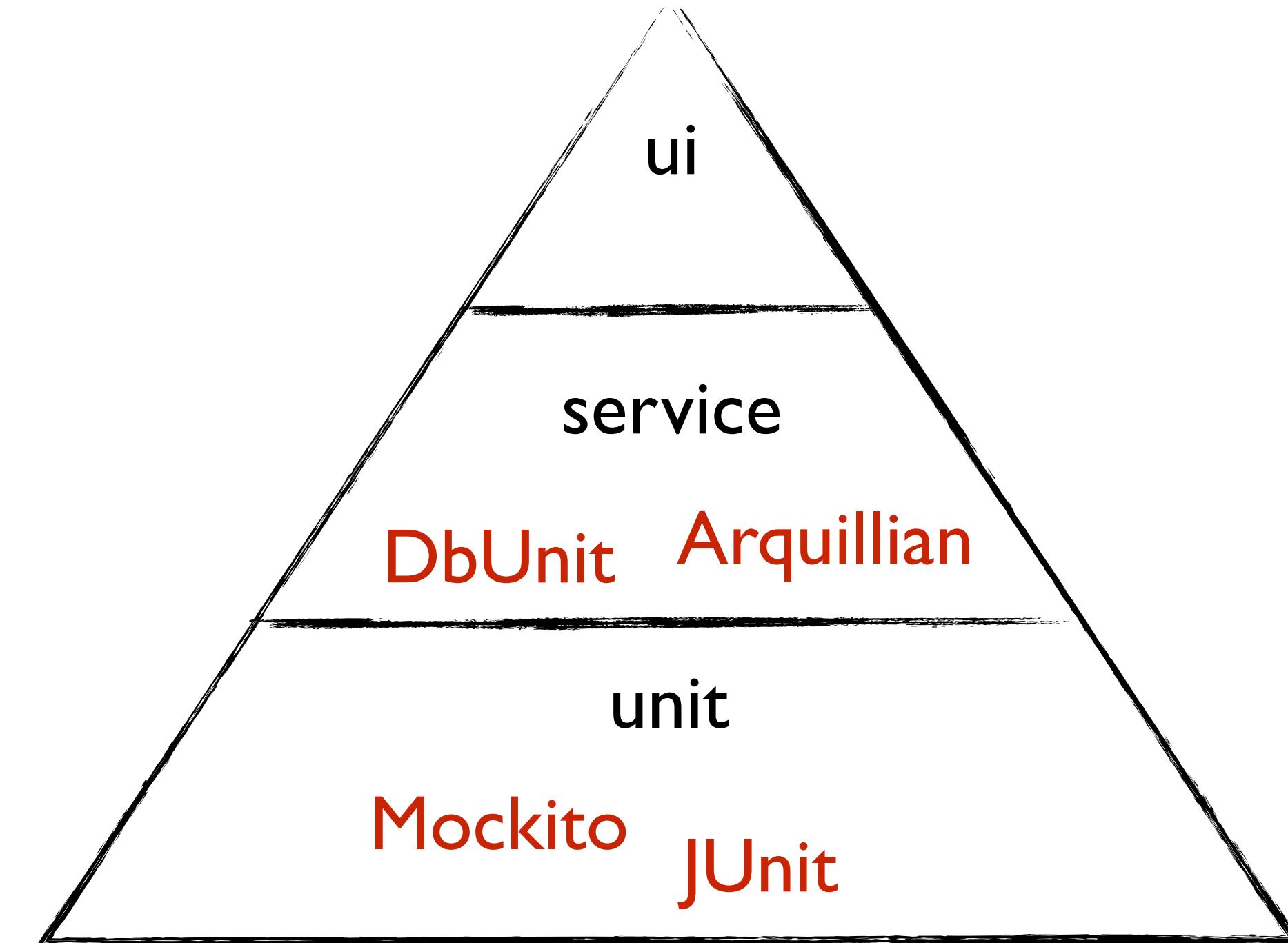
# Test Pyramid



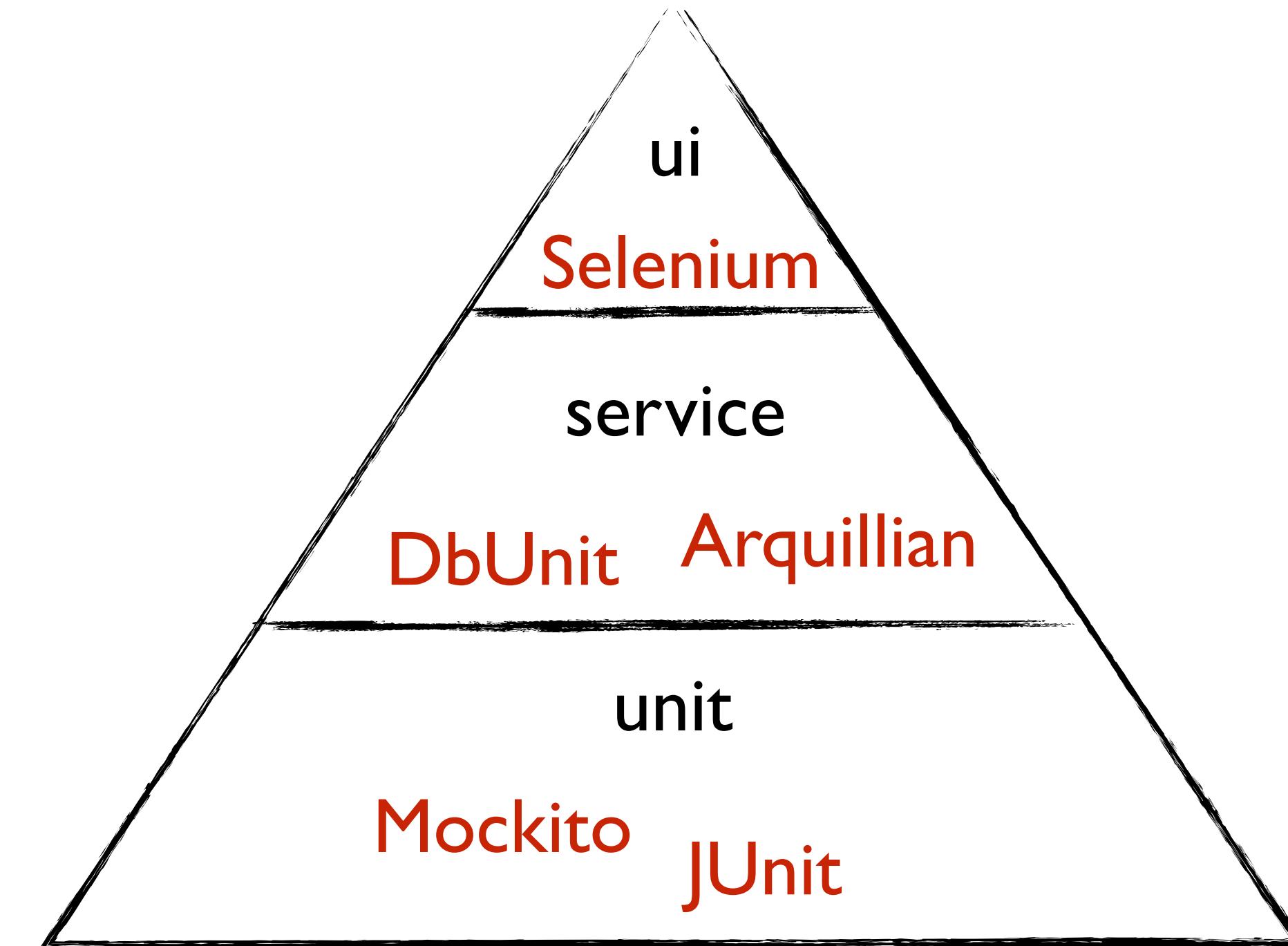
# Test Pyramid



# Test Pyramid

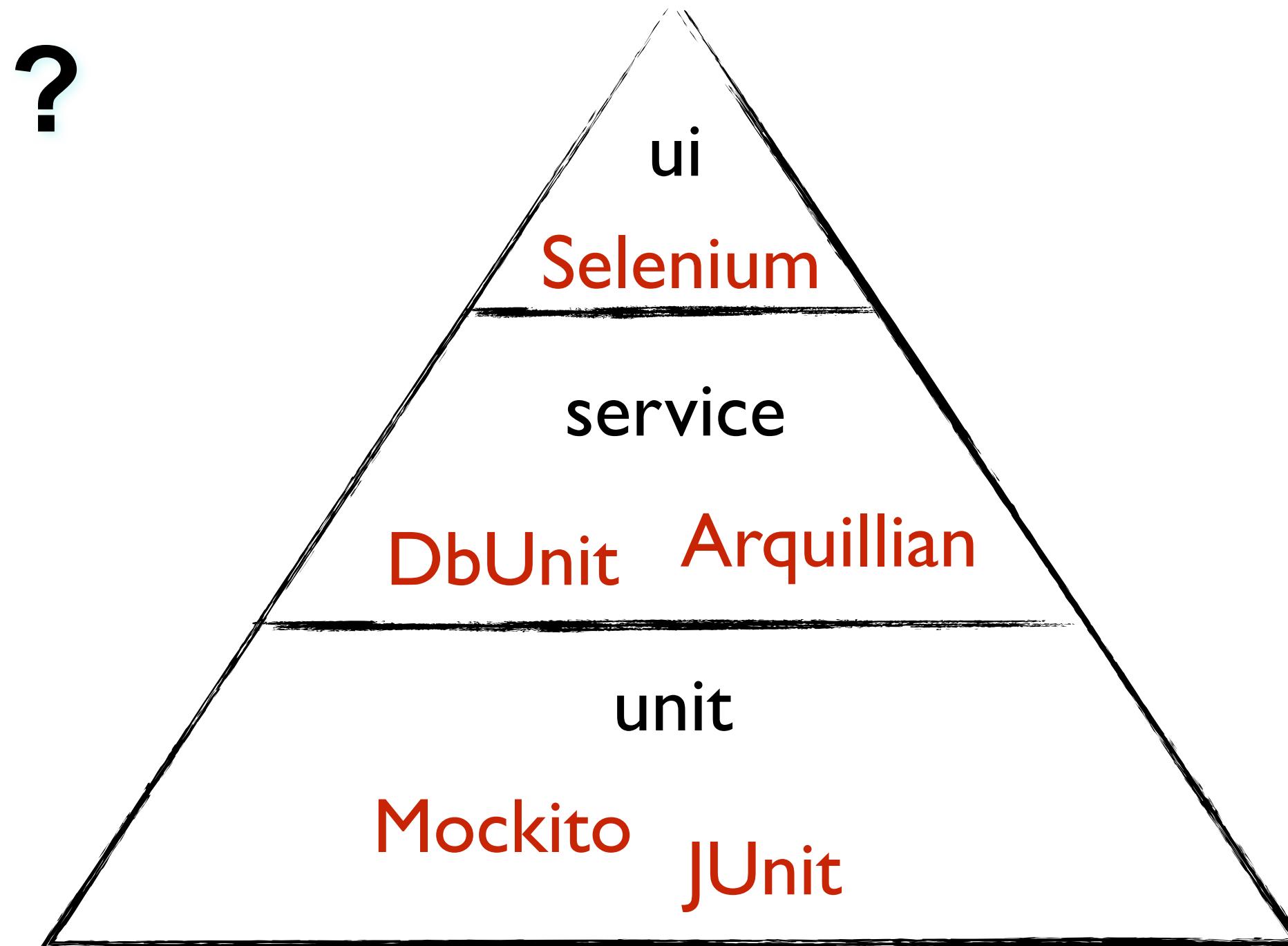


# Test Pyramid



# Test Pyramid

TDD ?



# WRITE TEST THAT FAILS

Show patient overview

PERSONS
SOCIAL SECURITY NUMBER
98051822561
87081220062

New

Add

NEW PE

SOCIAL SECURITY

BIRTH DATE

GENDER

LENGTH (CM)

WEIGHT (KG)

EXAMINED ON

START  
HERE

OK CANCEL

Show patient detail

PERSON 98051822561

SOCIAL SECURITY NUMBER	98051822561
BIRTH DATE	1998-05-18
GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2015-10-05
AGE	25.15

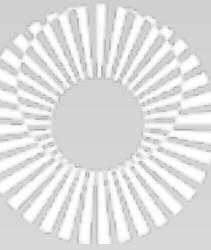
ADD EXAMINATION CANCEL

Add examination data

NEW EXAMINATION

LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

OK



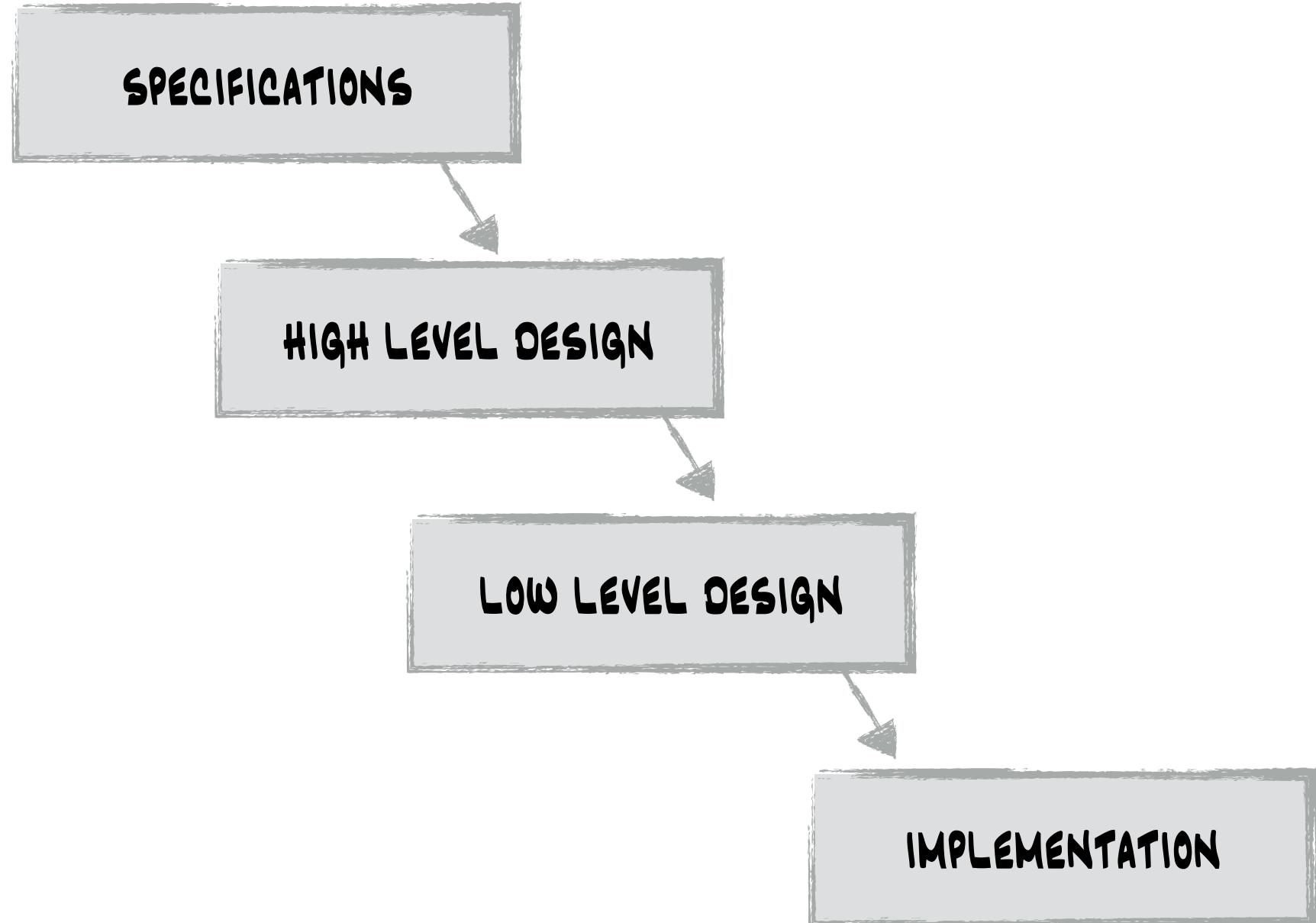
# BDD

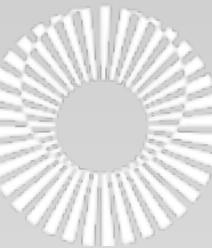
- Behaviour Driven Development
- = TDD done right
- Think behaviour: 'should' instead of 'test'
- Ubiquitous Language

*Where we choose to call it BDD or ATDD or Specification by Example, we want the same result – a shared common understanding of what is to be built to try to build the ‘thing’ right the first time. We know it never will be, but the less rework, the better.*

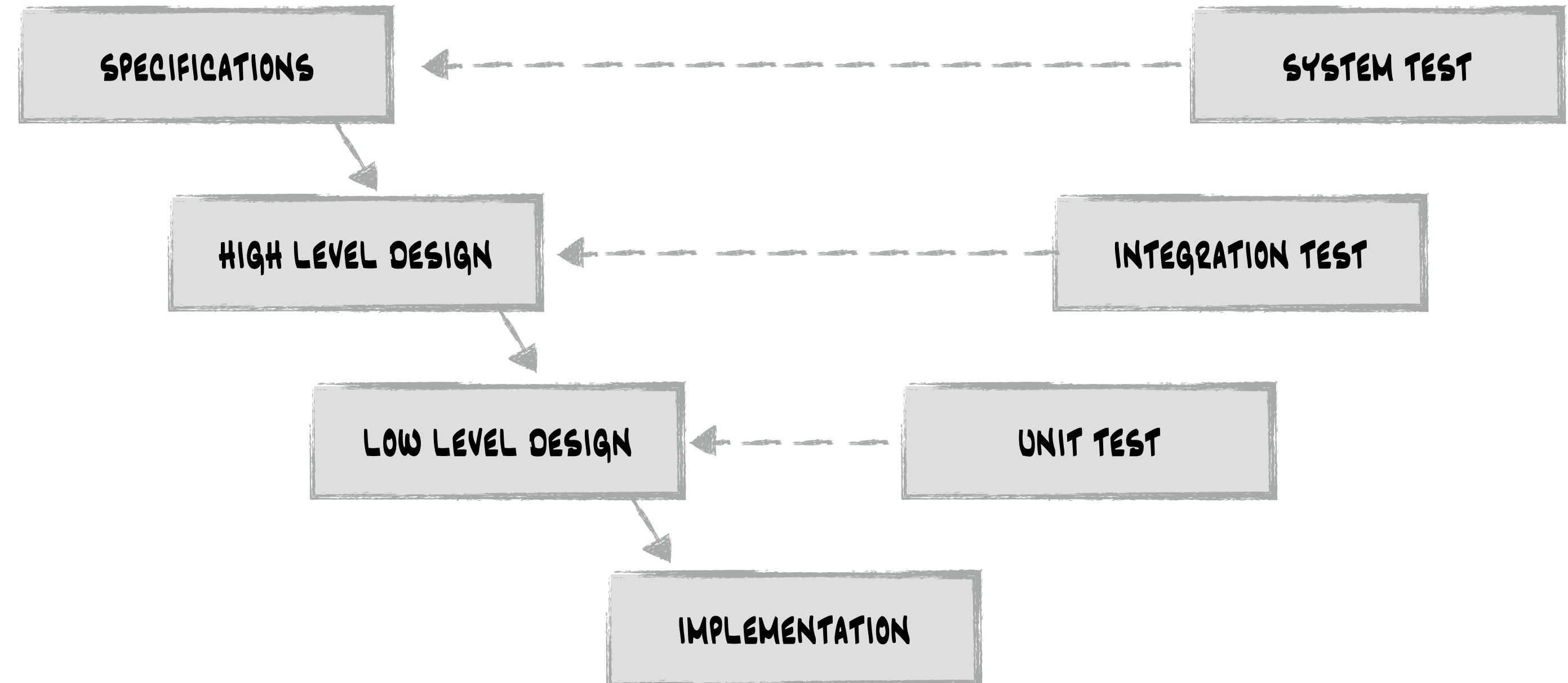
*Janet Gregory*

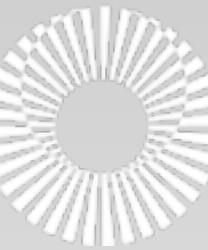
# V-model



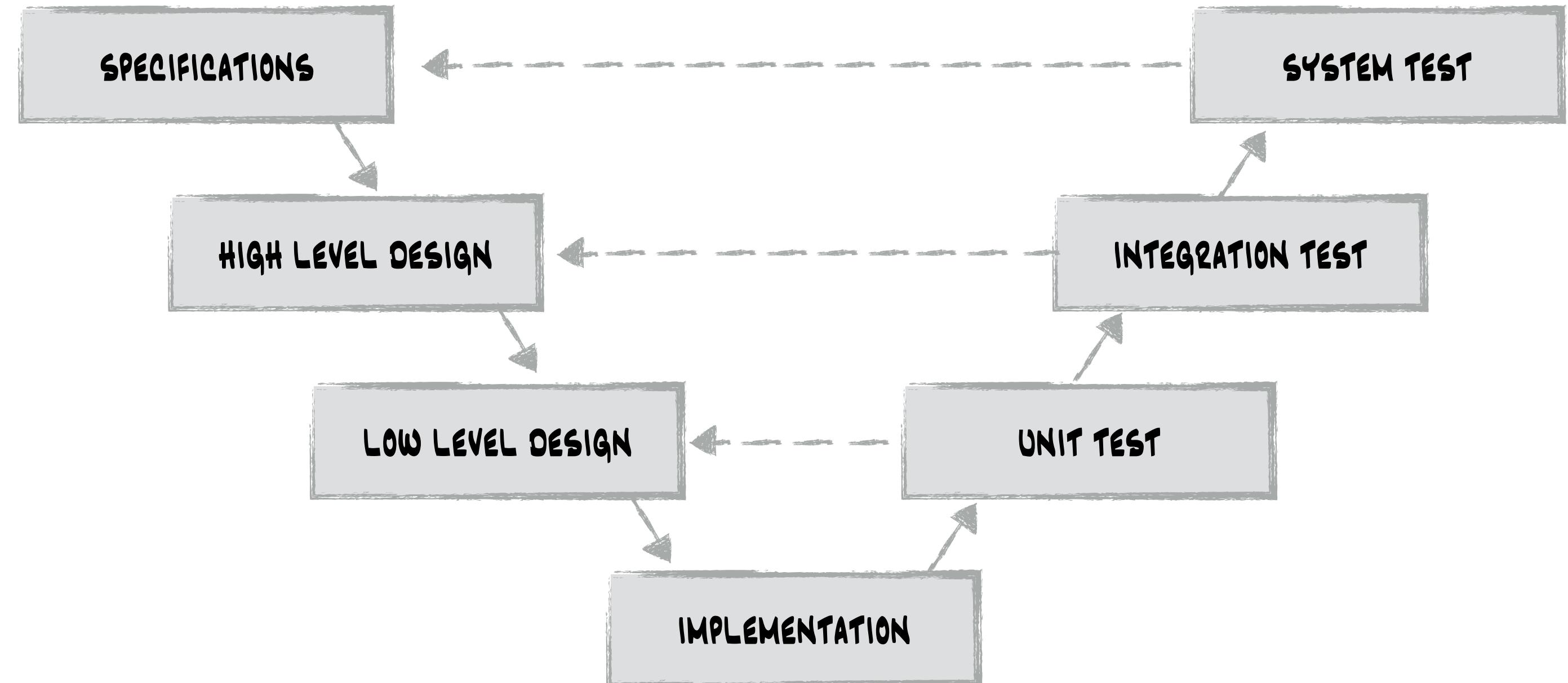


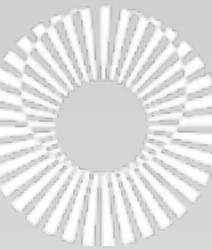
# V-model



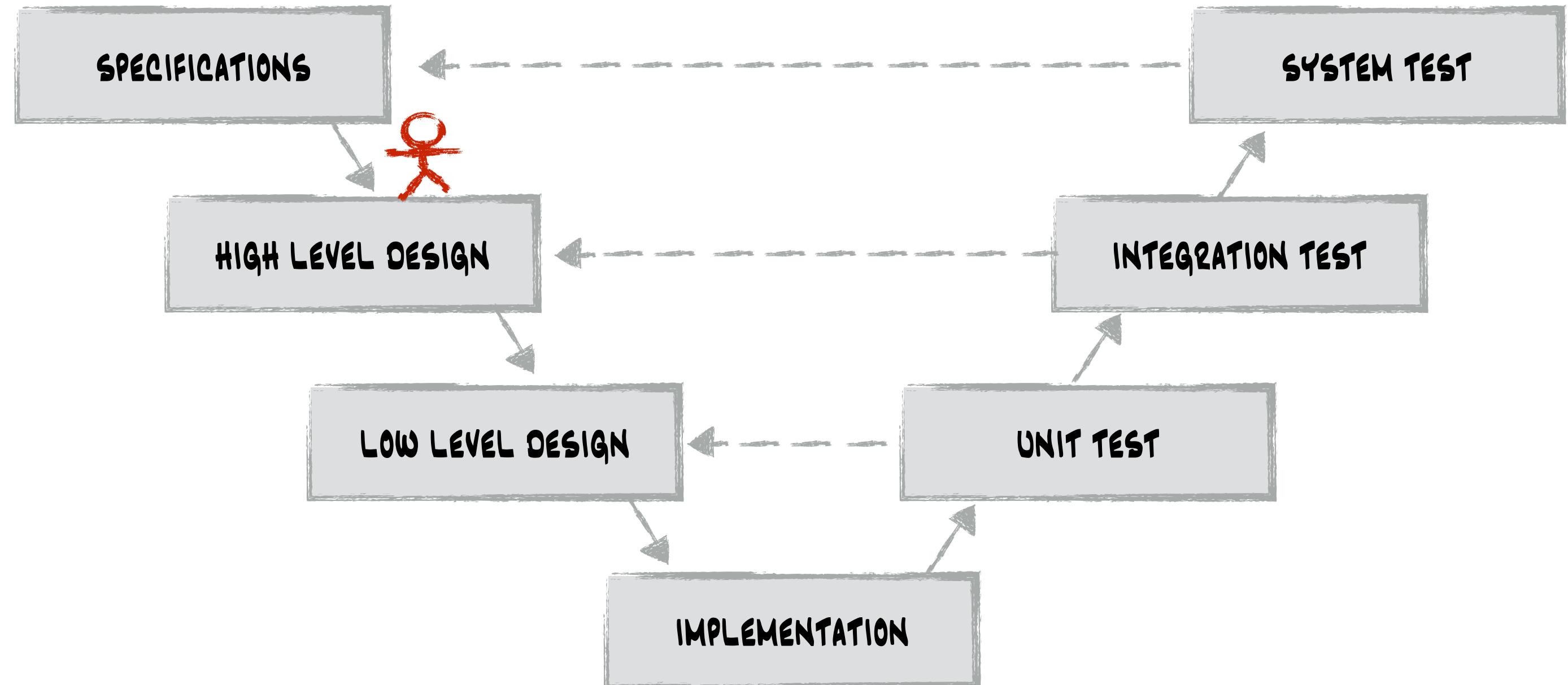


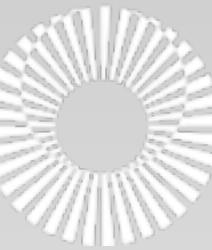
# V-model



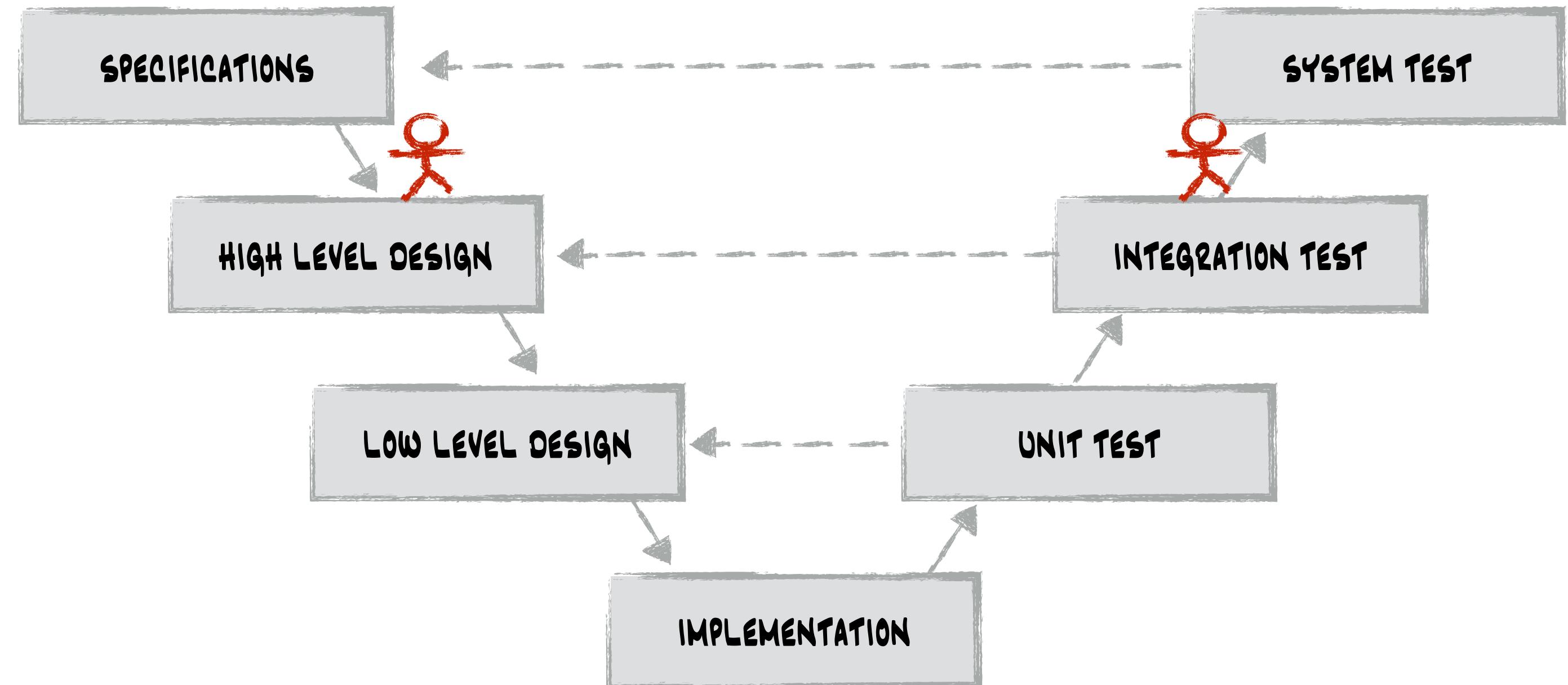


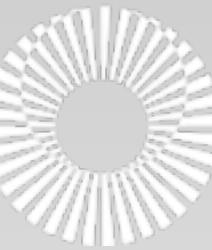
# V-model



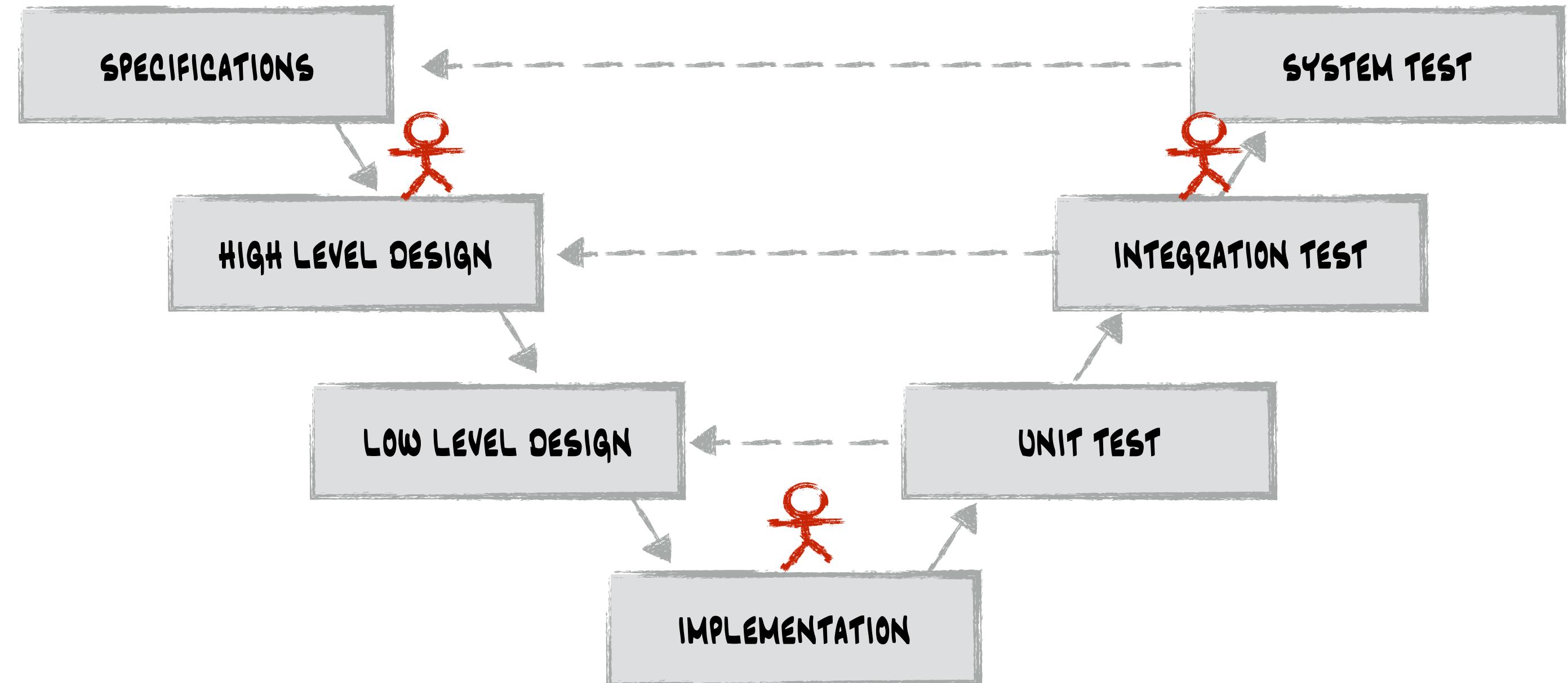


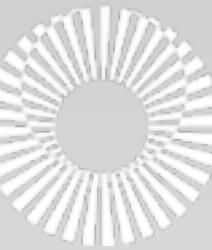
# V-model



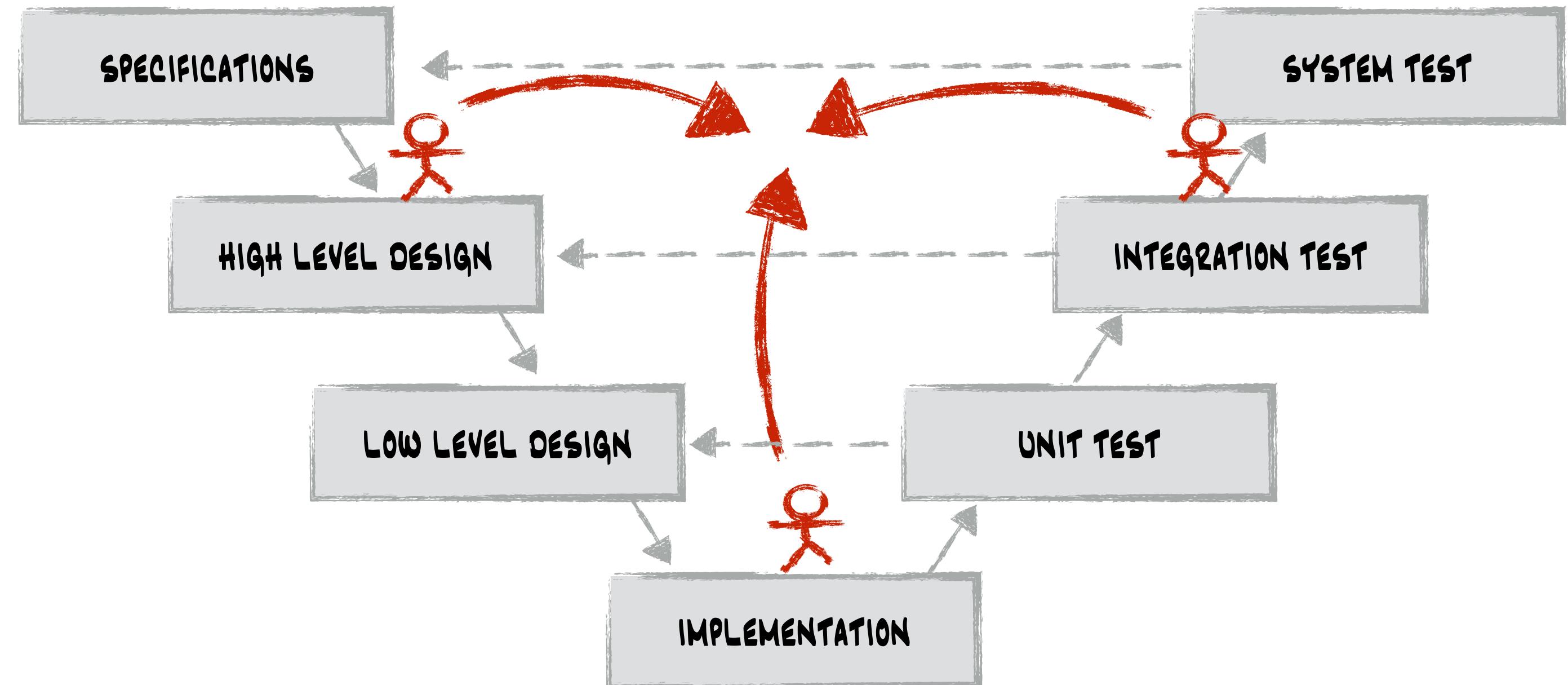


# V-model





# V-model



# The three amigos

## Business analist/Developer/Tester



# Template User Story

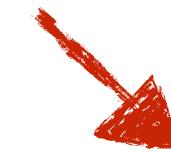
- In order to *<receive benefit>*,
- As a *<role>*,
- I want to *<goal/desire>*.

# Template Specifications

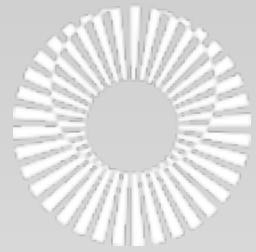
- Given *<some initial context>*,
- When *<an event occurs>*,
- Then *<ensure some outcomes>*.

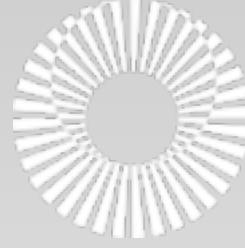
# Template Specifications

- Given *<some initial context>*,
- When *<an event occurs>*,
- Then *<ensure some outcomes>*.



Gherkin language



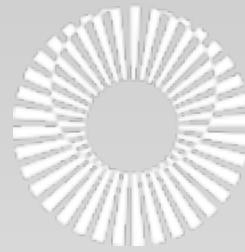


NARRATIVE

In order to track the physical condition of a patient

As a caretaker

I want to register the data of a new physical examination



**NARRATIVE**

In order to track the physical condition of a patient  
As a caretaker  
I want to register the data of a new physical examination

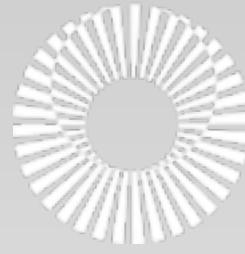
**ACCEPTANCE CRITERIA**

I can register the length, weight and date of the examination  
An error message is given if not all fields are provided  
A patient cannot have more than one examination for a given date  
An error message is given if invalid data is provided:

- Length should be between 50 and 300 cm
- Weight should be between 300 and 700000 gr
- Date should not be in the future



**Acceptance criteria  
for story**



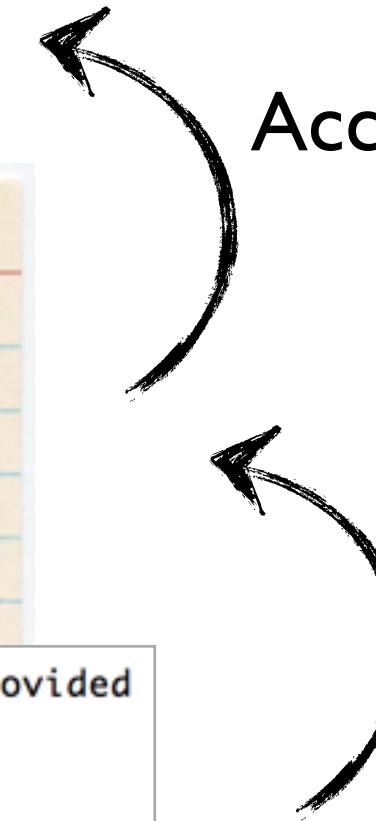
NARRATIVE

In order to track the physical condition of a patient  
As a caretaker  
I want to register the data of a new physical examination

ACCEPTANCE CRITERIA

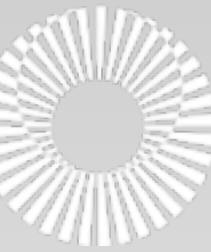
I can register the length, weight and date of the examination  
An error message is given if not all fields are provided  
A patient cannot have more than one examination for a given date  
An error message is given if invalid data is provided:  
• Length should be between 50 and 300 cm

**Scenario:** An error message is given if an invalid length is provided  
**Given** a patient that is registered  
**When** I provide an invalid value -100 for the length  
And choose to save the data  
**Then** an error message "Length should be between 50 and 300" is given  
And the data are not added to the physical history of the patient



Acceptance criteria  
for story

Scenarios for  
acceptance criteria

**NARRATIVE**

In order to track the physical condition of a patient  
As a caretaker  
I want to register the data of a new physical examination

**ACCEPTANCE CRITERIA**

I can register the length, weight and date of the examination  
An error message is given if not all fields are provided  
A patient cannot have more than one examination for a given date  
An error message is given if invalid data is provided:  
• Length should be between 50 and 300 cm

**Scenario:** An error message is given if an invalid length is provided

**Given** a patient that is registered  
**When** I provide an invalid value -100 for the length  
**And** choose to save the data  
**Then** an error message "Length should be between 50 and 300" is given  
**And** the data are not added to the physical history of the patient

**Scenario:** An error message is given if an invalid length is provided

**Given** a patient that is registered  
**When** I provide an invalid value <length> for the length  
**And** choose to save the data  
**Then** an error <message> is given  
**And** the data are not added to the physical history of the

**Examples:**

length message	
49	Length should be at least 50 cm
-180	Length should be at least 50 cm
301	Length should be maximum 300 cm

**Acceptance criteria  
for story**



**Scenarios for  
acceptance criteria**



**Examples for  
scenarios**



# User story: Show patient details

**Narrative:**

In order to check the physical condition of a patient

As a caretaker

I want to consult his/her personal details

# User story: Show patient details

Acceptance criteria:

- The social security number, birthdate, gender, length, weight, examination date and bmi are given
- The physical data of the most recent examination are given
- The bmi is rounded to 2 digits
- An error message is given if the patient cannot be found

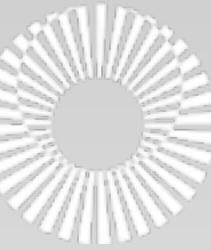
# User story: Show patient details

## Acceptance criteria:

- The social security number, birthdate, gender, length, weight, examination date and bmi are given
- The physical data of the most recent examination are given
- The bmi is rounded to 2 digits
- An error message is given if the patient cannot be found

# 1. Write scenario

## 2. Make scenario executable



# Other tools

## Show patient details

### Narrative

In order to check the physical condition of a patient

As a caretaker

I want to consult his/her personal details

### Acceptance criteria

#### Scenario: the personal details of a registered patient are given

Given a patient with the social security number 93051822361, gender male and birthdate 1993-05-18

And on 2000-04-10 the patient had a length of 180 cm and a weight of 75000 gr

And the patient is registered

When I ask for the details of the patient using his social security number

Then the personal details social security number, gender and birthdate are given

And the examination details length, weight and last examination date are given

And the calculated bmi 23.15 is given

### Mock-up

Person 93051822361	
SOCIAL SECURITY NUMBER	93051822361
BIRTH DATE	1993-05-18
SEX/GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2000-04-10
BMI	23.15
Cancel	

#### Scenario: the physical data of the most recent examination are given

Given a patient with the social security number 93051822361

And on 2000-04-10 the patient had a length of 180 cm and a weight of 75000 gr

And the patient is registered

And on 2000-04-17 the patient had a length of 180 cm and a weight of 80000 gr but these data were added later

When I ask for the details of the patient using his social security number

Then the length 180, weight 80000, and date of the examination date 2000-04-17 are given

And the calculated bmi 24.69 is based on these data

### Scenario Outline: the bmi is rounded to 2 digits

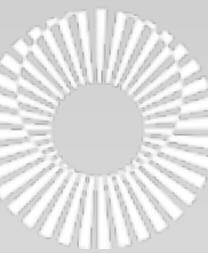
Given a patient that is registered with a length in cm and weight in gr

When I ask for details of the patient

Then the bmi given is rounded to 2 digits

### Examples

length	weight	bmi
160	65000	25.39
160	65001	25.39
160	65009	25.39
180	75000	23.15
180	75000	23.15



# ScalaTest Results

Run completed in 24 seconds, 188 milliseconds.  
Total number of tests run: 9  
Suites: completed 3, aborted 0  
Tests: succeeded 8, failed 1, canceled 0, ignored 0, pending 1

Category	Count	Percentage
Succeeded	8	(80%)
Failed	1	(10%)
Canceled	0	(0%)
Ignored	0	(0%)
Pending	1	(10%)

Succeeded  Failed  Canceled  Ignored  Pending

Suite	Duration (ms.)	Succeeded	Failed	Canceled	Ignored	Pending	Total
PersonServiceJavaApiSpec	282	2	1	0	0	0	3
ShowPatientDetailsIT	23624	2	0	0	0	1	3
ShowPatientDetails	35	4	0	0	0	0	4

## PersonService

Tests: total 3, succeeded 2, failed 1, canceled 0, ignored 0, pending 0

getPerson

- should retrieve a person from the repository if the personId is provided
- should throw an `IllegalArgumentException` if an unknown personId is provided \*\*\* FAILED \*\*\*

[\(Show Details\)](#)

- should throw an `IllegalArgumentException` if no personId is provided

Suite ID org.ucll.demo.service.api

# Selection

## *Criteria*

- Activity:

< 1 year

- Maturity:

$\geq 3$  years



ScalaTest  
specs2  
Cucumber-JVM  
jnario  
specrunner  
selenium  
Concordion  
robot framework  
Fabasoft.app.test  
reFit  
FitNesse  
thucydides  
capybara  
Twist  
Imacros  
JBehave  
spock  
BaDaDam  
EasyB  
GreenPepper  
tumbler-glass  
GivWhenZen  
Jnarrate  
jbee  
ZiBreve/FitLibrary  
JDave  
jaspec  
bdd4j  
Watij/webspec  
narrative  
broccoli-test-engine  
spectacular  
bunit  
bdoc  
beanspec  
instinct  
instinct  
javastubs  
greenbridge  
StoryTestIQ  
javaspec  
easyspec  
rbehave  
gspec  
Fit

# Selection

## *Criteria*

- Activity:  
 $< 1$  year

- Maturity:  
 $\geq 3$  years



- |                          |                |
|--------------------------|----------------|
| <input type="checkbox"/> | Concordion     |
| <input type="checkbox"/> | Cucumber       |
| <input type="checkbox"/> | EasyB          |
| <input type="checkbox"/> | Jbehave        |
| <input type="checkbox"/> | JNario         |
| <input type="checkbox"/> | Fitnessse      |
| <input type="checkbox"/> | RobotFramework |
| <input type="checkbox"/> | Serenity       |
| <input type="checkbox"/> | ScalaTest      |
| <input type="checkbox"/> | Specs2         |
| <input type="checkbox"/> | Spock          |
| <input type="checkbox"/> | Twist          |

ScalaTest  
specs2  
Cucumber-JVM  
jnario  
specrunner  
selenium  
Concordion  
robot framework  
Fabasoft.app.test  
reFit  
FitNesse  
thucydides  
capybara  
Twist  
Imacros  
JBehave  
spock  
BaDaDam  
EasyB  
GreenPepper  
tumbler-glass  
GivWhenZen  
Jnarrate  
jbee  
ZiBreve/FitLibrary  
JDave  
jaspec  
bdd4j  
Watij/webspec  
narrative  
broccoli-test-engine  
spectacular  
bunit  
bdoc  
beanspec  
instinct  
instinct  
javastubs  
greenbridge  
StoryTestIQ  
javaspec  
easyspec  
rbehave  
gspec  
Fit

# Selection

## *Criteria*

- Activity:  
 $< 1$  year

- Maturity:  
 $\geq 3$  years



<input checked="" type="checkbox"/>	Concordion
<input checked="" type="checkbox"/>	Cucumber
<input type="checkbox"/>	<del>EasyB</del>
<input checked="" type="checkbox"/>	JBehave
<input type="checkbox"/>	<del>JNario</del>
<input type="checkbox"/>	<del>Fitnessse</del>
<input checked="" type="checkbox"/>	RobotFramework
<input checked="" type="checkbox"/>	Serenity
<input checked="" type="checkbox"/>	ScalaTest
<input checked="" type="checkbox"/>	Specs2
<input checked="" type="checkbox"/>	Spock
<input type="checkbox"/>	<del>Twist</del>

ScalaTest  
specs2  
Cucumber-JVM  
jnario  
specrunner  
selenium  
Concordion  
robot framework  
Fabasoft.app.test  
reFit  
FitNesse  
thucydides  
capybara  
Twist  
Imacros  
JBehave  
spock  
BaDaDam  
EasyB  
GreenPepper  
tumbler-glass  
GivWhenZen  
Jnarrate  
jbee  
ZiBreve/FitLibrary  
JDave  
jaspec  
bdd4j  
Watij/webspec  
narrative  
broccoli-test-engine  
spectacular  
bunit  
bdoc  
beanspec  
instinct  
instinct  
javastubs  
greenbridge  
StoryTestIQ  
javaspec  
easyspec  
rbehave  
gspec  
Fit

# Selection

## *Criteria*

- Activity:  
 $< 1$  year

- Maturity:  
 $\geq 3$  years



**Gauge ?**  
 **GreenPepper ? ...**

- |                                     |                       |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | <b>Concordion</b>     |
| <input checked="" type="checkbox"/> | <b>Cucumber</b>       |
| <input type="checkbox"/>            | <del>EasyB</del>      |
| <input checked="" type="checkbox"/> | <b>JBehave</b>        |
| <input type="checkbox"/>            | <del>JNario</del>     |
| <input type="checkbox"/>            | <del>Fitnessse</del>  |
| <input checked="" type="checkbox"/> | <b>RobotFramework</b> |
| <input checked="" type="checkbox"/> | <b>Serenity</b>       |
| <input checked="" type="checkbox"/> | <b>ScalaTest</b>      |
| <input checked="" type="checkbox"/> | <b>Specs2</b>         |
| <input checked="" type="checkbox"/> | <b>Spock</b>          |
| <input type="checkbox"/>            | <del>Twist</del>      |

ScalaTest  
specs2  
Cucumber-JVM  
jnario  
specrunner  
selenium  
Concordion  
robot framework  
Fabasoft.app.test  
reFit  
FitNesse  
thucydides  
capybara  
Twist  
Imacros  
JBehave  
spock  
BaDaDam  
EasyB  
GreenPepper  
tumbler-glass  
GivWhenZen  
Jnarrate  
jbee  
ZiBreve/FitLibrary  
JDave  
jaspec  
bdd4j  
Watij/webspec  
narrative  
broccoli-test-engine  
spectacular  
bunit  
bdoc  
beanspec  
instinct  
instinct  
javastubs  
greenbridge  
StoryTestIQ  
javaspec  
easyspec  
rbehave  
gspec  
Fit

# Lab - tools

Serenity	Cucumber Jbehave	Concordion	ScalaTest Specs2 Spock
----------	---------------------	------------	------------------------------

RobotFramework

# Lab - tools

Serenity	Cucumber Jbehave	Concordion	ScalaTest Specs2 Spock
----------	---------------------	------------	------------------------------



Given  
When  
Then

RobotFramework

# Lab - tools

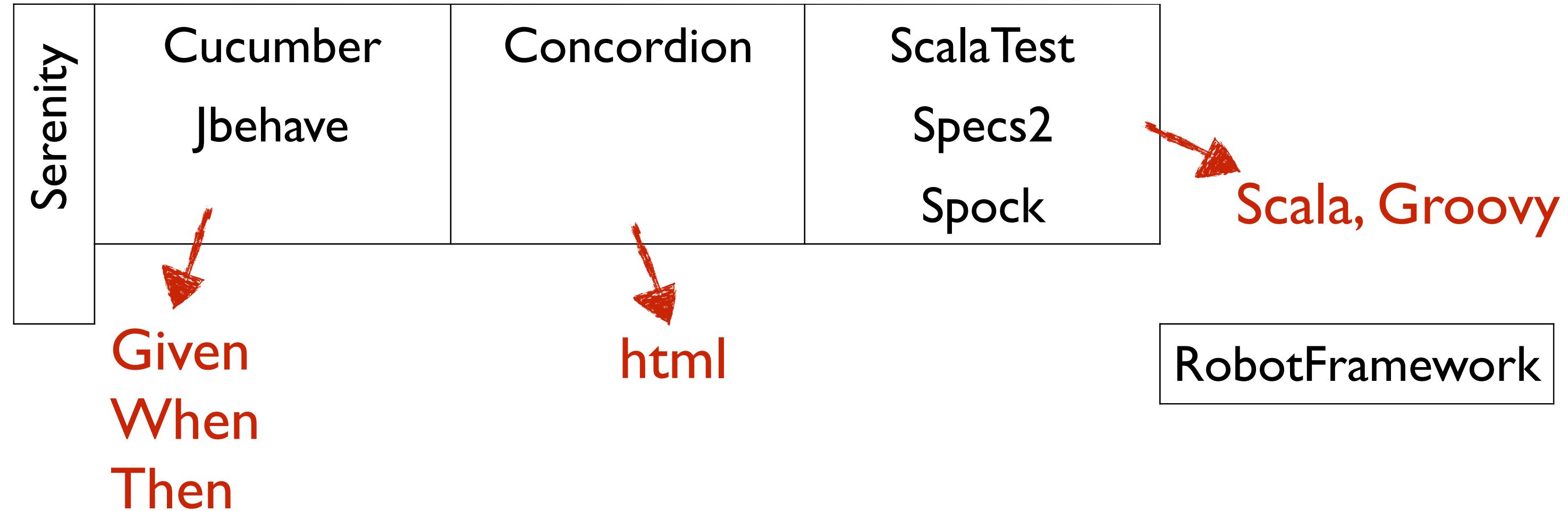
Serenity	Cucumber Jbehave	Concordion	ScalaTest Specs2 Spock
----------	---------------------	------------	------------------------------

Given  
When  
Then

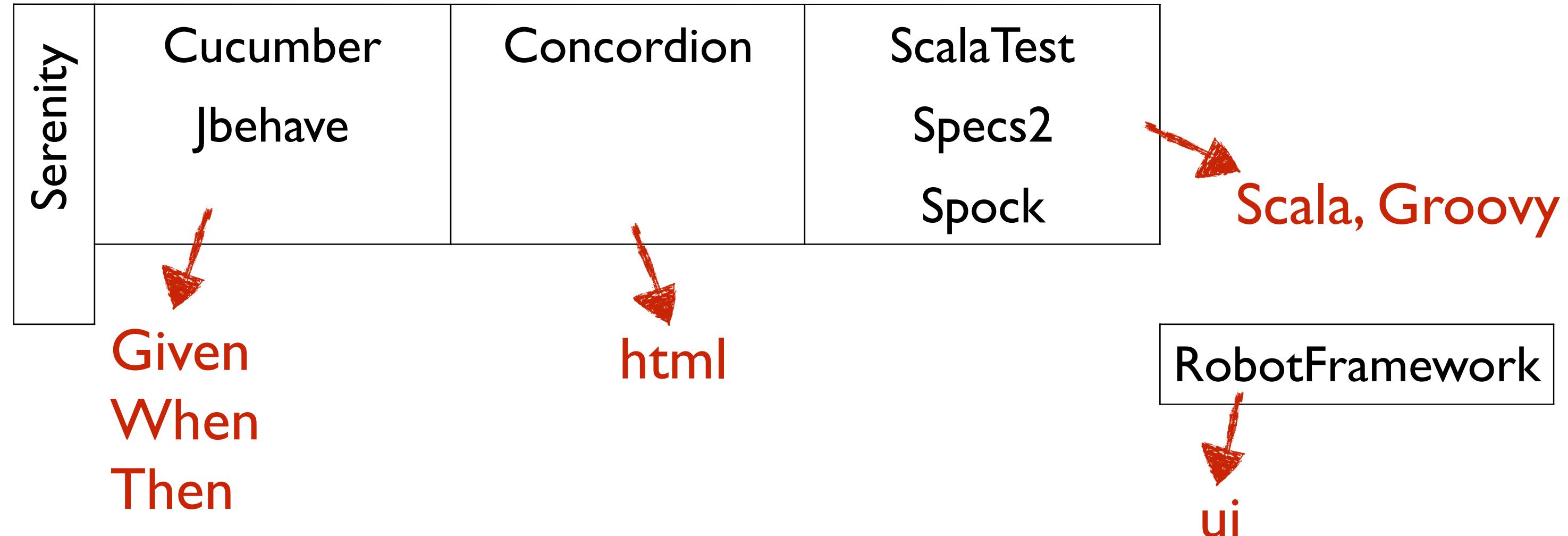
html

RobotFramework

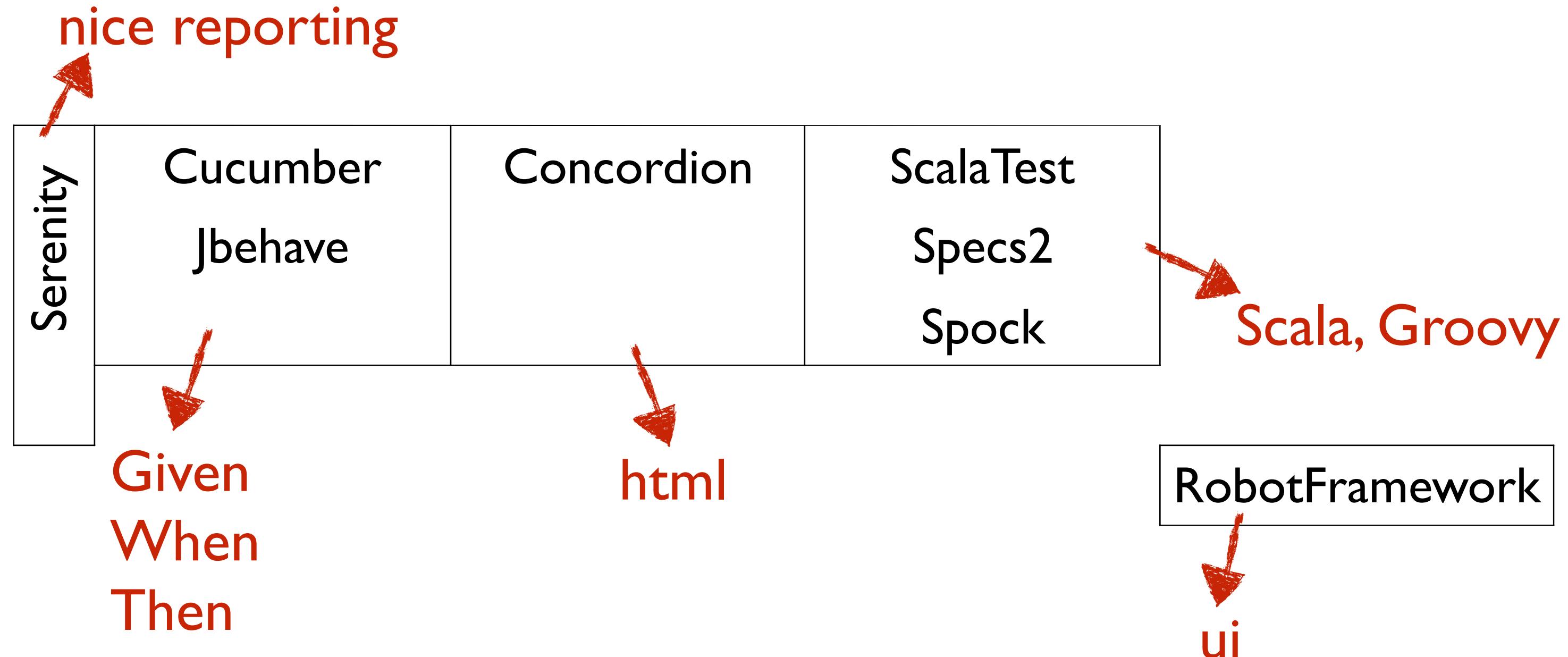
# Lab - tools



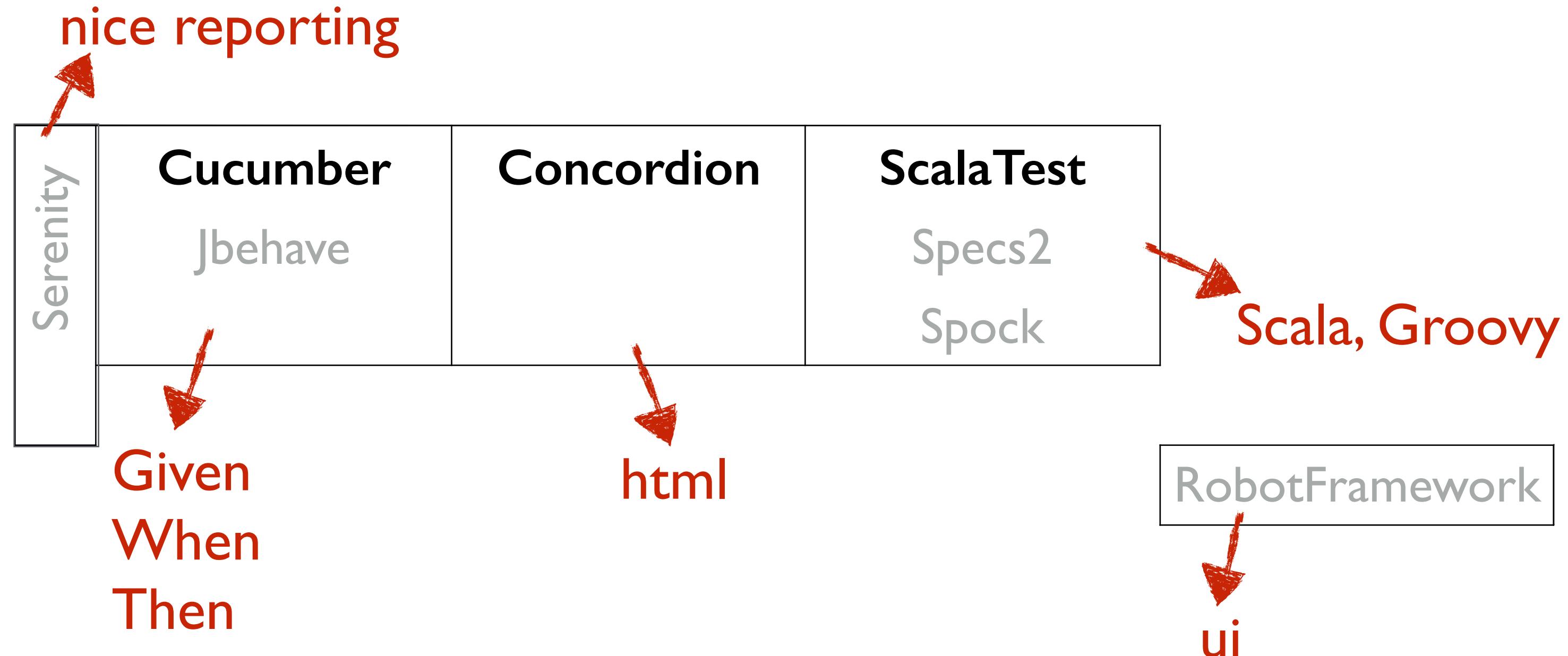
# Lab - tools

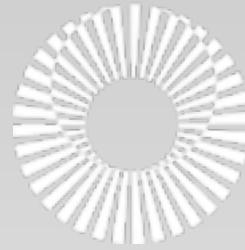


# Lab - tools



# Lab - tools

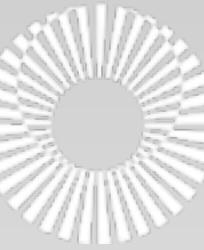




# Today

Install

1



# Today

step-by-step

Show patient detail

PERSON 93051822361

SOCIAL SECURITY NUMBER	93051822361
BIRTH DATE	1993-05-18
GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2015-10-05
SMI	23.15

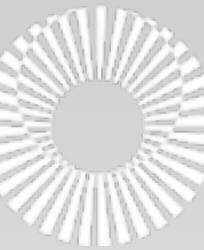
**Add EXAMINATION** **CANCEL**



Install

1

2



# Today

step-by-step

Install

1

Show patient detail

PERSON 93051822361

SOCIAL SECURITY NUMBER	93051822361
BIRTH DATE	1993-05-18
GENDER	MALE
LENGTH (CM)	180
WEIGHT (KG)	75000
EXAMINED ON	2015-10-05
BMI	23.15

**ADD EXAMINATION** **CANCEL**

explore

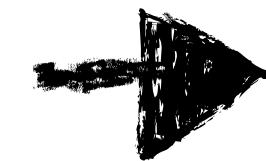
Add examination data

NEW EXAMINATION

LENGTH (CM)	
WEIGHT (KG)	
EXAMINED ON	2015-10-05

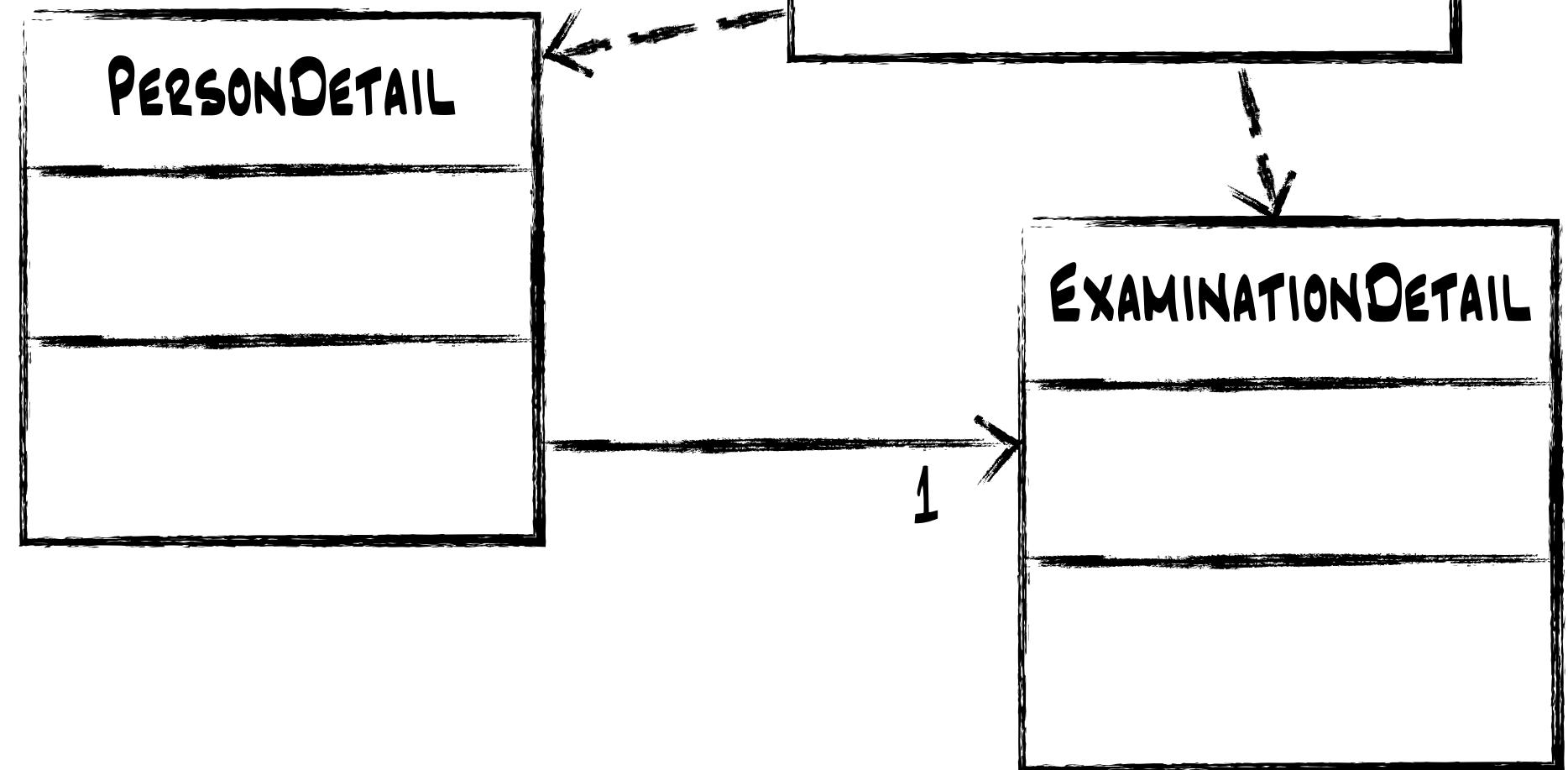
**OK**

# System Under Test



PERSONSERVICEJAVA API

+GETPERSON(STRING):PERSONDETAIL  
+ADDEXAMINATION(EXAMINATIONDETAIL)



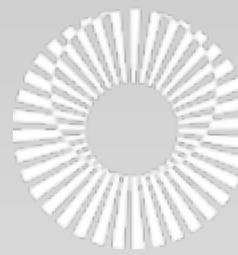
Keep it simple:

- POJOs
- DB = static map!

# Documentation

- <https://github.com/thinfir/think-first>





# Depends...

- Size of project
- Involvement customer
- Skills of the team
- Goal
  - acceptance testing?
  - unit testing?
  - ui driven?
- ...

# Cucumber

Feature: Show patient details

In order to check the physical condition of a patient

As a caretaker

I want to consult his/her personal details

Scenario: the personal details of a registered patient are given

Given a patient with the social security number "93051822361", gender "male" and birthdate "1993-05-18"

And on "2000-04-10" the patient had a length of 180 cm and a weight of 75000 gr

And the patient is registered

When I ask for the details of the patient using his social security number

Then the personal details social security number, gender and birthdate are given

And the examination details length, weight and last examination date are given

And the calculated bmi 23.15 is given

```
@Given("^the patient is registered$")
public void the_patient_is_registered() throws Throwable {
    service.addPerson(patient);
}

@When("^I ask for the details of the patient using his social sec")
public void i_ask_for_the_details_of_the_patient_using_his_social
    detailsRetrieved = service.getPerson(this.socialSecurityNumbe
}
...
...
```

# Cucumber - JBehave

- + no technical knowledge needed to write specifications
- + well documented
- + separation of concerns
- + step generation
  
- overhead
- less suitable for unit testing

# Cucumber vs. JBehave

- It's a neck-and-neck race
- Slightly in favor of Cucumber
  - easy to learn
  - nice reporting

<http://mkolisnyk.blogspot.be/2013/03/jbehave-vs-cucumber-jvm-comparison.html>

# Concordion

## Acceptance criteria

### Scenario: the personal details

```
<div>
```

```
  <p concordion:execute="initializePersonDetails(#socialSecurityNumber, #gender, #birthDate)">
```

```
    Given a patient with the social security number
```

```
      <label concordion:set="#socialSecurityNumber">93051822361</label>,
```

```
      gender <label concordion:set="#gender">male</label> and birthdate
```

```
      <label concordion:set="#birthDate">1993-05-18</label>
```

```
  </p>
```

```
  <p concordion:execute="registerPatient()">
```

```
    And the patient is registered
```

```
  </p>
```

```
  <p concordion:execute="askPatientDetails()">
```

```
    When I ask for the details of the patient using his social security number
```

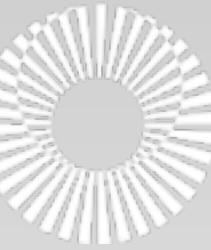
```
  </p>
```

```
  <p concordion:assertTrue="verifyPatientDetails()">
```

```
    Then the personal details social security number, gender and birthdate are given
```

```
  </p>
```

```
public void registerPatient () {  
    service.addPerson(patient);  
}  
  
public void askPatientDetails () {  
    detailsRetrieved = service.getPerson(this.socialSecurityNumber);  
}  
...
```



# Concordion

- + not limited to *Given When Then* template
- + loose coupling between specifications and test code
- + possibility to include other resources next to the scenario's
  
- some knowledge of HTML required to write specifications
- no code generation
- reporting very basic

# ScalaTest - Specs2 - Spock

- + less overhead
- + specific DSL for unit testing
- + more developer centric
  
- more developer centric
- other language (Scala or Groovy)

# ScalaTest

```
scenario("the personal details of a registered patient are given"){

    Given("a patient with the social security number 93051822361, gender male and birthdate 1993-05-18")
    val testData = fixture()

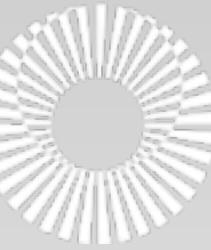
    And("the patient is registered")
    service.addPerson(testData.patient)

    When("I ask for the details of the patient using his social security number")
    val detailsRetrieved = service.getPerson(testData.socialSecurityNumber)

    Then("the personal details social security number, gender and birthdate are given")
    assert(testData.socialSecurityNumber == detailsRetrieved.getSocialSecurityNumber)
    assert(testData.gender == detailsRetrieved.getGender)
    assert(differNoMoreThanFewSeconds(testData.birthDate, detailsRetrieved.getBirthdate))

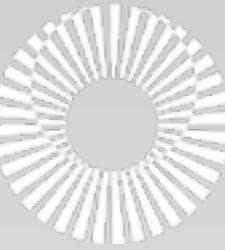
    And("the calculated bmi 23.15 is given")
    assert(23.15 != detailsRetrieved.getBmi);
}

}
```



# ScalaTest vs. Specs2 vs. Spock

- Language
  - Scala: ScalaTest, Specs2
  - Groovy: Spock
- Slightly favor of ScalaTest:
  - easier for Java developers
  - nice reporting



# Serenity

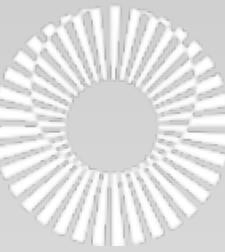
- Former Thucydides
    - + all the advantages
    - + nice reporting
    - + selenium integration
  - Wrapper around
    - JUnit
    - Cucumber
    - JBehave
- more complex

# Robot Framework

- keyword driven
  - + separate file with test cases
  - + plain text
  - focus on ui
  - not integrated in IDEs

# Conclusion

- It is not about the tool ...
- it's about BDD (ATDD, Specification by example, ...)
- ... but a tool can help
  - communication
  - living documentation
- To be continued...



# Special thanks...

to our own testers



ACA IT-Solutions



students UCLL