

Лекция 7. Репортинг. BDD подход.

Тестирование. Практическое применение.

Брекелов Всеволод 11.10.2020

Что такое репорт/отчет?

Почему не сделать один для всех?

Что придумывали для создания отчетов?



Хочу супер
приложение

Заказчик

Представитель компании



Все
протестируали,
вот готово

Отчет для клиентов



Все
протестировали,
вот готово

Отчет для Продукт Оунера / Менеджера



Все
протестировали,
вот готово

Отчет для команды разработки

Отчет для команды интеграции



Отчет для фронтенд команды



Отчет для команды тестирования

**Все хотят видеть разные
отчеты**

Test Report

Test Cycle System Test

EXECUTED	PASSED FAILED <i>(Total) TESTS EXECUTED (PASSED + FAILED)</i>	130 0
PENDING		0
IN PROGRESS		0
BLOCKED		0
<i>(Sub-Total) TEST PLANNED</i> <i>(PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED)</i>		130

Functions	Description	% TCs Executed	% TCs Passed	TCs pending	Priority	Remarks
New Customer	Check new Customer is created	100%	100%	0	High	
Edit Customer	Check Customer can be edited	100%	100%	0	High	
New Account	Check New account is added	100%	100%	0	High	
Edit Account	Check Account is edit	100%	100%	0	High	
Delete Account	Verify Account is delete	100%	100%	0	High	
Delete customer	Verify Customer is Deleted	100%	100%	0	High	
Mini Statement	Verify Ministatement is generated	100%	100%	0	High	
Customized Statement	Check Customized Statement is generated	100%	100%	0	High	

SBTM Report Template v.1.1

ramona.tripa@allom.ro

File Edit View Insert Format Data Tools Help SBTM Report All changes saved in Drive

Comments Share

Last refreshed: Tue May 21 2013 17:50:29 GMT+0300 (EEST).

Last refreshed: Tue May 21 2013 17:50:29 GMT+0300 (EEST).

Exploratory Testing Summary Report

Overall Summary

Total Sessions	4
Total Time (hours)	4.72
Total Bugs	6
Time on Test (hours)	0.76
Time on Setup (hours)	0.23
Time on Bugs (hours)	0.35
Time Off Charter (hours)	0.22
Total Questions	12
Total Next Time Ideas	4

Test/Bugs/Setup/OffCharter

A pie chart titled "Test/Bugs/Setup/OffCharter" showing the distribution of time. The chart is divided into four segments: Green (0.76 hours, 48.7%), Red (0.35 hours, 22.4%), Blue (0.23 hours, 14.7%), and Yellow (0.22 hours, 14.7%).

Category	Time (hours)	Percentage (%)
Time on Test	0.76	48.7%
Time on Bugs	0.35	22.4%
Time on Setup	0.23	14.7%
Time Off Charter	0.22	14.7%

Testing Summary by Area

A bar chart titled "Testing Summary by Area" comparing three categories: HELP, GENERAL, and TIME CALCULATIONS. The Y-axis represents the count of tests or bugs.

Area	Count
HELP	1.5
GENERAL	3.0
TIME CALCULATIONS	0.5

Summary Sample 1 Sample 2 Sample 3 Sample 4

Показывают общее состояние проекта Для любителей олд скула

Отлично Для менеджеров

О, люблю колхоз
Формочки
Excel
Супер!



Руками собирать отчеты в Excel?

Это вы серьезно?



Jira. Zephyr

The screenshot shows the Jira Zephyr interface for a project named "SM Portfolio Project". The top navigation bar includes links for Dashboards, Projects, Issues, Boards, Tests, and a prominent "Create" button, which is highlighted with a red box. The main content area displays a "Cycle Summary" report under the "Tests" section. Key metrics shown include:

- Total Tests: 32
- Total Unexecuted: 14
- Total Executed: 37
- Total Unscheduled: 15

Below these metrics are three tables:

- Tests By Versions:**

Name	Count
Unscheduled	9
Version 1.0	12
Version 2.0	11

Page 1 of 1 | Prev | Next
- Tests By Components:**

Name	Count
No Component	7
Search	14
Engine Rendering	9
Mobile	9
User Accounts	8
Download	2

Page 1 of 1 | Prev | Next
- Tests By Labels:**

Name	Count
No Label	21
Demo	11

At the bottom, there are two summary charts:

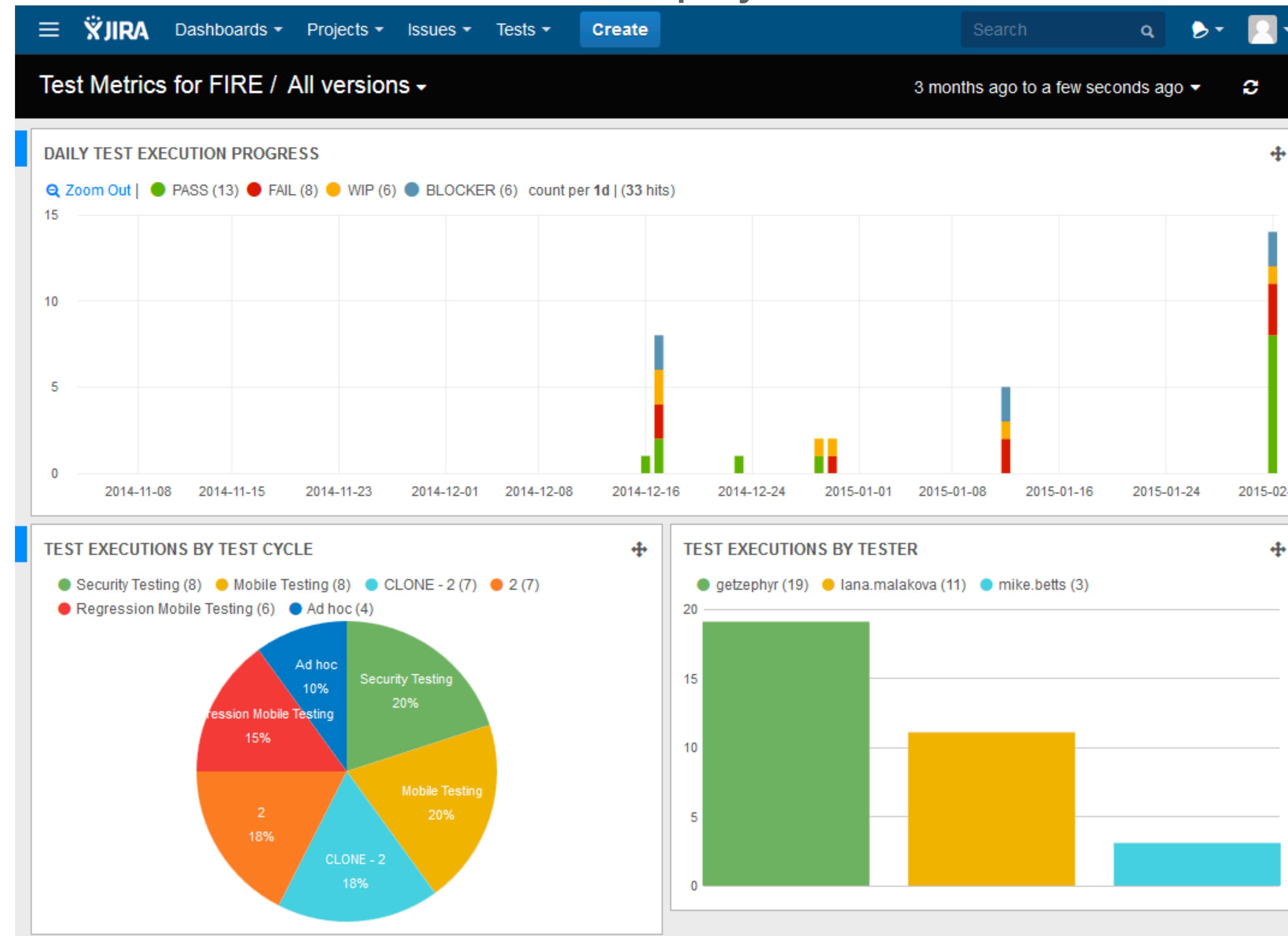
- Test Creation: 30 Day Summary:** A chart showing a value of 4.0.
- Test Execution: 30 Day Summary:** A chart showing a value of 2.0.

On the left sidebar, under the "Tests" section, there is a "Custom Fields" section with a note: "Add a link to useful information for your whole team to see." and a "+ Add link" button.

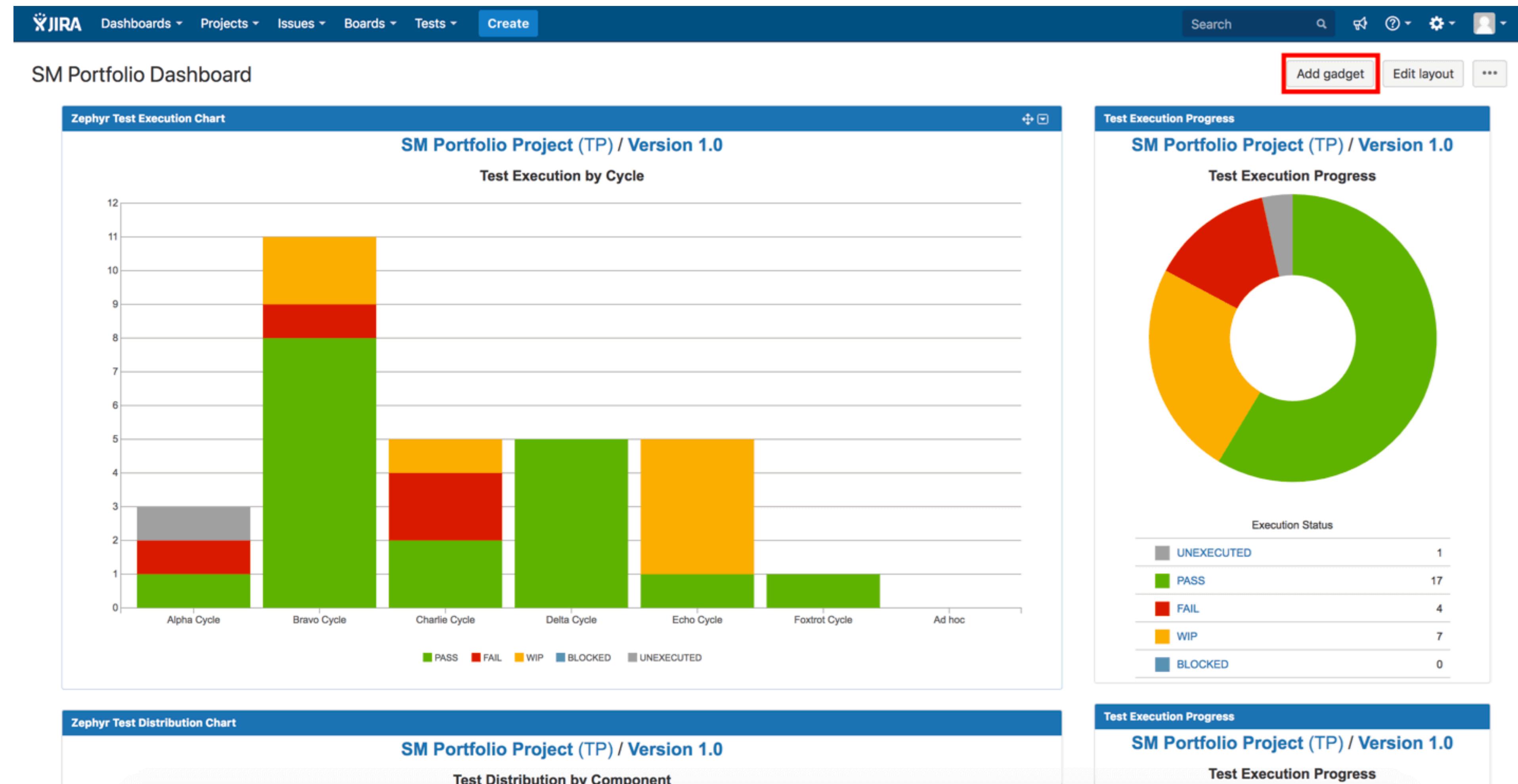
Jira. Zephyr



Jira. Zephyr



Jira. Zephyr



Показывают общее состояние проекта

Отлично Для менеджеров



Тестируем вручную
Смотрим в Jira
Все удобно



Открывать Jira
Искать какие-то отчеты

= (



Junit report

file:///Users/pankaj/Documents/github/journaldev/JUnit5-Example/target/site/surefire-report.html

Project Documentation ▾ Project Information ▾ Project Reports Surefire Report Built by: Maven

Surefire Report

Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
66	0	0	8	87.879%	0.762

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
com.journaldev.annotations	2	0	0	0	100%	0.001
com.journaldev.displayname	3	0	0	0	100%	0.001
com.journaldev.parameterizedtests	33	0	0	0	100%	0.179
com.journaldev	16	0	0	2	87.5%	0.037
com.journaldev.disabletests	12	0	0	6	50%	0.544

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

com.journaldev.annotations

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
MyUtilsTest	2	0	0	0	100%	0.001

com.journaldev.displayname

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
DisplayNameTest	3	0	0	0	100%	0.001

com.journaldev.parameterizedtests

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
ParameterizedTests	33	0	0	0	100%	0.179

com.journaldev

Class	Tests	Errors	Failures	Skipped	Success Rate	Time
JUnit5Sample4Test\$InnerClass	1	0	0	0	100%	0.003
JUnit5Sample4Test\$InnerClass\$InnerMostClass	1	0	0	0	100%	0.002
JUnit5Sample1Test	3	0	0	1	66.667%	0.002
JUnit5Sample3Test	5	0	0	1	80%	0.018
JUnit5Sample2Test	6	0	0	0	100%	0.012

	Code Coverage									
	Classes and Traits		Functions and Methods				Lines			
Total		0.00%	0 / 1	<div style="width: 75%;"><div style="width: 75%;"></div></div>	75.00%	3 / 4	CRAP	<div style="width: 87.5%;"><div style="width: 87.5%;"></div></div>	87.50%	7 / 8
User		0.00%	0 / 1	<div style="width: 75%;"><div style="width: 75%;"></div></div>	75.00%	3 / 4	5.05	<div style="width: 87.5%;"><div style="width: 87.5%;"></div></div>	87.50%	7 / 8
<code>__construct(array \$user)</code>				<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	1 / 1	1	<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	2 / 2
<code>setPassword(\$password)</code>				<div style="width: 0%;"><div style="width: 0%;"></div></div>	0.00%	0 / 1	2.06	<div style="width: 75%;"><div style="width: 75%;"></div></div>	75.00%	3 / 4
<code>getUser()</code>				<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	1 / 1	1	<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	1 / 1
<code>cryptPassword(\$password)</code>				<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	1 / 1	1	<div style="width: 100%;"><div style="width: 100%;"></div></div>	100.00%	1 / 1

```

1 <?php
2
3 namespace phpUnitTutorial;
4
5 class User
6 {
7     const MIN_PASS_LENGTH = 4;
8
9     private $user = array();
10
11    public function __construct(array $user)
12    {
13        $this->user = $user;
14    }
15
16    public function setPassword($password)
17    {
18        if (strlen($password) < self::MIN_PASS_LENGTH) {
19            return false;
20        }
21
22        $this->user['password'] = $this->cryptPassword($password);
23
24        return true;
25    }

```

Jest report

```
2. zsh
[cpojer: ~/Projects/jest/examples/getting_started (master)]$ jest --coverage
PASS ./sum.test.js
  ✓ adds 1 + 2 to equal 3 (4ms)

-----|-----|-----|-----|-----|-----|
File    | %Stmts | %Branch | %Funcs | %Lines | Uncovered Lines |
-----|-----|-----|-----|-----|-----|
All files | 100   | 100   | 100   | 100   | 
sum.js    | 100   | 100   | 100   | 100   | 
-----|-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.133s
Ran all test suites.
[cpojer: ~/Projects/jest/examples/getting_started (master)]$
```

Jest report

```
● ● ● 2. node
PASS packages/jest-matchers/src/_tests_/matchers-test.js
PASS packages/jest-editor-support/src/_tests_/parsers/TypeScriptParser-test.js
PASS packages/jest-cli/src/reporters/_tests_/VerboseReporter-test.js
PASS packages/eslint-plugin-jest/src/rules/_tests_/no-focused-tests-test.js
PASS packages/jest-matchers/src/_tests_/spyMatchers-test.js
PASS packages/jest-runtime/src/_tests_/Resolve-browser-test.js
PASS packages/jest-runtime/src/_tests_/Runtime-internalModule-test.js
PASS packages/jest-snapshot/src/_tests_/utils-test.js
PASS packages/jest-editor-support/src/_tests_/parsers/BabylonParser-test.js
PASS packages/jest-diff/src/_tests_/diff-test.js
PASS packages/jest-cli/src/lib/_tests_/setWatchMode-test.js
PASS packages/pretty-format/src/_tests_/AsymmetricMatcher-test.js

RUNS packages/jest-matcher-utils/src/_tests_/index-test.js
RUNS packages/jest-config/src/_tests_/normalize-test.js
RUNS packages/jest-runtime/src/_tests_/Runtime-genMockFromModule-test.js
RUNS packages/jest-haste-map/src/_tests_/index-test.js
RUNS packages/jest-editor-support/src/_tests_/Runner-test.js
RUNS packages/jest-matchers/src/_tests_/asymmetric-matchers-test.js
RUNS packages/jest-snapshot/src/_tests_/matcher-test.js

Test Suites: 39 passed, 39 of 76 total
Tests:       709 passed, 709 total
Snapshots:   418 passed, 418 total
Time:        14s, estimated 19s

```

**Не всем понятны.
Быстро собираются.
Не содержат лишней информации.**

Отлично QA Automation/Developer





О норм!
Все ясно!

Может быть не очень понятно



Нам ничего не понятно, собирайте Excel отчет

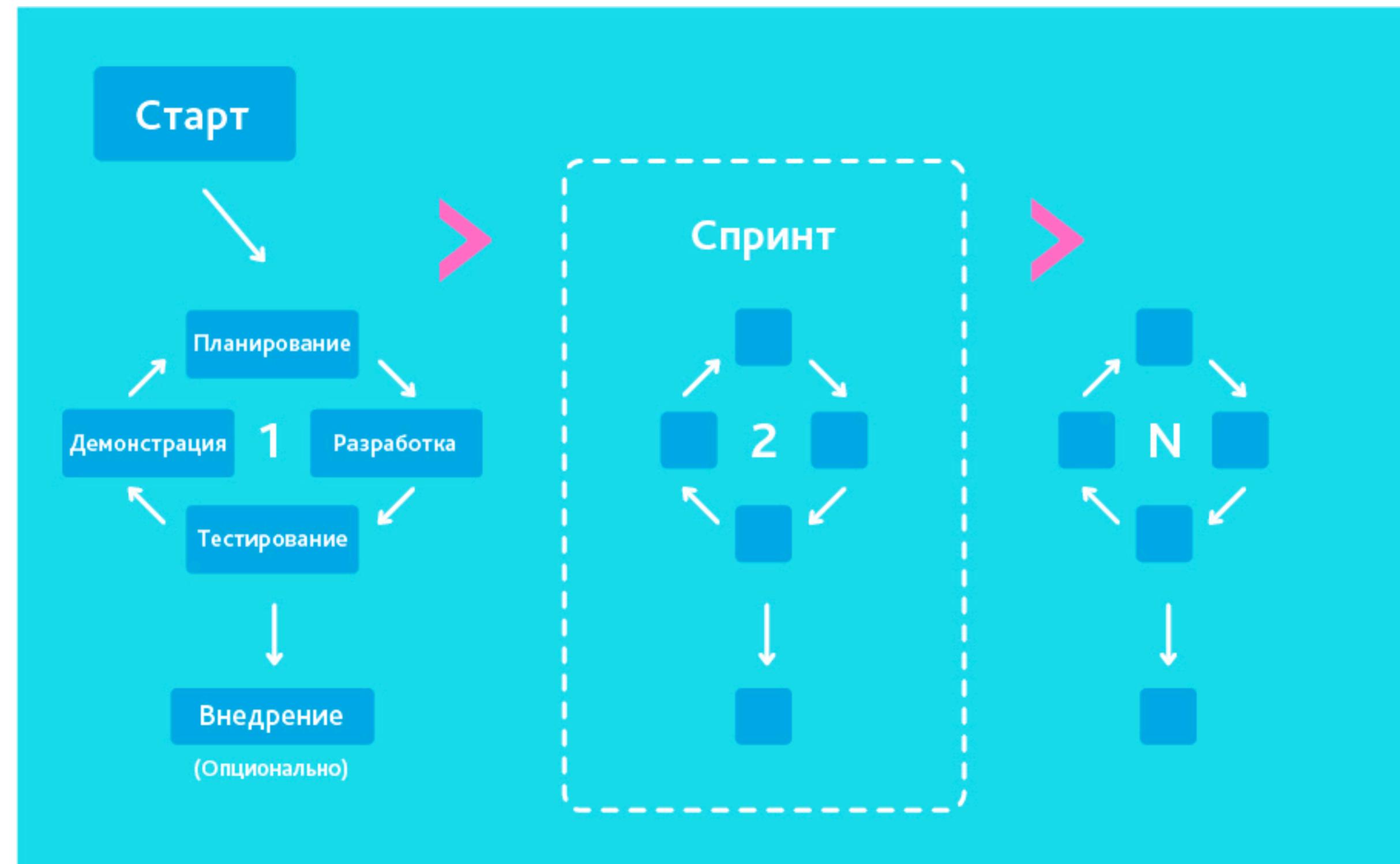


**Низкоуровневый
отчет**

**Высокоуровневый
отчет**

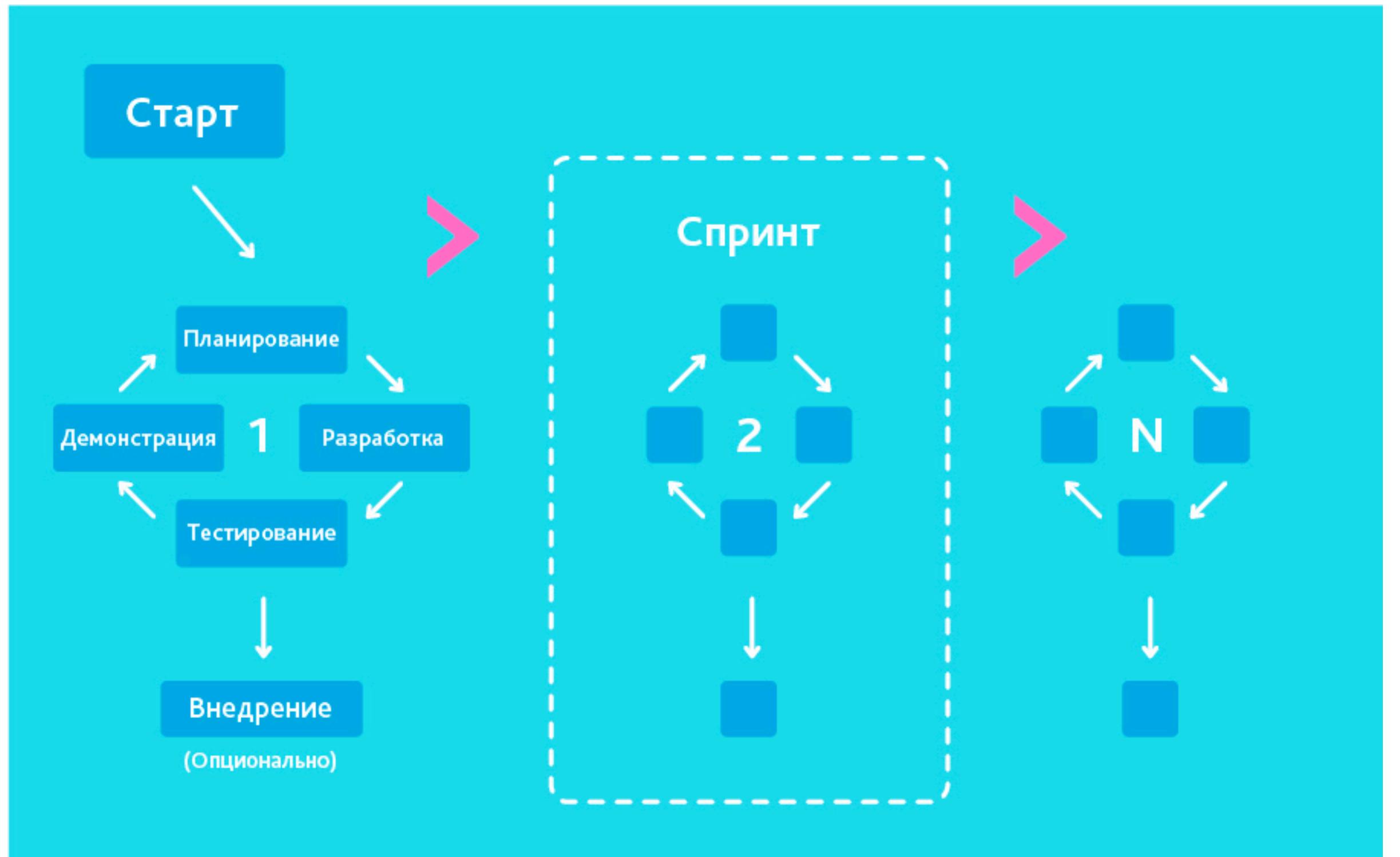
Кастомный отчет

Agile модель



<https://habr.com/ru/company/edison/blog/269789/>

Agile модель



Низкоуровневый
отчет



Высокоуровневый
отчет

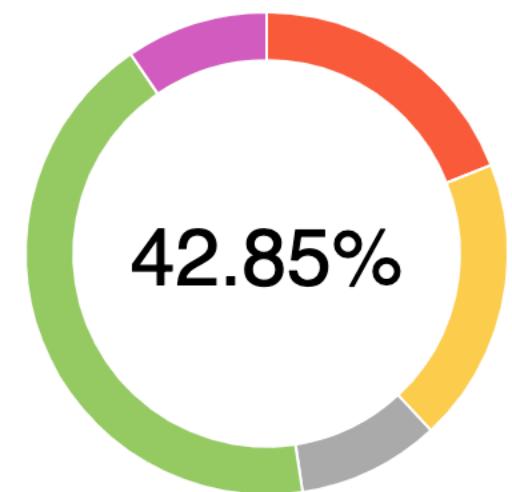


Кастомный отчет

**Автоматизируем все, что
можно**

ALLURE REPORT 10/10/2014
17:50:39 - 18:05:53 (15m 13s)

21
test cases



SUITES 9 items total

my.company.ManyInfoTest	2	1
my.company.SimpleTest	1	1
Search tests	1	3
my.company.NonAsciiTest	1	1
my.company.BeforeClassFailTest	1	
my.company.AlwaysPassingTest	2	2
Custom Widget page	1	
other.PassingTest	1	
my.company.AllureMigrationTest	1	

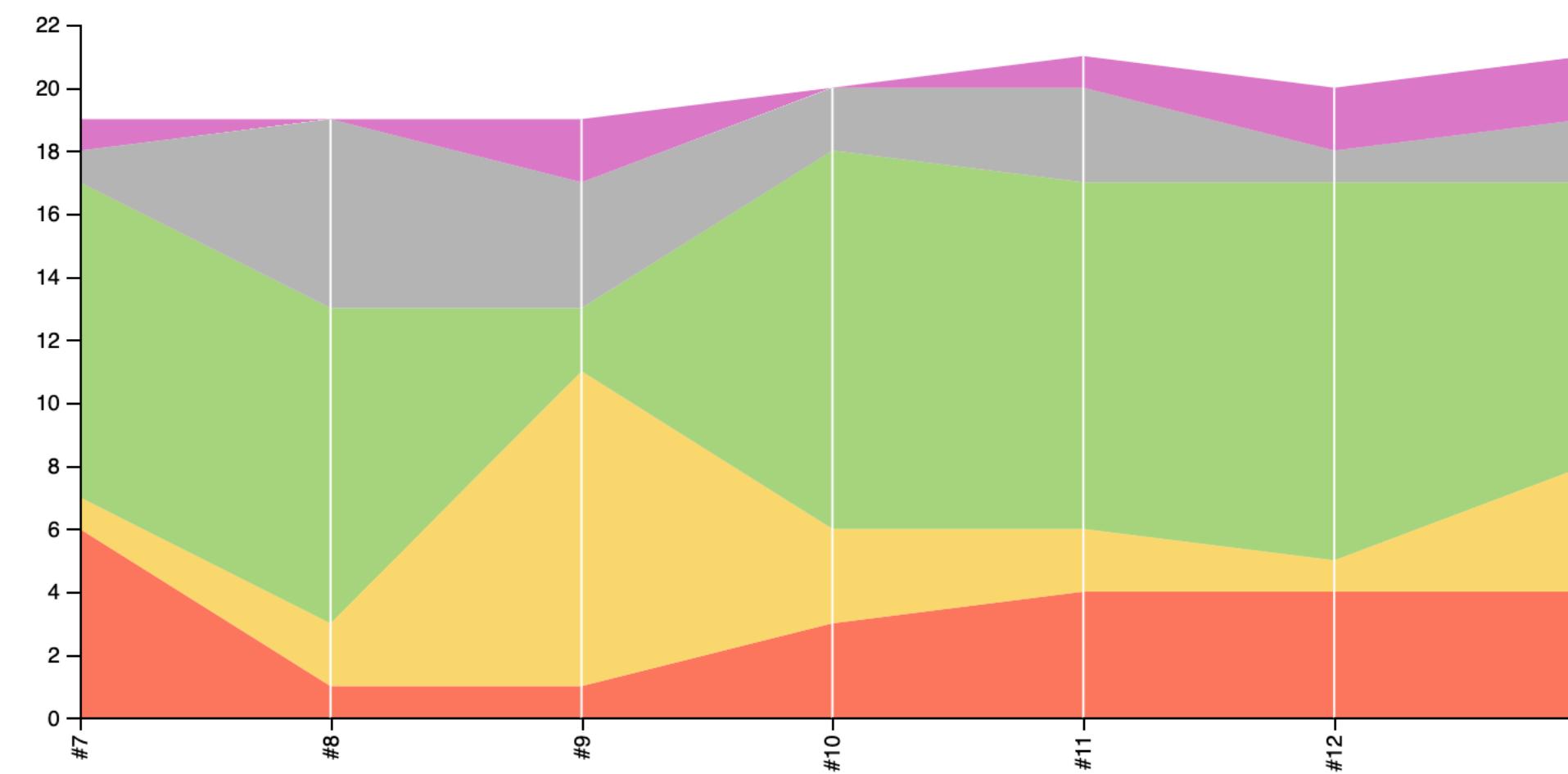
Show all

ENVIRONMENT

PATH

/Users/charlie/.rvm/gems/ruby-2.0.0-
p481/bin:/Users/charlie/.rvm/gems/ruby-2.0.0-
p481@global/bin:/Users/charlie/.rvm/rubies/ruby-2.0.0-
p481/bin:/opt/local/bin:/opt/local/sbin:/usr/bin:/bin:/usr/sbin:
/sbin:/usr/local/bin:/usr/local/git/bin:/usr/local/MacGPG2/bi
n:/usr/texbin:/usr/local/Cellar/maven/3.0.5/libexec/bin:/Libr

TREND



CATEGORIES 5 items total

Product defects	4
Test defects	2
Infrastructure problems	1
Outdated tests	1
Ignored tests	1

Show all

EXECUTORS



allure-report_deploy#13 ↗

<https://github.com/allure-examples/allure-examples>

 .github/workflows

 allure-cucumber-jvm

 allure-java-commons

 allure-javascript

 allure-junit4

 allure-junit5

 allure-kotlin-junit5

 allure-python

 allure-testng

 allure-typescript

Удобно, что интегрируется с Jira
Можно увидеть много деталей



Все понятно
Находится на Jenkins
Можно собрать локально



Красиво, но можем не понять как заходить на Jenkins



Что придумывали для создания отчетов?

И не только отчетов

BDD (Behavior Driven Development) — Разработка на основе поведения.

<https://habr.com/ru/post/139674/>

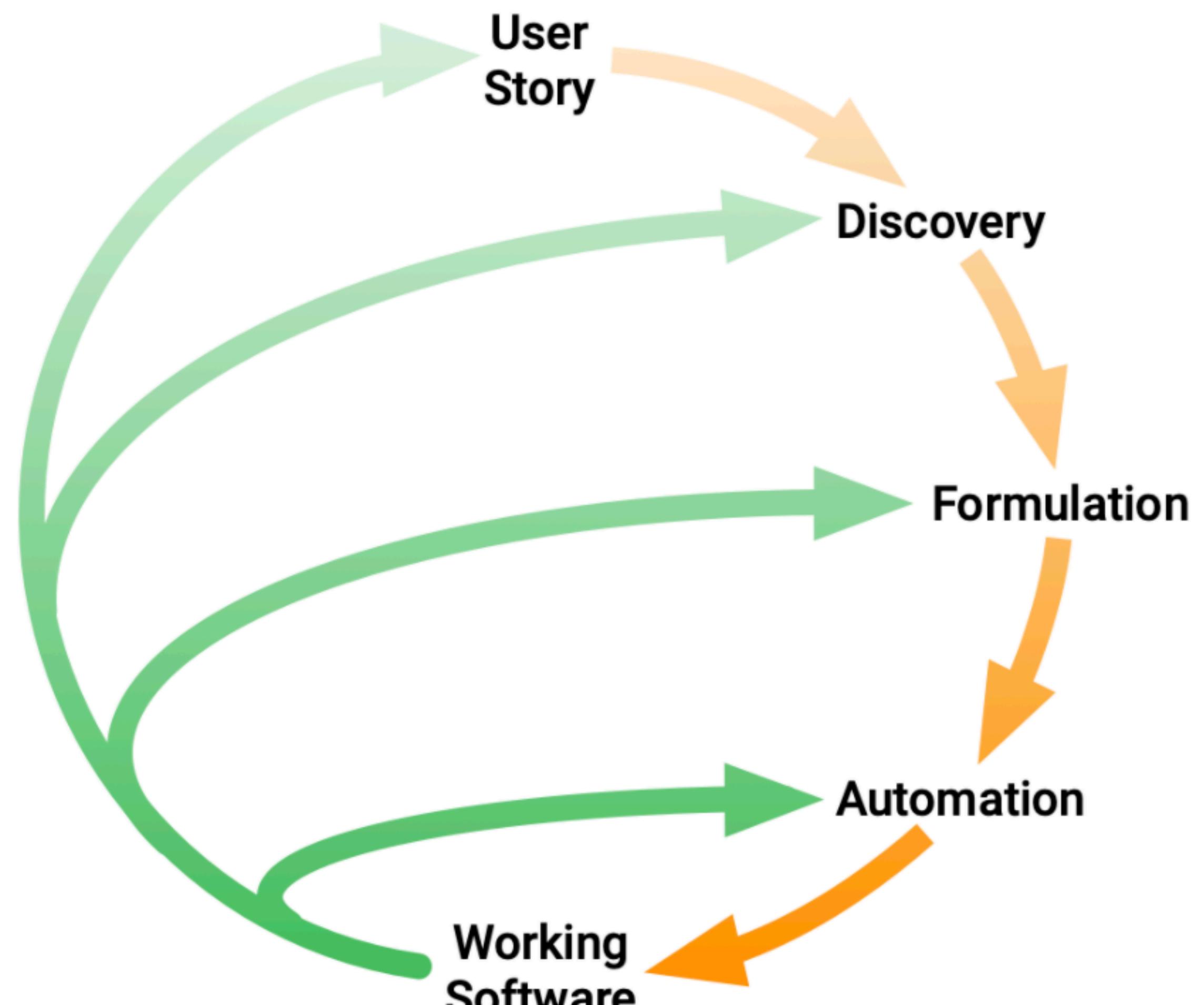
[Link to article](#)

Кто использовал TDD?

BDD и Gherkin

<https://habr.com/ru/post/275013/>

<https://cucumber.io/docs/gherkin/reference/>



Discovery, Formulation and Automation

Scenario Outline: eating

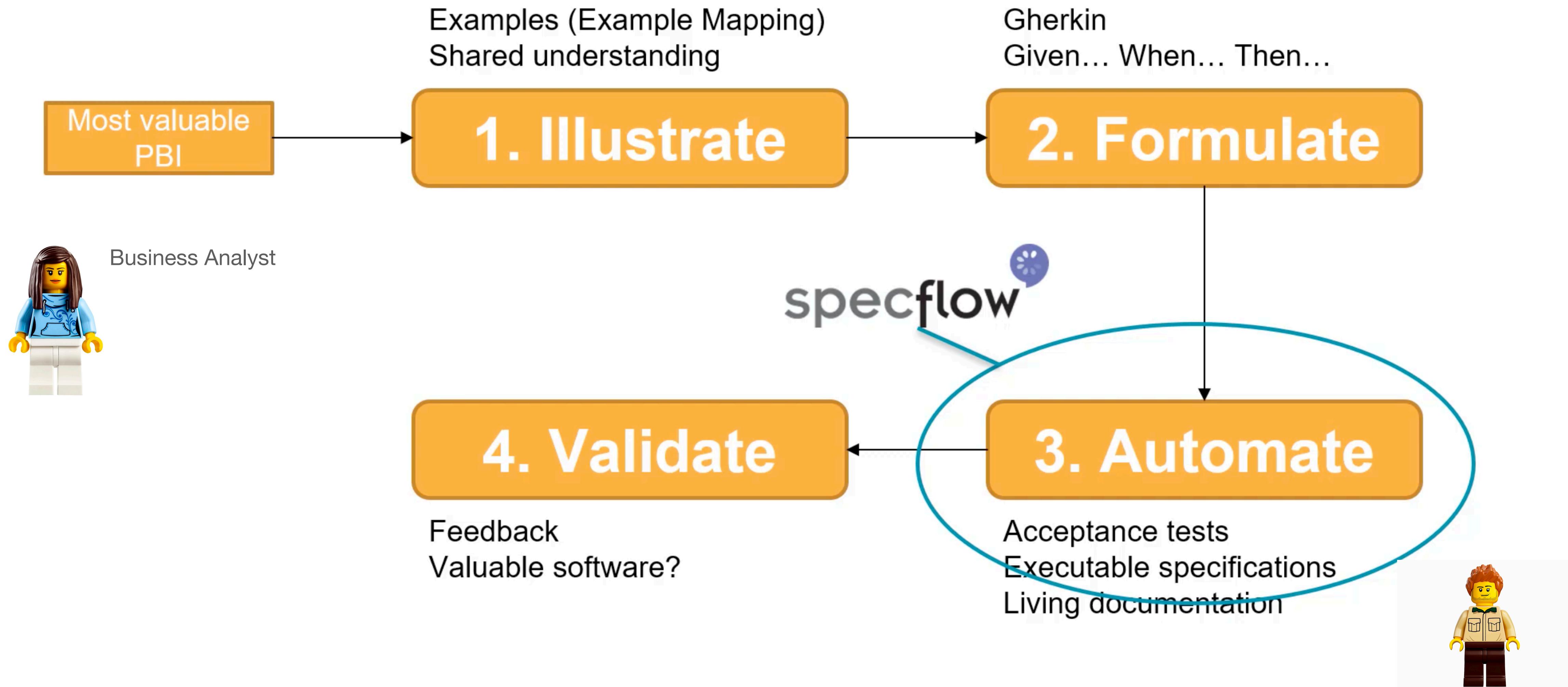
Given there are <start> cucumbers

When I eat <eat> cucumbers

Then I should have <left> cucumbers

Examples:

	start	eat	left
	12	5	7
	20	5	15



Feature:

In order to avoid silly mistakes
As a math idiot
I want to be told the sum of two numbers

Scenario: Add two numbers

Given I have entered 50 into the calculator
And I have entered 70 into the calculator
When I press add
Then the result should be 120 on the screen

```
...
[Given(@"I have entered (.*) into the calculator")]
public void GivenIHaveEnteredIntoTheCalculator(int p0)
{
}

[When(@"I press add")
public void WhenIPressAdd()
{
}

[Then(@"the result should be (.*) on the screen")]
public void ThenTheResultShouldBeOnTheScreen(int p0)
{
}

...
```

1. Write story

Plain
text

Scenario: A trader is alerted of status

Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status should be OFF
Writing Stories
When stock is traded at 16.0
Then the alert status should be ON

2. Map steps to Java

POJO

```
public class TraderSteps {  
    private TradingService service; // Injected  
    private Stock stock; // Created  
  
    @Given("a stock and a threshold of $threshold")  
    public void aStock(double threshold) {  
        stock = service.newStock("STK", threshold);  
    }  
    @When("the stock is traded at price $price")  
    public void theStockIsTraded(double price) {  
        stock.tradeAt(price);  
    }  
    @Then("the alert status is $status")  
    public void theAlertStatusIs(String status) {  
        assertThat(stock.getStatus().name(), equalTo(status));  
    }  
}
```

3. Configure Stories

Only
once

```
public class TraderStories extends JUnitStories {  
  
    public Configuration configuration() {  
        return new MostUsefulConfiguration()  
            .useStoryLoader(new LoadFromClasspath(this.getClass()))  
            .useStoryReporterBuilder(new StoryReporterBuilder()  
                .withCodeLocation(codeLocationFromClass(this.getClass()))  
                .withFormats(CONSOLE, TXT, HTML, XML));  
    }  
  
    public List<CandidateSteps> candidateSteps() {  
        return new InstanceStepsFactory(configuration(),  
            new TraderSteps(new TradingService())).createCandidateSteps();  
    }  
  
    protected List<String> storyPaths() {  
        return new StoryFinder().findPaths(codeLocationFromClass(this.getClass()),  
            "**/*.story");  
    }  
}
```

4. Run Stories

With
any of



<https://cucumber.io/>

<https://jbehave.org/>

<https://mochajs.org/>

<https://www.chaijs.com/api/bdd/>

Но так ли все хорошо с BDD фреймворками?

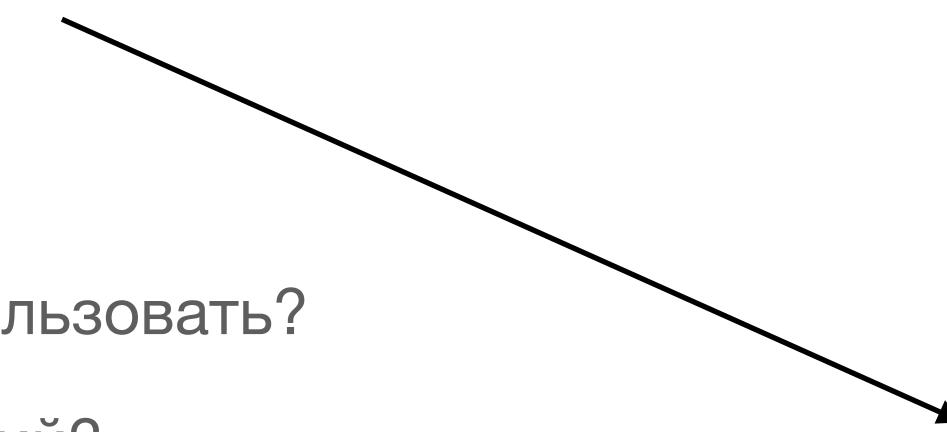
Создание пользовательских историй



А где создает?

А какие слова(ключевые) может использовать?

А использует систему контроля версий?



Написание новых обвязок



Объясняет аналитику что нельзя писать

Объясняет аналитику где допустил ошибку

Пишет сценарии за аналитика



Там много букв, очень тяжело понимать

**Стиль BDD ок
Фреймворки – под вопросом**

**Автоматизированные отчеты вида
Allure подходят и разработчикам и
менеджерам и тестировщикам**