2012 CEE-SEC(R)

Software Engineering Conference in Russia

On Requirements for Acceptance Testing Automation Tools in Behavior Driven Software Development

Evgeny Pyshkin

Maxim Mozgovoy

Mikhail Glukhikh

St. Petersburg Polytechnical University
The University of Aizu
Joint Group in Information Retrieval & Software Research

2012 CEE-SEC(R) Разработка ПО

О требованиях к средствам автоматизации приемочных тестов при использовании подхода «разработка, управляемая описанием поведения»

Евгений Пышкин

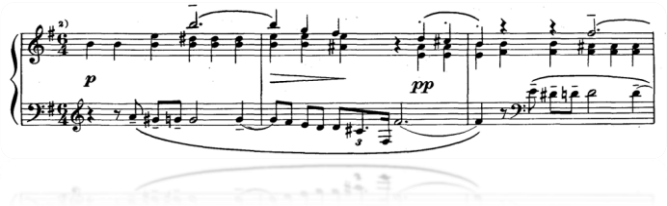
Максим Мозговой

Михаил Глухих

Санкт-Петербургский Государственный политехнический университет
Университет Айзу

Совместная научно-исследовательская группа «Информационный поиск и программное обеспечение»

Classified

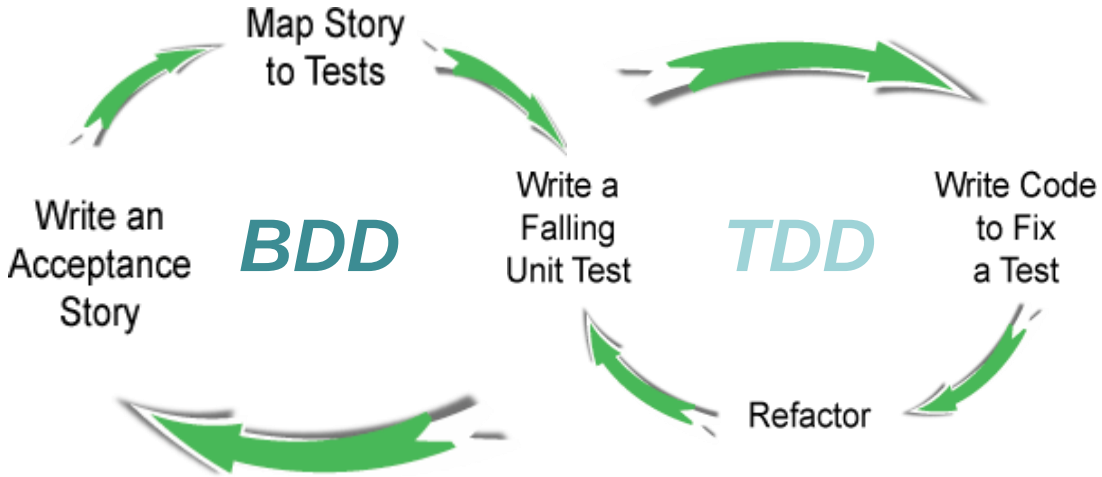


- ***Joint Group of Information Retrieval and Software Research***
  - Problems of higher education in field of programming and software engineering
  - Software development and testing
  - Information retrieval, especially MIR
  - Natural language processing and virtual learning labs
  - Software reliability…

http://kspt.ftk.spbstu.ru/research/science/jgirsr/en/

КСПТ | КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ ФТК СПбГПУ

Why did we write today's paper

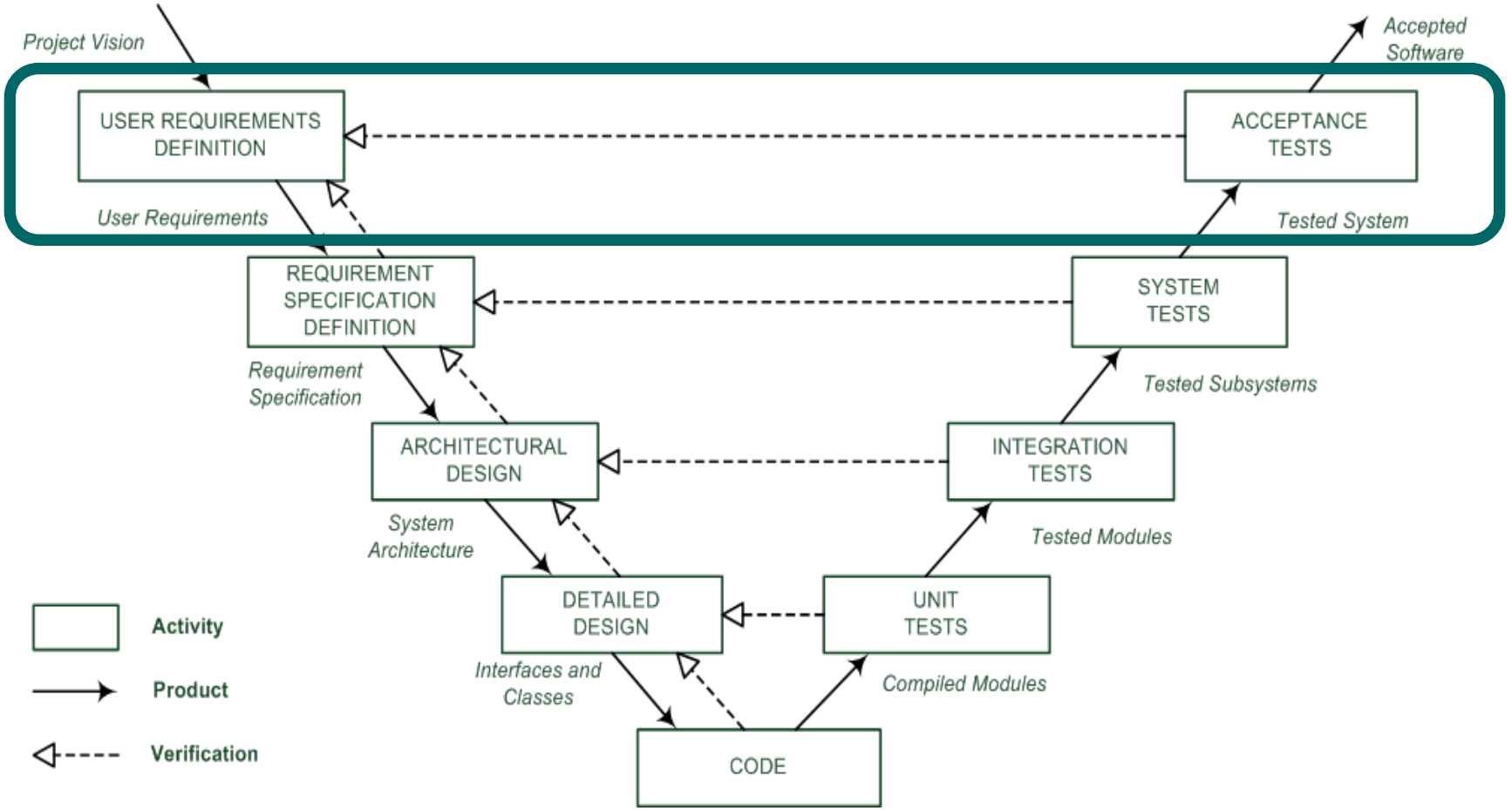


- Know more about BDD solutions
- Try to use
- Learning from practice:
  - Testing technique vs. development practice
- Maturing approach
  - Debates & definitions

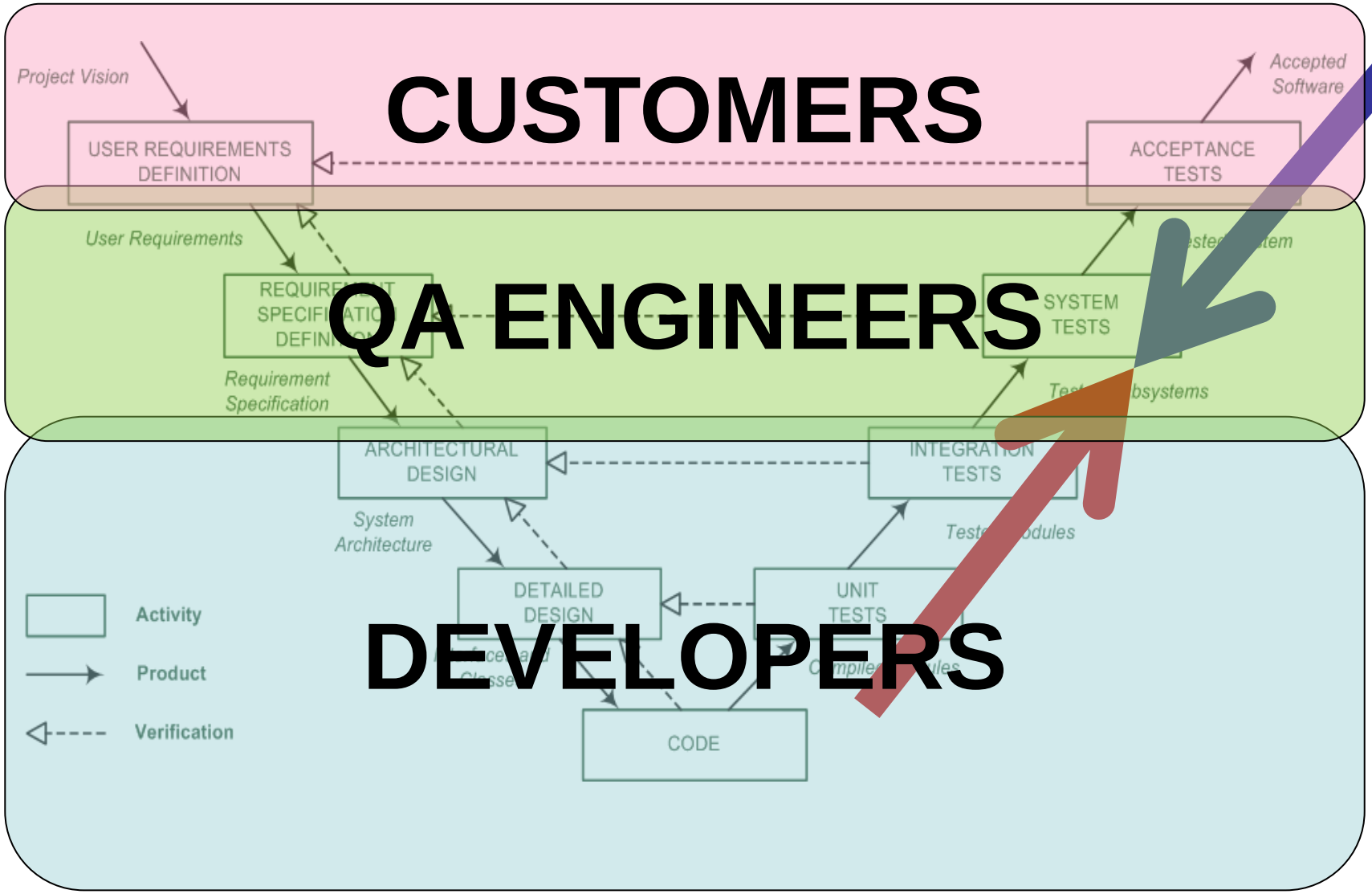
Acceptance Testing Automation in BDD

- What are acceptance tests?
- What are main difficulties about acceptance tests?
- What's the BDD?
- How it works?
- What did we do?

What are acceptance tests?

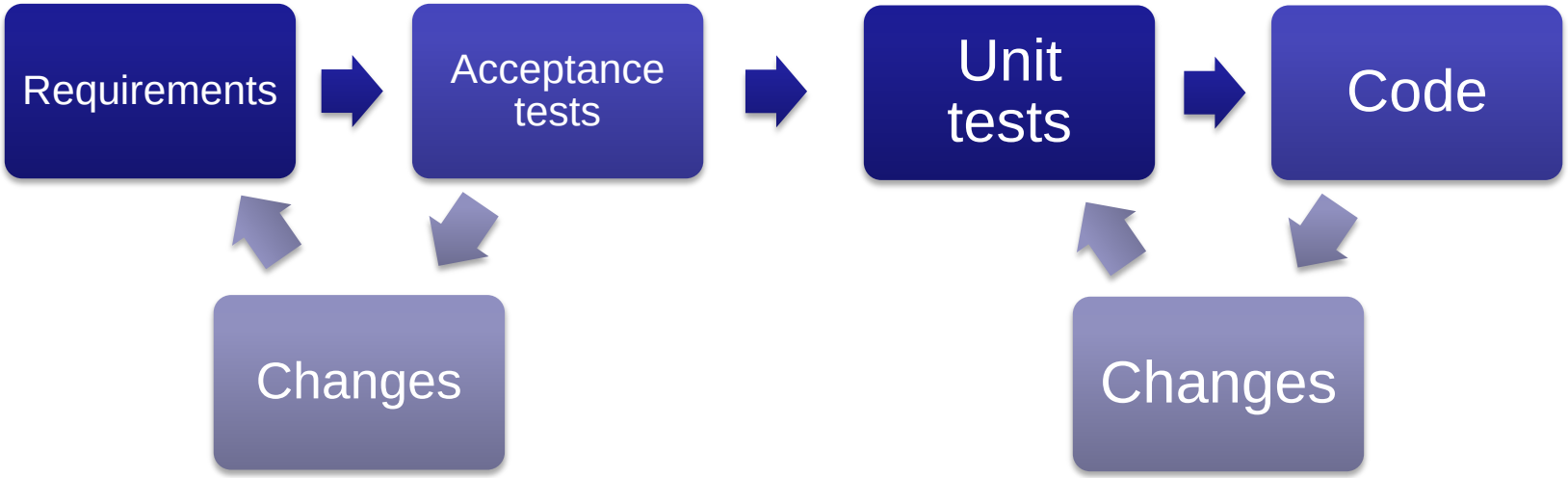


The main idea of BDD



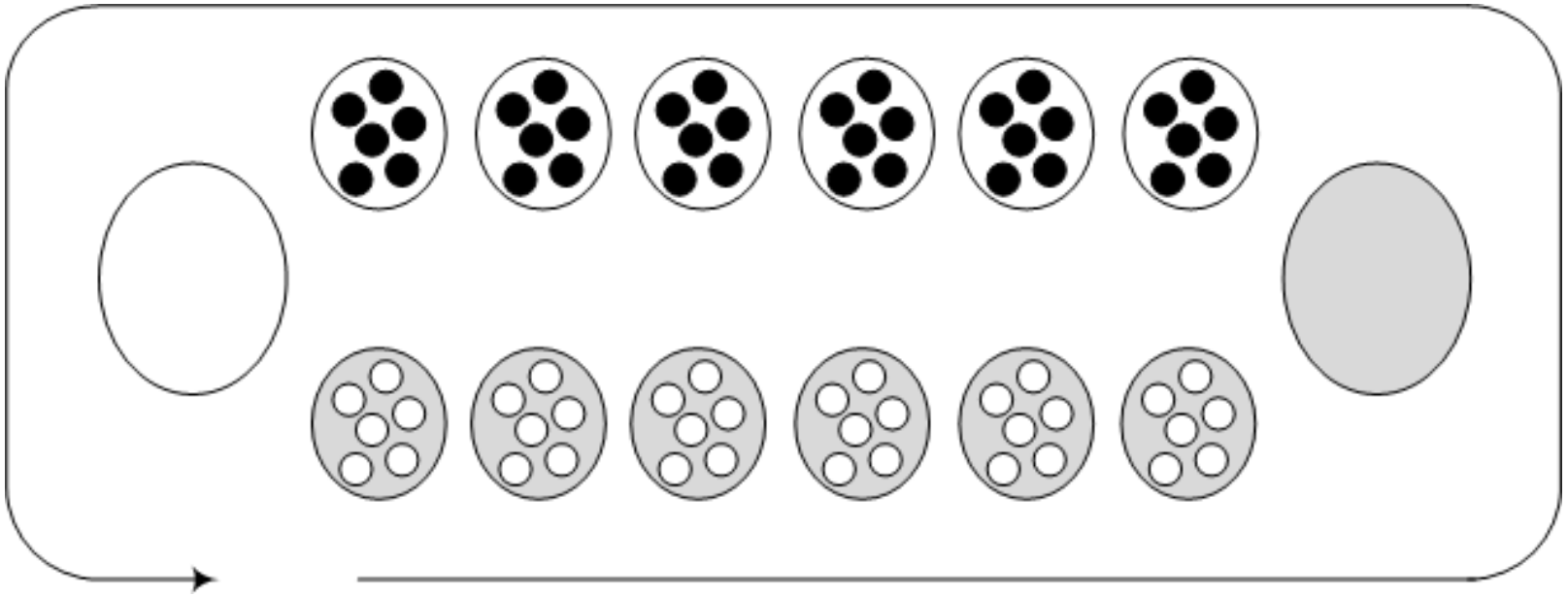
What are main difficulties?

- To run tests we have to run code
- To test code we have to know the code
- Customers don’t know anything about the code
- But they want to be sure that the program fits the requirements



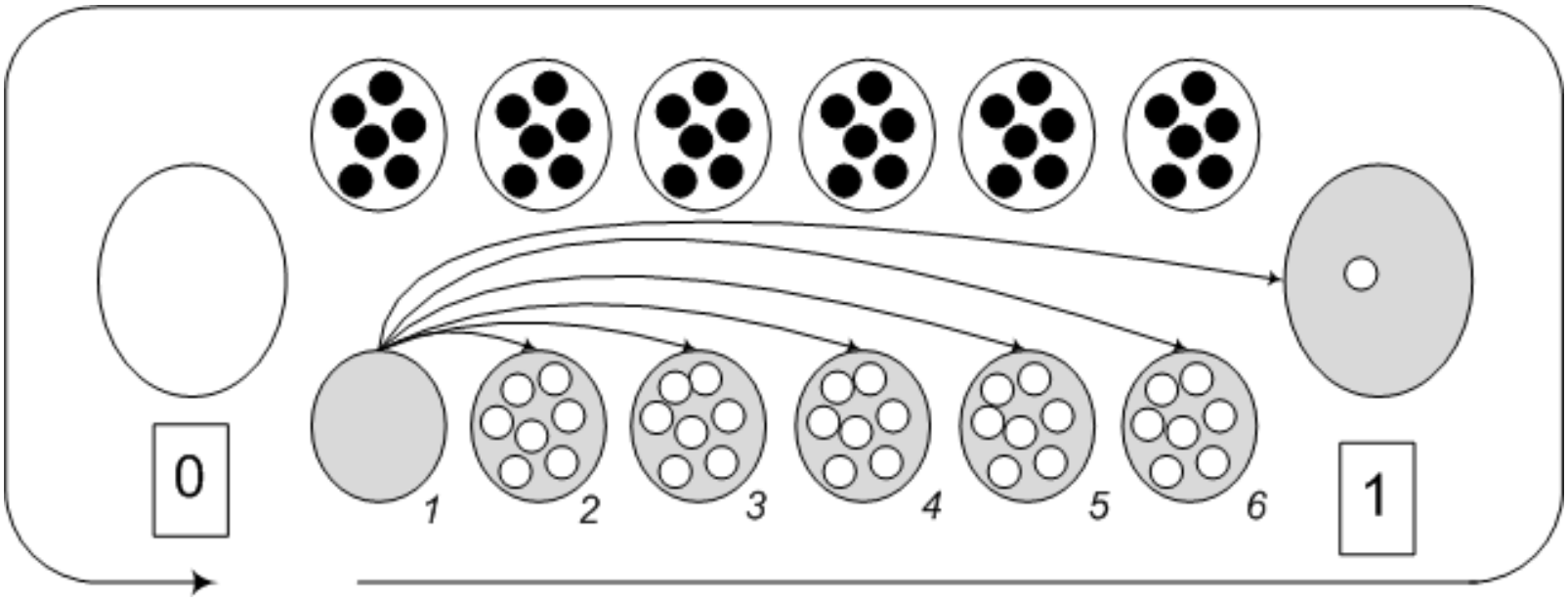
What's the BDD?

- Suppose we create software for Kalah game

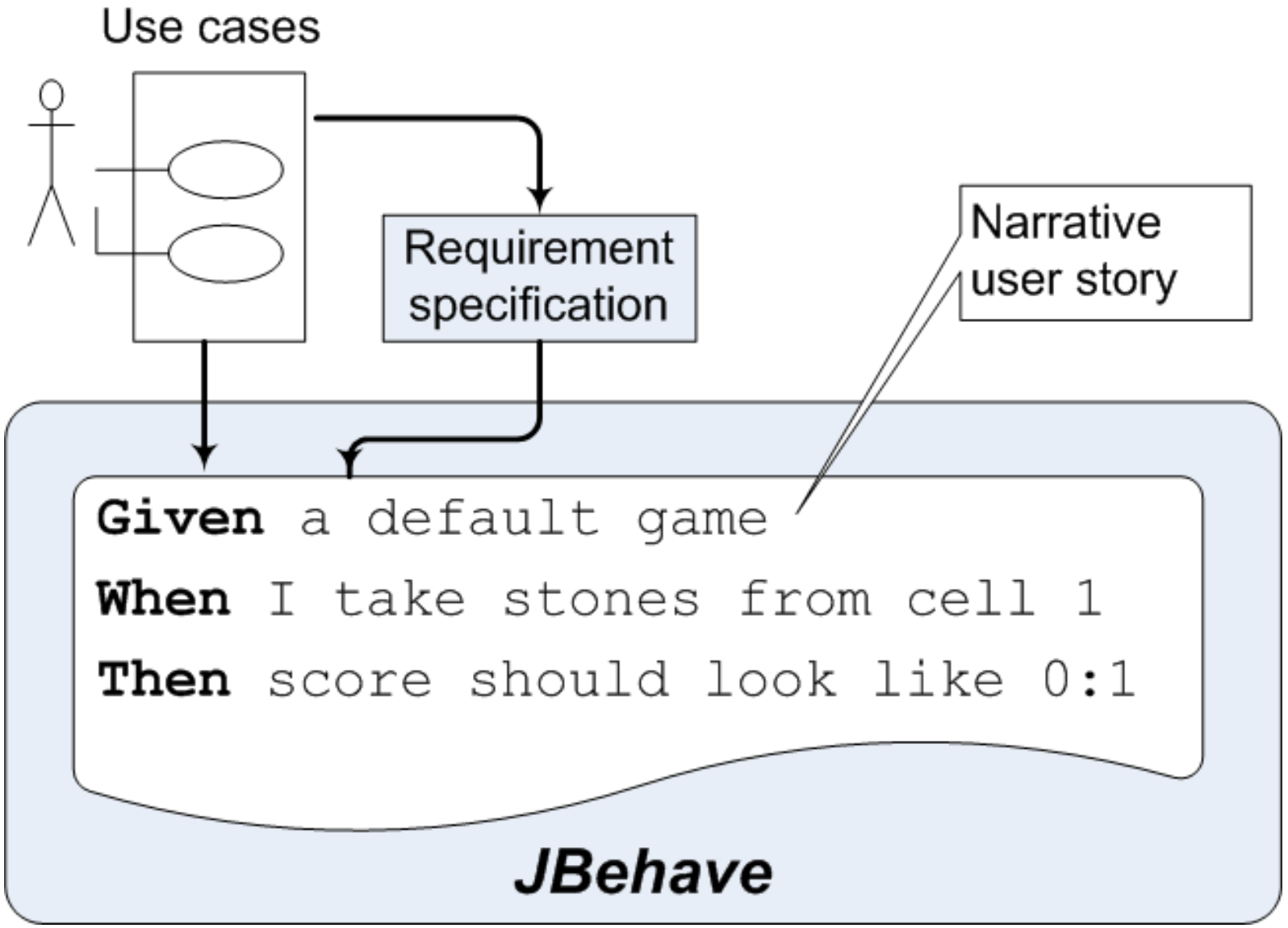


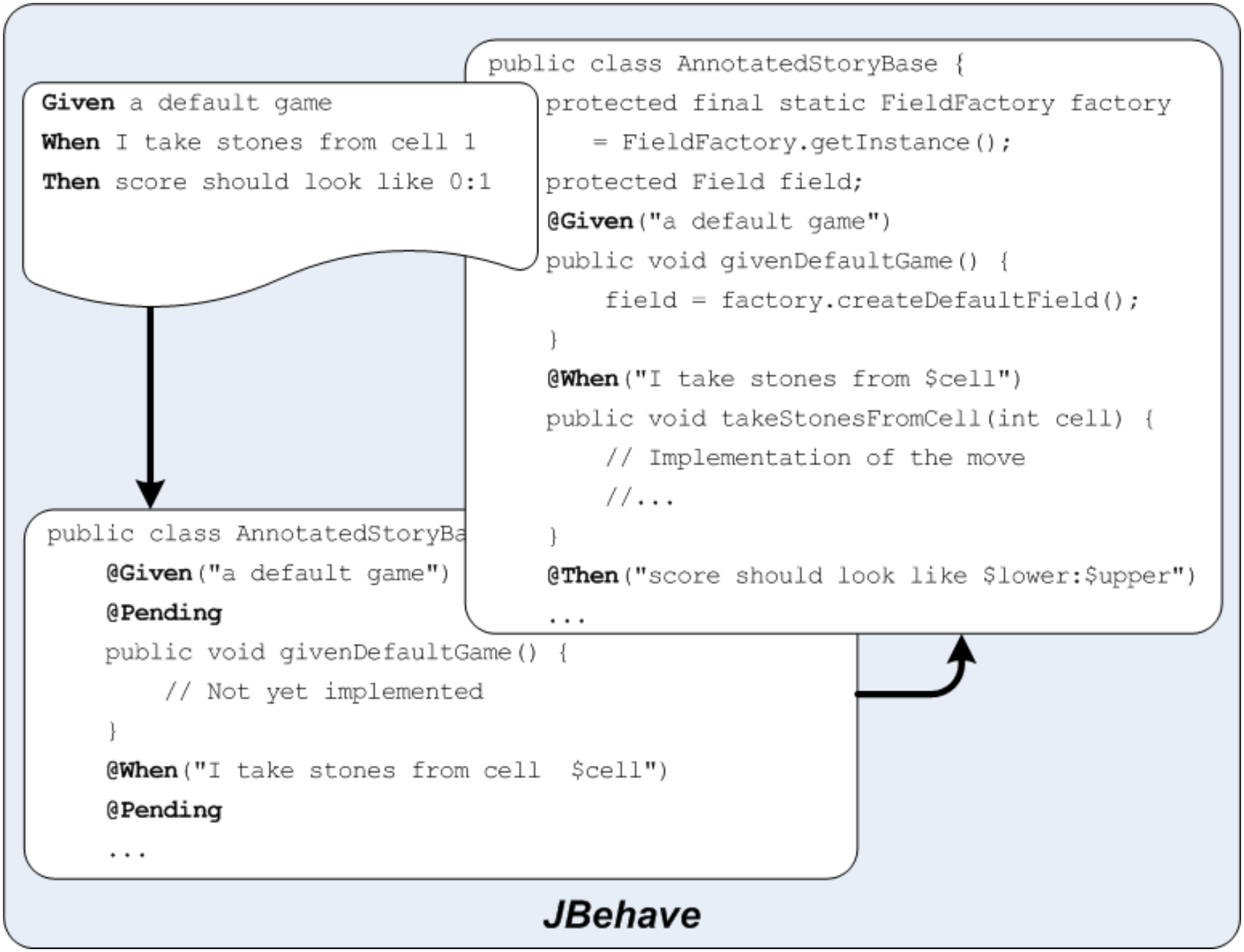
What's the BDD?

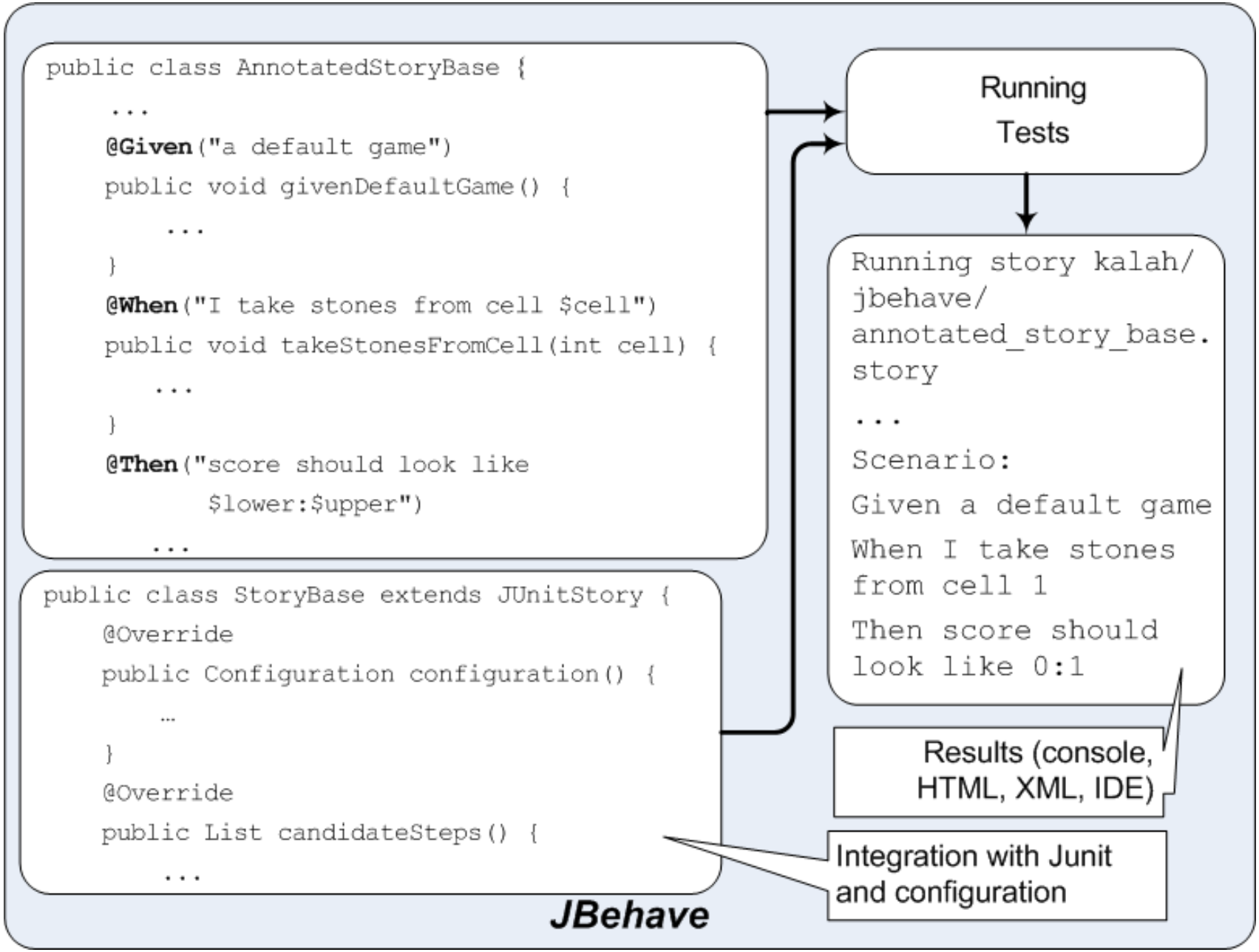
- We know the rules



How it works?

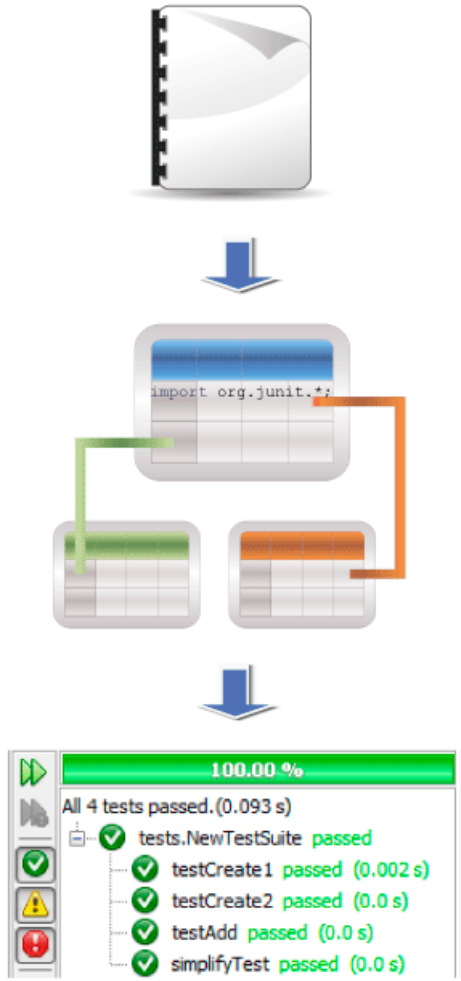

Use cases
Requirement specification
Narrative user story
Given a default game
When I take stones from cell 1
Then score should look like 0:1
JBehave


Given a default game
When I take stones from cell 1
Then score should look like 0:1
public class AnnotatedStoryBase {
protected final static FieldFactory factory
= FieldFactory.getInstance();
protected Field field;
@Given("a default game")
public void givenDefaultGame() {
field = factory.createDefaultField();
}
@When("I take stones from $cell")
public void takeStonesFromCell(int cell) {
// Implementation of the move
//...
}
@Then("score should look like $lower:$upper")
...
public class AnnotatedStoryBa
@Given("a default game")
@Pending
public void givenDefaultGame() {
// Not yet implemented
}
@When("I take stones from cell $cell")
@Pending
...
JBehave


public class AnnotatedStoryBase {
...
@Given("a default game")
public void givenDefaultGame() {
...
}
@When("I take stones from cell $cell")
public void takeStonesFromCell(int cell) {
...
}
@Then("score should look like
$lower:$upper")
...
public class StoryBase extends JUnitStory {
@Override
public Configuration configuration() {
...
}
@Override
public List candidateSteps() {
...
Running
Tests
Running story kalah/
jbehave/
annotated_story_base.
story
...
Scenario:
Given a default game
When I take stones
from cell 1
Then score should
look like 0:1
Results (console,
HTML, XML, IDE)
Integration with Junit
and configuration
JBehave

How it works?

- Let's summarize:
  1. Stories
  2. Mapping to test classes
  3. Configuration
  4. Running tests
  5. Reports



What did we do?

- Analysis of the BDD communication schema
- Study of the BDD tools characteristics
  - Tools & features
  - Integration with IDEs
- Requirement analysis for a BDD supporting tool

jbehave Cucumber specflow PRAGMATIC BDD FOR .NET StoryQ {P} specs²

BDD tools

Toolkit	Analyzed Characteristics			
	Supported languages	***User stories as plain text***	***Mapping rules***	***Automated mapping to the unit tests***
JBehave	Java	Yes	Yes	No
NBehave	.NET	Yes	Yes	No
RSpec	Ruby	No	No	No
MSpec	C#	No	No	No
Cucumber	Ruby, Java, Python, .NET, C++, etc.	Yes	Yes	No
StoryQ	.NET	Yes	Yes	No
SpecFlow	.NET	Yes	Yes	Yes
CBehave	C	Yes	Part.	No

Challenges & Considerations

- Units tests are easy to automate, conversions are not
- Even if acceptance tests don't change after changing requirements, the conversions may change
- Are we always able to define behavior without diving into the code?
- From unstructured native language to the simplified "automatable" language

Integration with IDEs

Toolkit	IDE Integration Features				
	Deployment	***IDE integration***	***IDE templates***	***Debug***	***Unit tests***
JBehave	jar	No	No	Part.	JUnit
NBehave	Install	Plug-in for Visual Studio	No	Part.	NUnit MbUnit XUnit MSTest
StoryQ	dll	No	No	No	Visual Studio Unit Testing
SpecFlow	Install	Visual Studio	Yes	Yes	NUnit, Visual Studio Unit Testing
CBehave	source code	No	No	No	Own

Requirements: BDD inspired

Tracing and debugging the test executions by marked-up scenarios	
Test run reporting	
Back trace to the story from the test run	
Conversion of narrative stories to the marked-up scenarios	
Conversion from the marked-up scenarios to the unit tests	

Requirements: BDD inspired

“Running” user stories	
Conversion from the unit tests to the marked-up scenarios	
Including meta-information to the stories	
Marking-up scenarios	
More…	

Summary

- State of art
  - *BDD tools are still oriented to the developers' side in a greater degree*
  - *In many published cases the test stories and the marked-up scenarios were composed by the same engineers*
  - *Additional work for engineers*

Summary

- BDD ideas are *great* but implemented at surface level

Image from propeopie Blogging about web design and development

Summary

- BDD ideas are *agile* ☺ but implemented at surface level

- Even modest improvements can greatly increase the overall usability of BDD-supporting instruments

2012 CEE-SEC(R)

Software Engineering Conference in Russia

Thank you!

Evgeny Pyshkin

Maxim Mozgovoy

Mikhail Glukhikh

St. Petersburg Polytechnical University
The University of Aizu
Joint Group in Information Retrieval & Software Research