



# Introduction to Behavior Driven Development

Gáspár Nagy

coach • trainer • bdd addict • creator of specflow

@gasparnagy • gaspar@specsolutions.eu

# unknown unknown



# Agenda

- Agile Testing
- Specification by Example
- BDD
- Demo

# What is Agile Testing?





A what?

Hey! There is a bug  
flying around you!

# Agile development landscape

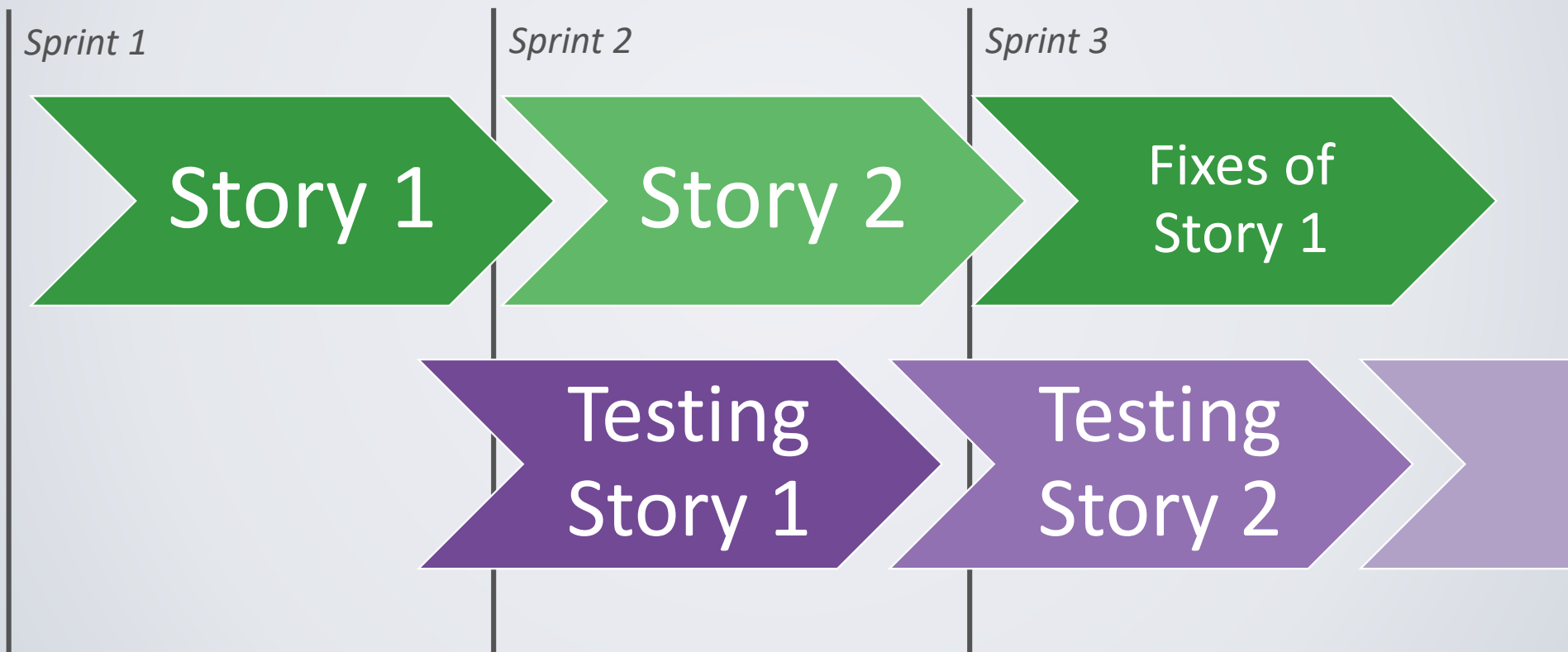
Agile Project  
Mgmt.  
(Scrum, Kanban)

Agile Testing  
(bug prevention,  
exploratory  
testing, context-  
driven)

Agile  
Requirement  
Analysis  
(User Story,  
Acceptance  
Criteria)

Agile  
Engineering  
(XP, TDD, pairing)

# Waterfall testing in Scrum

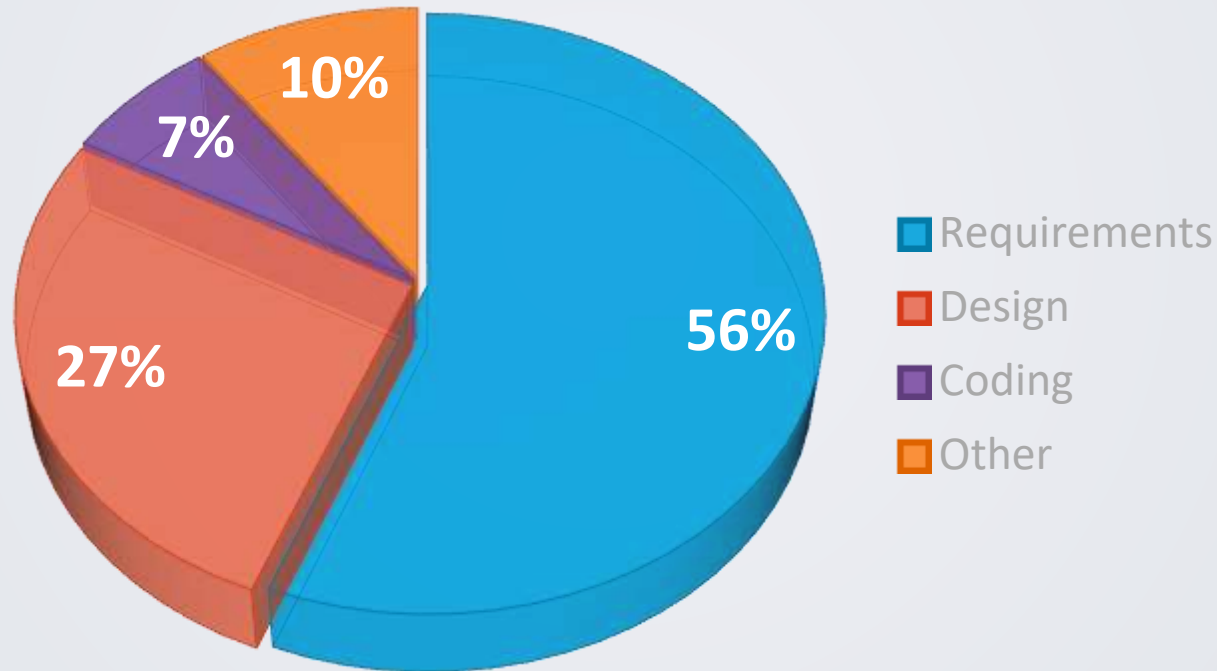


# Agile Testing: Shift from “Bug Hunting” to “Bug Prevention”





# Where do the bugs come from?



Specification is supported by testers


Testers think in tests

Tests are examples

**Specification** is supported by testers

Testers think in tests

Tests are **Examples**



Why Examples?

# Agile mini-waterfall

As a team  
I want to get scores for my  
answers  
So that I can compare my  
results with other teams

*implement*

*feedback*

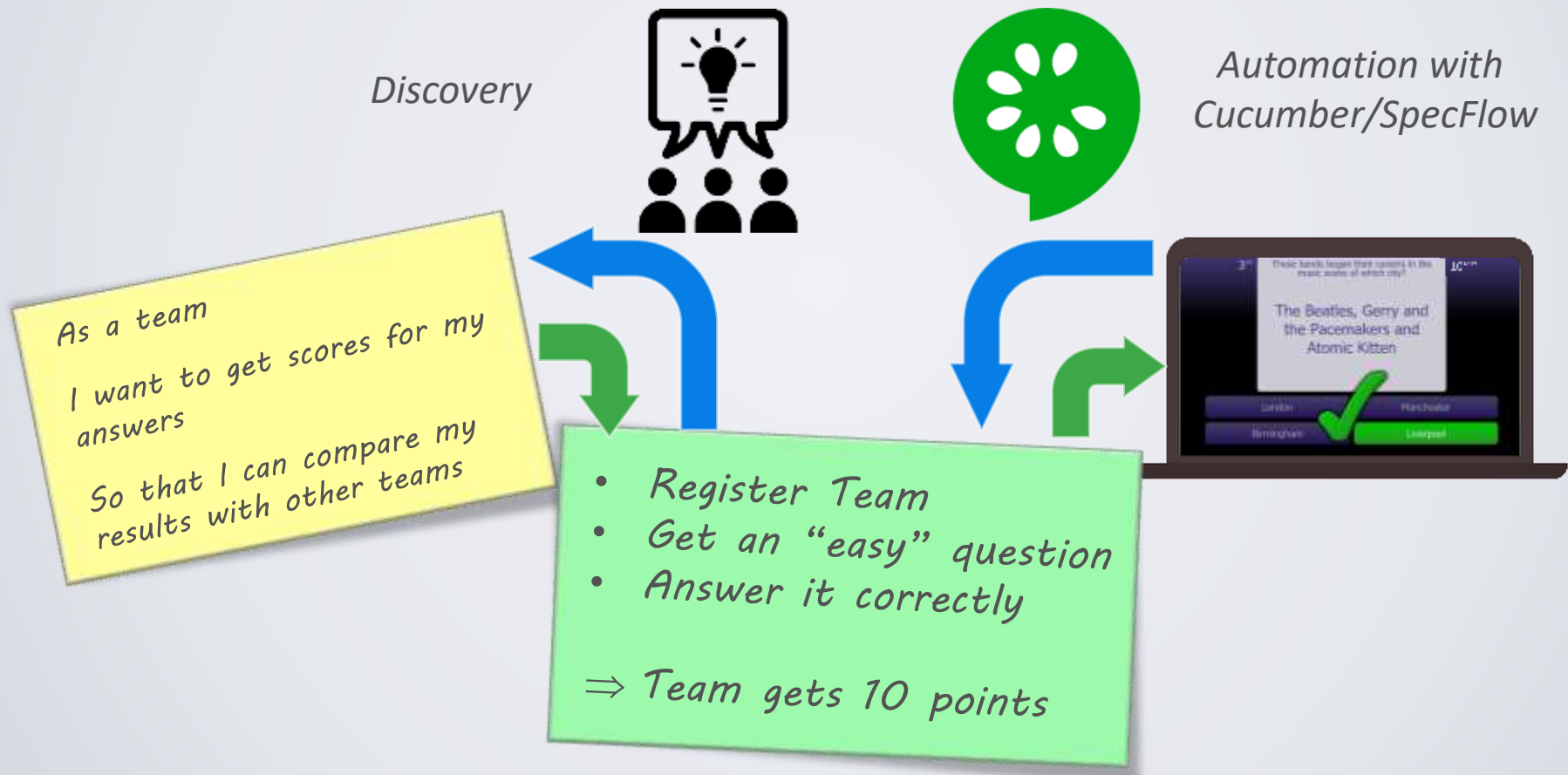
```
[HttpPost]
public void QuestionTest()
{
    string questionTitle = "Test question " + Guid.NewGuid().ToString("N");
    var wait = new WebDriverWait(driver, new TimeSpan(0, 0, 5));

    driver.Navigate().GoToUrl(baseUrl + "/" + "#");
    wait.Until(d => d.FindElement(By.LinkText("Home")));
    Assert.AreEqual("Stackoverflow | Stamplay app", driver.Title);
    driver.FindElement(By.LinkText("Login")).Click();
    Thread.Sleep(2000);
    driver.FindElement(By.LinkText("Ask a question")).Click();
    Thread.Sleep(500);
    driver.FindElement(By.XPath("//input[@type='text']")).Clear();
    driver.FindElement(By.XPath("//input[@type='text']")).SendKeys(questionTitle);

    driver.FindElement(By.ClassName("ta-text")).FindElement(By.ClassName("ta-t")).Click();
    driver.FindElement(By.XPath("//input[@type='text']")).Clear();
    driver.FindElement(By.XPath("//input[@type='text']")).SendKeys("spec");
    driver.FindElement(By.XPath("//div[4]/div/div")).Click();
    Thread.Sleep(2000);
    Assert.AreEqual("Stackoverflow | Stamplay app", driver.Title);
}
```



# Examples link requirements to software



# This is an example!

- *Register Team*
- *Get an “easy” question*
- *Answer it correctly*

*⇒ Team gets 10 points*

# This is an example!

**Scenario:** Correct easy answer scores 10

**Given** I register a team

**When** I submit a correct easy answer

**Then** my score should be 10

*Illustration for  
the requirements*

*Automated test*

*Documentation  
of the behavior*

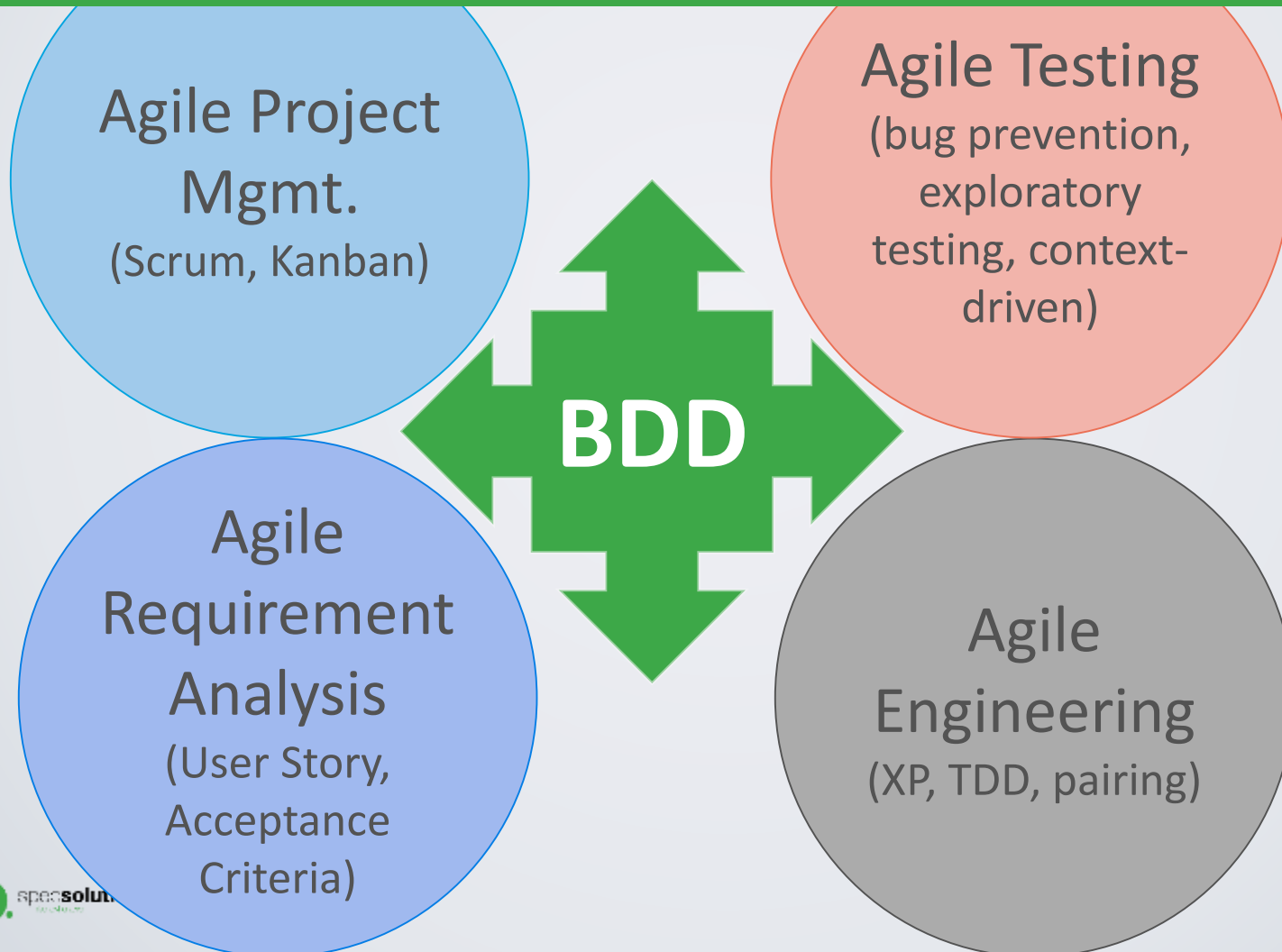
# What is BDD?

*Specification by  
Example*

*Acceptance Test Driven  
Development (ATDD)*

*Behavior Driven  
Development*

# BDD is a process for practicing Specification by Example





***Behavior Driven Development*** is about  
understanding & validating  
business requirements  
through illustrative examples

# The BDD Approach

## Discovery

Shared understanding is established through collaboration and structured conversations

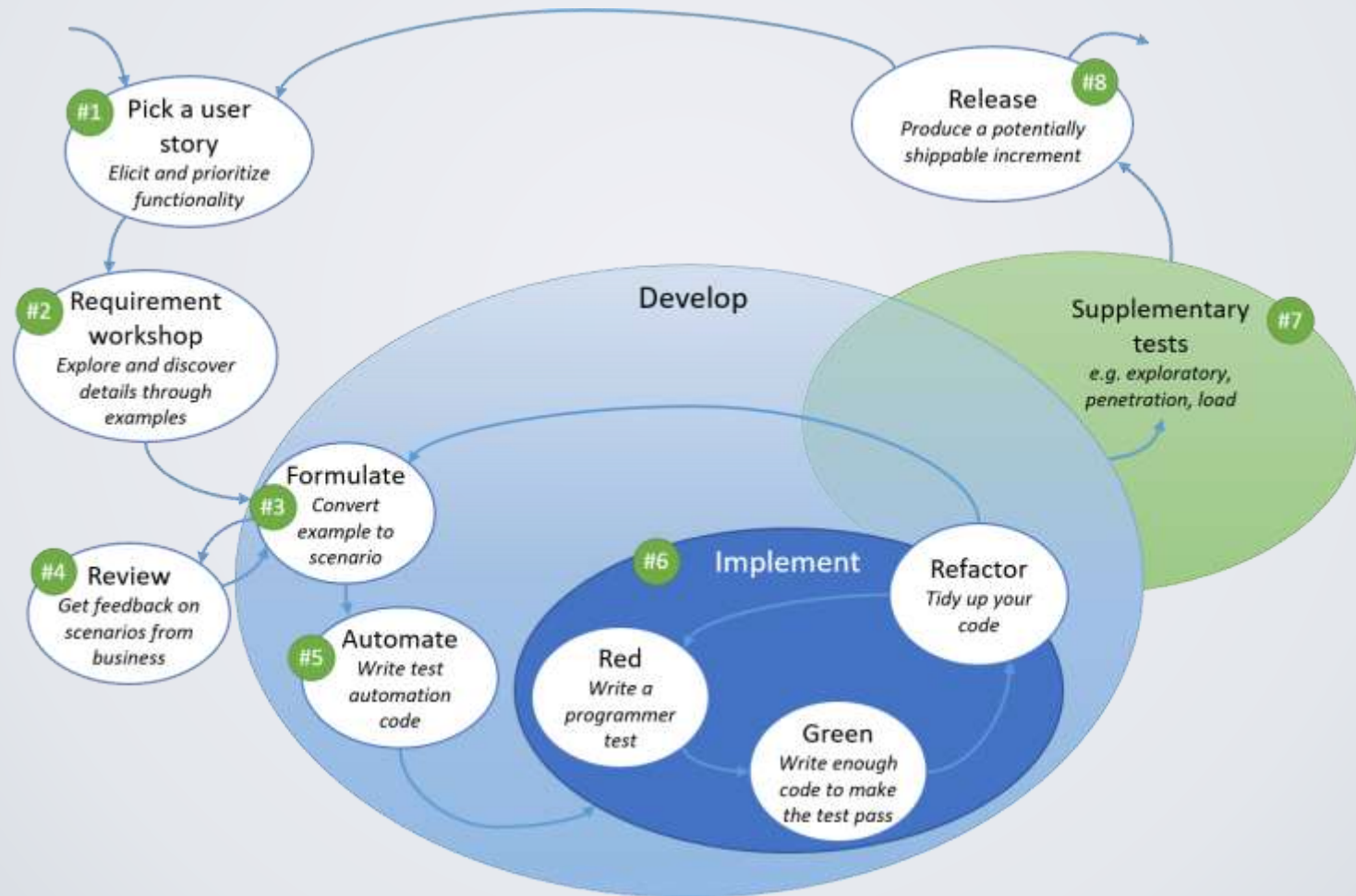
## Formulation

Examples of system behaviour are documented as scenarios

## Automation

Scenarios are automated to be able to verify the system behaviour

# BDD tasks and activities



# Break down stories to rules (acceptance criteria)

As a potential customer  
*I want to **collect books in a shopping cart***  
*So that I can order several books at once.*

*Find checkable rules!*

Books can be added to shopping cart.

Books can be removed from shopping cart.

Shopping cart should be empty when entering the shop.

The same book can be added multiple times to the shopping basket.

...

# Understand rules by examples

The same book can be added multiple times to the shopping basket

- *Shopping basket: 1 pcs of "Clean Code"*
- *Browse "Clean Code"*
- *Add again*

⇒ ???



# Understand rules by examples

The same book can be added multiple times to the shopping basket

- *Shopping basket: 1 pcs of "Clean Code"*
- *Browse "Clean Code"*
- *Add again*

⇒ *Shopping basket: 2 pcs of "Clean Code"*

- Positive/negative
- Sometimes belong to multiple rules
- Not necessarily text

# Scenarios: Examples in Gherkin

As a shop visitor  
*I want to* **collect books in my shopping basket**  
*so that* I can purchase multiple books at once.

**B** The same book can be added multiple times to the shopping basket

*Given* my shopping cart contains 1 copy of "Clean Code"

*When* I add the book "Clean Code" to my shopping cart

*Then* my shopping cart should contain 2 copies of "Clean Code"

=> Shopping basket: 2 pcs of "Clean Code"

# Scenarios drive the development

*Implement automation & product*



**Scenario:** The same book can be added multiple times to the shopping basket

**Given** my shopping basket contains 1 copy of “Clean Code”

**When** I add the book “Clean Code” to my shopping basket

**Then** my shopping basket should contain 2 copies of “Clean Code”

```
[HttpPost]
public ActionResult Answer(int answer)
{
    TriviaEntities db = new TriviaEntities();
    var question = db.FindQuestion(CurrentQuestion);

    if (question.Type == QuestionType.Easy)
    {
        db.AddScore(question, user, 10);
    }
}
```



# Scenarios drive the development

*Implement automation & product*

```
[HttpPost]
public ActionResult Answer(int answer)
{
    TriviaEntities db = new TriviaEntities();
    var question = db.FindQuestion(CurrentQuestion);

    if (question.Type == QuestionType.Easy)
    {
        db.AddScore(question, user, 10);
    }
}
```

**Scenario:** The same book can be added multiple times to the shopping basket

**Given** my shopping basket contains 1 copy of “Clean Code”

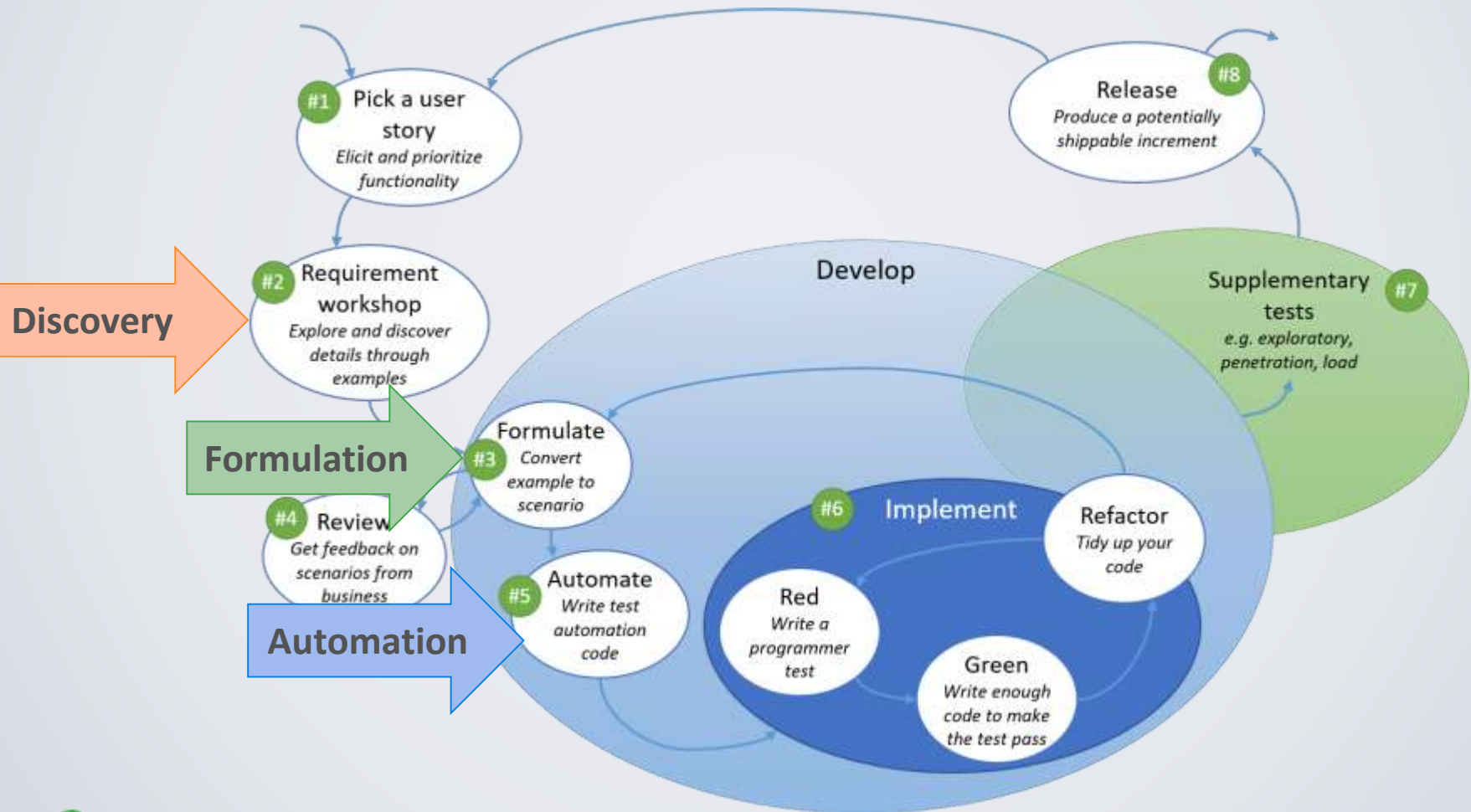
**When** I add the book “Clean Code” to my shopping basket

**Then** my shopping basket should contain 2 copies of “Clean Code”



*Derive further tests, exploratory testing*

# BDD tasks and activities





# The BDD Approach

## Discovery

Shared understanding is established through collaboration and structured conversations

## Formulation

Examples of system behaviour are documented as scenarios

## Automation

Scenarios are automated to be able to verify the system behaviour

# Demo: SpecFlow in Action

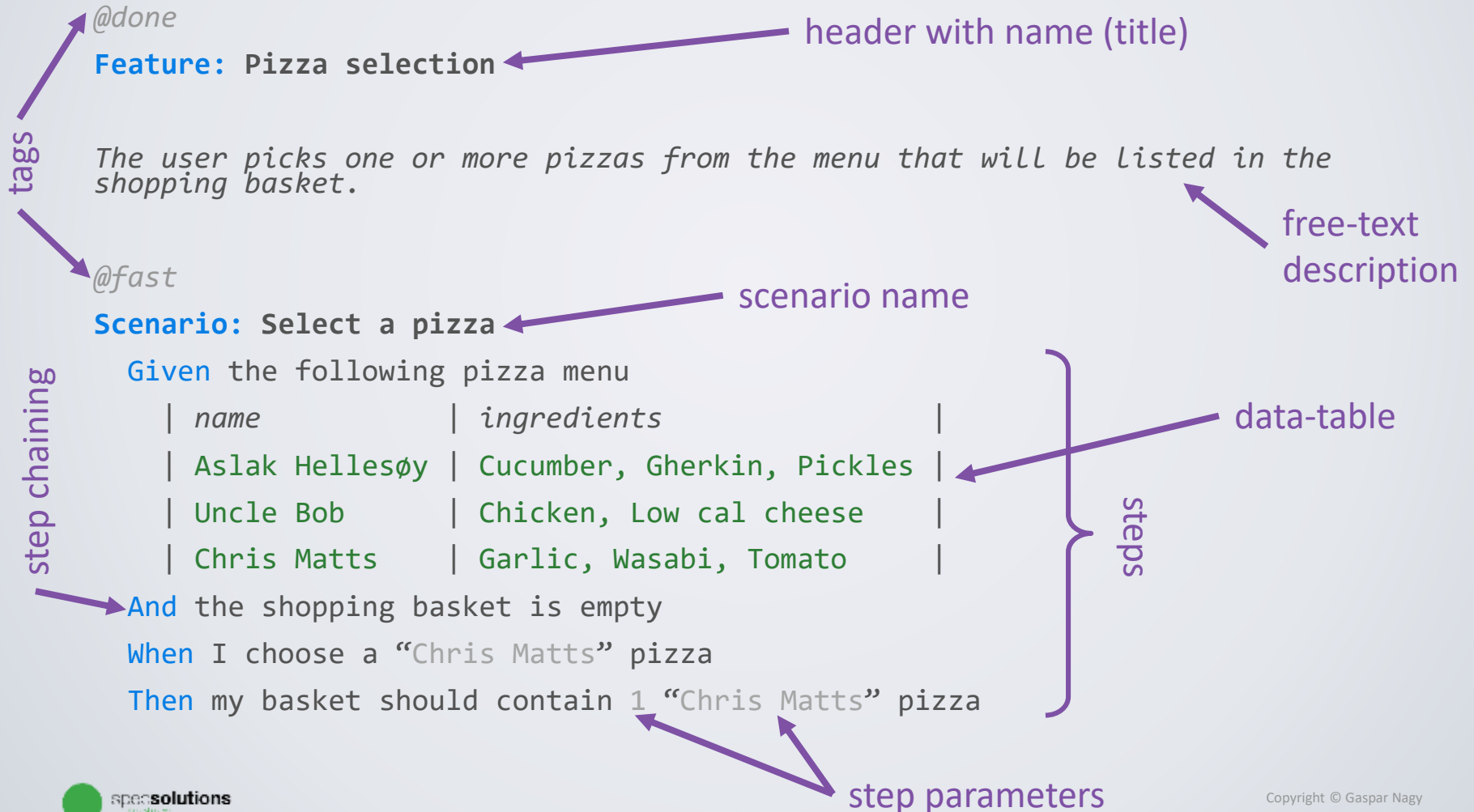
## Gherkin syntax elements

- Feature
- Scenario
- Steps
- Step arguments

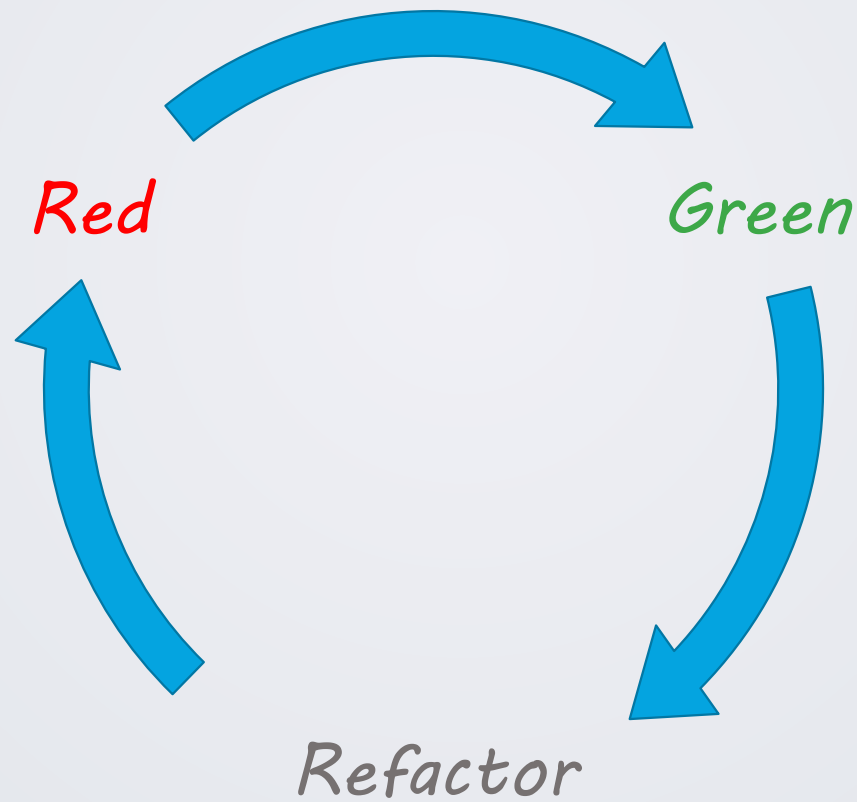
## Executing Gherkin Scenarios

Live connection of scenarios with the app

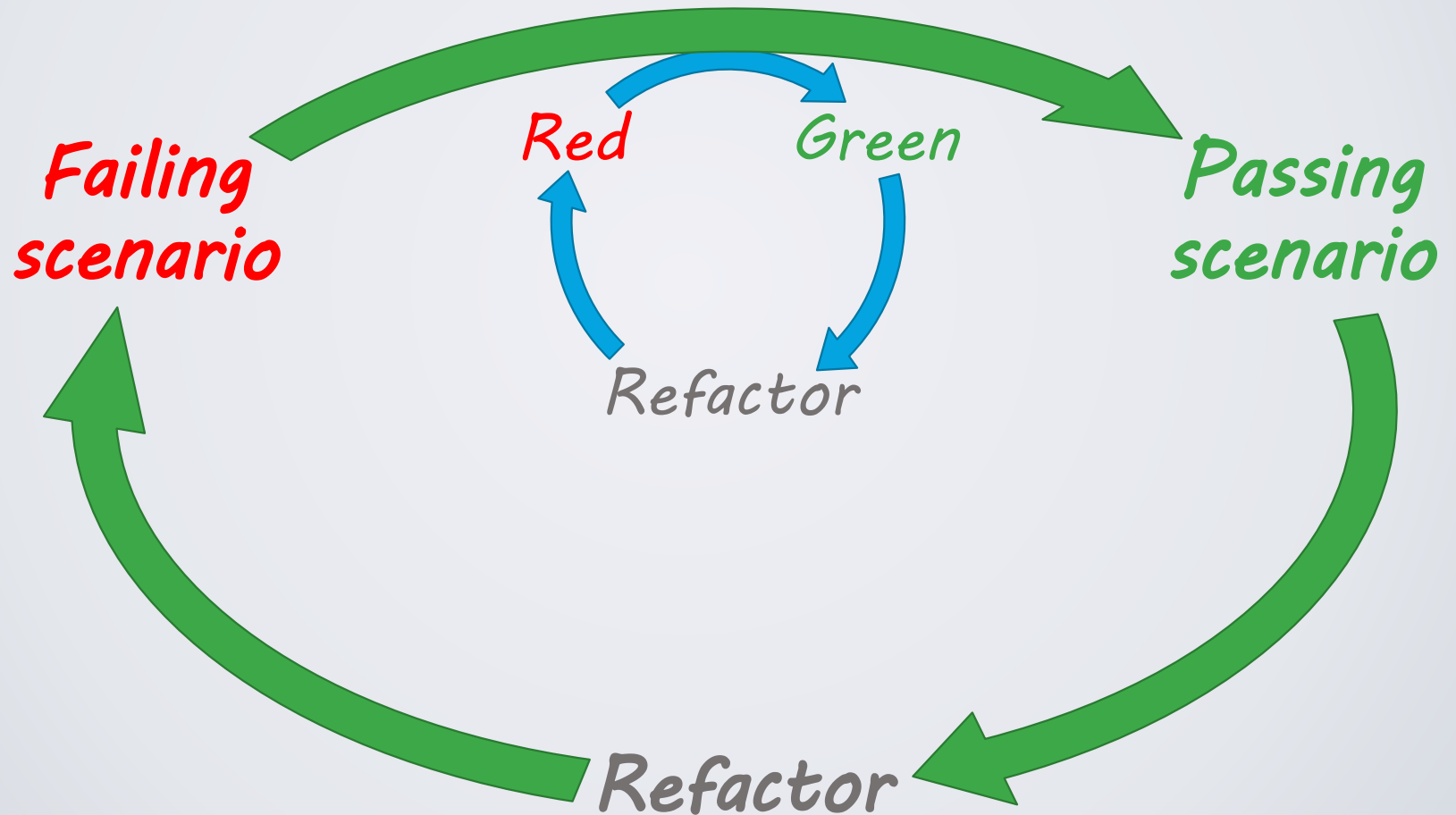
# Gherkin Structure



# The TDD cycle



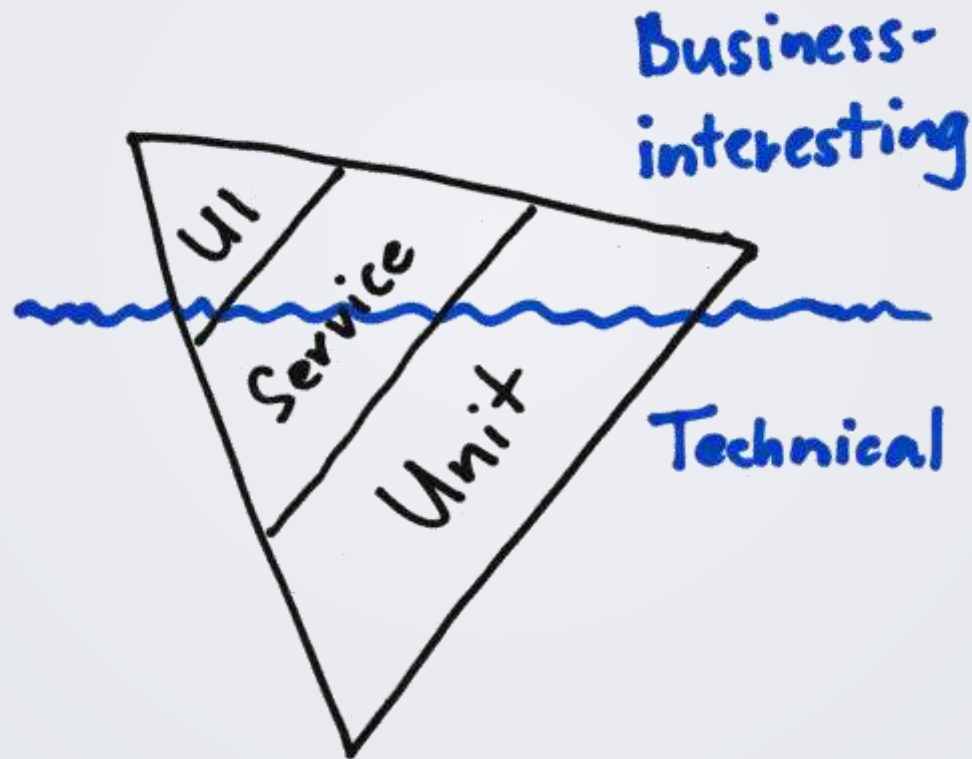
# The BDD (ATDD) cycle



# The Testing Pyramid



# Scenarios in the testing pyramid – the Testing Iceberg





bddaddict.com





# Thank you!

Gáspár Nagy

coach • trainer • bdd addict • creator of specflow

@gasparnagy • gaspar@specsolutions.eu



spec**solutions**  
given.when.then.