bns it
Łódź, ul. Piotrkowska 60
tel. 42 630 24 42
www.bnsit.pl

Test-Driven Development, Specification by Example and Behaviour-Driven Development [TBDDSBE]

Description

Keeping updated requirments documentation well understood and aligned with real customer needs is a very challenging goal. Specification by Example and Behaviour-Driven Development are great mental tools that can help us to achieve it. Describing how to specify requirements, how to document them and then naturally link to code implementation make it revolutionary approach that holistically unifies those artifacts. Putting all this things to work requires technical tools and techniques and this is where Test-Driven Development comes in. Focusing on the target goals and turning into well defined test, which precedes implementation and refactoring, makes this approach really refreshing and extremely effective.

All these metods put together results in great design and well organized architecture.

Here you can find wikipedia definitions:

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards.

In software engineering, behavior-driven development (BDD) is a software development process that emerged from test-driven development (TDD).Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design to provide software development and management teams with shared tools and a shared process to collaborate on software development.

Specification by example (SBE) is a collaborative approach to defining requirements and business-oriented functional tests for software products based on capturing and illustrating requirements using realistic examples instead of abstract statements. It is applied in the context of agile software development methods, in particular behavior-driven development. This approach is particularly successful for managing requirements and functional tests on large-scale projects of significant domain and organisational complexity.

bns it
Łódź, ul. Piotrkowska 60
tel. 42 630 24 42
www.bnsit.pl

DETAILED PROGRAMME

	Module name	Agenda
Day 1	**TDD overview**	• Cycle of programming process • Red-Green-Refactor cycle • Growing the software • Unit testing principles • First Test-Driven Development
	xUnit tool	• Creating xUnit tests • The most commonly used xUnit configurations: @Test [Test], @Before [SetUp], @After [TearDown], @Expected, [ExpectedException], @Ignore [Ignore] • Negative tests • Exception texting • TDD patterns • What to test?: state testing, behaviour testing
	xUnit Patterns	• Guard Assertion • Assertion Method • Delta Assertion • Custom Assertion • Behaviour Verification • Object Factory • Test Helper • Object Mother • Parameterized Tests • Extra constructor • Test-Specific subclass

	TDD principles	• Testing strategies
		  ○ Top – down
		  ○ Known - unknown

		○ Rainy day scenario – sunny day scenario • Implementation strategies ○ Faking it ○ Traingulation ○ Obvious implementation • TDD basic concept ○ Text Fixture ○ Test doubles (Stubs/Fakes/Mocks) ○ Testing state and interaction
Day 2	**Tests refactoring**	• Useful refactorings • Refactoring to the patterns • Testability principles • Unit test refactoring • Don't Repeat Yourself
	Source code testability	• Composition versus inheritance • Static elements, singletons • Dependency Injection, Dependency Injection • Layered architecture
	Mockito/Moq tools	• Mocking lifecycle • Testing behaviour • Stubbing
	Testing rest of the world	• Useful strategies • Adapters layer
Day 3	**Specification by Example**	• Difficulties with requirements management in traditional methods • Executable documentaton – a way to shorten feedback loop • Scope definitione based on objectives • Specification illustrated by examples • Specifying details

		• Automated validation • Validation frequency • Documentation evolution
	Behaviour-Driven Development	• Stories, descriptions, narrations • Story format and properties – As, I can, So that • Format i cechy historyjek użytkownika – As, I can, So that • Defining scenarios • Given, when, then • Organizing project with specifications • Gerkin language • Higher level requirements (Features/Epics) • Scenarios with Given, when, then template • Reusing scenario steps • Linking specification and code • BDD and incremental processes
Days 1-3	**Workshop**	• Practical application of TDD, SBE and BDD