modelus
knowledge. talent. results.

Driving Quality

Front-to-Back Test-Driven Development

Aleh Matus
Jacob Mulholland

OTUG - 4/17/2012

Aleh Matus

Aleh Matus is the founder of Modelus. He specializes in software design, business domain and mathematical modeling and has a passion for creating innovative technology solutions.

Jacob Mulholland

As Principal Consultant at Modelus, Jacob works with small clients and large enterprises in all facets of technical and business solution development. Recognizing huge gaps in technical skill-sets, Jacob has forged an effort to bring invaluable lessons of Design Patterns, DDD, TDD, and OO into the client-side UI/UX community. Through his OOUI effort, Jacob is redefining the thought-process and methodologies of client-side development.

modelus
knowledge. talent. results.

Test-Driven Development

- Never write a single line of code unless you have a failing automated test
- Eliminate duplication

Kent Beck, Test-Driven Development: By Example. Addison-Wesley Professional, 2003.

Objectives for Tests

- Speed
- Automation
- Availability
- Repeatability
- Clarity / single purpose / single reason to fail
- Isolation / side-effect free behavior

Single-Responsibility Principle

Every Test should have a single responsibility.

For Tests, we define Responsibility as a reason to fail. Every Test should a single reason to fail.

Objectives for Development

When working on development projects, focus on the following objectives:

- Support DDD philosophy
- Establish TDD techniques
- Create Distributed Development environment
- Favor set-based design over point-based design

Development Flow Principle

Allow complexity to develop naturally in the application while maintaining a sustainable development pace.

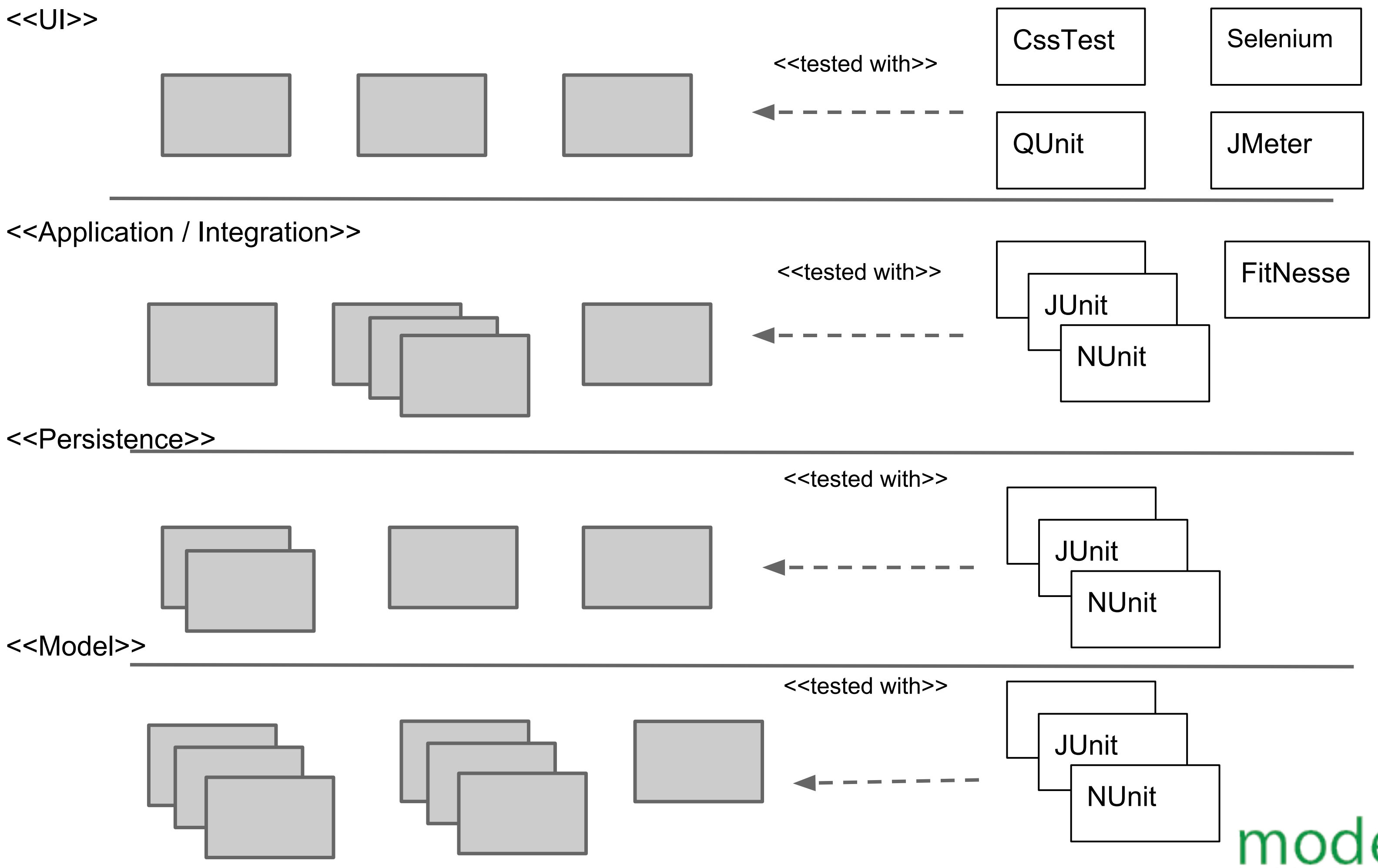
Release Unit Pattern

Release Unit is a collection of folders and files that are released together.

The source code is organized in Release Units that follow a domain context map.

All Release Units adopt a standard layout.

Release Unit Pattern



Model

Methods

- Class Design Principles
- Separation of Concerns
- Dependency Management
- Keep Classes free from Infrastructural Knowledge
- Automate Assertions via Reflection

Tools

- xUnit

Database

Methods

Testing Database via Repository Interfaces

Dedicated Database for each Test Project

- Local Database on Dev Machines
- Local Database on Build Agents

Using Transactions in Base Test Class for Persistence

Testing for Minimum and Maximum Conditions

Database Script Management

- Database Create Scripts for Release Units
- Database Update Scripts for Deployment

Tools

xUnit

JavaScript

Methods

- Separation of concerns
- Inversion of control
- Attribute: class is instance

Tools

- Web Browser
- QUnit

CSS

Methods

- Separation of concerns
- Attribute: class is instance
- Selectors are signatures
- Forego the cascade

Tools

- Web Browser
- SASS
- Compass
- CssTest

Questions

?

modelus
knowledge. talent. results.