Zen and the Art of Test Maintenance Workshop

As the code changes, the tests change too. This 1-day workshop is about the actual work we call "maintaining the tests".

We will discuss test relevancy and value, as the requirements change and when to (heaven forbid) abandon tests. We'll see cases where we need to change the level of existing tests (unit, API, UI or any other type), as we add and change functionality, and replace them with the appropriate level. We'll see how to approach tests from either test-first (BDD or TDD) or test-after. We'll refactor the tests to make them generic, as the code becomes more generic, and change the language they describe the examples.

As we go, we'll touch on what makes them "maintainable". Then make them so.

This 1-day workshop is highly interactive, and the exercises are similar to real world tests, including tests for APIs, database and 3rd party services. It is workshop for people who live in their tests and care for that quality of living.

The workshop is intended for: Testers, test automation engineers, developers and their team leaders.

Workshop goals

Following the workshop, attendees will be able to:

- Categorize and organize tests effectively
- Create, review and model existing and new test cases for a test plan
- Determine which kinds of tests to write for a feature (unit, integration, system, end-to-end)
- Create façades that are highly maintainable (like page objects, repositories, etc.)
- Evaluate tests based on their effectiveness, age, coverage, etc.
- Refactor code to make it more testable
- Refactor tests to make them more readable
- Use test-first effectively
- Add the next tests easily

Workshop agenda

- Effective test organization
- Where should we add the next test
- Modeling existing and planned tests
- Strategy for which types of tests to add
- Criteria for test effectiveness
- Patterns for better testability in code
- Seams and how to inject them
- Using existing frameworks and tools (Spring, DI, etc) to the maximum
- Refactoring the code before adding new tests
- Refactoring different types of tests
- Using test first approach on top of existing test suite

http://www.gilzilberfeld.com

Prerequisites

- Attendees need to have their own laptops ready and download the exercises beforehand.
- The training materials are targeted to either Java (Spring knowledge preferred) or C# (.Net core knowledge preferred) developers.
- Familiarity with unit testing is required.