

Building Quality Code with Microsoft Pex

T O O L S

Moles

Moles is for the definition of the **isolation** environment. It is included in Pex, it extends Pex with useful functions. Moles has a successor project called Fakes, but it does not support collaboration with Pex, yet.

Pex

Pex is an automated white-box unit **test generator** tool. It traverses all the possible execution paths of the source code, and tries to create inputs to traverse all of them. Unit tests are then generated and parameterized by Pex.

Code Contracts

Code Contract is a means to provide the ability to define **assumptions** regarding the code. Code contracts are used to guide the test generation algorithm. In addition code contracts are appropriate to be used as API documentation.

O V E R V I E W O F T H E A P P R O A C H

STEP

FUNCTION DEVELOPER

TEST DEVELOPER

SPECIFICATION

Interface definition of the module currently under development. Interface definition is based on the specification.

DEFINITION

(*OPTIONAL*) **Define** assumptions and assertions with Code Contracts for each method stub. Assumptions indicate invalid inputs, while assertions give Pex oracles and ideas for inputs.

Create parameterized unit tests (*PUTs*) for each method using the AAA pattern. Design the isolation environment for mocking (by analyzing the structure of the code).

IMPLEMENTATION

Implement business logic. It is important to *keep testability in mind*, such as defining interfaces for testing or choosing proper visibility parameters.

Implement

- Mocks and stubs,
- Test oracles based on the specification.

EXECUTION

Build the code. Optionally, try the simple Exploration function of Pex on the new build. This might help discovering simple bugs.

Implement necessary object factories. Run Pex on each PUT. Evaluate the results and send the *results* to the developers.

ITERATION

Fix uncovered problems based on the *test results*. Improve your code until reaching the required quality.

Run Pex again, where the code is modified. Evaluate the results and send feedback to the developers.

RESULT

The source code of the software is extensively tested. A **regression test suite** is also available to test future versions. We efficiently reduced the probability of major and minor bug occurrences in the tested projects.
We reached **higher code quality** in our software.

READINGS

<http://research.microsoft.com/en-us/projects/pex>
v.gd/c4t2fY • v.gd/m7DFY1 • v.gd/fdOTyV



CONTACT

Dávid Honfi, Zoltán Micskei
(<http://mit.bme.hu/~micskeiz>)

