LIES, DAMN LIES, & CODE COVERAGE
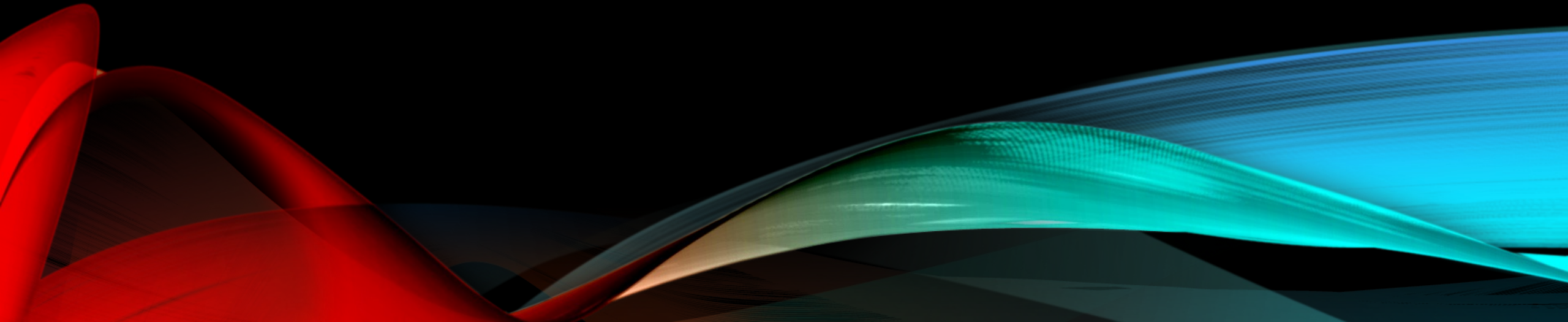
Using Mutation Test to Validate Unit Tests
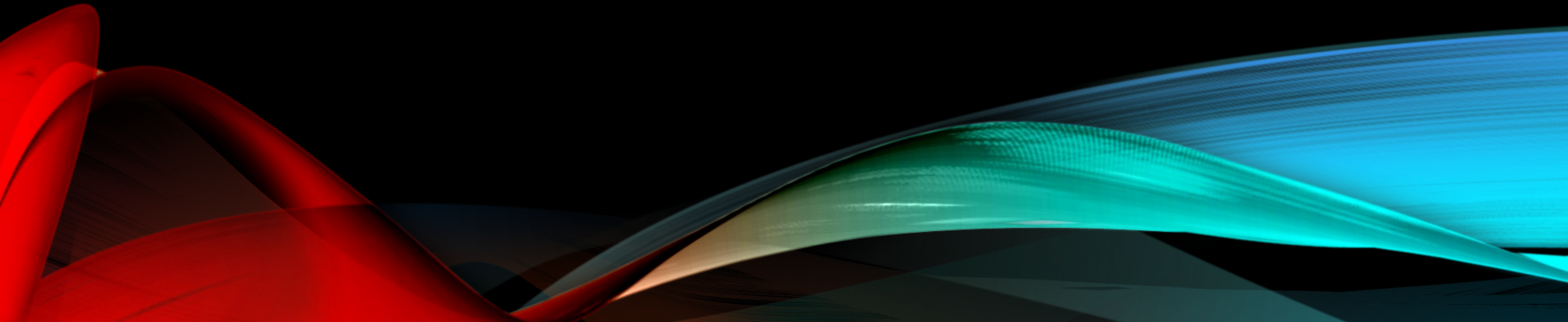
Matthew Heironimus

dev up 2024

SOFTWARE TESTING SURVEY

- Unit Testing
- Code Coverage Metrics
- Test Driven Development (TDD)

MANUAL TESTING EXAMPLE

- A change (mutation) is made to the code being tested.
- The test suite is run against the changed code.
  - If the tests fail, the mutant is killed.
  - If the tests pass, the mutant lived.
- Kill all the mutants!

MUTATION TESTING

- Mutation testing is a technique for assessing the effectiveness of a test suite.
- Mutation testing demonstrates a test suite will fail if a semantic change is made to the code.
- A semantic change is a change to the meaning of the code.
- Goal: A Semantically Stable Test Suite

SEMANTICALLY STABLE

- Why do we want Semantically Stable Tests?
- To eliminate the fear of changing the code
- We know at least one test will fail if we introduce a semantic change to our code.

MUTATION TESTING TOOLS

- Java
  - PIT - https://pitest.org/
  - Major mutation framework - https://mutation-testing.org/
- C#
  - Stryker - https://stryker-mutator.io/
  - VisualMutator - https://visualmutator.github.io/web/
- JavaScript / TypeScript
  - Stryker - https://stryker-mutator.io/

MUTATION TESTING TOOL EXAMPLE

OTHER BENEFITS

- Identifying Missing Test Cases
- Identifying Dead or Unneeded Code
- Identifying Bugs

ARE THERE OTHER WAYS?

Semantically Stable Test Suite

TEST DRIVEN DEVELOPMENT

- TDD typically produces very high code coverage
- TDD is a technique for creating a set of tests that are Semantically Stable

3 LAWS OF
TEST DRIVEN DEVELOPMENT

1. Do not write any production code unless it is to make a failing unit test pass.

2. Do not write any more of a unit test than is sufficient to fail (compilation failures are failures).

3. Do not write any more production code than is sufficient to pass the one failing unit test.

Robert C. Martin

MUTATION TESTING

- Mutation testing is a way to ensure the Semantic Stability of a test suite that was not written using TDD.

- Mutation testing can also be used to verify a set of tests written using TDD is Semantically Stable.

- Legacy Code
  - Code that was written by someone else
  - Code that was written without tests
  - Code that has tests, but you do not trust them
  - Code that you are afraid to change
- Semantic Stability = Freedom to Change the Code

LEGACY CODE

- Using code coverage and mutation testing tools, a set of semantically stable tests can be written for Legacy Code.

- Once we have our semantically stable tests, we can change Legacy Code without fear.

- (or at least with a little less fear)

REVIEW

- What is Mutation Testing?
- Why do Mutation Testing?
- How to do Mutation Testing

REFERENCES

- Stryker - https://stryker-mutator.io/
- Wikipedia - https://en.wikipedia.org/wiki/Mutation_testing
- The Clean Code Blog - https://blog.cleancoder.com/uncle-bob/2016/06/10/MutationTesting.html
- Video: Testing Like It's 1971, By Henry Coles - https://vimeo.com/145201725

CODE SAMPLES & EXAMPLES

- https://github.com/MHeironimus/devup-2024
- https://bit.ly/devup2024

QUESTIONS?

ADVANCED TOPICS

- Mutations & Test Timeouts
- Other Mutation Types
- Writing Custom Mutations