Xerris Bootcamp Series

Test Automation

xerris

ACCOLITE DIGITAL
Transforming The Future, Now

What is Test Automation

Test Automation

- Test automation is the process of developing a program to test the quality of another program.
- These tests verify the features of your system to ensure they meet the customer's expectations.
- Tests are repeatable, providing a safety net when refactoring the system.
- Tests act as documentation showing how the system reacts to the inputs given.
- Tests help architect the software to provide a higher-quality product.

What is Test Automation

What are the goals of Test Automation?

- Improved software quality.
- Act as a form of system documentation.
- Helps reduce risk.
- Easy to run from the command line.
- Easy to write and maintain.
- Require minimal maintenance as the system evolves.
- Easily integrated into the build pipeline.
- Quicker to market.

The Economy of Test Automation

Tests can be expensive to develop and maintain

- Upfront investment scares some off.
- The investment is paid back by higher-quality software.
- Shorter QA cycles.
- Fewer bugs are reported.
- The regression suite provides feedback during refactoring.
- Overall improvement in software design.
- Improved developer productivity and satisfaction.

The Economy of Test Automation

Getting over the hump
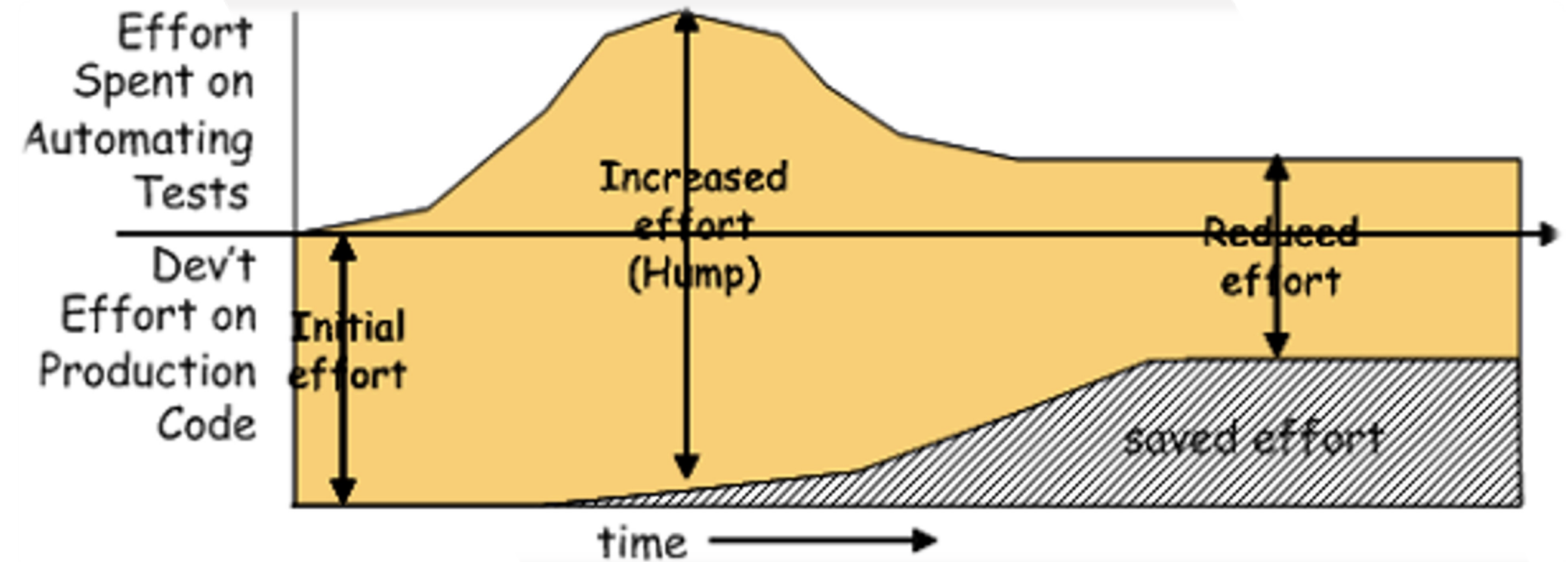
Types of Tests

User Interface Tests
- Tests that ensure the UI functions properly
- Can act as end-to-end tests
- Tend to be more fragile and expensive to maintain

Unit Tests
- tests a single component (class)

Integration Tests
- End-to-end Test
- Test that interacts with external dependencies
- Resources beyond your control

Acceptance Tests
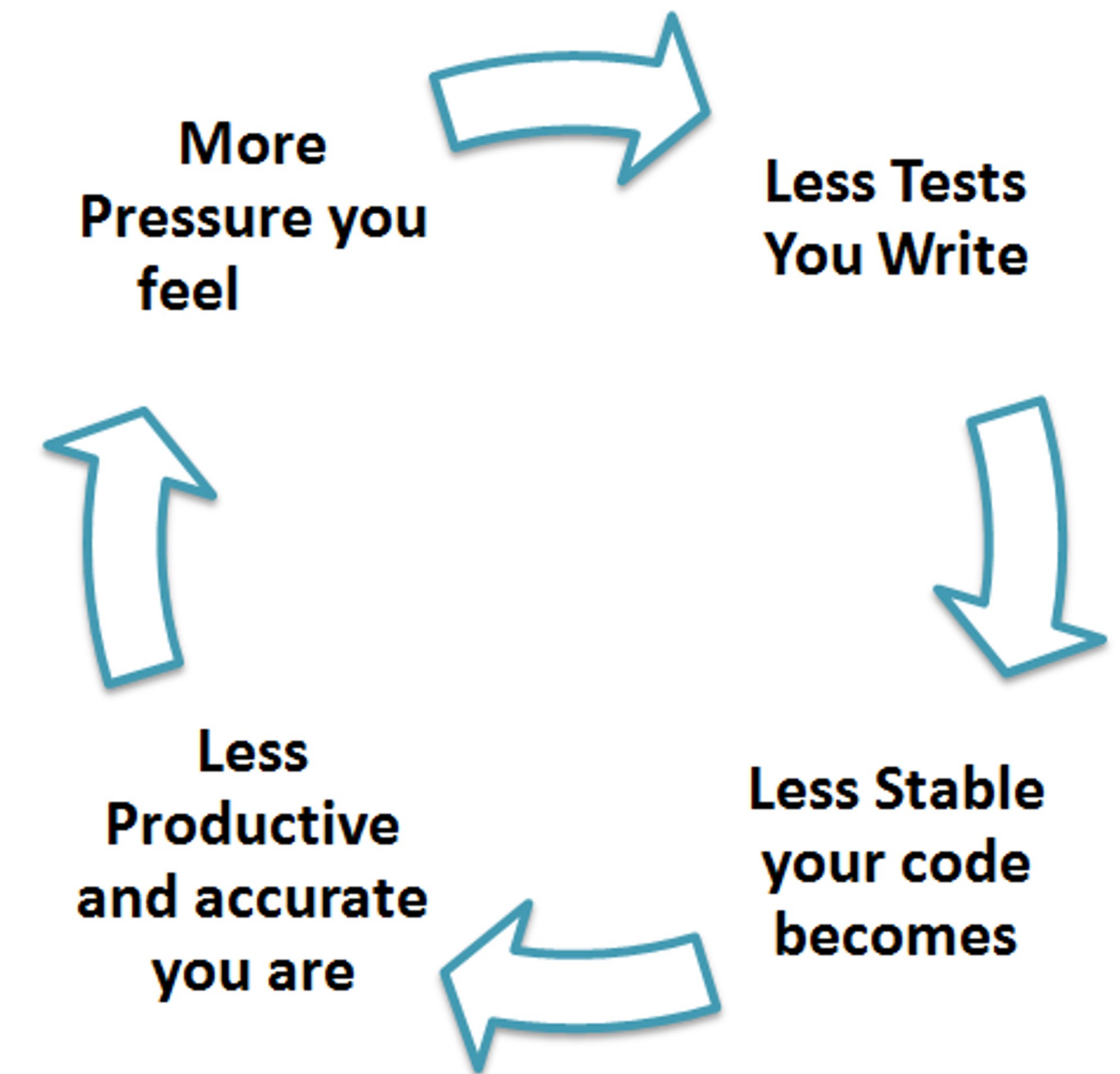- Determines if the feature delivers what was agreed upon.

Smoke Tests
- Used after a deployment to validate the deployment was a success.
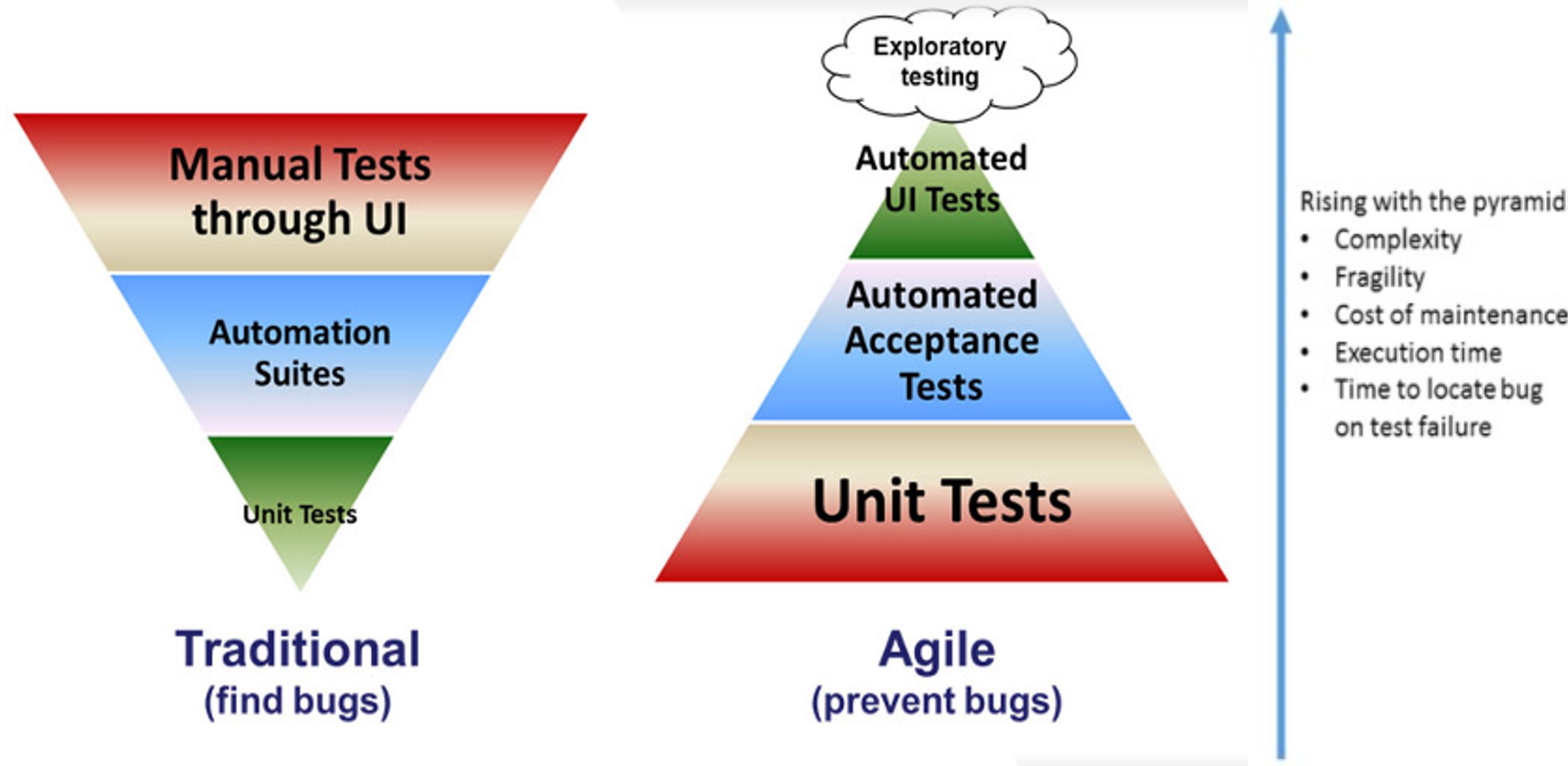- Usually non-destructive-type tests.

Unit Tests

What is a Unit Test?

- A "program" to test a single component of the overall system.
- Tests the correctness of an isolated unit (class).
- Written in the same language as the production software.
- Open-source frameworks provide great unit testing tools.
- Attempts to isolate a class from its dependencies to test it as a single unit.
- Mocking tools allow for this type of isolation.

More Pressure you feel

Less Tests You Write

Less Stable your code becomes

Less Productive and accurate you are

Test Automation

Traditional
(find bugs)

Manual Tests through UI

Automation Suites

Unit Tests

Agile
(prevent bugs)

Exploratory testing

Automated UI Tests

Automated Acceptance Tests

Unit Tests

Rising with the pyramid:
- Complexity
- Fragility
- Cost of maintenance
- Execution time
- Time to locate bug on test failure

Test Automation

What is the right balance of tests?

- Automated UI tests are more expensive and brittle.
- Service API tests provide great documentation for the service.
- Tests the orchestration between all the individual components (units).
- Acts as a boundary between UI -> Back-End.
- Unit tests each component and overall provide the most value (arguably).

Minimize late-cycle brittle UI driven tests

Focus on highly automated API centric testing

Establish a solid foundation of early stage Unit Tests