Test Automation

What is 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 expectations of the **customer**
- **Tests are repeatable** providing you 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

xerris

Goals of Test Automation

That are the goals for 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
- Quicker to market

xerris

Economics of Test Automation

Tests can be expensive to develop

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

xerris

Economics of Test Automation

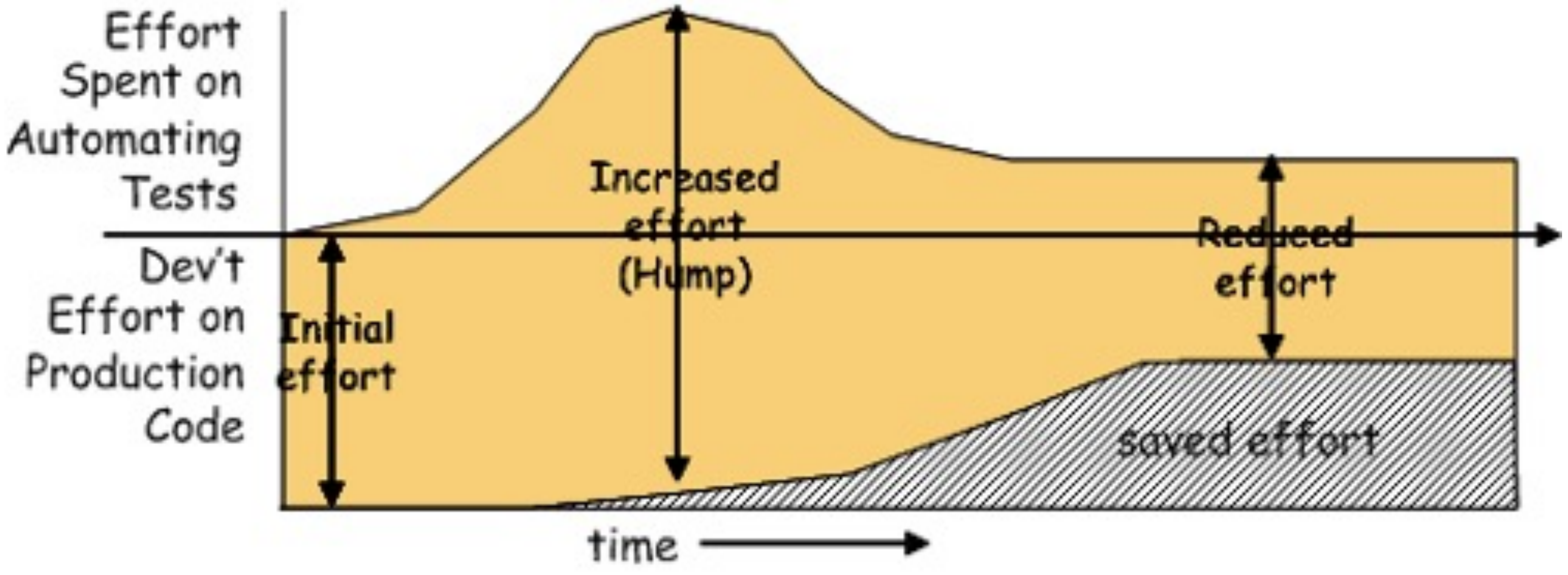
Tests can be expensive to develop

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

xerris

Economics of Test Automation

Getting over the "hump"



http://xunitpatterns.com/

Types of Tests

- 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.

xerris

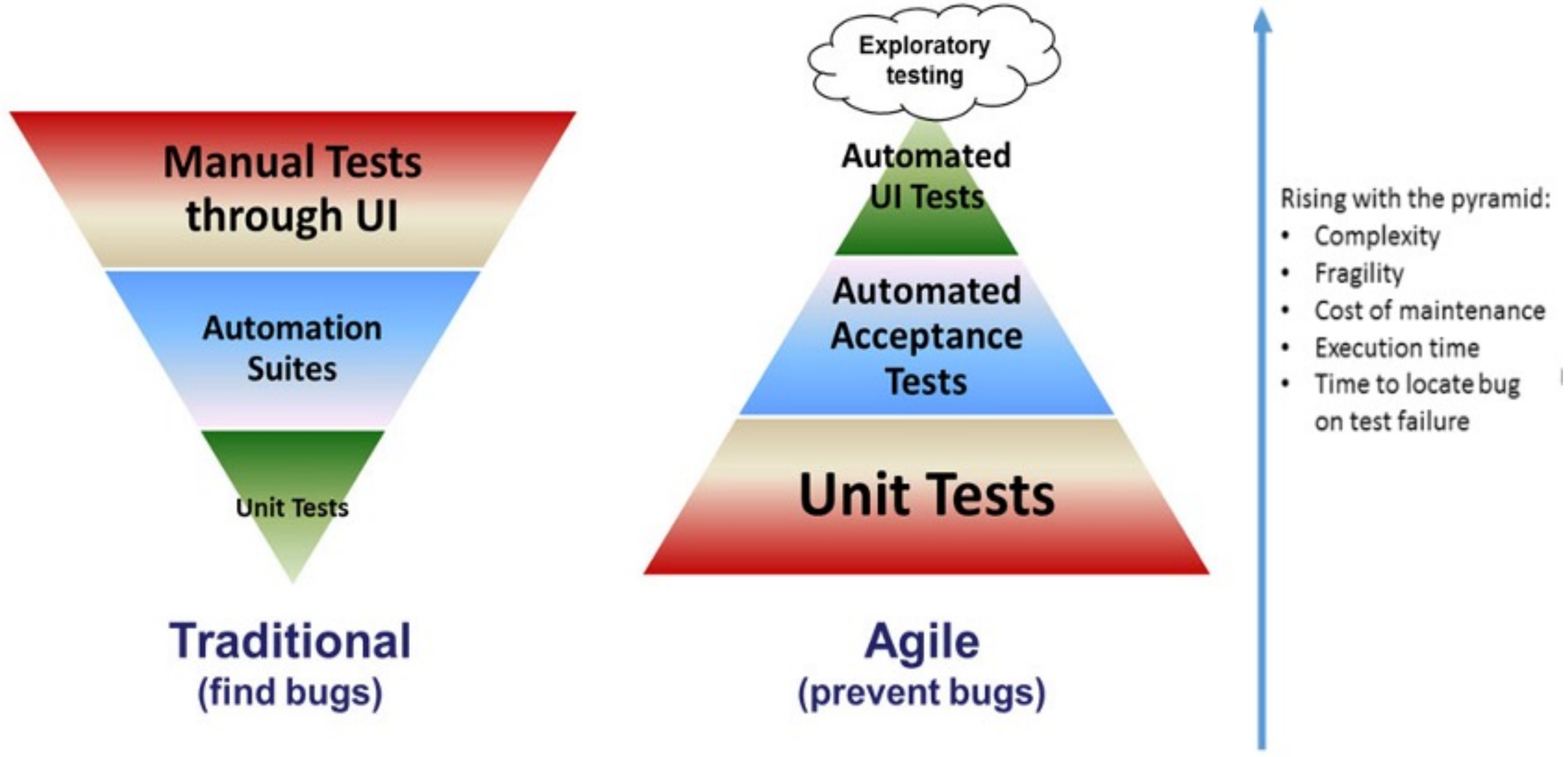
Unit Tests

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.

xerris

Testing Pyramid



Rising with the pyramid:

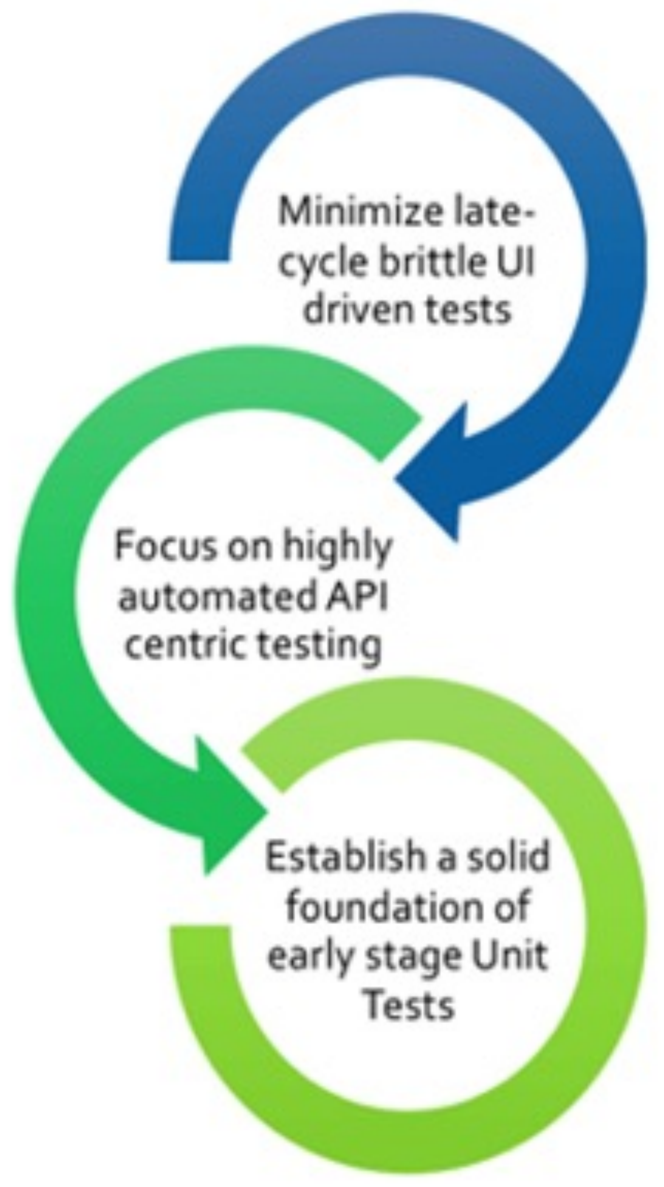
- Complexity
- Fragility
- Cost of maintenance
- Execution time
- Time to locate bug on test failure

xerris

Testing Pyramid

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 test 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

xerris