Dependency Testing: Testing the Uncontrollable

Eric Lane

PNSQC 2021
Hello. Thank you for being here.
What is Dependency Testing?

• A testing activity that validates application dependencies.
• These dependencies can be:
  • Environmental
  • Functional
  • Performative
• Dependency testing is not a type of test, it is the intent of the test.
Background

Modern Software Is:

- Complex
- Asynchronous
- Interconnected
- Highly Reliant on Web-Based Services

Each attribute adds innumerable combinations of interactions with dependencies
Approach

4 Simple Steps:

• Catalog Dependencies
• Application Refactoring
• Test Implementation
• Test Execution Triggers
Cataloging Dependencies

Discover Dependencies:

• Application Documentation
• Codebase
• Build and Deployment System

Record Dependency Information:

• Usage
• How changes are made
Application Refactoring

• Does the dependency have a service layer?
• Is that code that interacts with the dependency in a discrete class?
• Can dependency clients be configured without code change?
Test Implementation

- Determine appropriate approaches
  - Integration, Performance, Functional
- All tests map to an existing use case of the dependency
- Automated tests are tagged appropriately to be run whenever needed
Test Execution Triggers

Use a diverse array of execution triggers.

• Performance Triggers
• Change Triggers
• Scheduled Execution
An Example
Navigation Application

- Creates driving route between two addresses
- Routes users around reported traffic flow disruptions
Let’s walk through the steps:

- What are the application’s dependencies?
  - Mobile Device
  - Database
  - Map Library
- How are the dependencies integrated?
  - Can they be readily tested?
- What dependency tests do we create?
- When do the dependency tests run?
In Conclusion

• Applications rely on uncontrollable systems
• These systems can be tested to discover and mitigate errors
• Dependency testing is an activity, not a specific type of test
• Automation allows for diagnosis
Thank You