Exploratory Testing

Overview

- Today we'll look at the role manual testing plays in Software Quality Assurance
  - Specifically, we'll look at exploratory testing
- The majority of testing we have covered in this course has emphasized the benefits of automation:
  - Faster builds
  - Testing scalability
  - Test process consistency
- With exploratory testing we'll consider the benefits of manual testing and how it can be complimentary to test automation

Automated testing vs manual testing

The role of the tester

Automated Testing

Emphasis on writing code to test systems.

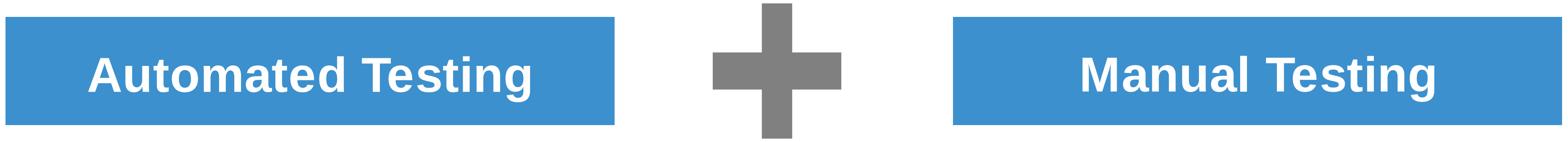
Manual Testing

Emphasis on interacting with the system under test directly by providing manual inputs through the user interface.

Automated testing + manual testing

How does manual testing complement automated testing?

- Despite our best efforts with systematic automated testing methods, bugs still make it into releases and are still found by end users
- Manual testing methods like exploratory testing can fill this gap by using real user data to manually test an application’s user interface and assess release readiness

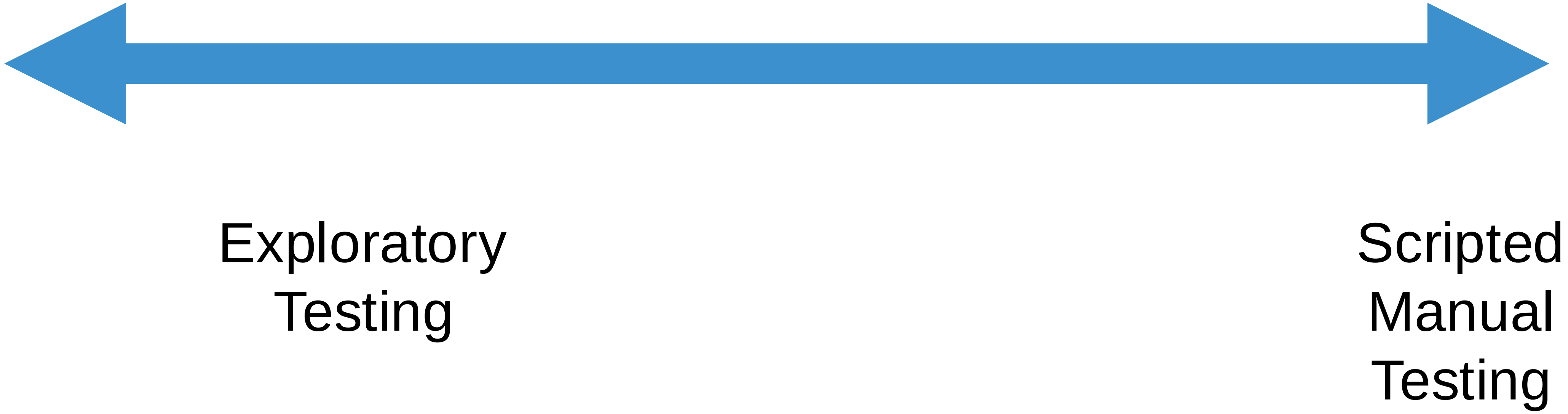


Manual Testing

"Manual testing is human-present testing. A human tester uses her brain, her fingers, and her wits to create the scenarios that will cause software either to fail or fulfill its mission... Manual testing is the best choice for finding bugs related to the underlying business logic of an application."

– James A. Whittaker, "Exploratory Software Testing"

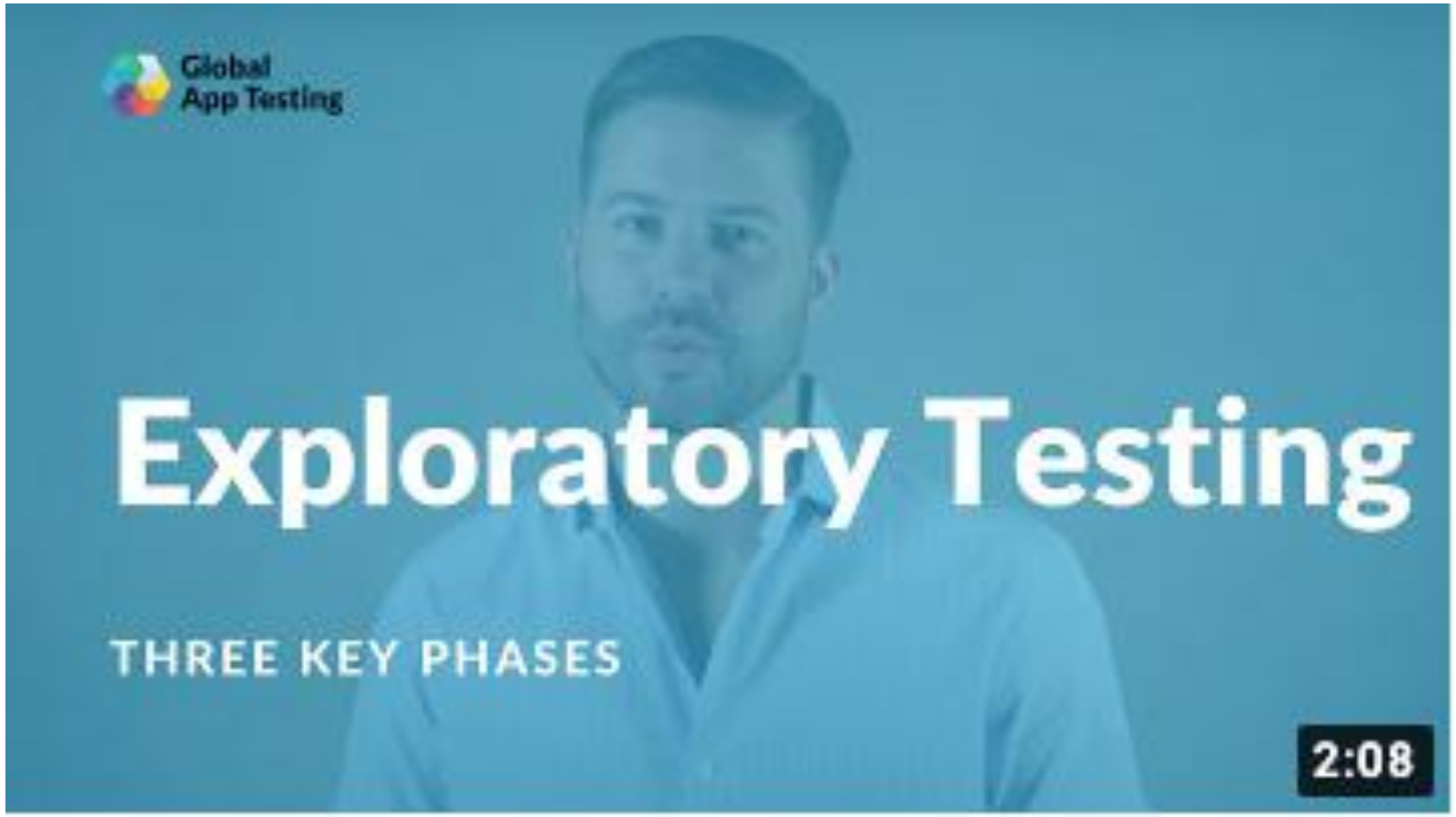
Range of Manual Testing



Exploratory Testing

- In most testing methods we have a test strategy, test plan, test case design including inputs and expected outputs.
- How do we plan and document exploratory testing?
  - Sometimes use flexible scripts
  - Use screen capture and keylogging to document tests in order to be able to rerun and reproduce bugs

The Three Phases of Exploratory Testing



https://www.youtube.com/watch?v=CJF-jCQeMys

Exploratory Testing in the Small

- **Goal: Make small decision correctly!**
- Focused on localized decisions that are made within a test.
- Need to understand how to select test inputs (+ve vs –ve input, input filters, input checks, special characters, output coverage)
- Need to test software state – data stored internally in a program (including the program counter)
  - Software state can help identify input combinations and sequences to test
- Code paths are another consideration when identifying test input
- Input data should be representative of real user data
- Consider the environment when testing

Exploratory Testing in the Large

- **Goal: Create a strategy to guide test design and application exploration**
- Focused, on how the application is explored (not on how a feature or decision in a test is made).
- Test planning shouldn't be based solely on covering features – this will potentially miss interesting interactions!
- Instead of using features as the basic unit of coverage in a test plan consider using tours – tours provide guidance and often consider interesting execution paths not covered by ad-hoc or feature-based manual testing

How I Use Exploratory Testing To Improve Mobile Apps

https://www.youtube.com/watch?v=t6RprWpphNU

Exploratory Testing

Summary

- Exploratory testing can be a complimentary technique to test automation and can help assess release readiness as well as identify bugs often found by users.

References

- Exploratory Software Testing by James A. Whittaker