CPPEN 422
Software Testing and Analysis

Manual Testing

Logistics

- **Groups are created**
- **GitHub invitations are sent**
  - **(check your email associated with your GitHub account)**
- **First assignment released!**
- **Labs are mandatory**

Testing Software

- **Manual testing**
  - fully manual, not repeatable, very ad hoc
- Automating test execution
  - Manually creating test cases
  - Automating test execution, repeatable
- Automatic test generation
  - Automatically generate test cases
  - Achieve high coverage
  - Oracle problem

Automatic Bug Detection?

- Stigma: Can the developer test?
- Respect: Automation is cool

- Oracle problem:
  - *How can we know that the software did what it was supposed to when we ran a given test case?*

Manual Testing

How would you test a given software system manually?

Manual Testing

How would you test a given software system manually?

Can we do it more systematically?

Exploratory Testing

Bug Prevention?

- Developers aiming at bug prevention?

Hard, since:

- Developer makes the worst tester
- Software at rest
- No data

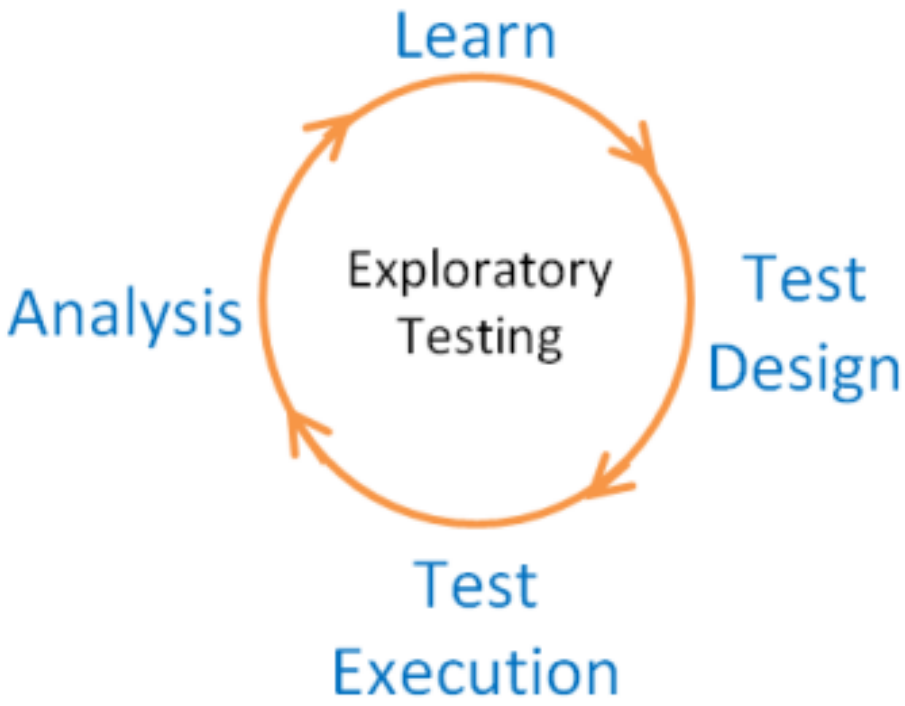
Manual Testing

- Human tester,
  - using brain, fingers, & wit
  - to create scenarios that
  - will cause software either to fail or to fulfill its mission.

- Take advantage of human cleverness
  - *No scripts:* ***Exploratory Testing***
  - Record test findings as you go

Exploratory software testing

- emphasizes the **personal freedom** and **responsibility** of the individual **tester**
- in test-related learning, test design, test execution, and test result interpretation
- runs in parallel throughout the project.


Learn
Exploratory Testing
Test Design
Test Execution
Analysis

Exploratory Testing: Goals

1. Gain an understanding of how an application works
   - Interface & functionality

2. Force the software to exhibit its capabilities

3. To find bugs before the users do!

What is Exploratory Testing not?

- Ad hoc testing
- Sloppy testing
- Careless testing
- Unstructured testing
- Undocumented testing
- Unskilled testing

Testing is all about Varying Things: (What things?)

Input:

- Input combinations
- Order of inputs
- Legal versus illegal
- Input filters & checks
- Normal versus special
- Default / user supplied

State

- History of stimuli

Paths

- Routes through system

Data

- Evolution over months

Environment

- Simulate the real world

The Touring Metaphor

- Tourist metaphor: explore Paris.
  - Strategy & goal setting
- Benefits from mix of **structure** & **freedom**
- Separate "districts" of functionality
  - Business, historical, entertainment, tourist, hotel,
  - Guidebook, money tour, landmark tour, intellectual tour (hard questions)

“Touring” Tests

Touring

- Guidebook tour
- Money tour
- Landmark tour
- Museum tour
- Rained-out tour
- Couch potato tour
- Antisocial tour

Testing

- Use the manual
- The money-generating features
- Key features
- Legacy features
- Start and then cancel operations
- Do as little as possible (all defaults)
- Known bad inputs

Charter Template

Document what you are exploring and finding

- **Charter name, time/date, duration, tester**
- **Target**
  Where are you exploring?
- **Data/Resources:**
  What data/resources are you using?
- **Test Notes:**
  Steps taken and observations (bugs, anomalies, tests)

Exercise:
Piazza Exploration

- Form groups of 2-3, around a laptop
  - 1 writes notes
  - 1 does the test execution
- Define a charter and `touring' guide
- Explore Piazza according to your charter.

Why Exploratory Testing?

- Less preparations
- Do not need complete specs or requirements
- Bugs found quickly
- Adaptable and flexible
- Creative, fun and stimulating

Limitations

- ?

Limitations of manual testing

- Black-box testing
- UI required (input/output)
- Not systematic
- Not really **measurable**
- Not easily **repeatable**