Specification Based Testing – Part 2

Exploratory Testing

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Understand role of exploratory testing

Analogy to Early Explorers

- Learn as much as possible prior to the exploration
- Develop a systematic strategy for exploring
- Keep track of where you have been
- Be observant of possible side effects
- Document findings carefully

Exploratory Testing

- Unlike scripted testing, testers explore the product and write test cases on the fly
- Tests are driven from both requirements and previous test results (continuous learning)
- There is potential to detect errors missed by scripted and automated tests

Exploratory Testing (Session Based Testing)

- **Pair of testers work together for 90 minute session**
- **Testing is focused on a charter / tour (what to test)**
  - Analogous to going on a tour in a city
  - Provides structure to exploring the system (application tour, feature tour, menu tour) while focusing on different types of errors you are looking for
- **Session Report is generated**
  - What was tested
  - Results
  - Bugs

Sample Tours

Requirements tour:

- Find all the information in the software that tells the user what the product or certain feature does. Does it explain it adequately? Do results reflect the claims made?

Complexity tour:

- Look for most complex features and data, in other words, all places where most inextricable bugs could lurk

Continuous use tour:

- Leave the system on for a prolonged period of time with multiple screens and files open. Observe what happens as disk and memory usage increase

Documentation tour:

- Tour the help section of your product and follow some instructions to see if they produce the results desired

Sample Tours

Feature tour:

- Try as many of the controls and features available on the application as possible

Inter-operability tour:

- Check if the system interacts as it should with third-party apps and whether data is shared and updated as it should

Scenario tour:

- Create a scenario (user story) that mimics the real-life interaction of a user with the system and play it out

Variability tour:

- Look for all the elements that can be changed or customized in the system and test different combinations of settings

Summary