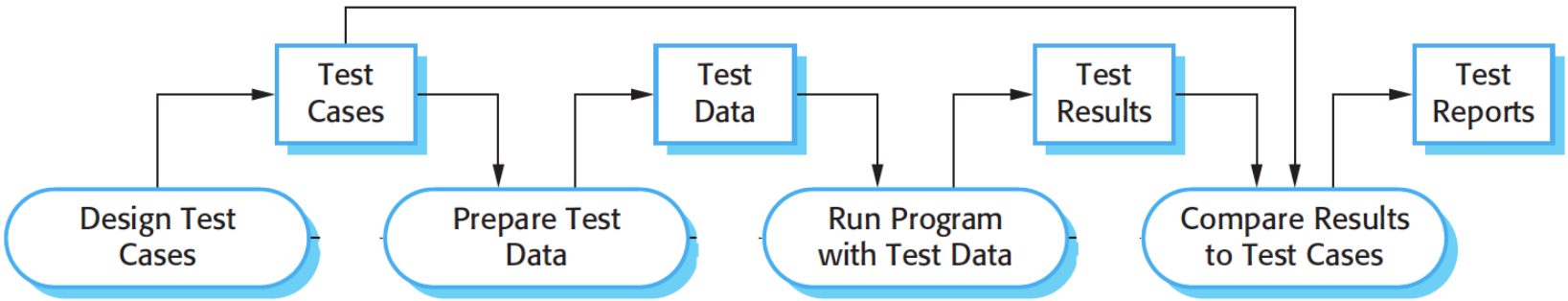
Software Testing

Testing Process

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Testing Process



Stages of Testing

- Development testing
- Release testing
  - Alpha testing
- User/Acceptance testing
  - Beta testing

Development Testing

- Development testing is primarily a defect testing process, where the aim of testing is to discover bugs in the software
- It is therefore usually interleaved with debugging – the process of locating problems with the code and changing the program to fix these problems

Development Testing

- Unit testing
- Component Testing
- System Testing

Unit Testing

- The process of testing program components
- Individual functions or methods are the simplest type of component
- Call these routines with different input parameters

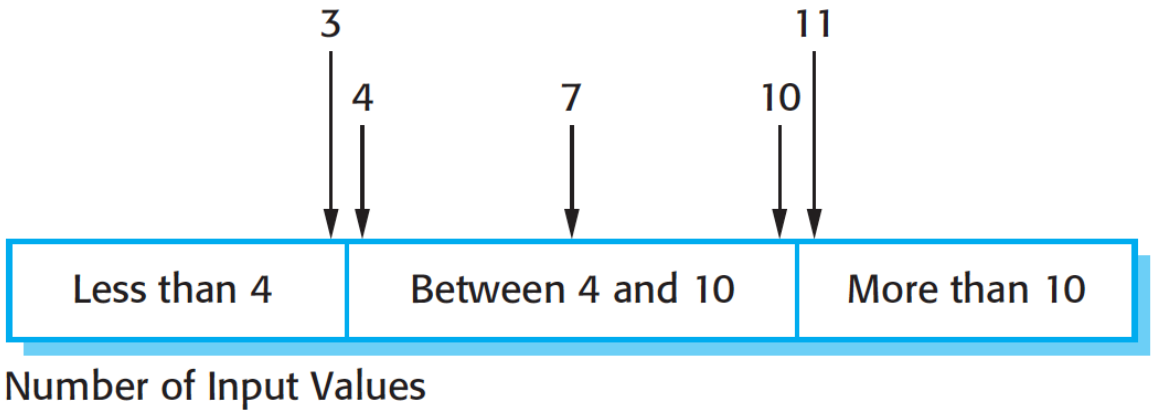
Choosing Unit Test Cases

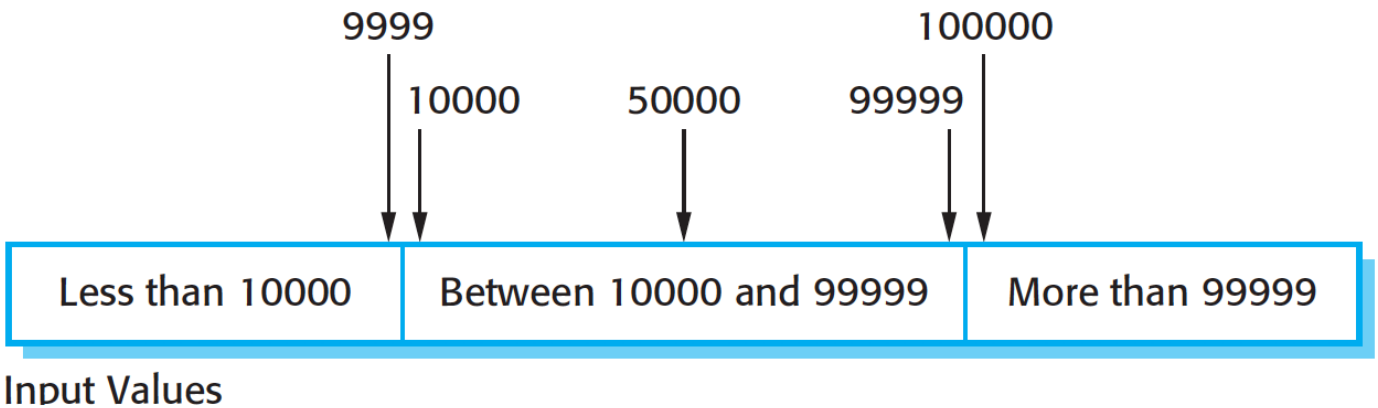
- **Guideline-based testing**
  - Use testing guidelines to choose test cases
  - Reflect previous experience of the kinds of errors that programmers often make when developing components
- **Partition testing**
  - Identify groups of inputs that have common characteristics and should be processed in the same way
  - Choose tests from within each of these groups

Example: Guidelines

- Choose inputs that force the system to generate all error messages
- Design inputs that cause input buffers to overflow
- Force invalid outputs to be generated
- Force computation results to be too large or too small

Example: Partition Testing





Black-Box Vs. White-Box Testing

- **Black-box testing**
  - Use the specification of a system to identify equivalence partitions
  - Don't need any knowledge of how the system works
- **White-box testing**
  - Look at the code of the program to find other possible tests
  - For example, your code may include exceptions to handle incorrect inputs
    - You can use this knowledge to identify 'exception partitions' —different ranges where the same exception handling should be applied

Component Testing

- Focuses on showing that the component interface behaves according to its specification
- Assumes that unit tests on the individual objects within the component have been completed.
- The test cases are not applied to the individual components but rather to the interface of the composite component created by combining these components

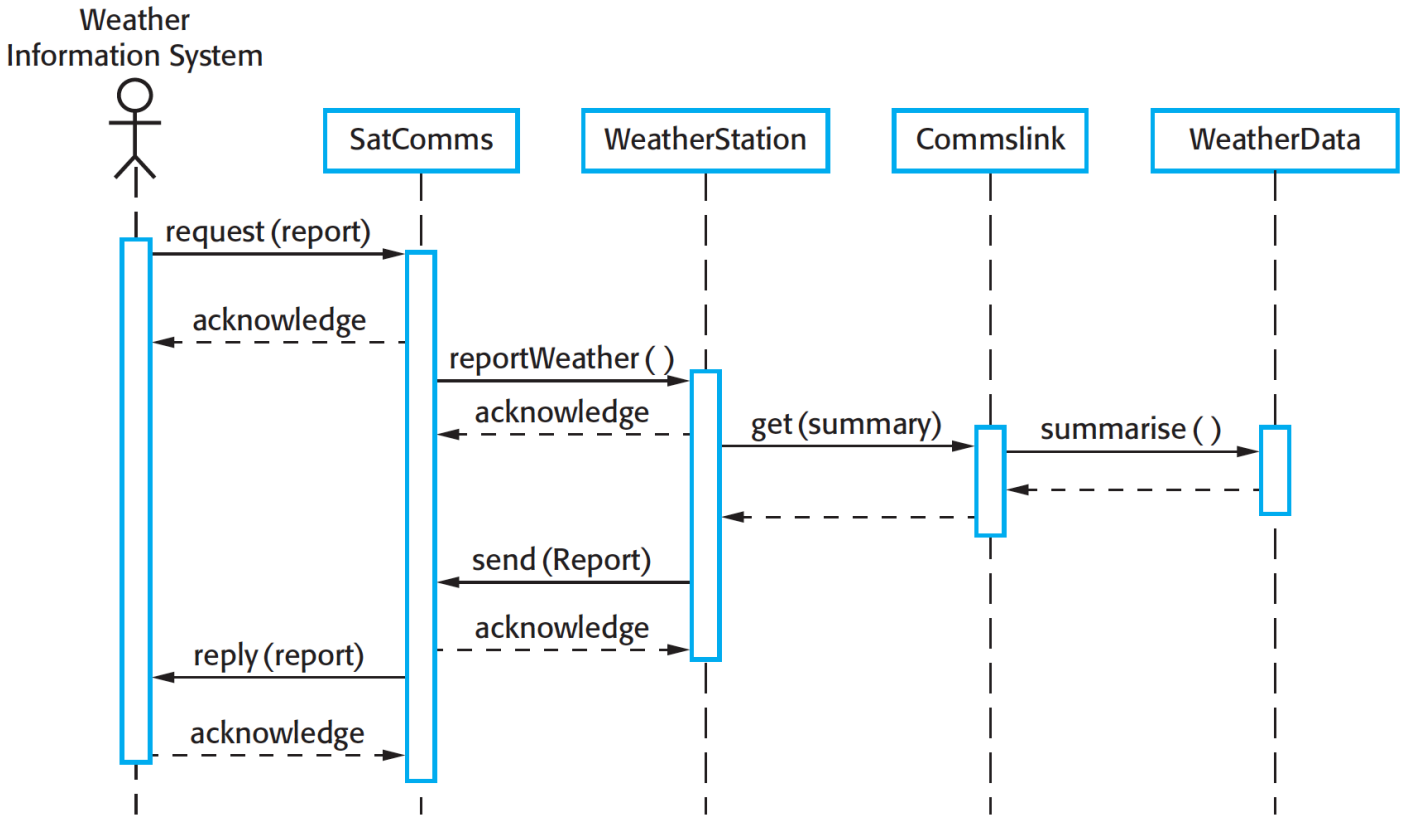
Component Testing Guidelines

- Design a set of tests in which the values of the parameters to the external components are at the extreme ends of their ranges
- Where pointers are passed across an interface, always test the interface with null pointer parameters
- Where a component is called through a procedural interface, design tests that deliberately cause the component to fail
- Use stress testing in message passing systems
- Where several components interact through shared memory, design tests that vary the order in which these components are activated

System Testing

- Check that the system meets the requirements of all of the system stakeholders
- Focus on testing the interactions between the components and objects that make up a system.
- Test performance, reliability
- Checks that the system does not do things that it shouldn't do, such as produce unwanted outputs

Example: System Testing



Test case:

SatComms:request –> WeatherStation:reportWeather –> Commslink:Get(summary) –> WeatherData:summarize

Software Testing

Testing Process

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur