

# Good Software

## External Measures

Usability

Correctness

Fitness for purpose

Performance/efficiency

Market value

Conformance to specifications

Reliability

Compatibility

## Internal Measures

Readability

Maintainability

Well documented

Testability

Integrity/consistency

Portability

Re-usability

## Management's Measures

Return on investment

# Types of Software

- Commercial off-the-shelf software
  - Individuals can buy a copy
- Custom
  - Created for a specific use by an organization that's not publicly available
- Open Source
  - Freely available, including the source code
  - Developed by communities

- Human interactive software

  - People use it

- Embedded software

  - Single user

  - Multi user

# Testing

- How do we know if software works correctly?

- Test it!

- "Testing shows the presence, not the absence, of bugs" - Edsger Dijkstra

- Why test?

- Increase confidence that code is correct

- Catch regression errors

  - Code maintenance might break existing features

  - Tests help ensure that what used to be working still works after updates

- What to test

- Devise test cases

- Boundary conditions (also called edge cases)

- What values are on the edge of different cases?

- Here is where errors often occur

- Unit tests

- Test individual functions and classes rather than complete programs

- Tests can be written as the functions and classes are written, allowing early debugging

## Test - Driven Development

- Write tests before code
- Forces you to think carefully about behavior of a unit before writing it
- Lets you run code as soon as you write it