Testing

- Automated Testing
- JUnit
- Test-Driven Development
- Test Coverage
- Integration Tests

Manual Testing

- The most basic form of testing
- Run the program yourself
- When an error occurs, write it down
- Ensure that the error can be reproduced

Issues with Manual Testing

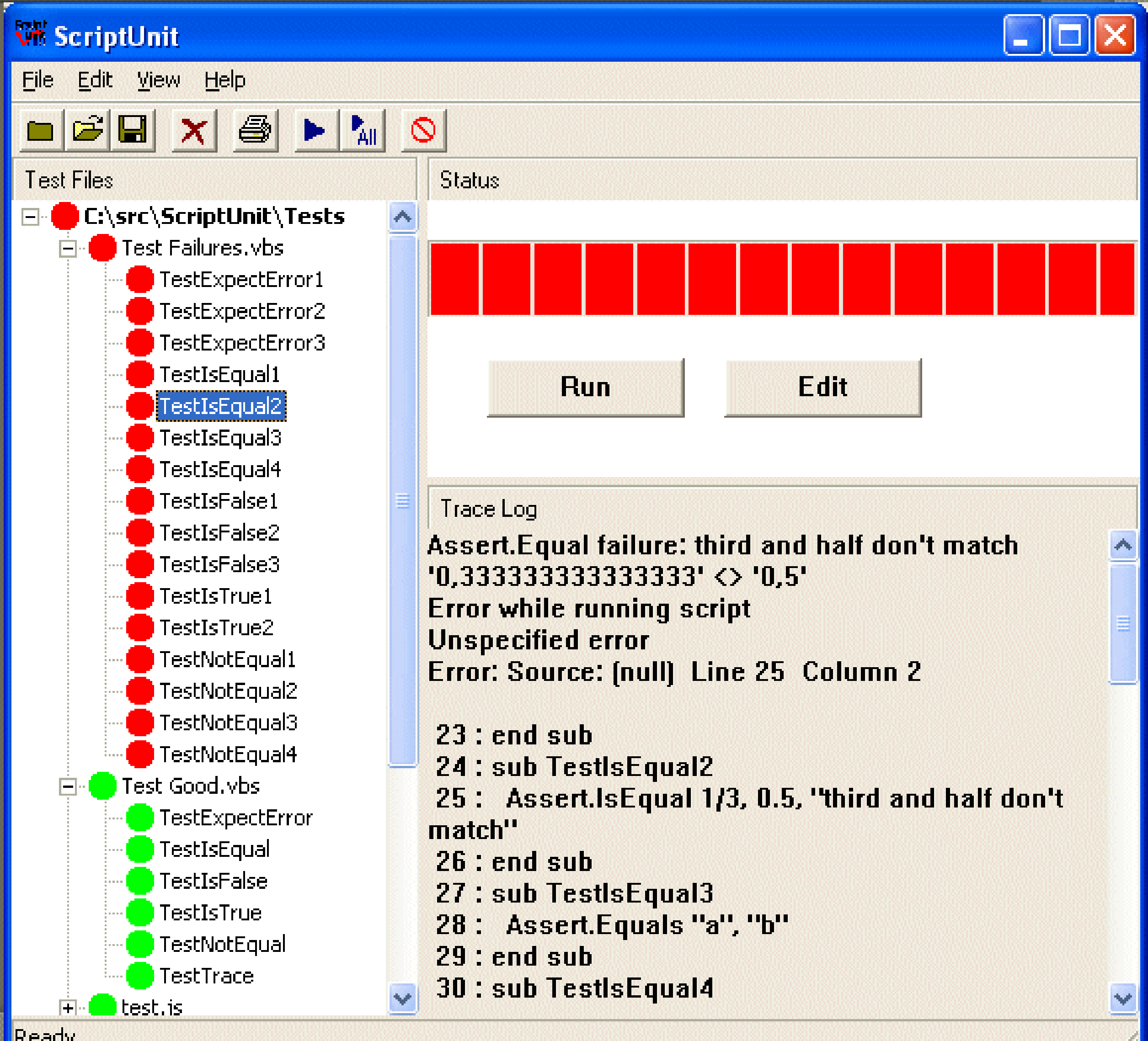
- To be effective, bugs must be tracked
- Bug reproduction steps must be carefully retained
- Have to re-test after changes
- Expensive and error-prone

Automated Testing

- Manual testing is still necessary for sanity checks
- But we can create *automated tests* to provide immediate feedback
- Tests are run for each important build
- We receive a checklist afterwards seeing how many, and which, tests succeeded and failed

Test Framework

- Tests are grouped into cases
- Cases are grouped into units



- ● C:\src\ScriptUnit\Tests
  - ● Test Failures.vbs
    - ● TestExpectError1
    - ● TestExpectError2
    - ● TestExpectError3
    - ● TestIsEqual1
    - ● TestIsEqual2
    - ● TestIsEqual3
    - ● TestIsEqual4
    - ● TestIsFalse1
    - ● TestIsFalse2
    - ● TestIsFalse3
    - ● TestIsTrue1
    - ● TestIsTrue2
    - ● TestNotEqual1
    - ● TestNotEqual2
    - ● TestNotEqual3
    - ● TestNotEqual4
  - ● Test Good.vbs
    - ● TestExpectError
    - ● TestIsEqual
    - ● TestIsFalse
    - ● TestIsTrue
    - ● TestNotEqual
    - ● TestTrace
  - ● test.js

```
Assert.Equal failure: third and half don't match
'0,333333333333333' <> '0,5'
Error while running script
Unspecified error
Error: Source: (null)  Line 25  Column 2

 23 : end sub
 24 : sub TestIsEqual2
 25 :   Assert.IsEqual 1/3, 0.5, "third and half don't
match"
 26 : end sub
 27 : sub TestIsEqual3
 28 :   Assert.Equals "a", "b"
 29 : end sub
 30 : sub TestIsEqual4
```

Unit Testing

- The most common kind of automated testing is Unit Testing
- Unit testing is a form of automated testing where you test a single class, module, or method
- A unit is the smallest testable portion of an application:
  - Each unit test tests one thing
  - We can test every function of the unit, and every meaningful case of the function

Regressions

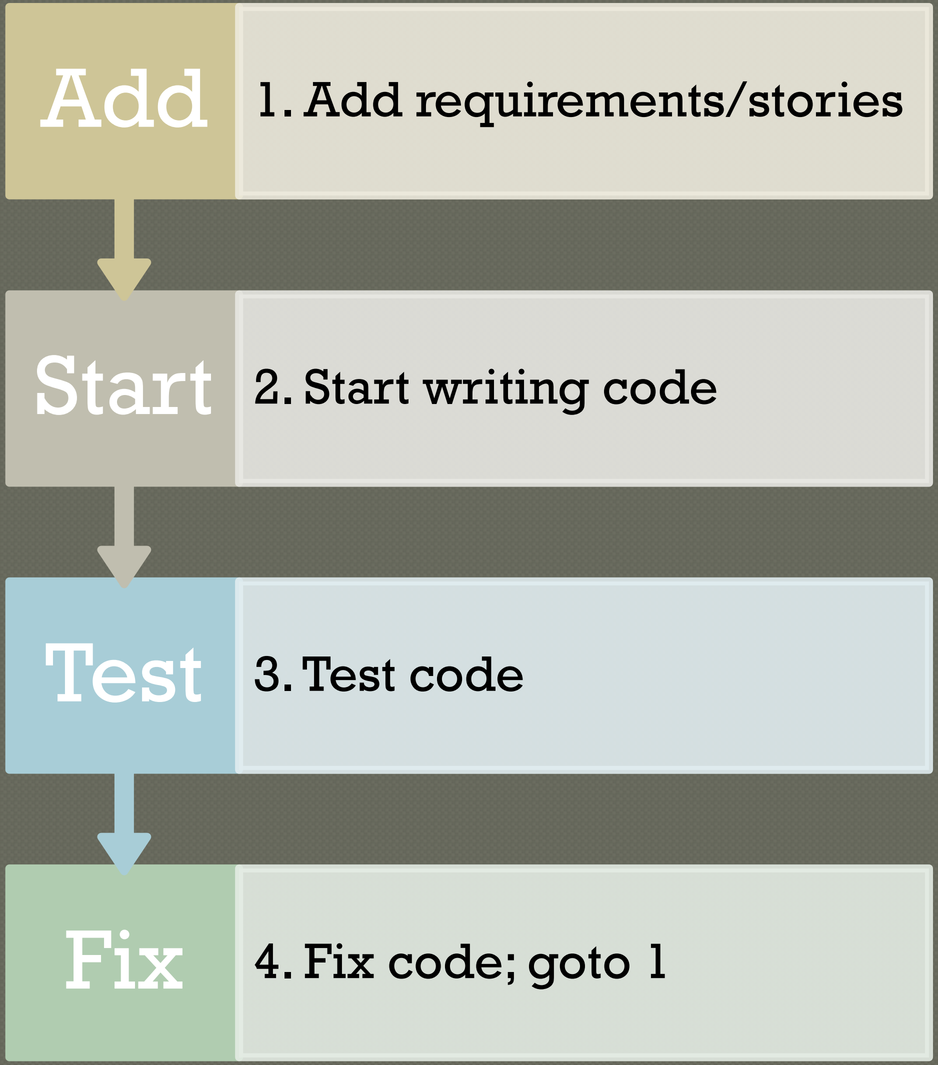
- It is extremely common for a new update to fix old bugs, but introduce new ones
- If a test passes, but then we update and it fails, this is called a regression
- Automated testing provides a mechanism to quickly discover regressions

Be Careful

- The natural instinct is to rely on automated testing
- It is NOT a replacement for manual testing
- And manual testing is not a replacement for code review

- In studies, code review catches more difficult-to-detect bugs than testing!
  - Juristo, N., & Vegas, S. (2003). *Functional testing, structural testing and code reading: What fault type do they each detect?*. In Empirical Methods and Studies in Software Engineering (pp. 208-232). Springer, Berlin, Heidelberg.

Normal Process



Add

1. Add requirements/stories

Start

2. Start writing code

Test

3. Test code

Fix

4. Fix code; goto 1