HTTP 5221

Security & Usability

Sean Doyle

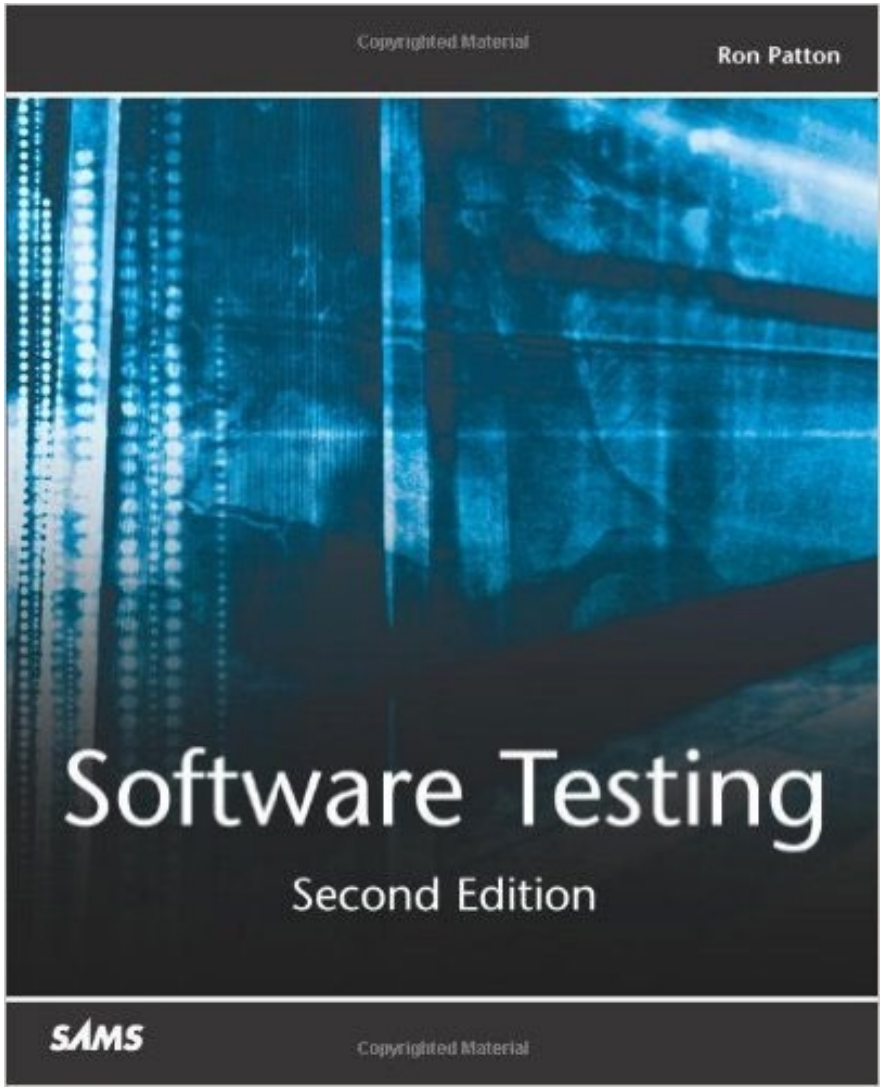
How do we minimize risk?

- Plan it
- Build it
- Test it
  - **Unit testing through to user testing**
- Protect it

TESTING YOUR CODE

Lesson 3

Source Book

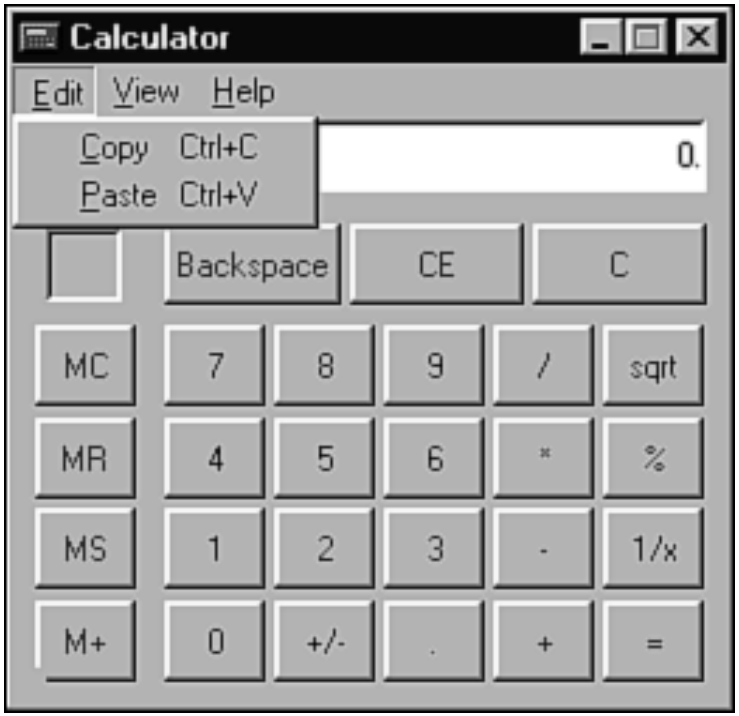

Copyrighted Material
Ron Patton
Software Testing
Second Edition
SAMS
Copyrighted Material

CHECKING FOR…

Does it do what is supposed to do?

Is there a way to make it do what it shouldn’t?

Striking a Balance



Test Implementation = Risk Management

Image source: Software Testing by Ron Patton

CHECK THE SPECS

A bug occurs when one of the following five rules is true:

- The software doesn't do something that the spec says it should do.
- The software does something that the spec says it shouldn't.
- The software does something that the spec doesn't mention.
- The software doesn't do something that the spec doesn't mention but should.
- The software is difficult to understand, hard to use, slow, or – in the software tester's eyes – will be viewed by the end user as just plain not right.

Preventing Bugs

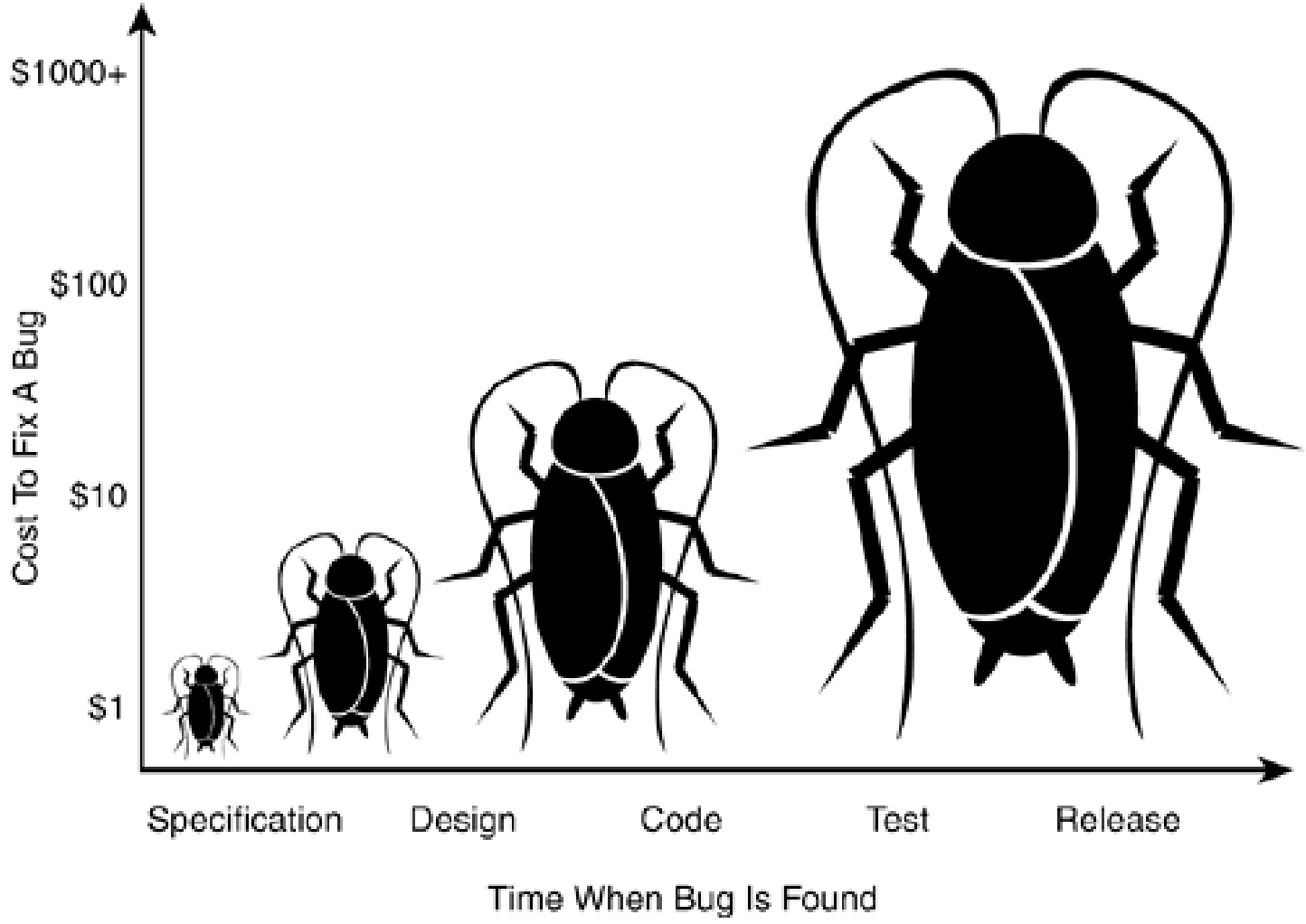


Image source: Software Testing by Ron Patton

THE GOAL OF SOFTWARE TESTING…

“The goal of a software tester is to find bugs, find them as early as possible, and make sure they get fixed.”

(Patton, p. 19)

Sources of Bugs

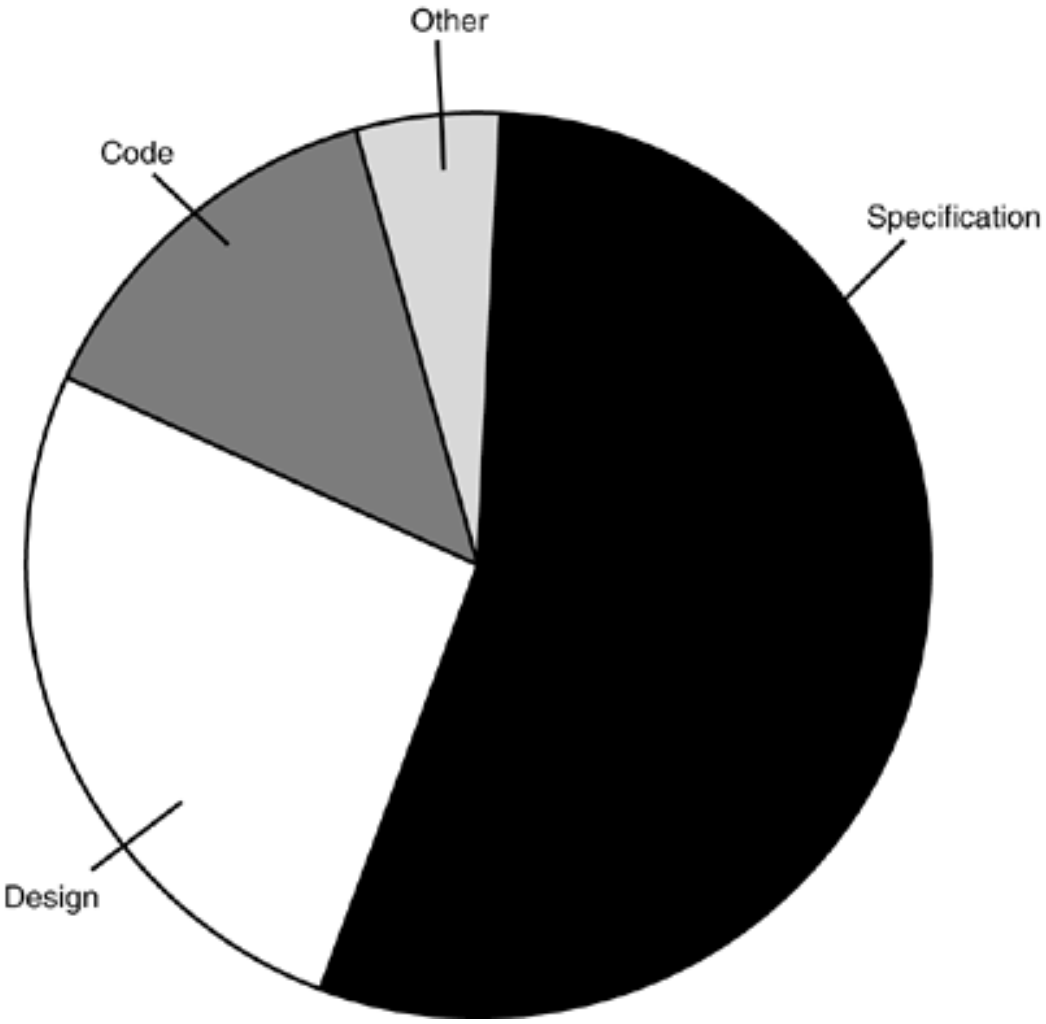


Image source: Software Testing by Ron Patton

CHECK THE SPECS FOR BUGS!

Words to watch for in a specification:

- **Always, Every, All, None, Never**
- **Certainly, Therefore, Clearly, Obviously, Evidently.**
- **Some, Sometimes, Often, Usually, Ordinarily, Customarily, Most, Mostly**
- **Etc., And So Forth, And So On, Such As**
- **Good, Fast, Cheap, Efficient, Small, Stable**
- **Handled, Processed, Rejected, Skipped, Eliminated**
- **If…Then…(but missing Else)**

TYPES OF TESTING

Static Testing & Dynamic Testing

- Dynamic testing is interacting with the application.
- Static testing does not involve interacting with the application. In software testing, this usually means reviewing the code. In a web context, it can mean testing the layout, checking responsive stages, or evaluating load times.

Black Box Testing

- Sometimes referred to as *functional testing* or *behavioral testing*
- *The tester cannot see the code, they only know what it is supposed to do.*

White Box Testing

- The tester has access to the code.
- The tester can conduct the testing with values that they know will push the limits (or break) the application.

Test-to-Pass & Test-to-Fail

- **Test-to-Pass** means you are testing the application with values that are known to be within the operational expectations and limits.
- **Test-to-Fail** means you are deliberately trying to 'break' the application to find bugs that occur outside of the expected boundaries.

Boundary Conditions

- Testing boundary conditions means you are testing just inside and outside the minimum and maximum expected values.

```
Enter the number of the month (1-12):
```

0 1 2 3 4 5 6 7 8 9 10 11 12 13

Unit Testing

- Unit Testing involves testing your application on a per function level.
- You create functions that will invoke and test your application functions.
- Pass in a set value and state what the result should be, then run the function and see if the result matches the expectation.

```
function myUnitTest(valueToTest, expectedResult) {
  //call function I want to test with valueToTest
  //did result match expectedResult?
}
```

Unit Testing Pattern

To create a unit test for a function called myFunction...

```
function myFunction_Test(valueToTest, expected) {
    //call function I want to test with valueToTest
    let result = myFunction(valueToTest);

    //did result match expected?
    if (result === expected) {
      return true;
    } else {
      return false;
    }
}//end myFunction_Test
```

Test-Driven Development

- An approach to Unit Testing where you create the test before you create any code.

Terminology Review

- Specification
- Bug
- Static Testing
- Dynamic Testing
- Black Box Testing
- White Box Testing
- Test-to-Pass
- Test-to-Fail
- Boundary Conditions
- Unit Testing
- Test-Driven Development

HTTP 5221

Security & Usability

Sean Doyle