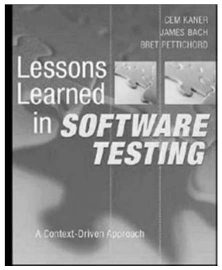
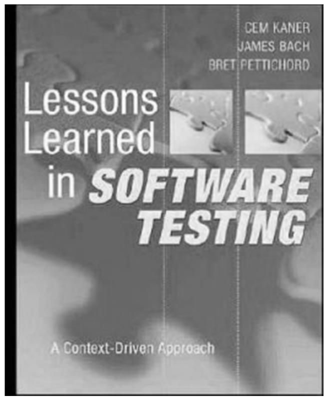

CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

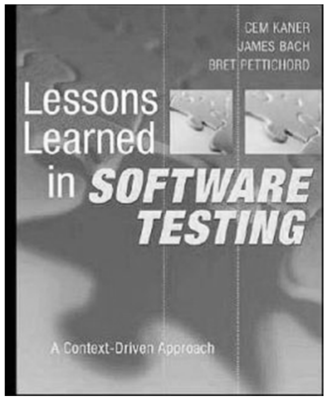
Theme 4: Reporting Bugs and Working with Others

Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

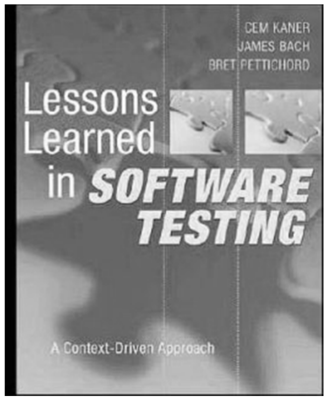
- Lesson 55: "You are what you write"
  - Bug reports are the main "product" of testers
  - **Bug reports::testers** as **source code::developers**
    - In heavily automated testing, your test code may also be a critical product, but it had better contribute to bug reports at some point
  - (Combining points from some other lessons)
    - You need to effectively make the case that *this* bug is worth giving up resources (money, programmer time, other development or bug fixing) to fix; you are the bug's *champion*
    - Be an honest champion!

Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

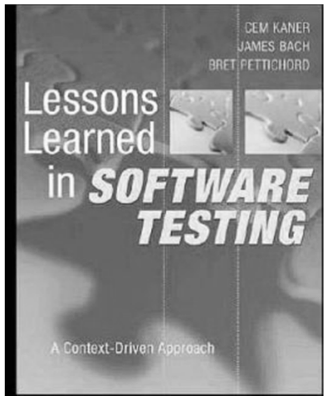
- Contents of a bug report (minimal)
  - Unique ID (name/number)
  - **What is the bug?**
  - How do you make the bug happen (BE SPECIFIC)?
    - If you have code that always produces the bug, include it!
    - If you can minimize (remember delta debugging?) do so
  - What version of the software was this detected on?
  - What is the estimated severity of the bug?
  - What is the estimated priority of the bug?

Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

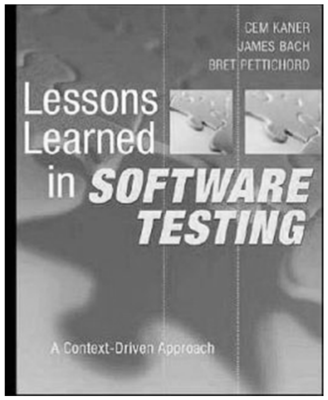
- Lesson 59: "Take the time to make your bug reports valuable"
  - Bug reports are the main "product" of testers
  - **Bug reports::testers** as **source code::developers**
    - In heavily automated testing, your test code may also be a critical product, but it had better contribute to bug reports at some point
  - If your reports aren't understandable and informative, this is like producing bad, buggy code

Reporting Bugs and…

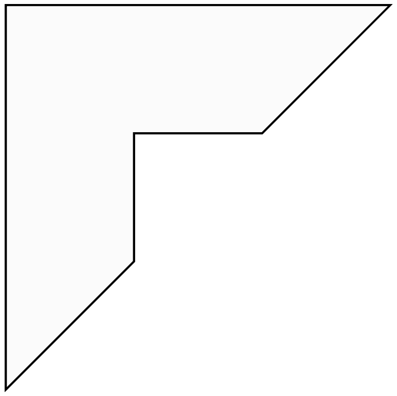

CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

- Lesson 68: “Never assume that an obvious bug has already been filed”
  - Everyone may make this assumption…
    - And the bug will never get filed!

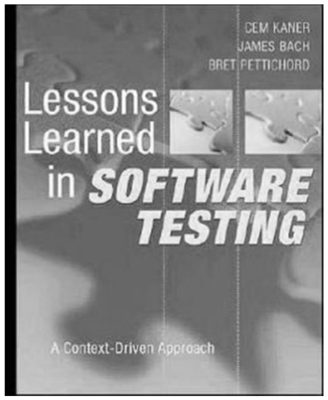
Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

- Lesson 71: “Uncorner your corner cases”
  - Programmers can sometimes ignore a test case that relies on particularly “odd” data:
    - You may try corner cases first since they are likely to fail
    - Once you find a bug, make sure you can’t reproduce it with a simpler/less weird input
      - If you can, report that version instead!

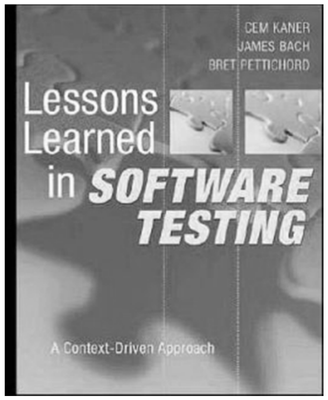


Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

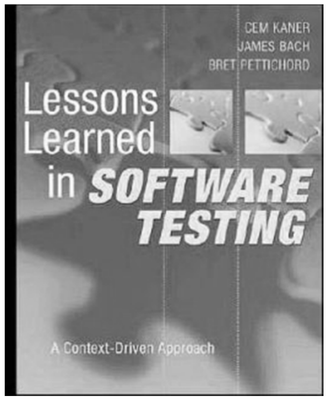
- Lesson 73: "Keep clear the difference between severity and priority"
  - *Severity* is about the impact of a bug
    - Severity is about worst-case scenarios, probabilities, risks
    - Examples of high severity bugs: security compromises, incorrect results used in financial calculations, bugs that stop all testing
  - *Priority* is about how soon a bug should be fixed
    - Changes with time and circumstances
  - High severity isn't always high priority:
    - If a bug corrupts any file saved in July 2010 only, it may not be important to fix
  - High priority isn't always high severity:
    - Misspelling the company's name

Reporting Bugs and…


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

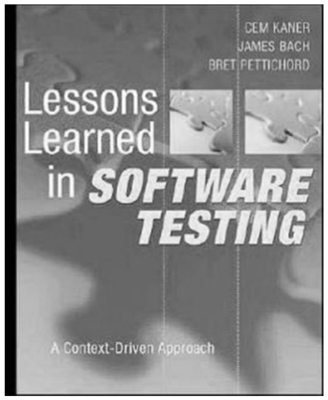
- Lesson 82: “Every bug deserves its own report”
- Lesson 83: “The summary line is the most important line in the bug report”
- Lesson 86: “Be careful of your tone. Every person you criticize will see the report”

Working with Others


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

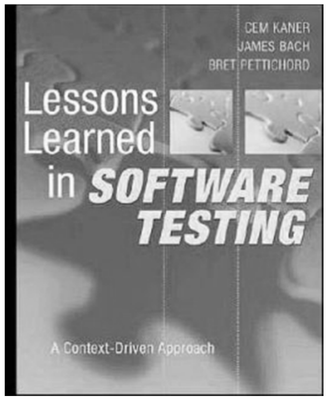
- Lesson 92: "The best approach may be to demonstrate your bugs to the programmers"
  - *Seeing is believing*
    - Don't interrupt!
    - Doesn't remove need for a written report, but can make initial report much better

Working with Others


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

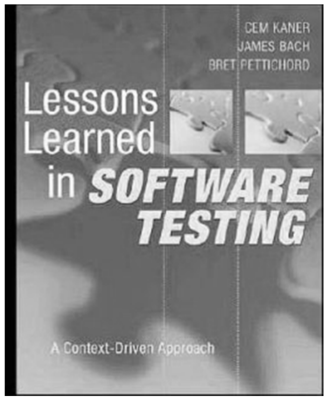
- Lesson 150: “Understand how programmers think”
  - Programmers tend to specialize
    - They often do not know the big picture very well
    - As a tester that may be your job
  - Programmers have a theory of the system
    - Report bugs in terms of programmers own models
  - Programmers often hate routine
    - They may think non-automated tests are “lame” or “wrong”

Working with Others


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

- Lesson 154: "Focus on the work, not the person"
  - Talk about the code and its bugs, not whether John Q. Programmer is a screw-up
    - Maybe he is, but that's not your job
    - Testing is not a management position, usually

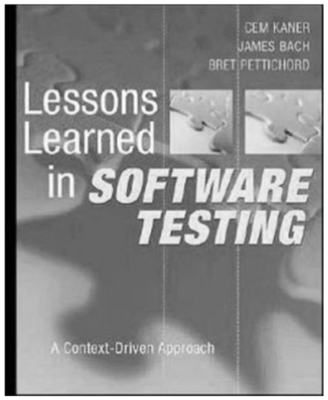
Working with Others


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

- Lesson 169: "Ask for testability features"
  - Code is not always as easy to test as it could be
    - If you don't ask, programmers won't think much about this aspect of coding
    - If you do ask, the worst that can happen is "no"

    - Programmers are often happy to make your job easier

Working with Others


CEM KANER
JAMES BACH
BRET PETTICHORD
Lessons Learned in SOFTWARE TESTING
A Context-Driven Approach

- Lesson 181: "Programmers are like tornadoes"
  - Programmers will do what they will do
  - At some companies that will be great
  - At other places, it may be a problem

  - You cannot solve the testing problem by declaring that programmers "can't act that way"
    - In the Midwest houses have basements because: tornadoes
    - Cannot get away with no basement by declaring tornadoes unreasonable