

Software Testing Pitfalls

Yegor Bugayenko

JPoint 2019

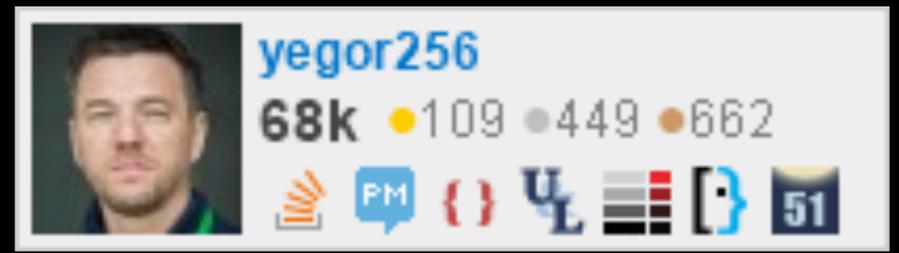
Moscow, Russia

5 April, 2019



@bloghacks





Elegant Objects

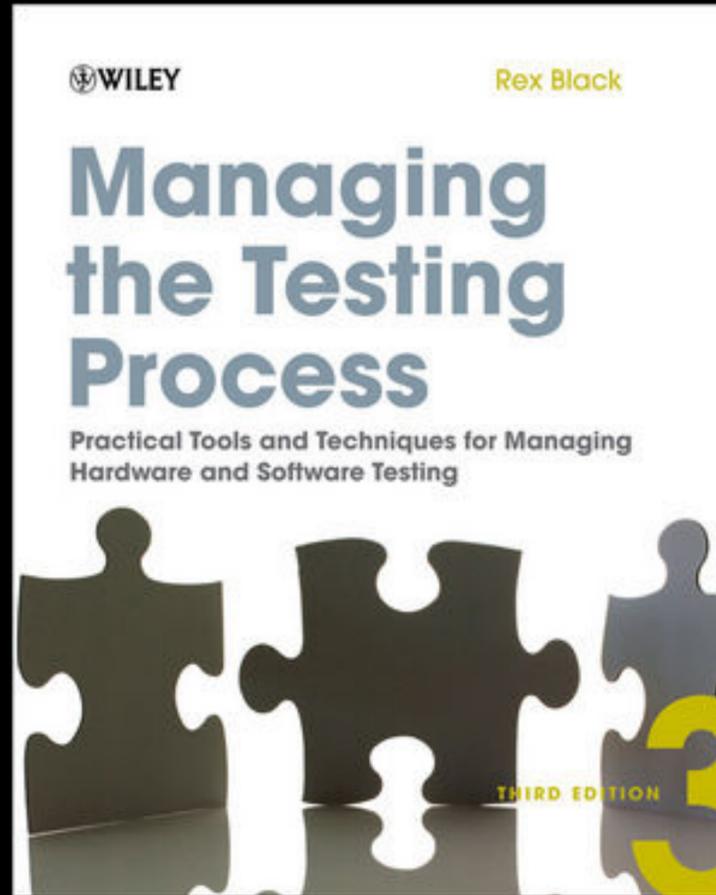


Quality



$$Q = \frac{F}{F + U}$$





Rex Black, 2009

Defect Detection Effectiveness



Software Defect Removal Efficiency

By

Capers Jones, President
Capers Jones & Associates LLC
Email: CJonesiii@cs.com

Abstract

The most important contributor to the quality of software-intensive systems is the quality of the software components. The most important single metric for software quality is that of defect removal efficiency (DRE). The DRE metric measures the percentage of bugs or defects found and removed prior to delivery of the software. The current U.S. average in 2011 is only about 85% of total defects removed. However, best in class projects can top 99% in defect removal efficiency. High levels of DRE cannot be achieved using testing alone. Pre-test inspections and static analysis are necessary to top 95% in defect removal efficiency.

Copyright© 2011 by Capers Jones & Associates LLC. All rights reserved.

Introduction

In the 1970's the author worked for IBM. Software applications were growing larger and more complex so quality was becoming a serious issue. IBM began a careful analysis of software quality. Measurements were taken of defects found in software requirements, design documents, source code, user manuals, and also "bad fixes" or secondary defects accidentally included in defect repairs.

At the same time IBM developed the function point metric, because it was necessary to analyze non-coding defects and non-coding development activities as well. After several years of data collection, it was possible to determine the relative contribution of various defect origins on total software defects. The total number of defects from all five sources was termed the "defect potential" of a software application.

Table 1 shows approximate U.S. averages from more than 13,000 projects. Table 1 shows the average volumes of defects found on software projects, and the average percentage of defects removed prior to delivery to customers:

Table 1: Defect Removal Efficiency by Origin of Defects Circa 2011
(Data Expressed in Terms of Defects per Function Point)

Defect Origins	Defect Potentials	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23



Capers Jones, 1996 Defect Removal Efficiency





Programmers



Testers



Good

Code

“A good programmer will

use for the code, programmer will

miss Bugs

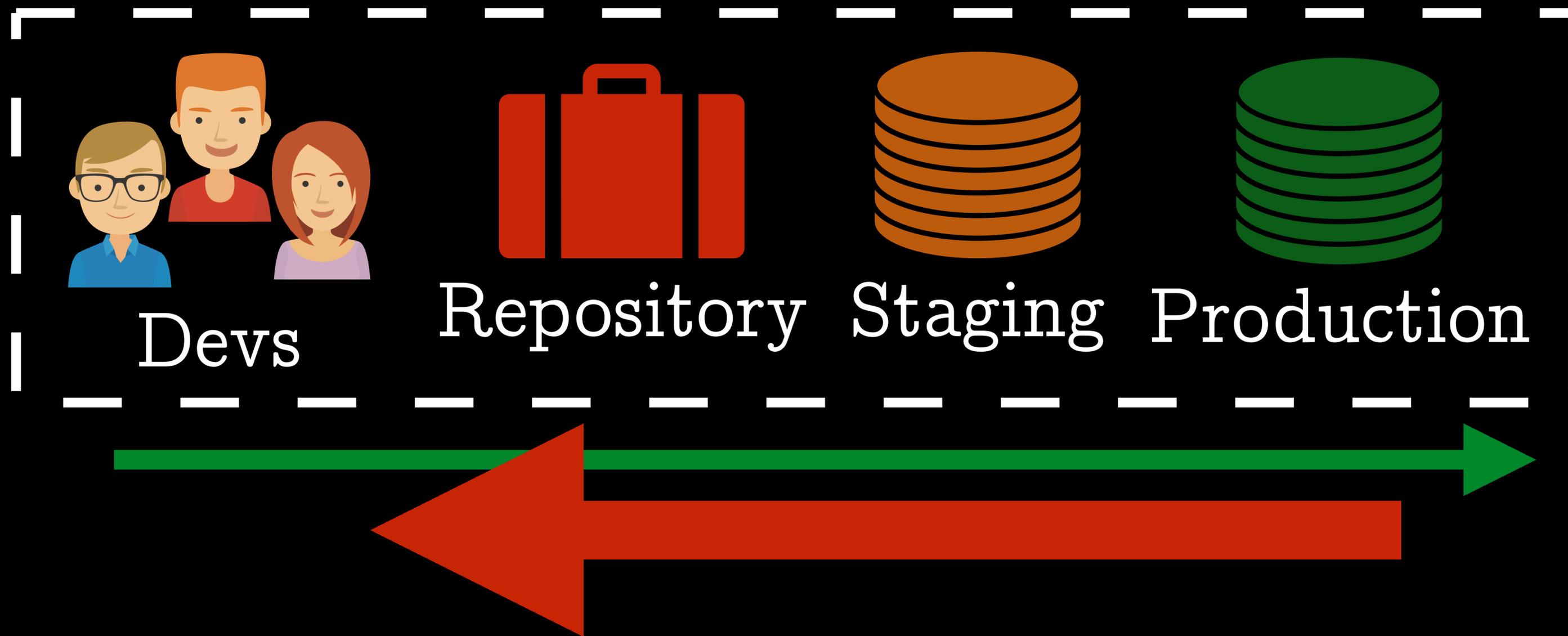
Wrong!



JOHN C. MUNSON, PH.D.



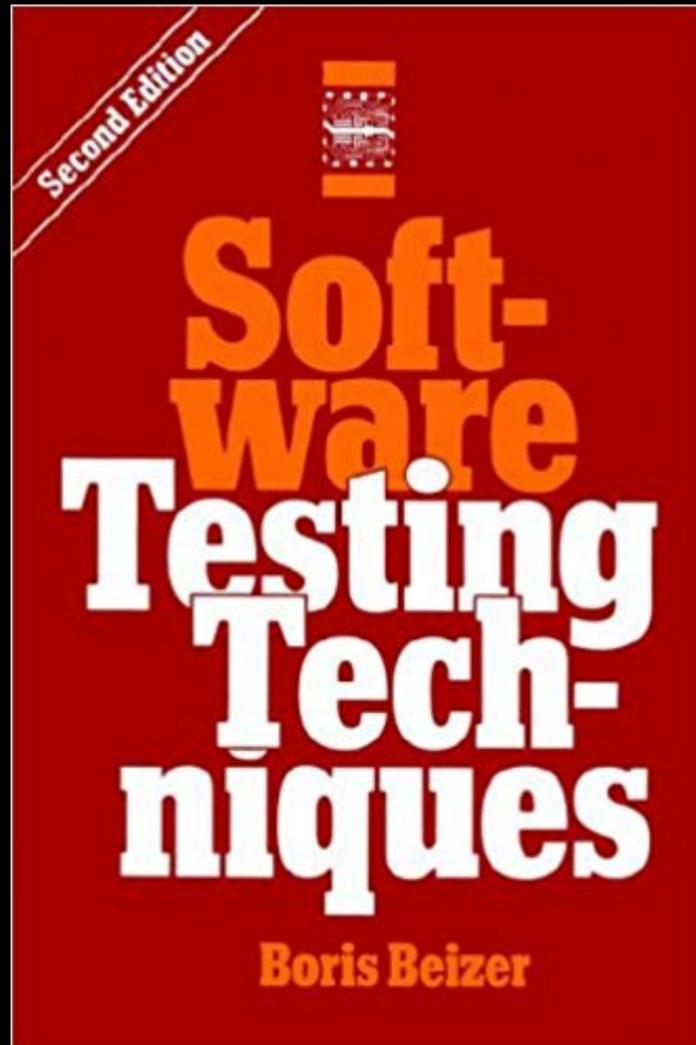
Pipeline





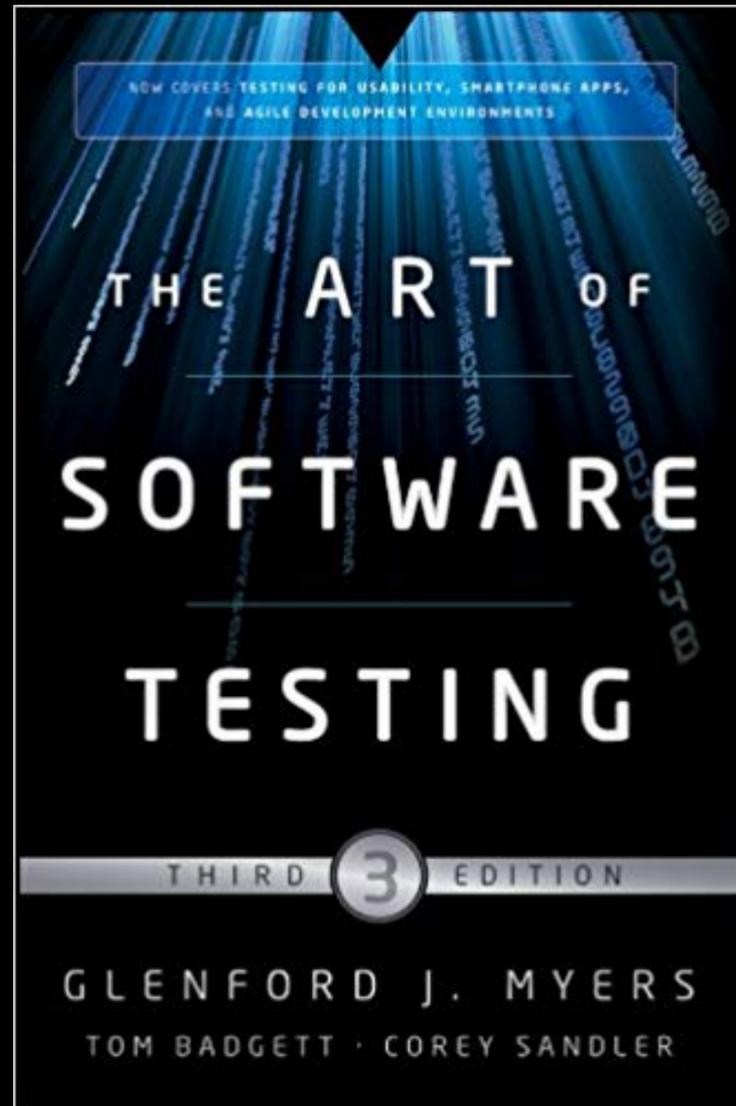
James A. Whittaker: “Software testing is the process of executing a software system to determine whether it matches its specification and executes in its intended environment.”





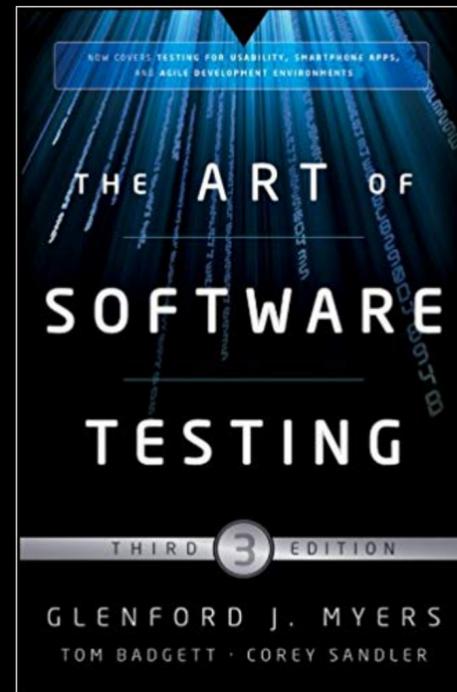
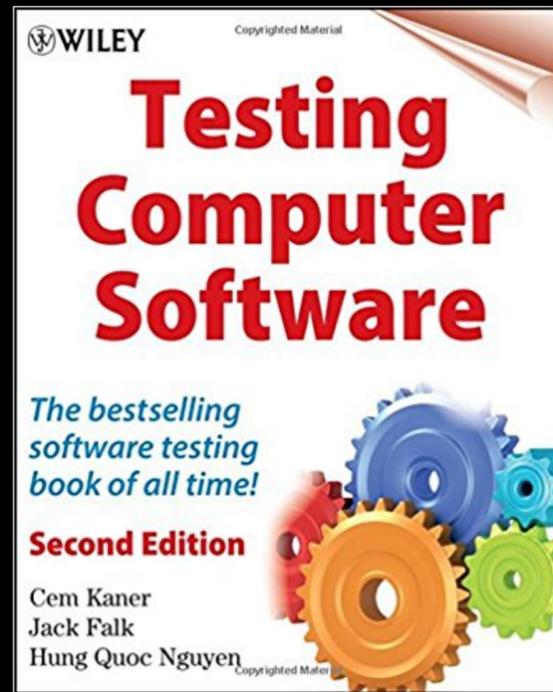
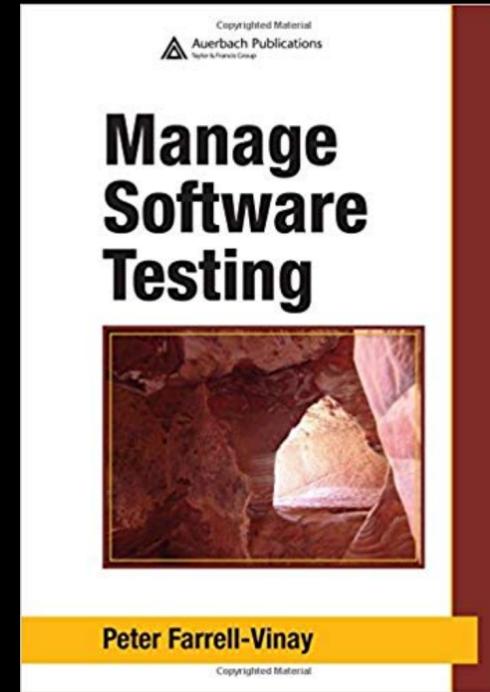
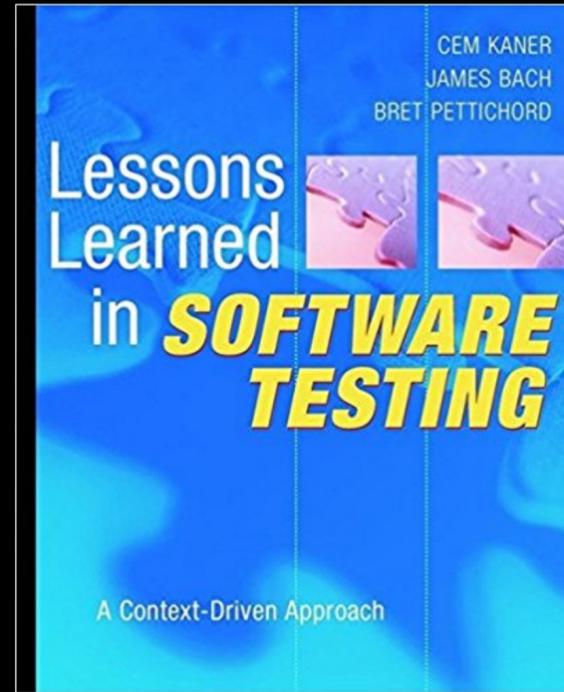
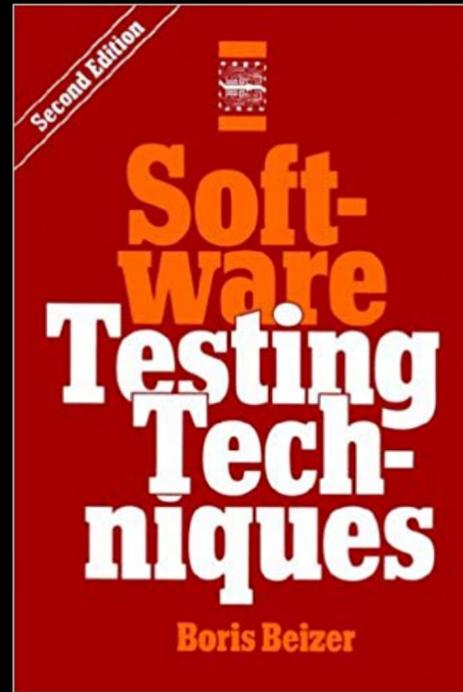
Boris Beizer: “If your objective is to demonstrate a high probability of working, that objective is best achieved by not testing at all!”





Glenford Myers: “despite the plethora of software testing tomes available on the market today, many developers seem to have an attitude that is counter to extensive testing. Testing is the process of executing a program with the intent of finding errors.”

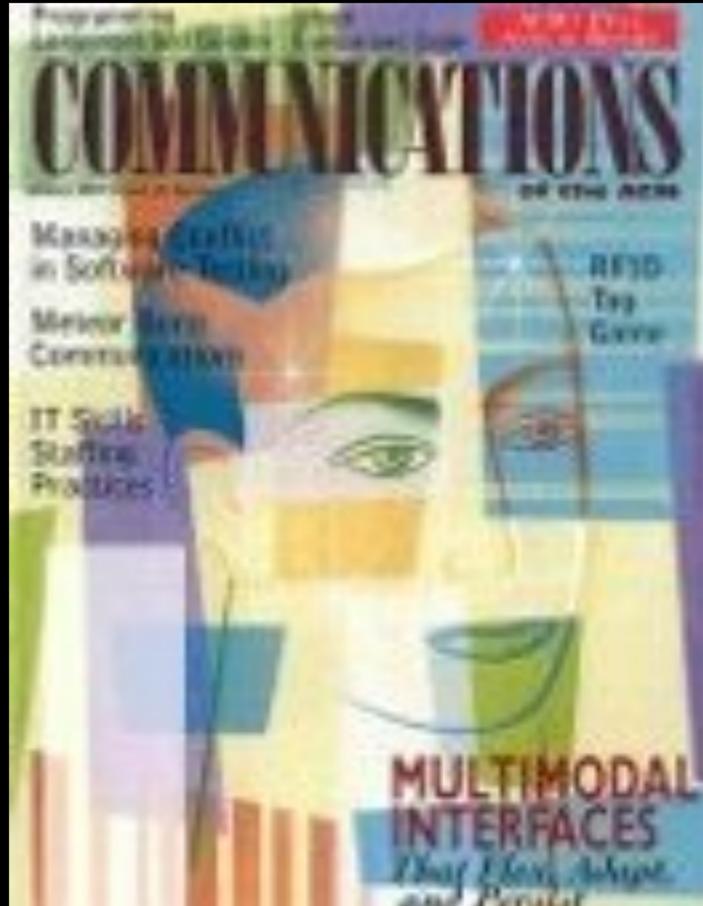






Testers are not
second-class citizens.

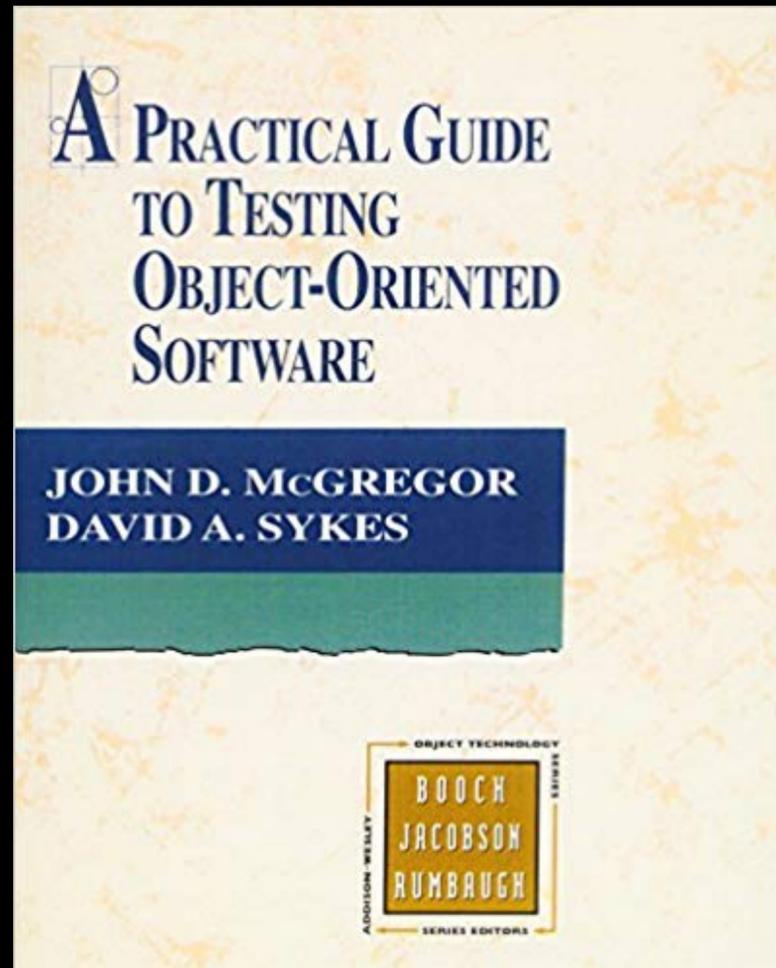




Cynthia Cohen: “The lack of status and support makes the tester’s job more difficult and time consuming, as the struggle for recognition becomes part of the job itself”

Managing Conflict in Software Testing, Communications of the ACM, Volume 47, Issue 1, 2004





John D. McGregor: “Being a good tester is harder than being a good developer because testing requires not only a very good understanding of the development process and its products, but it also demands an ability to anticipate likely faults and errors.”





Salaries



Seniority



More!



Tester

Senior

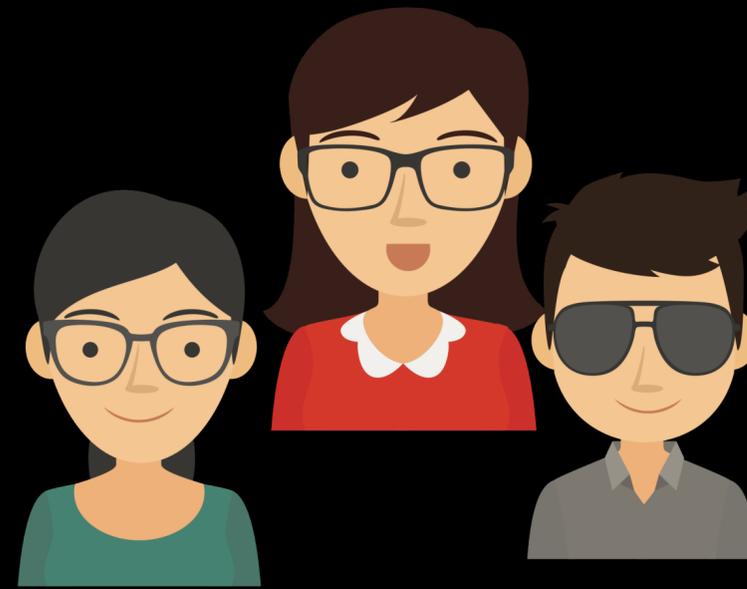
Middle

Junior





Testers don't tell us
when to release.



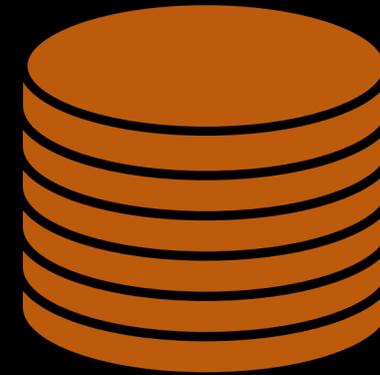
Testers



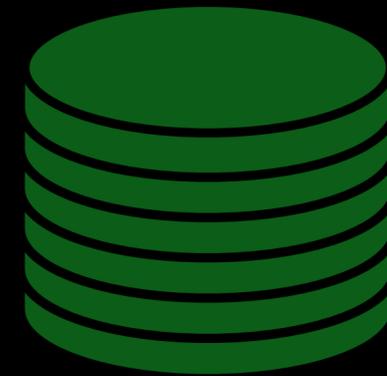
Devs



Repository

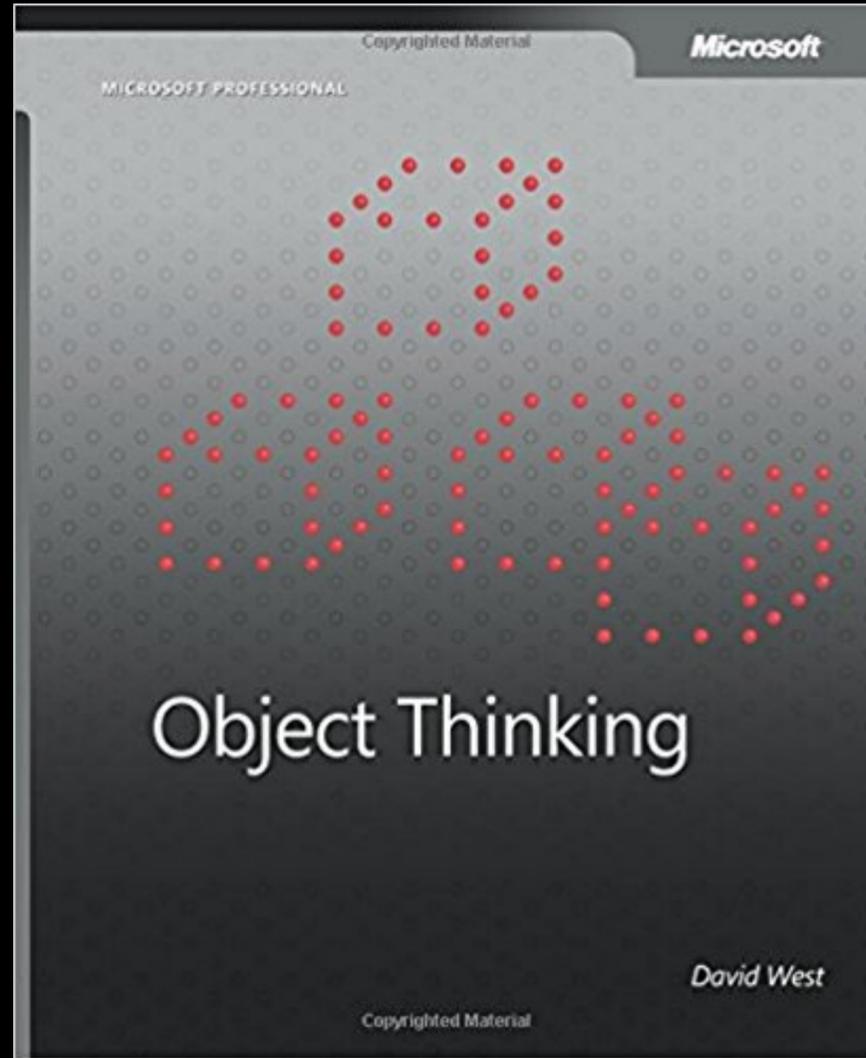


Staging



Production





David West: “Software is released for use, not when it is known to be correct, but when the rate of discovering errors slows down to one that management considers acceptable.”



PM



Devs



Testers



Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2008 Proceedings Americas Conference on Information Systems (AMCIS)

2008

Sources of Conflict Between Developers and Testers in Software Development

Xihui Zhang
University of Memphis, xihui.zhang@memphis.edu

Jasbir S. Dhaliwal
University of Memphis, jdhalilw@memphis.edu

Mark L. Gillenson
University of Memphis, mark.gillenson@memphis.edu

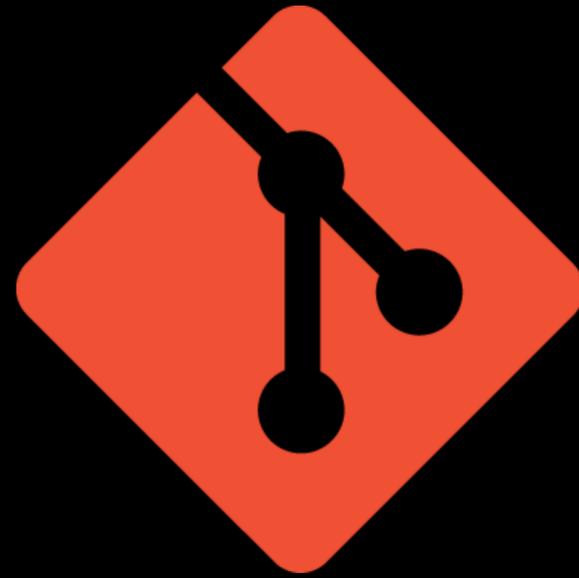
Gertrude Moeller
FedEx Express, gmoeller1@fedex.com

Follow this and additional works at: <http://aisel.aisnet.org/amcis2008>

Recommended Citation
Zhang, Xihui; Dhaliwal, Jasbir S.; Gillenson, Mark L.; and Moeller, Gertrude, "Sources of Conflict Between Developers and Testers in Software Development" (2008). *AMCIS 2008 Proceedings*. 313.
<http://aisel.aisnet.org/amcis2008/313>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Xihui Zhang: “The software testing process is inherently adversarial, setting the stage for inevitable developer-tester conflict.”



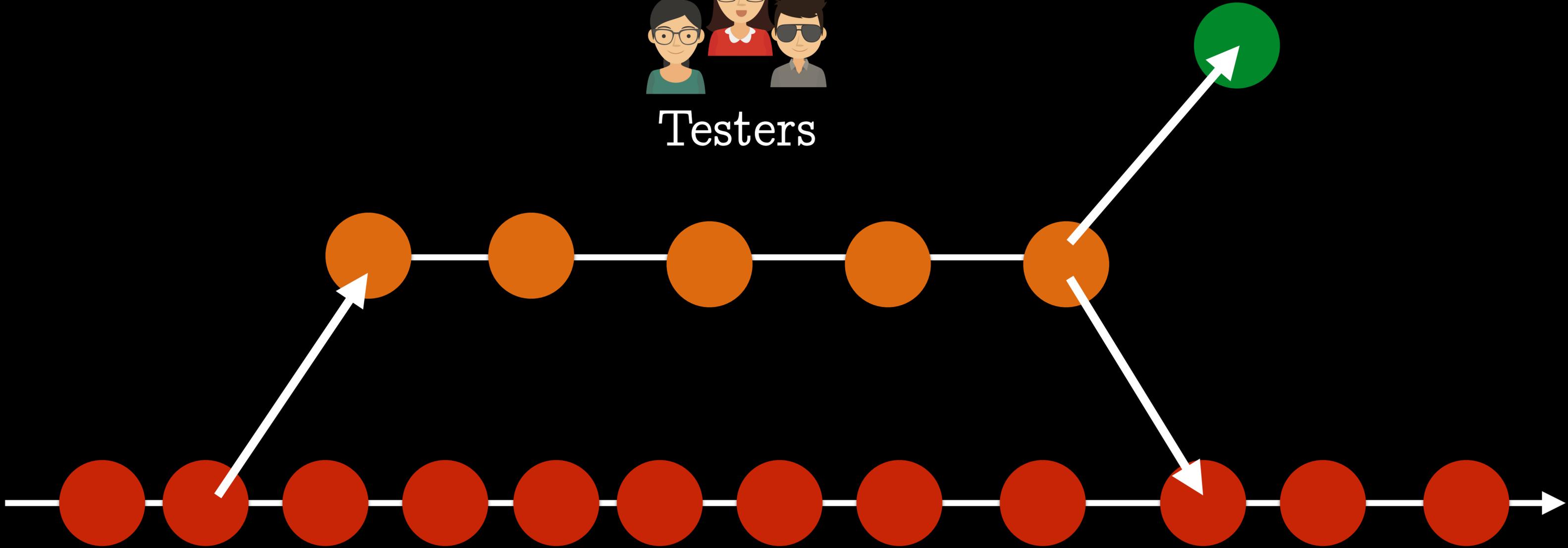
GitFlow

by Vincent Driessen





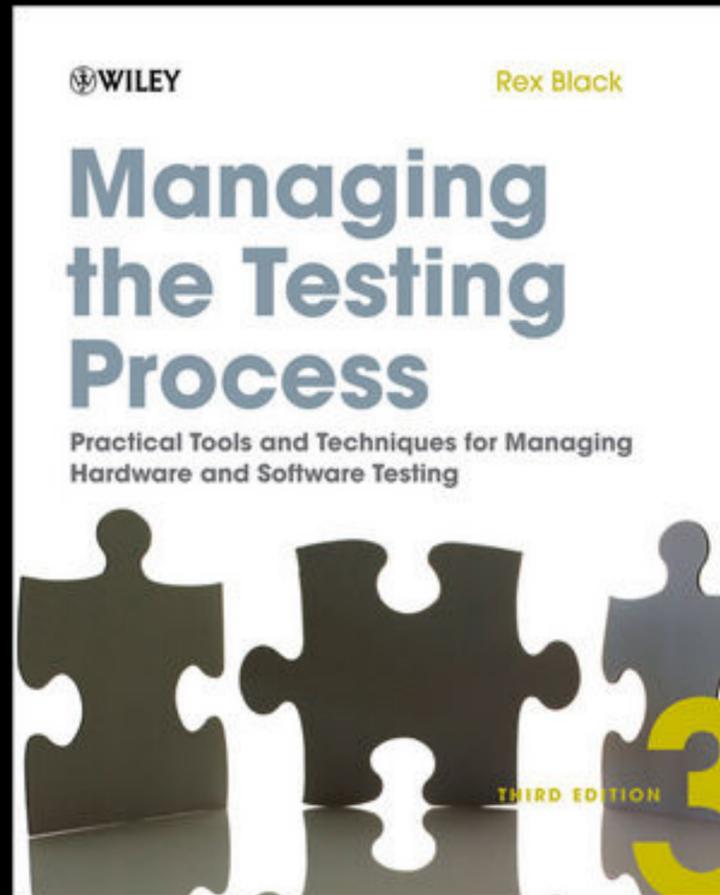
Testers



3

Testing is finished
when enough bugs
are found.





Rex Black: “Suppose, though, that you could estimate the total number of bugs in the system under test. Or, perhaps you could measure the bug-finding effectiveness of your test system. There are three techniques for solving these problems...”



Time

Scripts

Bugs



Guess

Budget

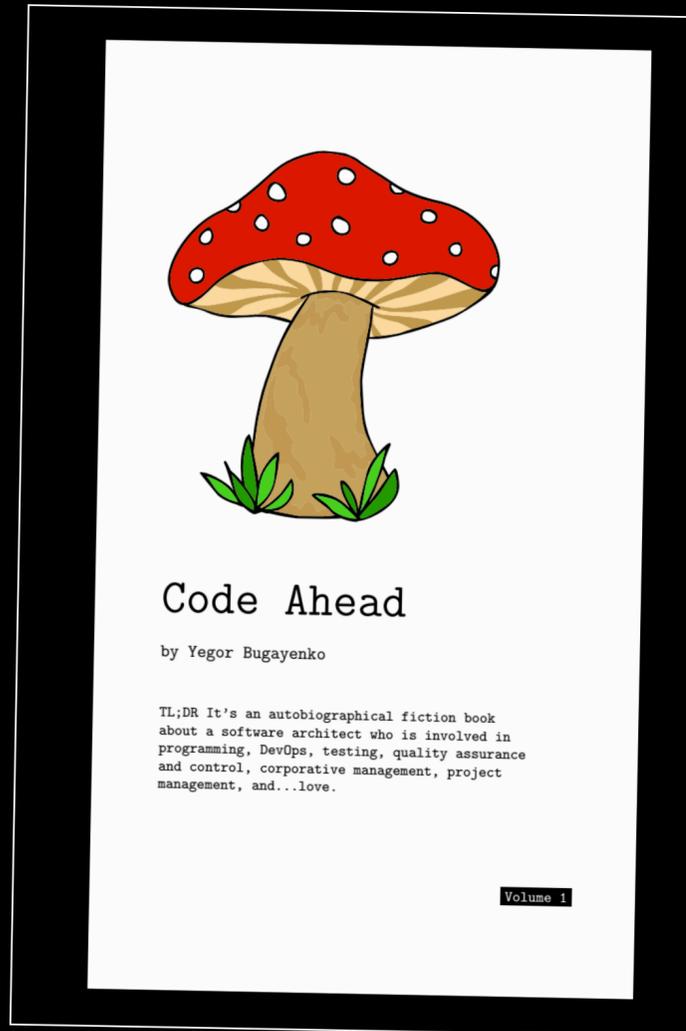
Experience



4

Testers must be rewarded for the bugs they find.





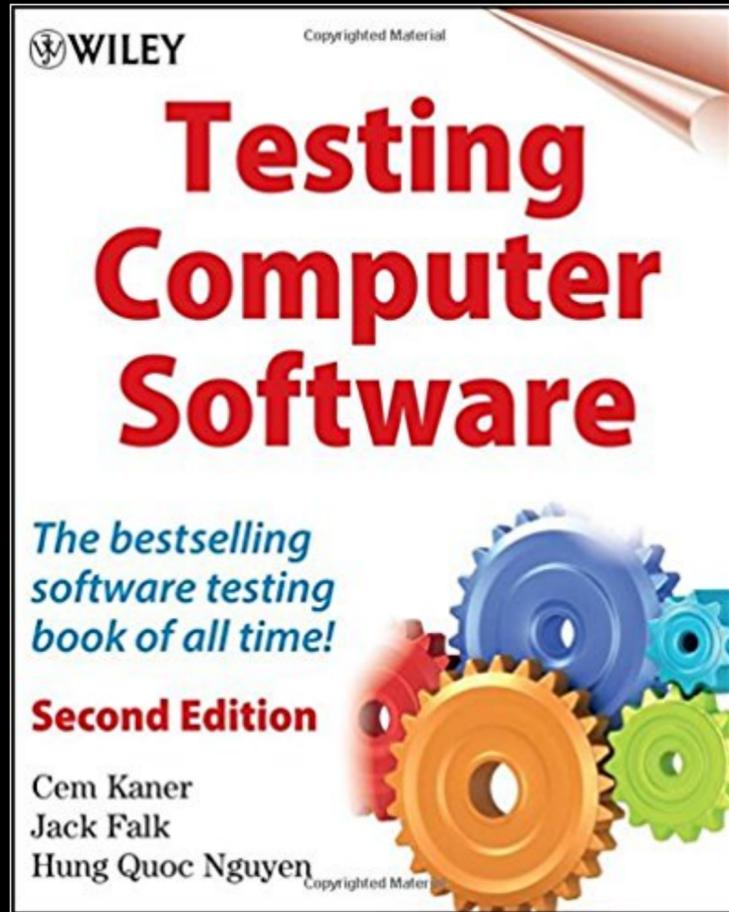
Yegor Bugayenko: “The best way to motivate testers to find more and better bugs is to pay them for each one.”

Price

Quality

Time





Cem Kaner: “The best tester isn’t the one who finds the most bugs or who embarrasses the most programmers. The best tester is the one who gets the most bugs fixed.”





@yegor256



@yegor256news