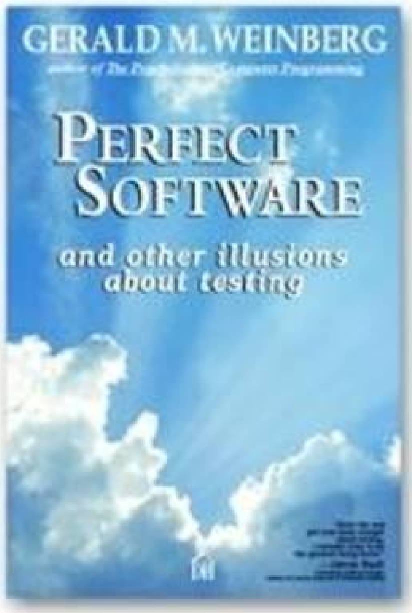

Perfect Software--And Other Illusions about Testing

Gerald M. Weinberg

Perfect Software--And Other Illusions about Testing

Gerald M. Weinberg

Perfect Software--And Other Illusions about Testing Gerald M. Weinberg
Editorial Reviews - Perfect Software

Perfect Software--And Other Illusions about Testing Details

Date	:	Published January 1st 2008 by Dorset House Publishing Company, Incorporated
ISBN	:	9780932633699
Author	:	Gerald M. Weinberg
Format	:	Hardcover 182 pages
Genre	:	Nonfiction, Science, Computer Science, Software

From Reader Review Perfect Software--And Other Illusions about Testing for online ebook

Matt Diephouse says

There was some good info in here: about the inability to exhaustively test, how to classify bugs, etc. But I found the bug hard to get through and poorly organized. (The latter was due largely to the poor formatting of the Kindle version, I expect.)

The writing switched between unnecessary, contrived narrative and a more bullet-point style. I found both very distraction, and I'm not sure which I liked least.

Željko Filipin says

A very short review on my blog: http://filipin.eu/perfect-software-an...

Howard says

Once you've been in the testing industry for a while, you may start to forget the questions you had when you started. You may be astounded when someone asks why you insist on testing. You may have forgotten what it's like to wonder why someone should test something. You may not actually know why you're testing - it's just something you do. It takes the insight of crossing paths with dozens or hundreds of naive souls working in the computer world all asking the same questions over and over again to understand why we test and what it takes to convince others of the value added. Weinberg has crossed paths with many, has earned respect, and his insights are quite eye opening. While not quite a page-turner, I did enjoy reading this. More than how to test, this is a book about the most difficult part of testing - interacting with others. When you are a tester, everybody (developers, managers, clients, other testers) are at times against you. Never the less, we have to work with them. Carry with the knowledge and support that Weinberg offers and you will not be alone.

Karin Axelsson says

"No amount of computing power can compensate for brainless testing."

Henriette says

I thoroughly enjoyed this book. It was written in a way that made it easy and entertaining enough to read in a day. It answers some questions about software testing, but more importantly reminds us of questions we should be asking when testing. If you are familiar with James Bach then you will see many similarities and even references to James Bach in this book. This book is a real reminder of why testers are needed and why testers can never be replaced by computers and therefore a recommended read for anyone working as a tester or with testers.

Igor says

Great read. Definitely would recommend it to anyone interested in software development, not just tje testing.

Gilbio says

An interesting journey around the world of software development, with a particular focus on testing activities. An hard landing for who think that the perfect in this world really exists! As the author underlines "When we want so much for everything to go well, it's all too easy to inject our fantasies into our data"

Ilya says

Iron ore mine of testing wisdom. Reading experience is stunning as is the case with all Gerald's books (at least those I read).

Charles says

Exhaustive testing of software is generally impossible; in all but the most simple of cases the number of possible paths is effectively infinite. By that I mean that while the number is finite, it is so large that only a tiny percentage could ever be tested. So in practice the number is infinite. Since humans are prone to error and the number of possible paths where a bug can be introduced into a program is also very large, the bug-free (perfect) program is an unattainable goal. It is no fun, but that is just the way things are.
Being driven by optimism and necessity, humans still create millions of lines of code every year and many of these programs are released into the untamed wilderness known as the customer base. Therefore, conscientious developers have no choice but to develop a testing program that will have a reasonable chance of identifying the most significant and likely bugs in the software.
As Weinberg points out, some people do make the choice to do very little testing, sometimes relying on an approach like "If we get lucky, no one will experience that bug." Others simply redefine the bug as a feature and move on. Neither of these approaches is ethical nor will it provide you with legal cover. In a very witty yet understated style, Weinberg describes some of the circumstances he has encountered where people simply denied there was a problem and if that failed tried to hoist the problem onto someone else.
As is the case in most areas of software development, the solution lies more in the realm of group psychology rather than technical expertise. Having authored the classis "The Psychology of Computer Programming" Weinberg is an expert at finding what solutions are available to this seemingly impossible problem. It starts with eliminating all illusions and delusions and accepting the inherent limitations of what software testing can do. After that, one develops a testing process that is executed concurrently with the development of the software. The first step is to test the modules by creating a set of reasonable coverage test values. Varying from module to module, this set covers some of the most likely entries as well as the bizarre ones and the ones on the limits.
Once the tests on the modules are complete, the next step is to integrate the modules into a functional unit.

Once the unit is stable and can be run, an additional set of reasonable coverage test operations is created and executed. In each case, detailed knowledge of what the software is supposed to is used to develop the sets of test values.
Weinberg is also forceful in debunking the current state of the automated testing of software, a tactic that he dismisses. The strongest argument against using software to test software is grounded in human psychology, namely the easily adopted assumption that if it came from a computer then it must be right. This belief in the infallibility of technology has a strong tendency to dumb-down humans and that is to be avoided if at all possible.
As stated before, the quality software developer has no choice but to test their product. If you want to maximize your likelihood of releasing a quality product, then your first step is for everyone in your team to read and discuss this book. Doing so will clear a lot of the yucky stuff out of your path. So while you may still step in it, the pile you squish is either of your own making or one of the few that Weinberg does not cover.

Published in the online "Journal of Object Technology" reprinted with permission

Janice says

Helped me answer exam questions for my CSQA. It's a great resource and makes sense.

Elizabeth says

Gerry Weinberg succinctly described many of the interpersonal struggles at my job and reinforced much of the stuff James Bach's preached in person. His explanations of what testing cannot do and why estimates at the beginning of a project are always too optimistic were spot on. This was a good reminder that testing is a waste of time if you're not going to do anything with the information you uncover.

The sections of supposed journal entries or dialogue did not help shed light on the material. A more conventional format would have served it better.

Sergiusz Golec says

+ Wit & vivid stories.
- Part of this book is a copy paste (or a paraphrase) from other books. Like "More Secrets of Consulting" by Gerald M. Weinberg.

Oleksii Burdin says

Great reading on how software is developed and what testers can do about it.

Matthew says

Note: Not my first time through this book.

This book should be mandatory reading for all software development professionals, particularly managers. Gerry has a great way of communicating the value of testing, the pitfalls of common approaches to development and test management, as well as what testers should be focusing on when it comes to communication.