How Google Tests Software

James A. Whittaker, Jason Arbon, Jeff Carollo

DOI: 10.1145/2347696.2347723
http://doi.acm.org/10.1145/2347696.2347723

How Google Tests Software is written by James Whittaker, Jason Arbon and Jeff Carollo, and published by Addison-Wesley, © 2012, (paperback), ISBN 978-0-321-80302-3, 281 pp.,0-321-80302-7 (paperback), $34.99

This book is about how the people (developers and managers) at Google engineered an organization that incorporates quality into its foundations. This is no mean task, considering that software quality is one of the cornerstones of any sustained software product.

As I read the book, the logical underpinnings of the test and quality practices become clearer. These principles are important for those who need to know the reasoning and context, if they wish to do something similar in their own organization.

For example, starting from one principle "Quality != Testing" (P1, pp 5), the reader will see how additional principles are derived and used. P1 distinguishes between Quality as defect prevention and Testing as defect assessment. Strategies like Crawl, Walk, Run (pp 10) introduces testing at many stages, so that each test stage provides feedback of quality to the next stage.

P1also leads to "Quality is owned by those who write the software", P2. The advice "Don't hire too many testers" (pp 4) is intended to focus the responsibility and not diffuse it to the traditional QA/Test group. This also avoids fatal flaw number 1: testers have become a crutch for developers (pp 249).

Thus, Google has learned to discard the traditional role of QA tester and created new roles called Software Engineer in Test (SET) and Test Engineer (TE). The foreword by Patrick Copeland mentions that "If I was going to change testing at Google, I needed to change what it means to be a tester" (pp xviii). In this way, a small number of test engineers are empowered to support a large number of developers. This contrast with the traditional and competing idea (from factory analogies) that software developers are expensive and their time are not well spent if they were to test their own software.

Google views development and test as the same discipline. This elevation of Test/and Quality status is an ongoing challenge. The book discusses the challenges of raising the hiring requirements for Test and Quality. By making testing resources a business resource to be justified, not a right of project teams, Google makes good use of competition and peer pressure to get all developers to adopt best quality and testing practices. In fact, "teams unwilling to commit to writing small tests and getting good unit-level coverage should be left to dig their graves in peace" (pp 190). This is because no test resource would be willing to join a team that does not adopt good quality practices.

There are other practices that synergistically strengthen these ideas. There are too many to write about, but here is a laundry list :

- Testing awards (pp xiv)
- This book is part of new employee orientation (pp 2)
- Earning code readabilities (pp 19)
- Peer Bonus (pp 19)
- 20% Time (pp 19, footnote)
- Test Certified Program (pp 54)
- Test Certified Mentors (pp 56)
- Ease of Project Transfers (pp 190 footnote 2)
- Open Transfer Process (pp 207)

It should be emphasized that Google's state of practice is an on-going process and the book is only a snapshot of what has worked/or not worked at the time of writing. There is a spirit of innovation and experimentation that the book seeks to impart. The many references to open sourced testing projects of Google is an invitation to participate, even if you do not work at Google.

For those who like a mystery, the Google's secret sauce in Testing is actually revealed in page 225.

Structure: Structurally, the book consists of 5 chapters. Chapter 1 provides the basic underpinnings, although the principles are scattered throughout the book. Ch2, 3 and 4 deal with the spe-cific roles and responsibilities of Software Engineer in Test (SET), Test Engineer (TE) and Test Engineering Manager (TEM) . The last chapter looks into the future.

I have also found the two forewords by Alberto Savoia and Patrick Copeland very useful and enlightening. The former explained the role of testing best as “attention to quality without derailing development or the speed at which we got things done” (pp xv). The latter provided a condensed summary of the early days of testing at Google and the difficulties of introducing the new roles and quality initiatives.

A pragmatic example, The book is full of pragmatic details of the realities of software testing and how Google attempts to deal with them. For example, consider the venerable test plan. We start with a quote from page 70 “As test documentation goes, test plans have the briefest actual lifespan of any test artifact”. It goes on to describe the ideal characteristics of a useful testplan and the use of Attribute Component Capability (ACC) to make it easy to create a minimal but complete and useful testplan.

Who should read this book? Other than the usual suspects - professional software engineers, software test and automation engineers and software project managers, I would like to name two other categories - venture capital investors of software companies and software job seekers.

Both of these have the need to be able to assess the viability of the company they are going to invest or join and this book provides material to ask questions to ascertain the maturity and empowerment at the engineers’ level, of the quality and testing processes used by the company.

The section “Interviewing SETs” (pp 62) and “Recruiting TEs” 9 pp 127) give insight into the job requirements and discusses what traits Google is looking for..

Google Testing Blog on “How Google Tests Software”

It would be amiss to mention that some of the material for the book was derived from the Google Testing Blog. There is a series of seven blogs on the topic “How Google Tests Software” that readers will find interesting too. Search the web and there are other resources and videos on this topic too.

Postscript: What happened to the authors? At the time the review was written:

- James Whittaker has left Google and returned to Microsoft as a Partner Development Manager at Microsoft.
- Jason Arbon is an engineering director at utest.com
- Jeff Carollo is a senior partner at Google.

Reviewed by Kwee Heong Tan, software test engineer at Jeppesen Corporation, tank@acm.org