

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/384145162>

"Book review: M. Fewster and D. Graham, Software Test Automation: Effective Use of Test Execution Tools, Addison Wesley, New York, NY, USA (1999)", ACM Computing Reviews

Article · December 1999

CITATIONS

0

READS

45

1 author:



[T.H. Tse](#)

The University of Hong Kong

261 PUBLICATIONS 5,782 CITATIONS

SEE PROFILE

**Book Review: Fewster M. and Graham D.
*Software Test Automation:
Effective Use of Test Execution Tools.*
ACM Press/Addison-Wesley Publ. Co.
New York, NY, USA (1999).**

T.H. Tse,
Department of Computer Science,
The University of Hong Kong,
Pokfulam, Hong Kong

Test execution tools are becoming more and more popular. This book is designed for testers and technical managers who would like to use such tools effectively. It is especially useful for “those who already have a test execution automation tool but are having problems or are not achieving the benefit they should.”

The book is divided into two parts. Part 1 discusses test automation techniques. Chapter 1 introduces the basic concepts in software testing. Chapter 2 clarifies the common misconception that equates test automation with capture replay. Scripts are as important in test automation as programs are in software. Furthermore, it is rightly argued that testing automation without an automatic comparison of results is only input automation. Based on the discussions in this chapter, common techniques for scripting and automatic comparison are examined in chapters 3 and 4. Chapter 5 presents the basic components and structures of testware. Chapter 6 illustrates how pre- and post-processing in software testing can be automated. Testware maintenance is discussed in chapter 7. Chapter 8 highlights the metrics for measuring the effectiveness of software testing and testing automation. Other important issues, such as which tests to automate first, judging of passes and failures, and the monitoring of progress, are covered in chapter 9. Chapter 10 outlines the procedure related to testware selection. The last chapter of Part 1 highlights the importance of testware implementation issues.

I have no hesitation in strongly recommending Part 1 of the book. It is the first comprehensive text on test automation and is a must-read for anyone who is serious about the profession. Every concept in software testing and automation is explained in a precise and concise manner. Both the outline guide at the beginning of Part 1 and the summaries at the end of each chapter are very useful.

My only concern is the assumption, as summarized on page 102, that “when automating test cases, the expected outcomes have either to be prepared in advance or generated by capturing the actual outcomes of a test run. In the latter case the captured outcomes must be verified manually and saved as the expected outcomes for further runs of the automated tests.” Readers interested in advanced mathematical techniques may refer to Blum and Kannan [1] for an alternative

method that does not require a manual computation of the expected outcomes in the first place, though this technique may only be applicable to special situations with known mathematical properties.

Part 2 contains 17 chapters devoted to case studies and other experience reports on test automation. I appreciate the usefulness of reading experience reports by guest authors that may deviate from the academic views in Part 1. On the other hand, the authors' levels of experience vary widely. Some guest authors discuss fifth-generation test tools, while others are not fully convinced of the benefits of automation in software testing. Conflicting comments by the guest authors may confuse novices considering test automation.

For example, chapter 16 suggests that automated testing definitely made testing more efficient, but did not play a part in finding bugs. The vast majority of the bugs — dozens of them — were discovered during the manual process of preparing automatic test scripts. The concluding remarks explain that this can still be seen as evidence for the effectiveness of “automating testing”, as distinct from “automated testing”. I cannot help wondering whether the author is only being cynical. If the testers had been just as careful in preparing manual test scripts, the same bugs would have been discovered during the process.

Some of the recommendations in this section are too general to be useful. For example, chapter 14 states that we must be very clear about what automated testing can and cannot do before we introduce it. However, the general characteristics of automated testing as listed on page 350 apply to both success and failure stories. What makes a successful case different from a failure? This is exactly what readers would like to learn from the book.

I cannot determine whether readers will welcome a combination of conflicting views from authors with different levels of experience. Perhaps readers will have to decide for themselves. Whatever your view about Part 2, however, the book is definitely worth its price, even if only Part 1 is considered. Perhaps we should regard Part 2 as a bonus, rather than be too critical about it.