

How to test



Mike Talks

How To Test

Mike Talks

This book is for sale at <http://leanpub.com/howtotest>

This version was published on 2016-04-14



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2015 - 2016 Mike Talks

Contents

Dedication	1
Introduction	2
How to use this book	3
Chapter 1 - Why does the world need testers?	6
For more - get the full book	9

Dedication

To nurture is to love. In memory of my beloved grandmother, Elsie Taylor, who always saw the potential in me.



Introduction



A few years ago, I was exchanging emails with another test manager named Bernice Ruhland - and we were having a similar problem. How do you help to develop someone who is new to testing?

The problem is this - there are a lot of books out there, but they can be quite long, and they represent a range of opinions. Oh yes, there is a spectrum of feeling about “how to test”, let me start by saying that. There are also a lot of training courses out there, some claim they can certify you as a professional tester, for a hefty price. However some of those courses are not the automatic entry into the profession that they claim to be.

We wanted to make a quick, easy-to-work through guide for people new to testing. Our target would be people who were considering a career in testing. But it would also be of use of people within IT who wanted to know more about the core skills of testing.

The book also needed to be free. Both Bernice and I have received

a lot of help from other testers around the world on a spectrum of subjects, and we wanted to “*pay it forward*”. Neither does the book end with any recommendation for any book or project we have a vested interest in.

We had great plans, but never quite the time to write it. Thanks to an invitation by The Summer Of Tech in Wellington to teach a workshop on testing to a group of technology students, I finally got the kick-in-the-ass I needed to sit down and write, and this book you’re reading is the end-product.

I hope it gives you the material you need to think, experiment and try out the core skills and thinking of the testing profession. If you read this and want to know more, there are some suggestions at the end for next steps.

Some people see testing as a potential easy in-road to the world of IT and potentially programming. But to myself and others, software testing is its own unique and challenging career path which is full of its own reward.

Just remember - everyone benefits from a kick-in-the-ass at some point

Mike Talks

September 2015

How to use this book

Yes, I’m quite certain you know how to use a book, but this one is a little different. It’s been designed with two kinds of end user in mind.

To the student tester

So you’re just starting out, or thinking of starting out on a career as a tester. Welcome!

This book is designed to teach you some of the nuts and bolts of testing - the day-to-day tasks of testing. As such there is some material which I have deliberately missed out - talking in detail about software development lifecycles such as waterfall or agile for instance. I'm really trying to focus on the skills and knowledge needed to get you up and contributing as a manual tester on a team.

Some of the sections include proposed activities for you to explore, trying out different social media registration, login and account management. I really encourage you to have a go at these exercises as a way you'll get the most out of this book, by trying yourself to apply some of the principles.

If you've just started at a new company, I encourage you to find time with a more senior tester and go through what you've done, and see if they have any feedback, especially ideas for anything you might have missed, or for a different way of doing an action. If not, find another person who's learning, and share your ideas, debriefing each other. Either way, you get to benefit from another's viewpoint and experience - typically you'll come to find there's an awesome power to collaborating with people in this way.

To the test leader/mentor/manager

So you've got a newbie starting next week, and you don't have anything prepared to help develop them through those first few weeks? Well, this is the book for you! I advise you to read through it yourself first, and encourage you to follow up on some of the exercises with your new starter. It should take them a couple of days to work through the book and the exercises, dependant on how intensively you need them to go through it.

This book is free, but if you do find it useful and valuable, I just ask that you contribute to the spirit of "*pay it forward*" that this book is all about. Consider making a charity donation from either yourself or your company. I've previously asked for donations

to the childrenâ€™s healthcare charity [The Starship Foundation¹](https://www.starship.org.nz/).

¹<https://www.starship.org.nz/>

Chapter 1 - Why does the world need testers?

As human beings we're incredible, imaginative, creative beings. We've made art, built huge buildings and visited every planet in our solar system.

But there's another side to being human, and that's we're not as infallible as we'd like to think. We might know basic arithmetic, we might know the difference between their, they're and there - but even so, once in a while our concentration will be off a little, and there it will be. *A little "oops".*

In software programming, a simple grammar mistake of using a ":" instead of a ";" can cause problems. If we're lucky the code will simply refuse to build. If we're less lucky, that mistake will go unnoticed until that line is executed, and even then may pass unseen.

One huge problem is we're really poor at noticing our own mistakes. And this is compounded in software by an effect called "*confirmation bias*".

Confirmation Bias

Let's just take a moment so you understand what confirmation bias is. There's a really good example on the recent New Horizons flyby of Pluto which I was guilty of.

I hold a Bachelor of Science in Astronomy, so as you can image, was getting somewhat excited at the initial pictures we got whilst the probe was still some way off. I knew from my degree how prevalent cratering is through the solar system - we see it most clearly on the Moon, where the lack of any weather system preserves these

features almost indefinitely. But they're there on pretty much every solid body we've been to.

When I saw some of those early pictures of Pluto then I saw, without a doubt vague outlines of craters. Only there was a problem, after close encounter, everyone was talking about the strange lack of craters. How had I got it so wrong?



NASA Image

Confirmation bias works like this - if you look at a picture like that, which is slightly fuzzy, if you know that craters exist all throughout the solar system. Then you are going to see shapes which your mind will recognise as craters. Your mind is looking to confirm what it already thinks is there. The problem is, there are no craters there, you've created a mirage of geology in your mind, and you've fooled yourself. And this in many ways is how much magic and illusion works.

Confirmation Bias In Software

If a programmer has been working on a piece of software for a couple of days, they will have invested their time and energy on every aspect they know of resolving the problem. But this means when they come to test it, they won't bring any new ideas to the

table. They will test it using the scenarios they've planned out in their head for it to operate for.

Like me and my craters on Pluto, they're expecting to see it work, so they'll only notice it failing if it's really obvious. Some bugs though can be more subtle, and more importantly can only be seen when trying different scenario.

Testing has developed as a career path because we've found some people are very good at using software, trying scenarios and describing problems they encounter. Often these people aren't developers and share very few core skills with developers.

In many ways it's the same as how an author for a book will need to use a proof reader for their material who will look at it with fresh eyes, and challenge any areas which don't make sense to them.

[Which reminds me – if you do find any obvious mistakes in this book – do drop me a line]

For more - get the full book

Hopefully you've enjoyed the book so far. To read more, please get the full book - it's free, and always will be!

Getting the book allows me to know you've downloaded it, and helps me to weigh up my audience. It lets me know that people are actually reading it.

If you do enjoy it - or have any ideas, do drop me a line.

Regards,

Mike