THE
Tester's
Pocketbook

Paul Gerrard

The Tester's Pocketbook

Paul Gerrard

This book is for sale at
http://leanpub.com/testerspocketbook

This version was published on 2014-08-11

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.

To My Family

Contents

CONTENTS

Paul Gerrard

Paul Gerrard is a consultant, teacher, author, webmaster, programmer, tester, conference speaker, rowing coach and publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them.

Educated at the universities of Oxford and Imperial College London, he is a Principal of Gerrard Consulting Limited, the host of the UK Test Management Forum and the Programme Chair for the 2014 EuroSTAR testing conference.

In 2010 he won the EuroSTAR Testing Excellence Award and in 2013 he won the inaugural TESTA Lifetime Achievement Award.

Publishers note

Every possible effort has been made to ensure that the information contained in this book is accurate at the time of going to press, and the publishers and author cannot accept any responsibility for any errors or omissions, however caused. No responsibility for loss or damage occasioned by any person acting, or refraining from action, as a result of the material in this publication can be accepted by the editor, the publisher or the author.

case of reprographic reproduction in accordance with the terms and licenses issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1T 4LP. Enquiries concerning reproduction outside these terms should be sent to the publishers.

The right of Paul Gerrard to be identified as the author of this work has been asserted by him in accordance with the Copyright, Design and Patents Act 1988.

The views expressed in this book are those of the author.

Typeset by Paul Gerrard

Preface

Aims of the Pocketbook

This little Pocketbook has two aims.

The first aim is to provide a brief introduction to the discipline called *testing*. Have you just been told you are responsible for testing something? Perhaps it is the implementation of a computer system in your business. But you have never done this sort of thing before! Quite a lot of the information and training on testing is technical, bureaucratic, complicated or dated. This Pocketbook presents a set of *Test Axioms* that will help you determine what your mission should be and how to gain commitment and understanding from your management. The technical stuff might then

make more sense to you[1].

The second aim of this Pocketbook is to provide a handy reference, an aide memoire or prompter for testing practitioners. But it isn't a pocket dictionary or summary of procedures, techniques or processes. When you are stuck for what to do next, or believe there's something wrong in your own or someone else's testing or you want to understand their testing or improve it, this Pocketbook might be of help. It will prompt you to ask some germane questions of yourself, your team, your management, stakeholder or supplier.

Testing is easy, isn't it?

Some people think so. These people have never had to think about it seriously beyond it being a task on a project plan to be completed as quickly as possible before 'going live' (whatever that means). But there are good reasons

---

[1]Very few references to the popular testing books are given in the Pocketbook (not even to my own book). Visit the book's website testers-pocketbook.com for a 'further reading list'.

why everyone should think testing is easy, or at least is second-nature. You see, everyone became a tester[2] before they could walk or talk.

Just about everything a baby can grasp is put into their mouths within seconds. It is tasted. It is *tested*. Babies and children learn incredibly rapidly through play. Play and testing are very closely related. At some level, all 'good' testing requires curiosity and imagination. So what could be hard about play?

Many things make testing hard. As we grow and mature, curiosity and imagination are often stifled and our natural testing skills are dulled. Usually, we don't test for our own benefit; rather, we test on behalf of other people whose needs, preferences, prejudices and assumptions mislead us. The things we are asked to test are not simple; systems, even small ones, can be immensely complex to understand, to configure and to interpret.

---

[2]From now on, I'll use the term tester to denote the role of someone who does testing. There are many testers who test full time, but testing is also a part-time role for a designers, developers and users.

Context-neutral

Although my own background is in software testing, I have tried throughout the Pocketbook to avoid the use of terms like software, programmer, program, module etc. because testing is about so much more than software. Instead of software, I have used the word system to label *what* we test. I've tried to make the text as generic as possible so that it supports the testing of any system. Does this mean the Pocketbook describes the approach to testing anything? Well, that would be neat but highly unlikely[3].

In my career, I've tested programs, software systems, interfaces, hardware, firmware, buildings, furniture, telephones, concrete and people. This Pocketbook attempts to describe the thought processes associated with testing systems in the broadest sense and is not specific to any one milieu, technology, organisation, culture. It is context-neutral.

---

[3]If you are testing something unusual and the book helps you to formulate a test approach – please let me know.

By the way, I'll discuss testing mainly in the context of projects, but testing isn't restricted to systems development projects. Testing might also take place in a maintenance or business-as-usual environment to ensure repairs or other changes to production systems work correctly. Testing might also be used to explore how a system works for the purpose of an evaluation.

This Pocketbook won't make you a demon tester. It won't teach you anything about test process, test case design or methodology. What it will do is introduce (or remind you of) the pre-requisites, thinking processes and decision-making you need to do a half-decent job.

Test Axioms

I introduced the idea of Test Axioms in posts on my blog in the spring of 2008[4]. Over a few months their definitions evolved and in May 2008 I summarised the thinking behind

---

[4]My blog can be found on my website: http://gerrardconsulting.com

them and tabulated 16 proposed axioms. Further evolution has occurred and with some changes, they have been used in this Pocketbook.

There was quite a reaction to the proposed axioms. Some people rejected the idea, saying there were no such things. Others were more supportive and offered new axioms or alternate definitions. You can see these posts and links to other peoples' comments on my blog.

I believe that there are a set of rules or principles that provide a framework for all testing. *But there is no single agreed definition of test.* This is mostly because every consultant and author has tended to write their own definition to suit their own purposes (and I am as guilty as the rest of them). As an industry, we are hamstrung because of this. Our clients are confused, and we get distracted by discussions on definitions because of not-invented-here mentalities and our competitive instincts. Context-neutral definitions are proposed in Chapter 1.

An axiom is something believed to be true, but

cannot be proven in any practical way. It could be disproven by experiment or experience and we should be prepared to be proven wrong and welcome attempts to do this[5].

But some people object to the notion of Test Axioms and say that nothing in testing is certain. There are no axioms. All testing rules, principles, techniques, approaches etc. are heuristic. *Heuristics have value in some contexts, but are limited in application, usefulness, accuracy etc. in other contexts. They are limited or fallible in known ways.*

Here is my proposal for Test Axioms restated in a different way. I have tried to define them in a way that testers can, for all practical purposes, regard them as axiomatic. If anyone devises a testing context where the axioms are violated, we need to think again: Perhaps the axiom should be scrapped or changed or its scope of applicability defined.

---

[5]This usage is consistent with many other famous examples: the Definitions in Euclid's Elements, Newton's laws of motion, the US Declaration of Independence present sets of beliefs without proof or corroboration. Most have subsequently been shown to be imperfect, but they continue to work for most practical purposes.

Reviewers have generously commented on the early drafts of the book and provided many suggestions and helpful criticisms. *But so far, I have not received any concrete examples that invalidate the Test Axioms as stated in the Pocketbook.*

The Test Axioms are an attempt to provide a context-neutral set of rules for testing that identify the critical thinking processes and motivations for all test approaches.

Stakeholder obsessed

Testing is an information or intelligence-gathering[6] activity performed on behalf of (people who I will call) testing stakeholders. The manager who asked you to test could be your most important stakeholder (ask them!) They think testing is important enough to get someone as important as you involved – but might not

---

[6]In 2001, I coined the term Project Intelligence to represent the information, data, and analysed outcomes (the evidence) from testing. (Project intelligence is analogous to battlefield intelligence in a military campaign).

be able to articulate why they see it as an important role.

You might use this Pocketbook to help you have a rational dialogue with them, to home in on the important issues you need to resolve to enable you to do the right job.

By the way, if you are testing the products of your own efforts, *you could be your own stakeholder*. Your approach to testing your own products or systems will be focused on what *you and others*, as stakeholders, want to learn about those products or systems.

Most systems that need testing have stakeholders whose interests do not coincide perfectly. We cannot test everything, so we need to help them to make choices. We need to develop a good relationship with stakeholders to build consensus, buy-in and trust in our test approach. Since most stakeholders are non-technical, the language we use must be simple and direct. The Test Axiom definitions are just that.

Structure

Chapter 1 sets the scene and introduces the foundations of testing. Testing is a natural human activity; we test systems; the purpose of testing is presented. If we take an axiomatic stance to testing, we need context-, process-, technology- (and other things) –neutral definitions of test and testing. My chosen definitions close the chapter.

Chapter 2 introduces the First Equation of Testing as a way of understanding how an approach to testing can be formulated. The fundamental development process is introduced. Test Axioms, Context, Values and Thinking are presented as the building blocks of a test approach.

In Chapters 3, 4, and 5 the Test Axioms are presented under chapter headings Stakeholder, Design and Delivery respectively. Each axiom is discussed to present the reasons why the axiom exists and what its impact on a test approach can be. Some relationships between the axioms are identified.

Chapter 6 discusses how testing improvement can be achieved. The meaning of improvement and brief overviews of model-based and goal-directed improvement approaches are provided. The change-management process is introduced and a popular '8-step' process provides an example approach.

Chapter 7 provides some brief guidelines for conducting an assessment of testing in an organisation. The axioms are then listed, one per page complete with 'seed' questions for each that can help your investigations. (They could also help you to challenge your own test approach).

Chapter 8 presents some closing remarks and summarises how the Test Axioms can be used to define test approaches.

An index is included at the end of the Pocketbook. Definitions of key terms are identified by bold page numbers in the main text.

Acknowledgements

Although the writing of this Pocketbook is a solo effort, it is the concentrated result of many testing assignments starting in 1992, countless conversations with my friends in testing at the UK Test Management Forum, EuroSTAR, STAR East and West, LEWT, software testing specialist interest groups SIGIST (UK), TestNet (Holland), SAST (Sweden) and Den Norske Dataforening (Norway) and tough questioning in training courses, blogs and email discussions. You all know who you are :O).

In particular, I'd like to thank all of the Tester Retreat attendees, especially my good friends Isabel Evans, Peter Morgan, Graham Thomas and Neil Thompson for inspiring the initial work on Test Axioms and providing feedback and guidance.

I invited a lot of people to review the Pocketbook and most reviewers responded quickly and with a lot of comments[7]. I apologise for

---

[7]Serves me right.

taking so long to use them. All of the comments were considered – some conflicted with each other and some conflicted with my aim of being context-neutral, so not every contribution made it into the final draft.

Many, many thanks to: Julien Bensaid, Bart Broekman, Bob van de Burgt, Fiona Charles, Graham Dwyer, Julia Gerrard, Derk-Jan de Grood, Tim Koomen, Alon Linetzki, Richard Marsden, Henk van Merode, Tone Molyneux, Maurice Siteur, Michael Stahl, John van Veen, Jeffrey Wannee, Adam White, Barbara van Wijk, Maarten Woolthuis and all those people who reviewed the first draft, gave verbal feedback and support. If I've omitted your name – many apologies and please let me know so I can thank you in future editions.

Thanks and much love, as always, to my (tolerant and smarter-than-me) family, Julia, Lizzie and Max.

Any errors or omissions are my fault entirely. This is a work in progress. Please let me know how I can improve it. My email address is paul@gerrardconsulting.com.

Further information can be found on the book's website:

http://testers-pocketbook.com

Before we begin...

*A generous and elevated mind is
distinguished by nothing more certainly than
an eminent degree of curiosity*

Samuel Johnson

Cogito ergo pertento

I think, therefore I test

How many testers does it take to change a light bulb?

None. We keep our testers in the dark.

1. What do we Mean by Testing?

We Test Systems

The word *system* conjures up different images to different people. To people in the IT community, a system is one or more computer applications (or sub-systems) that provide services to customers, partners or end-users. A software company might view their software product, which is a component or sub-system of larger systems as a system. But a software-centric view is rather too narrow for our purposes.

A *systems approach* takes a much broader view and a system might include the computer hardware, software and infrastructure,

but might also include the procedures, partners, management, users and many other assets.

We'll use a rather abstract definition of system from now on.

A system is a combination of related parts organized into a complex whole.

A system can be anything from a mobile phone to a guided missile; a web site to a global entertainment network; an electric shaver to an airport; a team, company, culture or society.

Every system needs some testing; every test requires some disciplined thinking; testers are the people who think, and make testing happen.

We are all Testers; We are all Tested

We are all testers. We test all the time. We test drive a car before we buy it. We test our food before we swallow it. We re-read our

emails before we send them. We download software and try it out before we pay for it. We try on clothes and ask for a friend's opinion before we commit. We visit and inspect houses before make an offer to buy them. We court prospective mates – we test for evidence of fitness for mating and raising offspring!

In every case, our behaviour is affected by the outcome of a test. If the outcome is negative, we don't buy, commit, swallow or even propose marriage. If we are risk-averse, if the outcome is neutral or uninformative, we might decide we need more information, new tests, and test again.

During the course of our lives, we are all tested or assessed in countless ways. We take school tests, examinations, IQ and aptitude tests; we fill in forms for tax assessments, new bank accounts and credit checks; we are subjected to health checks, eye tests and driving tests. Interns and apprentices are hired on the basis of successful completion of a probationary period.

We might not like it but on the successful out-

come of these tests depend our grades, licence to drive, job offers and many other aspects of our very livelihood.

Everyone has their standards, and we all assess other people and things against them all the time[1].

The Purpose of Testing

At the most fundamental level, the purpose of testing is to gather information to learn about some aspect of a system and potentially make a decision based on the outcome of one or more tests. Consequently, testing is an information business, a people business and a business business. Let me explain.

Testers (or system builders[2], when they are

---

[1]Years ago, I was sat alone in a Euston Railway Station Restaurant. A small man (he turned out to be a professional jockey on his travels) approached me and asked could he join me? We enjoyed a long, diverse conversation. As we parted I asked him why he chose to talk to me rather than one of the other travellers. He said, "Your shoes were clean and your tie is tied properly". I obviously passed his test.

[2]From now on, I will use the term developer to denote the people who build systems.

testing) do not build; they do not put defects in and do not take defects out. In this respect, testers do not improve quality or add value to the systems they test. They are however, responsible for providing the most valuable information required by developers (to fix defects), project management (to understand and manage achievement) and stakeholders (to be assured). In this one respect, testing is all-powerful – it is the single source of knowledge of achievement in systems projects[3].

Testing is a people business. Most testers need excellent interpersonal skills, particularly communication skills. Testers and those who manage them need at one time or another to communicate with, negotiate with, influence and advise end users, developers, technical/environmental support, business and stakeholder management, internal and external auditors, outsource companies, customers, client services, and product managers. Interpersonal

---

[3]I'm using Projects as the context for testing throughout the book because that is the most common context. However, testing can exist in a maintenance context or as a learning or evaluation exercise (to better understand a packaged-solution) for example. It doesn't make much difference to the thought processes involved.

skills are, in many ways, more important than technical skills for testers.

A business business? Testing exists to detect defects and so protect end users, but there is a higher goal: to inform stakeholder and management decision making. The stakeholders who must decide to release, accept or reject a system and the managers who must close, delay, re-think or re-plan systems projects are critically dependent on the information produced by testers. In this respect, testers are a great ally of their stakeholders.

When we execute a test, if our aim is to understand aspects of the system under test to make a decision, we can treat the test as an information gathering or learning process. We design tests to gather specific information that we do not currently have.

The Definition of Test

It's about time I defined the most important little word in this Pocketbook.

This Pocketbook is about that word – test – used as a noun and as a verb. It's about testing as an activity and the outcome of that activity. It's about the people or organisations who commission that activity and those who use the results. Very much it's about the people who call themselves testers and the complex systems on which we work.

It's a small word, but it turns out to be a large subject. So let's get our definition right.

We need a definition of test that is context-neutral so I looked up test in the dictionary.com website. Of the many pages of references to the word test and its applications in many areas, the definition from the American Heritage Dictionary is the most appropriate.

Test: (noun) a procedure for critical evaluation; a means of determining the presence, quality, or truth of something; a trial

This statement seems to capture the essence of what is meant by a test – but there are three variations. Well this isn't so bad I think, as all

three taken together give us the foundations we need. Let's take a closer look at each one.

A procedure for critical evaluation

Critical evaluation involves a skilful judgement as to the truth or merit of something. A test is a procedure, usually with a series of steps that include some form of preparation, execution, results gathering, analysis and interpretation. This isn't a definitive description of a test procedure. There could be more steps and one could break these main steps down further.

The procedure doesn't necessarily require prepared documentation, but many tests are so documented[4]. The important thing is that there is a perspicacious thought process at the heart of a test.

This thought process is driven by the need to evaluate the system under test with respect to its adequacy, consistency, behaviour, accuracy, reliability or any other significant aspect

---

[4]The merits and demerits of planned, scripted testing compared with ad-hoc, unscripted exploratory testing are the cause of some debate.

or property.

A means of determining the presence, quality, or truth of something

A test could determine the presence (or absence) of something easily enough, but quality is a different matter: the term is loaded with emotional connotations, but we are rescued by the dictionary.

A quality can be, "an essential or distinctive characteristic, property, or attribute"[5]. Now we can see that a test can reveal these properties.

Can a test determine the truth of something? Well this makes good sense too. Typically, we need to test an assertion such as, "this system meets some requirement"[6] or "this system be-

---

[5]From dictionary.com. Note that I'm not using the term quality to reflect the relationship between a user or stakeholder and a product. Quality is like beauty – in the eye of the user. I won't be drawn into discussions of how one measures it. I'll avoid using the word wherever possible from now on.

[6]A requirement is a singular need of what a particular system should do. Sets of requirements may be documented in detail or at a high level, but requirements can never be complete. Implicit requirements reside in the heads of users, stakeholders, developers and testers.

haves in such a way" or "this system is acceptable" and so on. There's a certain amount of subjective judgement involved but we can see that a test or tests could provide evidence for someone to exercise that judgement and make a decision.

A trial

The notion of a trial implies that the process of testing a system will help us to evaluate that system with respect to its qualities. The purpose of such an evaluation is normally to make a decision.

The decision might be to accept or reject the system, but it might also be to expose its shortcomings so they can be remedied in some way. A test might also influence an individual or organisation to change direction – to rethink a design; to relax or change a requirement; to scrap a component and start again; to buy rather than build or build rather than buy.

A natural way of looking at a system under test is that it is on trial, and will be judged in some way.

The definition of testing

From our definition of the noun test, we can derive a verb easily enough.

Test: (verb) to critically evaluate; to determine the presence, quality, or truth of something; to conduct a trial.

So far so good[7].

---

[7]But not that good, really. Unfortunately, the testing profession is dogged with terminological problems. The words test and testing can be preceded and prefixed by countless words that qualify them. Re-test, user test, acceptance test can all be both nouns and verbs, for example! Always check the context of the words test and testing to be sure you understand what is meant by them.

Gerrard Consulting

Gerrard Consulting has been delivering testing services since 1984 and we have gained a reputation for innovative independent thinking based upon an honest open relationships with our clients. Often, clients come to us because of specific problems which aren't easily resolved using more 'normal' textbook solutions.

We offer Test Assurance, Test Consultancy and Supplier Management Services.

We have developed the Business Story Method™ and offer an SaaS product: businessstorymanager.com. The method focuses on using Business Stories as a lingua franca between stakeholders, developers and testers as specifications, as tests of requirements and as the source for Behaviour- and Test-Driven Development

approaches. We also provide a free to use sub-set of functionality in the SP.QA tool.

We host the Test Managers Forum which is run as a not-for-profit venture and is geared towards supporting experienced Test Managers and Technical Testers. Our events are discussion based (rather than presentation based) sessions and we allow much time for networking and information exchange. The website is http://uktmf.com.

If you have a testing problem you would like us to look at, we are very happy to come to your site and run a workshop to explore the problem at no cost to you. If we think we can help you, we'll make a more formal proposal.

Our website is http://gerrardconsulting.com.

Further Information

Contacting the Author

Should you have any questions or wish to discuss any of the issues raised in this Pocketbook, or perhaps you would like help in improving testing in your organisation, please feel free to contact Paul Gerrard.

Email: paul@gerrardconsulting.com

Tel: +44(0)7940 547894

Ordering copies of the Tester's Pocketbook

Hard copies of this Pocketbook can be ordered from the Gerrard Consulting website:

http://gerrardconsulting.com

Custom versions of The Tester's Pocketbooks

Some companies like a custom version of the pocketbook with their name on the cover. If you would like a customised version of the Pocketbook for your team, we are happy to create one for you. Perhaps you want include company information or policy in the Pocketbook? The material in customised Pocketbooks will remain copyrighted to its respective authors.

Do get in touch if this is of interest.