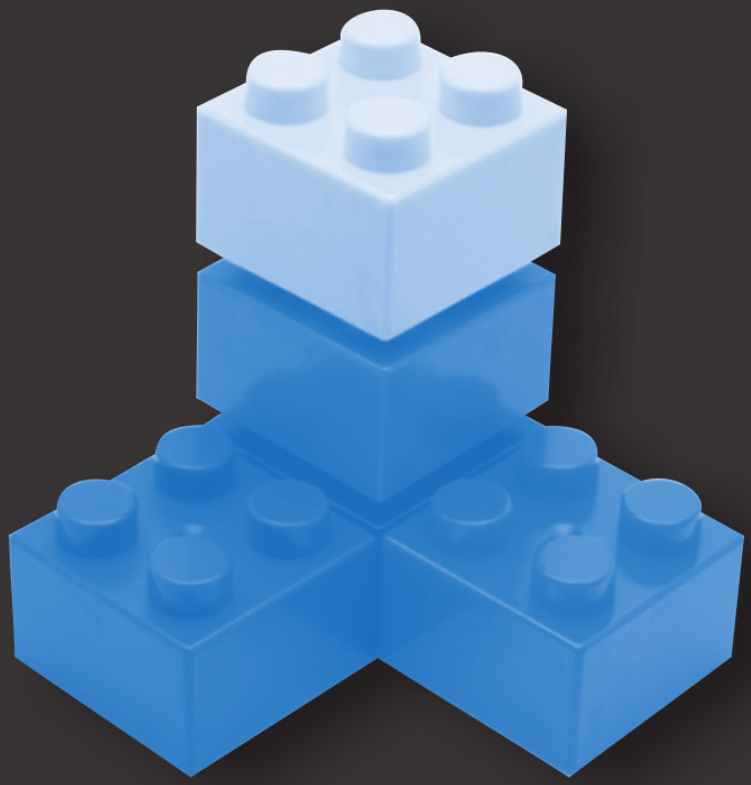
Anton Smirnov

How to build test automation framework



How to build a test automation framework.

Learn to create a framework for automation testing fundamentals fast.

This version was published on 2021-02-22

Contact details:

antony.s.smirnov@gmail.com

Related Websites:

- How to build a test automation framework: https://test-engineer.site/
- Author's Software Testing Blog: https://test-engineer.site/

Every 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.

Table of Contents

Introduction.

Pretty often you can see test automation framework successfully running tests and reporting results but not doing what it's supposed to do: providing a reliable way for team members to build automated tests, and get reliable results.

This often happens when a test automation framework is built without planning in advance and understanding how it will be used.

At first, the team realizes that they need automated tests. One of the engineers decides to take care of it (or gets assigned) — using the tools they are familiar with; they automate the first bunch of tests.

Since initially, it's a proof of concept, some things are being implemented via the fastest and most obvious solution, which is not always utilizing the industry's best practices. Such solutions introduce technical debt. If not addressed early, the impact of technical debt grows once the framework is expanded.

As a result, few iterations later, the team gets a test automation framework that can pretty well-run tests that were in the mind of the author building it. But making a step aside, expanding coverage to additional features, or trying to get other engineers owning tests creation via such framework becomes a challenging task.

Have you ever wondered how to set up a test automation framework? Well, in this book you will learn about everything you'll need to successfully create such a framework. We're going to look at the pros and cons of preconfigured testing environments and those that are created dynamically.

This book is based on more than 7+ years of experience in the field of test automation. During this time, a huge collection of solved questions has accumulated, and the problems and difficulties characteristic of many beginners have become clearly visible. In the course of working in different places, I have repeatedly had to create a framework for testing automation from scratch. It was obvious and reasonable for me to summarize this material in the form of a book that will help novice testers quickly build an automation testing framework on a project and avoid many annoying mistakes.

This book does not aim to fully disclose the entire subject area with all its nuances, so do not take it as a textbook or Handbook — for decades of development testing has accumulated such a volume of data that its formal presentation is not enough, and a dozen books.

Also, reading just this one book is not enough to become a "senior automated testing engineer". Then why do we need this book?

First, this book is worth reading if you are determined to engage in automated testing – it will be useful as a "very beginner" and have some experience in automation.

Secondly, this book can and should be used as reference material.

Thirdly, this book — a kind of "map", which has links to many external sources of information (which can be useful even experienced automation engineer), as well as many examples with explanations.

This book is not intended for people with high experience in test automation. From time to time, I use a learning approach and try to “chew” all the approaches and build the stages step by step.

Some people more experienced in software test automation also having may find it slow, boring, and monotonous.

This book is intended for people who first approach the creation of an automation testing framework, especially if their goal is to add automation to their test approach.

First of all, I wrote this book for a tester with experience in the field of “manual” software testing, the purpose of which is to move to a higher level in the tester career.

Summary:

We can safely say that this book is a kind of guide for beginners in the field of automation software testing.

I have a huge knowledge of the field of test automation. I also have quite a lot of experience building automation on a project from scratch.

I have repeatedly had to develop and implement the framework of testing automation on projects.

The learning approach focuses on a huge chunk of theory on building the automation testing framework. The book also discusses the theory of test automation in detail.

However, the direction of automation to support testing is no longer limited to testing, so this book is suitable for anyone who wants to improve the use of automation: managers, business analysts, users, and, of course, testers.

Testers use different approaches for testing on projects. I remember when I first started doing testing, I was drawing information from traditional books and was unnecessarily confused by some concepts that I rarely had to use. And most of the books, to my great regret, did not address the aspects and approaches to test automation. Most books on testing begin by showing how you can test a software product with basic approaches. But I do not consider the approaches and implementations of test automation at the testing stage.

My main goal is to help you start building an automation testing framework using a strategy and have the basic knowledge you need to do so.

This book focuses on theory rather than a lot of additional libraries, because once you have the basics, building a library and learning how to use it becomes a matter of reading the documentation.

This book is not an "exhaustive" introduction. This is a guide to getting started in building an automation testing framework. I focused on the examples.

I argue that in order to start implementing an automation testing framework, you need a basic set of knowledge in testing and management to start adding value to automation projects.

In fact, when I started creating the automation testing framework first, I used only the initial level of knowledge in the field of testing and development.

I also want the book to be small and accessible so that people actually read it and apply the approaches described in it in practice.

Acknowledgments.

This book was created as a “work in progress” on **leanpub.com**. My thanks go to everyone who bought the book in its early stages, this provided the continued motivation to create something that added value, and then spends the extra time needed to add polish and readability.

I am also grateful to every QA engineer that I have worked with who took the time to explain their approach. You helped me observe what a good QA engineer does and how they work. The fact that you were good, forced me to ‘up my game’ and improve both my coding and testing skills. All mistakes in this book are my fault.

«After many years of my work, I realized a very important thing. Little changes in this world. New technologies come and go. But the basic principles remain the same».

I wrote my first test framework back in 2014. It is still in use today. Yes, it added new versions, new modules. But the structure and the idea has not changed.

It got me to think back in 2014 were well-known principles laid down in this framework. These principles to this day are absolutely the same, for 7+ years nothing has changed. I have not once analyzed existing information, existing frameworks. And my conclusion is just conceptual nothing has changed. All articles and books of 2014 are still relevant. Yes, we live a little at a different time, at a different speed, cloud, mobile devices, technologies are developing at a great speed, but the basic principles remain.

And it is about these basic principles I want to tell you in this book.

In this book, there will not be much programming or description of a huge number of patterns and instructions on how to write code. In this book, I have tried to focus more on the concept of how and with what vision you need to approach the creation of an automation testing framework.

I will try to answer the following questions:

What is an automation testing framework?

What components does the automation testing framework consist of?

What should I take into account when building an automation testing framework?

How to make a large and good automation testing framework?

However, it is worth making some clarity. For some reason, there is not always a need to write an automated test framework. Sometimes you just need to write a few automated scripts using Selenium WebDriver. And this situation is absolutely normal. However, in some projects, this is sometimes not enough and you have to create an automated test framework. Much depends on the requirements and external factors.

In this book, I will not write about commercial products. But all the rules that exist for building a non-commercial automated framework similarly apply to commercial ones. As a rule, the only difference is that commercial products give a large package of components at once and they are usually already all integrated together. The user of this product, as a rule, can no longer add anything new. But the architectural approaches of commercial frameworks are similar to non-commercial ones.

In this book, most of the examples will be in Java. But since the book talks about architectural principles, they apply to any programming language.