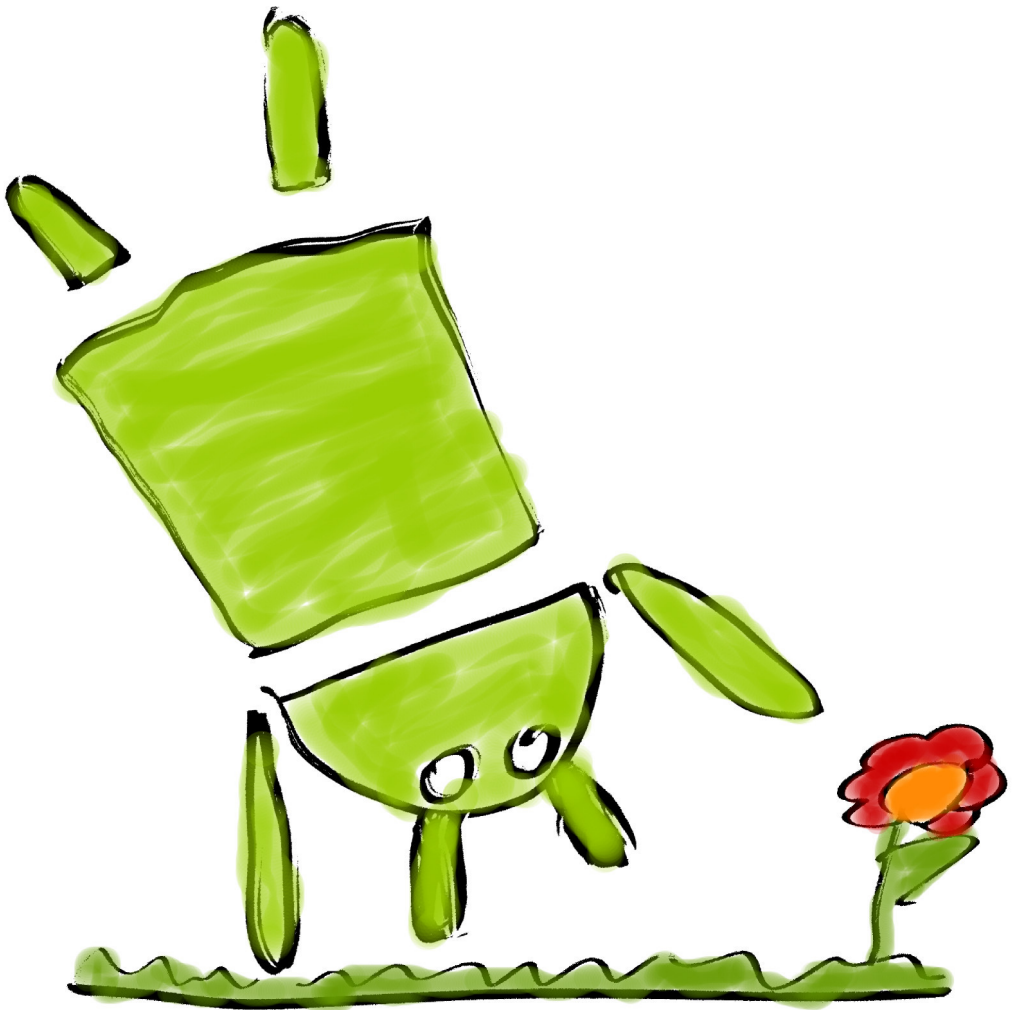


# Agile Android Software Development

Writing Apps faster & better using Agile



*Étienne Savard*



# Contents

<b>Legal and Disclaimers</b>	<b>7</b>
License . . . . .	7
How to support this book . . . . .	7
Limited Warranty . . . . .	8
Limitation of Liability . . . . .	8
Trademarks Disclaimer . . . . .	8
<b>Acknowledgments</b>	<b>9</b>
Donators . . . . .	9
Cover image and illustrations . . . . .	10
<b>Preface</b>	<b>11</b>
About this book . . . . .	12
Who should read this book . . . . .	12
Philosophy and guidelines . . . . .	13
How this book is organized . . . . .	14
How to use this book . . . . .	15
Conventions used in this book . . . . .	15
Recommended references . . . . .	16
About the author . . . . .	16
<b>Part 1 : The Agile Android Toolbox</b>	<b>19</b>
Getting Started with Docker . . . . .	20
Introduction to Android Studio . . . . .	23

Quick Intro to Git . . . . .	23
Getting to know Gradle . . . . .	23
Android Emulators . . . . .	23
Testing Android on real devices . . . . .	23
Configuring our test environment . . . . .	23
Testing the Test Environment . . . . .	23
Our first red-green-refactor loop . . . . .	24
Static Code Analysis Setup . . . . .	24
<b>Part 2 : Automation 101</b>	<b>25</b>
Droplet setup on Digital Ocean . . . . .	26
Setting up Jenkins . . . . .	26
Useful Jenkins plugins for Android Development . . . . .	26
launching a first CI job . . . . .	26
GitLab Installation on our Droplet . . . . .	26
Integration between GitLab and Jenkins . . . . .	26
Setting up a first CI job for Fizz Buzz . . . . .	26
Adding TDD to Fizz Buzz . . . . .	26
Publishing APKs in Jenkins for beta testing . . . . .	27
SonarQube Installation and Setup . . . . .	27
Integration between SonarQube and Jenkins . . . . .	27
Setting up email notifications . . . . .	27
Receiving build notifications on Android . . . . .	27
Securing Web access using SSL . . . . .	27
<b>Part 3 : Agile Software Project Management</b>	<b>29</b>
Agile project management tools . . . . .	30
Setting up Redmine . . . . .	30
Redmine Integration with our environment . . . . .	30
Installing the BackLog plugin in Redmine . . . . .	30
Quick Introduction to the Backlog plugin . . . . .	30
App design mockup . . . . .	30

CONTENTS	5
BDD and Android	30
Simplifying using code injection	30
Android source code modularization	31
Automatization using Gradle	31
<b>Part 4 : Lean Android</b>	<b>33</b>
DRY-ing out the Android development process	34
Setting up an efficient Build-Measure-Learn loop	34
<b>Part 5 : Scaling</b>	<b>35</b>
Automation strategy	36
On-demand Jenkins slaves	36
Automated deployment	36
Large scale testing on real devices	36
Team collaboration	36
Offshore development : how to find great collaborators	36
Multiple target using the same source code	36
<b>Part 6 : Advanced Topics</b>	<b>37</b>
Reducing time to market	38
Monetization	38
Low cost marketing tips	38
How to deal with legacy apps	38
<b>Appendix : About technological watch and agility</b>	<b>39</b>
How to observe a technological watch	40
Selecting the best tool for the job	43
The Agile Android Developer Philosophy	48



# Legal and Disclaimers

## License



Agile Android Software Development by Etienne Savard is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

For commercial use of this book, please contact the author for inquiries (see the book website for contact information): <http://agiledroid.com>

## How to support this book

### Donations

This book is free (as in free speech, not free beer). [Donations are welcome](#), and a great way to show your support for my work. All donators will be credited in the acknowledgments section of this book. You can give any amount you like.

### Opinions

Please share your opinions and comments about the book by answering [this short survey](http://goo.gl/4xosAm): <http://goo.gl/4xosAm>

### Referral Links

In the electronic versions of this book (pdf, mobi, epub, etc.), you will find referral links in the text. These links allow the author to get paid if you end up buying the products to which they refer. At any time, you can decide to copy the link text in your web browser rather than clicking the hyperlink to directly access the product page or service.

## **Limited Warranty**

Symbiosoft warrants that the ebook files are free from defects in the operational sense that they can be read by a PDF Reader, ePub reader, Kindle, or other. EXCEPT FOR THIS EXPRESS LIMITED WARRANTY, SYMBIOSOFT MAKES AND YOU RECEIVE NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND SYMBIOSOFT SPECIFICALLY DISCLAIMS ANY OTHER WARRANTY INCLUDING THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS OR A PARTICULAR PURPOSE. SYMBIOSOFT DOES NOT WARRANT THAT THE OPERATION OF THE EBOOK WILL BE UNINTERRUPTED OR ERROR FREE. If the ebook was purchased in the United States, the above exclusions may not apply to you as some states do not allow the exclusion of implied warranties. In addition to the above warranty rights, you may also have other rights that vary from state to state.

## **Limitation of Liability**

IN NO EVENT WILL SYMBIOSOFT BE LIABLE FOR ANY DAMAGES, WHETHER ARISING FOR TORT OR CONTRACT, INCLUDING LOSS OF DATA, LOST PROFITS, OR OTHER SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE EBOOK.

## **Trademarks Disclaimer**

Symbiosoft logo and name are trademarks of Symbiosoft (9250-2780 QUEBEC INC.).

Product names, logos, brands and other trademarks referred to within this work are the property of their respective trademark holders. These trademark holders are not affiliated with Symbiosoft, our products, or our websites. They do not sponsor or endorse our materials. Below is a partial listing of these trademarks and their owners. This list is subject to change without notice.

Android is a trademark of Google Inc. Portions are modifications based on work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

Opinions expressed in this eBook are those of Etienne Savard, and may not be shared by Symbiosoft's customers.



# Acknowledgments

## Donators

You will find below a list of all donators that contributed money to support this free book. Your support to this project means a lot, and best efforts were deployed to meet your expectations while writing this book.

Thank you!

## Early adopters

Below are the first donators when this book was still in its infancy. Thank you to have believed in this project early on. You drived me forward during this journey.

- David Conde Ortega
- Ryan De Gruyter
- Francis Perreault
- Douglas Carvalho
- Elaine Dias Batista
- [Juan Jose Sanchez Silos](#)
- François Roy
- Francis Toth
- Carmen Radu
- Joe Jimenez
- Kevin Berglund

## Kickstarter campain contributors

Coming soon!

**How to be listed here?**

You can add your name to this list too by [donating](#).

**Cover image and illustrations**

All illustrations of this book, unless specified otherwise, are the original creations of either my daughters, Lea and Marianne, or myself.

Lea draw the cover image for me on the whiteboard after she saw the Android mascot somewhere in my home office. I colored and edited the original drawing using GIMP.

Thanks Lea & Marianne.

# Preface

Android can be a lot of things depending on your point of view.

For Google, Android is a vehicle to expand their business into new areas where they were absent a few years ago. Google has evolved from a web search engine into a company with many product lines and businesses—from mobile phones, music, and books to video-on-demand. With the release of Android Lollipop, you can also add TV and wearables to that list.

For mobile manufacturers—such as Samsung, LG and HTC—Android is an Open Source mobile operating system (OS) they can tweak and use without paying royalties.<sup>1</sup>

For users, the Android OS is less visible but is nevertheless a critical piece of their daily activities. They depend on Android to send and receive emails, chat, take pictures, entertain, and much more. Average users are not power users, and as such they will want quality apps that will run without crashing.

For us as developers, Android is a way to express our creativity (outside of our 9–5 routine), learn mobile software development, and make some money on the side. Regardless of the reason that brings you to Android software development, you are (or will be) faced with the same problems that fellow Android developers encounter every day. These include: fragmentation, legacy code, bad code, ineffective monetization, and so forth. Most of these burdens can be avoided or mitigated by using the right tools and techniques combined with an agile mindset. These effective techniques and tools can help you enjoy Android programming even more by freeing up your time, so you can really develop—instead of just fixing problems, looking for answers on StackOverflow, avoiding pitfalls, or rewriting the same boiler plate code over and over again.<sup>2</sup>

By the end of this book, you should be a more productive and, therefore, happier Android software developer. You would be surprised what a happy developer can accomplish!

---

<sup>1</sup>If you omit the fact that Microsoft collected Android patent royalties from Android smartphone manufacturers. (<http://www.businessinsider.com/microsoft-earns-2-billion-per-year-from-android-patent-royalties-2013-11>)

<sup>2</sup>My definition of boiler plate code is this: Any code that doesn't add value to the final product. In our case, the product is an Android app. In Lean Startup terms, it's synonymous to waste (waste of time, resources, money).

## About this book

When I started learning Android programming, I sometimes found myself confronted by a daunting task and wondering, “Is there a more agile way to do this or that?” I’ve been searching for answers in Android programming books expecting to find something about the tools I use daily in my job but used in the context of mobile application development. Yet I have found nothing.

Why is that? There are a bunch of great tools out there that can make your Android software development experience more enjoyable. Many of these tools are free and Open Source, but they are not covered in any book. My theory is that most Android programming books are targeted at novices and seldom at intermediate-level Android developers (our audience here). I think you have to experience (and even suffer a bit) with Android app development before you start looking around for more agile, alternative ways of doing things.

In this book, we’re covering most of the tools normally described in Application Lifecycle Management (ALM) or DevOps<sup>3</sup> books, but used in the context of Android software development. Think of DevOps for a team of one!

I also share my experiences about monetization, marketing and other things not related to Android programming per se, but essential to successfully publishing an app in Google Play or other Android app stores.

Because of the number of topics covered in this book, I will not throw a bunch of code in your face (lucky you). Instead, I’ve selected a simple example that we will use from the first chapters until the end of the book—the FizzBuzz app.<sup>4</sup> The Fizz Buzz algorithm is great to teach developers how to do test-driven development (TDD), and it is simple enough to let us focus on the techniques rather than our problem-solving skills. We evolve this app from a wireframe concept on paper to a fully functioning app that we release to Google Play. All of the concepts demonstrated in this book are usable for other apps you develop. At the end of each chapter, there are some hints to guide you on how to use the concepts to your own benefit.

## Who should read this book

This book is targeted at experienced software developers who are familiar with at least one Object Oriented Programming (OOP) language, such as C++, C#, or Java. You should also have some basic understanding of source code control (preferably Git), shell scripting and Linux (if you know how to open a command shell, you will do great).

The techniques and tools shared in this book are presented mostly in the context of a team of one—that is, one indie developer who wants to take his Android development workflow to the next level. The concepts presented in this book can also apply to larger teams, but this is not our main audience.

To get the most from this book, you should already have read one or more entry-level books about Android programming (see the following [Recommended References](#) section), and you should

---

<sup>3</sup>For details, see the DevOps definition on Wikipedia: <http://en.wikipedia.org/wiki/DevOps>

<sup>4</sup>See the Fizz Buzz article on Wikipedia for details: [http://en.wikipedia.org/wiki/Fizz\\_buzz](http://en.wikipedia.org/wiki/Fizz_buzz)

already have released one or more Android apps to Google Play.

This book does not cover Android basics, but it does its best to address all those questions you may end up with after we present advanced topics. In the case where some intermediate to advanced knowledge is required to fully understand a topic, you are provided with references to look up.

Additionally, if you have some experience with Agile software projects, this will give you a head start, but it is not required.

Finally, if you are an experienced developer working full-time for your boss's software company—and dreaming of doing your own things—you will find food for thought in this book. Likewise, if you're looking to improve your Android programming workflow, learn new tools and techniques, develop more agile apps—and even make some money with Android—this book is definitely for you!

## Philosophy and guidelines

Big disclaimer: This book is, at times, very opinionated. Mea culpa, the core of this book comes from my years on the IT battlefield, and I've learned the hard way what works and what doesn't.

I beg you to differ and share your view about any subject covered in this book in the discussion group (see the following [How to use this book](#) section). For this book, I have positioned some tools that I find work great in the context of Android project development. They may not be the best tools on the market, but they are the best I have found to get the job done. (They also happen to be the tools I know and use every day). I will explain the process I use to select tools among many equivalents in the chapter on technological watch. You will be able to use these concepts beyond Android app development

I've developed the book examples on Linux and haven't fully tested these on Windows or OSX. However, due to the portable nature of Android (and Java), they should work out-of-the-box on any platform. Generally, for the sake of agility, I do my best to select tools that are platform-independent or web-based. If some examples or scripts don't work as expected in your development environment, I encourage you to submit a bug report to me with as much detail as possible (and if you happen to find a solution specific to a platform, send that as well). If I can help you, I will, or another reader might be able to give you a hand.

I designed the content of this book as a greenfield project (that is from scratch or without the legacy) and for a team of one or a "two-pizza team"<sup>5</sup> in mind. At times, the tools and processes I present will seem like killing a fly with a bazooka if you're a lone developer, but they will help you structure your work and keep you organized. In fact, if you end up growing your app business (and I hope you will!), you will be grateful that you have observed a Spartan discipline with your Android projects because the best practices already will be assimilated by you and your team, and they will serve as the cornerstone of your Android business.

---

<sup>5</sup>An expression coined by Jeff Bezos, CEO of Amazon. A small team (usually between 6 to 10 people) you can feed with two pizzas.

All designs, processes, examples, concepts, and such presented in this book had to meet the following criteria before being included:

- Must be scalable;
- Must respect the Don't Repeat Yourself (DRY) principle and be automated if possible;
- Must be agile (that is, tolerant and adaptable to change);
- Must reduce or (better) eliminate waste (for example, time, money, resources);
- Must not get in the way of the developer.

All of this was taken into account for the sake of helping you reduce the time-to-market for your apps and let you focus on developing great ones.

## How this book is organized

This book is organized in six parts:

1. The Agile Android Toolbox
2. Automation 101
3. Agile Software Project Management
4. Lean Android
5. Scaling
6. Advanced Topics

In Part 1, we set up and present the tools I propose to fill your Agile Android developer toolbox. By the end of Part 1, you will have set up a development environment that will help you build our example application. In Part 2, building upon what we've learned in the previous chapters, I teach you how to optimize your development process by automating tedious or repetitive tasks, so we can concentrate on tasks with added value in building our final app. Then, in Part 3, we look at software management using Agile tools and methodology. We also talk about subjects like behavior-driven development (BDD) and TDD to help us document our code and designs. In Part 4, we locate and eliminate sources of waste. We follow some theories provided by the Lean Startup movement and learn how to implement an effective feedback loop (Build-Measure-Learn loop) for Android software projects. In Part 5, I show you how to use cloud computing to your advantage and scale your builds and tests to multiple environments. Finally, in Part 6, I talk about more advanced topics, such as marketing for your apps, publishing, and monetization tips and techniques.

In the appendix, I present less technical subjects that are useful but not at the core of Agile Android software development. I present the techniques I use to monitor the evolution of the market, so I am alerted when new trends or opportunities surface. I also present how to select tools among equivalent alternatives.

I hope you enjoy the ride!

## How to use this book

The first three parts of this book are meant to be followed in order like a tutorial. First, we put in place a development environment to support the Agile workflow we propose. Then, in the subsequent parts, we use that environment to build our example application and to improve it as we progress. You can always skip some content if you are already familiar with it. The Git repository is designed to provide you with the example app at a working state corresponding to a specific chapter. So you can pick it up from there and continue with more advanced topics without getting lost. If we think a chapter can be skipped by the more advanced readers (without impacting their comprehension of the following chapters), you will see an optional icon marking that chapter (see the following [Conventions used in this book](#) section).

The book was written with the idea of releasing it to the community in parts as it was written. That way, I was able to build a Build-Measure-Learn loop from the beginning à la Lean Startup.<sup>6</sup> This book will continue to develop even after it is released, and it will be updated regularly using feedback from readers and by following the new trends the Android ecosystem brings us. So, as a complement to this book, I strongly encourage you to register—for free—to the following communities to stay informed of new events related to Agile Android Software Development:

- Google Group: <https://groups.google.com/forum/?hl=fr#!forum/agile-android-software-development>
- Google+ community: <https://plus.google.com/u/0/communities/106666596834203709065>
- Mailing List: <http://eepurl.com/XTV8H>

You should also register for a free account on GitHub to access the Agile Android Software Development repository : <https://github.com/Agile-Android-Software-Development>. You will be able to get all the examples, scripts, and much more from the Git repository.

## Conventions used in this book

The following conventions are used in this book:

Constant width

Used for source code listings and reference of class, variable, parameter names, and such referring to code snippets.

Constant width in bold

Indicates source code that was modified to illustrate an improvement to a previously presented source code listing.

Italic

Used when referring to URLs, filenames, file extensions, and to put emphasis on technical terms the first time they are cited in a chapter.

---

<sup>6</sup><http://theleanstartup.com/>



This icon indicates a tip or a suggestion.



This icon represents a notice or warning.



This icon illustrates that a section of text can be skipped and is provided only to offer some depth to a topic but is not essential to its understanding.

## Recommended references

If you need to deepen your knowledge or dust-off your Android programming skills, I recommend you the following books :

- Phillips, Bill, and Brian Hardy. **Android programming : the Big Nerd Ranch guide**. Atlanta, Ga: Big Nerd Ranch, 2013. Print. ISBN-10: 0321804333.
- Mednieks, Zigurd R. **Programming Android**. Farnham: O'Reilly, 2012. Print. ISBN-10: 9781449316648.
- Clifton, Ian G. **Android user interface design : turning ideas and sketches into beautifully designed apps**. Upper Saddle River, NJ: Addison-Wesley, 2013. Print. ISBN-10: 0321886739.
- Murphy, Mark L. **The Busy Coder's Guide to Android Development**. eBook. Website: <https://wares.commonsware.com>.

## About the author

Étienne Savard is a dad, a certified Professional Scrum Master (PSM), a pragmatic programmer, and an Open Source advocate. He has been a software developer for a long time. It all started on a TRS-80 Color Computer (when 64K was a lot of memory and 16 colors were enough). Since then, he has continued to pursue his quest to boldly go where no developer has gone before!

Étienne works as an IT consultant helping companies (from startups to large corporations) integrate best-of-breed tools and best practices into their software development process, including most of the same tools and practices presented in this book.

He started Android programming with the Gingerbread version of Android by reading books, following tutorials, and looking at a lot of source code. Étienne has also released a couple of apps



on Google Play with some success and a lot of trial and error. He enjoyed Android programming from the start because of its openness, community, and architecture. Etienne has found that the only missing part in the Android ecosystem is a more agile way of developing Android apps. This book is here to fill that gap.

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

