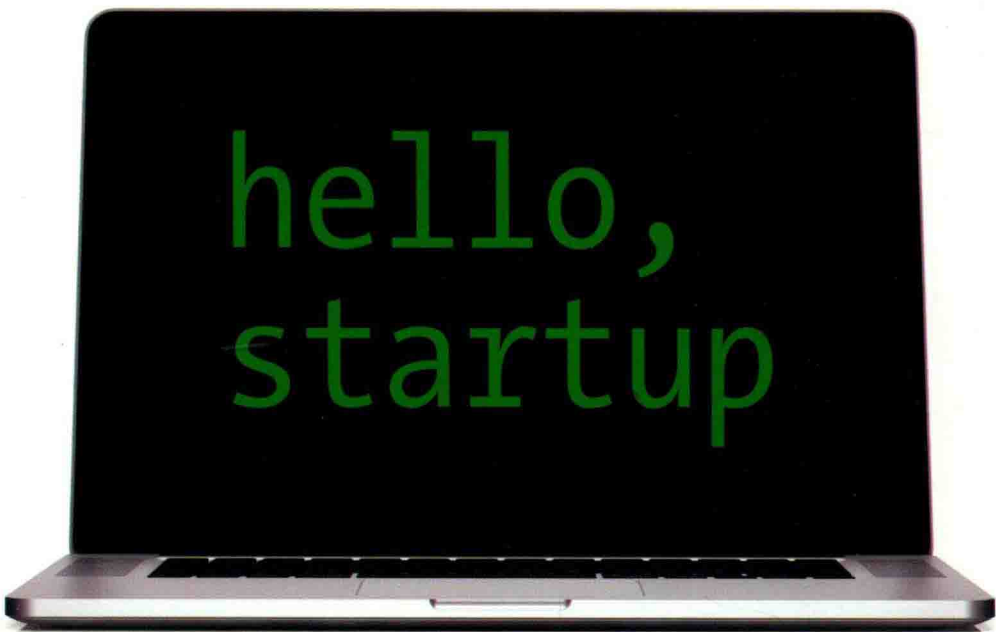


O'REILLY®

“我真希望在自己刚入行的时候能有这么一本书。”

—Jay Kreps, Confluent, Inc. CEO

Yevgeniy Brikman



hello,  
startup

你好, 创业公司

(影印版)

一本构建  
产品、技术、团队的  
程序员指南

东南大学出版社

---

你好, 创业公司 (影印版)

hello, startup

Yevgeniy Brikman 著

Beijing

O'REILLY®

东南大学出版社出版

南京 东南大学出版社

## 图书在版编目(CIP)数据

你好,创业公司:英文/(美)叶夫根尼·布里克曼(Yevgeniy Brikman)著. —影印本. —南京:东南大学出版社,2018.1

书名原文:Hello, Startup

ISBN 978-7-5641-7297-8

I. ①你… II. ①叶… III. ①企业管理—英文 IV. ①F272

中国版本图书馆CIP数据核字(2017)248388号  
图字:10-2017-347号

© 2015 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2018. Authorized reprint of the original English edition, 2017 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2015。

英文影印版由东南大学出版社出版 2018。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有,未得书面许可,本书的任何部分和全部不得以任何形式重制。

你好,创业公司(影印版)

---

出版发行:东南大学出版社

地址:南京四牌楼2号 邮编:210096

出版人:江建中

网址:<http://www.seupress.com>

电子邮件:[press@seupress.com](mailto:press@seupress.com)

印刷:常州市武进第三印刷有限公司

开本:787毫米×980毫米 16开本

印张:37.75

字数:652千字

版次:2018年1月第1版

印次:2018年1月第1次印刷

书号:ISBN 978-7-5641-7297-8

定价:122.00元

---

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

此为试读,需要完整PDF请访问:[www.ertongbook.com](http://www.ertongbook.com)

## Praise for *Hello, Startup*

There is a huge gap between a good CS education and what is considered “common knowledge” in start-up engineering teams. Most of us had to learn this stuff through blogs, co-workers, and, of course, the school of hard knocks. But *Hello, Startup* finally collects a lot of this wisdom into one place. I wish I’d had a book like this when I was getting my start in the industry.

—Jay Kreps, CEO of Confluent Inc.

As a startup founder, you are expected to learn an unbelievable amount of knowledge from an incredibly diverse set of disciplines in a very short period of time. While I treasure the experience of beating my head against the wall and eventually figuring things out, I wish I had this guide when I first got started.

—Boweï Gai, Co-Founder and CEO of CardMunch

Jim will give you a guided and well-informed look at what it takes to build a software startup. There is no jargon or fluff—just practical, simple, and proven advice, presented in an easily readable book. If you’ve ever wondered, “How do I come up with startup ideas?” or “What technology should I use for this project?” or “How do I get awesome startups to hire me?” then *Hello, Startup* is for you.

—Eugene Mirkin, Entrepreneur in Residence at Array Ventures

Like Jim, I started my career working for big, established companies. I'm now the co-founder of my own company, *prismic.io*, and this experience is teaching me a lot every single day. This book captures a lot of these learnings and explains not only how, but also why a startup can be a great way to  
unleash your true potential.

—*Sadek Drobi, Co-Founder of prismic.io*

If every computer science department gave out this book as a graduation gift, two good things would happen across the tech industry: the worst tech companies would go out of business, and the good ones would become dramatically more productive.

—*Brent Vincent, Founder of Adacio*

With the exploding interest in technology startups, *Hello, Startup* is a uniquely practical and entertaining book on how to develop startups. I wish I had this resource at the start of my entrepreneurial journey.

—*Sean Ammirati, Partner at Birchmere Ventures*

*To Mom, Dad, Lyalya, and Molly*

# Preface

```
main( ) {  
    printf("hello, world");  
}
```

When you learn a new programming language, the traditional first step is to go through a “Hello, World” tutorial that teaches you everything you need to know to get a basic program working, such as one that prints the text “Hello, World” to the screen. This book is the “Hello, World” tutorial for building products, technologies, and teams in a startup environment.

I wish I had a book like this when I was in college. By the time I graduated, I had a BS, a master’s, a bunch of internships—and absolutely no idea what I was doing.

One of the first big projects I built by myself was a desktop application for performance testing at Thomson Financial. I had no idea how to create a user interface, so I randomly sprinkled text fields, menus, and buttons across the screen. I had no idea how to reason about performance, so I randomly sprinkled caches and thread pools across the code. And I had no idea how to think about code maintenance, so I didn’t bother with tests or documentation, but I did manage to cram several thousand lines of code into one gigantic file.

My first project at TripAdvisor was to add new sort options to the web page that listed all the hotels in a city. It was a quick task—just enough to become familiar with the codebase—and I was able to get it done and pushed to production in my first week. Shortly thereafter, I was in my manager’s office for our first one-on-one meeting, and I watched as he clicked on the hotel listings for Paris, selected the new sort option, and waited. And waited. And waited. It took nearly two hours for the page to load. Well, it was probably closer to two minutes, but I’m pretty sure there is a law of special relativity that causes time to dilate when you’re sweating profusely and hoping you can melt through the floor and disap-

pear. Later that night—much later—I figured out that my fancy new code was making two database calls every time it compared hotels during the sorting process. It takes on the order of  $n \log n$  comparisons to sort  $n$  items, so for Paris, which has roughly 2,000 hotels, that works out to roughly 40,000 database calls for a single page load. I might not have melted that day, but our database server nearly did.

I remember lots of other nasty bugs, ugly code, uglier user interfaces, site outages, and late nights. But mostly, I remember having tons of questions and no easy way to find an answer. What technologies should I learn and use? Why should I bother with automated tests? How do I build a product that doesn't look terrible? How do I get people to use my product? How do I negotiate a job offer? Should I negotiate for more salary or more equity? What is equity, anyway? Should I work at a large company or join a startup?

I learned the answers to these questions and many others the hard way. I tried to capture what I learned—much of it the result of painful trial and error—in blog posts and talks (<http://www.ybrikman.com>). But after realizing that thousands of other developers were going through the same trial-and-error process, making the same mistakes, and still having nowhere to turn with the same questions, I decided it was time to do something more substantial. This book is the result. Of course, some lessons you can only learn by making your own mistakes, but for the rest, I hope *Hello, Startup* will save you a lot of pain by letting you learn from the mistakes of others.

One of my biggest mistakes was not paying enough attention to the startup world early in my career. My first few jobs were all at big, established institutions (Cisco Systems, Thomson Financial, Cornell University), and it was only later and somewhat accidentally that I made the jump into the startup world (LinkedIn, TripAdvisor). What I saw there astonished me. I learned more in my first few months at a startup than I had in all the years of work, internships, and schooling before that.

A startup is not just a smaller version of a larger company [Blank and Dorf 2012, xvii], much like quantum mechanics is not just a smaller version of classical mechanics. Classical mechanics describes the behavior of large objects (e.g., a baseball or planet) moving at relatively slow speeds with rules that are predictable and deterministic. Similarly, large companies tend to move slowly and live in a world with predictable rules because the customers and products are all known. Quantum mechanics describes the behavior of tiny objects (e.g., photons and electrons) moving at extremely high speeds with rules based on uncertainty,

probability, and non-determinism. Similarly, startups tend to move at a hectic pace to survive in an unpredictable world where nothing is known. While many people are familiar with classical mechanics and large companies, you won't have a complete picture of the world unless you also understand quantum mechanics and startups, and that means you'll need new ways of thinking and new ways of working.

In many ways, the way you work is the way you live, as you will spend half of your waking hours at work. Wouldn't you prefer to spend those hours doing something that makes you happy? I used to think that all software jobs involved endless cubicle farms, pointy-haired bosses, TPS reports, and enterprise code. Fortunately, there is an alternative, which I hope to show you in this book by introducing you to the way work gets done at some of the best startups in the world. The ideas from these companies will be useful even if you never join a startup, and will grow even more useful as startups become more and more ubiquitous.<sup>1</sup>

This book is based on my own experiences plus a considerable amount of research, including a series of interviews with programmers from some of the most successful startups of the last decade, such as Google, Facebook, Twitter, GitHub, Stripe, Instagram, Coursera, Foursquare, Pinterest, and Typesafe (the full list of interviewees is in "Interviews" on page 20). You'll find their stories and thoughts throughout the book. They offer a raw glimpse into startup life, with no marketing or PR spin—just programmers sharing their successes, mistakes, and advice.

## What will you find in this book?

My goal with *Hello, Startup* was to create a practical, actionable, how-to guide to startups. The book consists of three parts: Products, Technologies, and Teams. In the following sections, I've listed the chapters you'll find in each part, and for each chapter, I've outlined the concrete techniques, tools, and skills you will learn from reading it.

---

<sup>1</sup> In the United States alone, nearly 500,000 small businesses are created per month [Kauffman Index of Entrepreneurial Activity Interactive 2014] and they account for more than 66% of new jobs [Small Business Trends 2014].

## **PART I: PRODUCTS**

### *Chapter 1, Why Startups*

Why today, more than any other time in history, startups are your best opportunity to build products that touch millions of lives; what is a startup; why you should work at a startup; why you shouldn't work at a startup.

### *Chapter 2, Startup Ideas*

How to come up with startup ideas; idea journals, constraints, and pain points; ideas versus execution; Boyd's Law; how to use the customer development process to quickly and cheaply validate your ideas.

### *Chapter 3, Product Design*

Design skills everyone should learn; how to design a user interface that doesn't make users feel stupid; principles of user-centered design, including personas, emotional design, simplicity, and usability testing; principles of visual design, including copywriting, reuse, layout, typography, contrast and repetition, and colors; how to design a minimum viable product (MVP).

### *Chapter 4, Data and Distribution*

Metrics every startup should measure; data-driven product development; A/B testing; why the best product doesn't always win; marketing, viral growth, and sales strategies for startups.

## **PART II: TECHNOLOGIES**

### *Chapter 5, Picking a Tech Stack*

How to decide whether to build software in-house, buy a commercial product, or use open source; how to choose the initial tech stack; how to evolve a tech stack and rewrites; how to evaluate programming languages, frameworks, and databases.

### *Chapter 6, Clean Code*

Why a programmer's job is not to write code but to understand it; how code layout, naming, error handling, Don't Repeat Yourself (DRY), the Single Responsibility Principle (SRP), loose coupling, and high cohesion lead to code that's easier to understand; why functional programming leads to code that's easier to reuse; why refactoring is the essence of writing code well.

### *Chapter 7, Scalability*

How to scale a startup to more users and more developers; how to make changes to your code without being afraid; how Test-Driven Development (TDD) leads to better code; how to introduce design reviews, pair coding, and code reviews at your startup; why the readme is the most important file in your codebase; if you can't measure it, you can't fix it; how to do back-of-the-envelope calculations to reason about performance.

### *Chapter 8, Software Delivery*

What happens to the code after you write it; why you should use source control, an open source build system, and continuous integration; how to set up configuration management, automated deployment, and continuous delivery; how to instrument your code with logging, monitoring, and alerting.

## **PART III: TEAMS**

### *Chapter 9, Startup Culture*

Why you should define your company's mission and values; trade-offs between a management-driven hierarchy and a flat organization; the role of culture in hiring, promotions, and motivation; how to design the ideal office for programmers; trade-offs with working remotely; communication policies and process at a startup.

### *Chapter 10, Getting a Job at a Startup*

How to find a startup job using your network; how to get your résumé noticed; how to get good at interviews, whiteboard coding, and asking good questions; how to think about salary and equity; how to negotiate a job offer.

### *Chapter 11, Hiring for Your Startup*

Why people are the most important part of a startup; who to hire, including co-founders, early hires, generalists, and specialists; how to find great candidates (and how to perfect your company branding so that they can find you); why whiteboard coding is a terrible interview process (and the alternatives you should use instead); how to make an offer they can't refuse.

### *Chapter 12, Learning*

The most interesting software developer in the world; why you should write blog posts, articles, papers, and books; why you should talk at meetup

groups, tech talks, and conferences; why you should open source almost all of your code; why you should share almost everything you know.

## Key ideas

In addition to the concrete techniques, tools, and skills just listed, three key ideas that are essential for startup success are repeated again and again throughout the book: startups are about people, great companies are the result of evolution, and speed wins.

### STARTUPS ARE ABOUT PEOPLE

*The major problems of our work are not so much technological as sociological in nature.*

—[DEMARCO AND LISTER 1999, 4], TOM DEMARCO AND TIMOTHY LISTER, *PEOPLEWARE*

Most of what you learn in classes and books on startups—such as marketing plans, product design, system design, testing strategy, hiring plans, and organizational design—are just the outputs of a startup. You cannot fully understand startups solely by studying the outputs, just as the prisoners chained up in Plato’s Cave cannot fully understand the outside world solely by studying the shadows it casts on the wall in front of them [Plato 2008, Book VII].

This book will try to get you to step out of the cave and look not only at the outputs but the people who create them. You’ll not only learn how to come up with great product designs but also how to design products for people. You’ll see not only how to write effective automated tests but also why automated tests are essential for allowing people to change code without fear. And you’ll learn not only how great companies are organized but also why the most important part of building a great company is knowing how to find and motivate the right people.

### GREAT COMPANIES ARE THE RESULT OF EVOLUTION

*A complex system that works is invariably found to have evolved from a simple system that worked.*

—[BOOCH 1991, 11], JOHN GALL

When you see the neck of a giraffe, you have to realize that it isn’t long because of some plan put together by a divine being at the beginning of time. The neck started out small and grew longer over thousands of generations as the result of random mutations that happened to increase the giraffe’s odds of survival in a

particular environment. Likewise, when you see a successful company, you have to realize that it didn't succeed because of some plan the founders put together when they created the company. Most startups change and grow over the course of thousands of experiments, some of which happen to increase the company's odds of survival in a particular market, and the result usually looks little like what the founders originally envisioned.

Instead of worrying about how to come up with a perfect plan (think *Waterfall development*), this book will focus on how to build a startup using incremental and iterative development (think *Agile* and *Lean*). Whether you're building products, technologies, or teams, you'll see that the best way to start is by building the smallest thing that can possibly work—a *minimum viable product* (MVP)—and then gradually evolving it into something bigger based on feedback from your customers (for products), code reviews and tests (for technology), and employees (for teams).

## **SPEED WINS**

*The world is changing very fast. Big will not beat small anymore. It will be the fast beating the slow.*

—RUPERT MURDOCH

If great companies are the result of evolution and iterative development, then the company that can iterate the fastest will win. Many of the ideas in this book are about how to iterate faster—how to shorten the feedback loop so you can speed up your learning, and ultimately, your evolution. Customer development helps you find product/market fit faster. Clean code and automated tests let you build technology faster. A strong culture helps you build a team more quickly. And somewhat counterintuitively, you will see later in this book that building things faster will also allow you to build things that are higher quality. Speed wins.

## **This book covers a lot of ground**

Entire books, even multiple books, have been written about each of the topics and themes in *Hello, Startup*, so this book focuses on the most fundamental ideas to get you started—a “Hello, World” tutorial, so to speak—and refers you to additional resources for when you're ready to learn more. It's also worth mentioning the topics this book does *not* cover: namely, the legal and financial aspects of a startup. If you're interested in the details of writing business plans,

raising money from investors, or filing for an IPO, you'll want to read some of the books listed in "Recommended reading" on page 529.

Of course, reading a book on startups, by itself, can no more make you a great developer or founder than reading a book on exercise can make you a great weightlifter. An exercise book can teach you specific routines and exercises, but the first time you step up to a barbell, you'll still be a weakling. It's only after spending thousands of hours in a gym practicing, sweating, and applying what you read that you'll be able to move some serious weight. Likewise, the goal of this book is to teach you the tools and techniques you'll need in a startup, but you'll need to invest a lot of time practicing these techniques before you get them right.

That said, it's not really about right or wrong. All models are wrong, but some are useful [Box and Draper 1987, 424]. The tools and techniques in this book have proven useful for startups in the past and I hope you will find them useful in the future. Think of them not as canned solutions, but as a way to build a vocabulary for thinking about certain types of problems and coming up with your own solutions.

## Who should read this book?

**If you work in a startup or are considering joining the startup world, or you work in a large company and want to run it like a startup**, you should read this book. It will introduce you to all the basic ideas you need to build a successful company—and a successful career—in the world of rapidly changing, uncertain tech ventures. Although at its core this is a book for programmers, by programmers, only Part II, Technologies, is significantly technical. Part I, Products, and Part III, Teams, should be accessible to technical and non-technical audiences alike.

**If you're a programmer at the start of your career**, this book is for you. *Hello, Startup* is a collection of everything I wish I knew when I was in college—all the advice, tips, and tricks I wish someone had told me when I was starting my career. As a young programmer, you probably know a couple of programming languages; perhaps you've mastered a few libraries and frameworks; maybe you've built some small apps as part of schoolwork. Suddenly, someone is offering you money for your skills, and you will soon find out if what you learned in school holds up in the real world. Let me save you the suspense: it doesn't. You can find that out the hard way and repeat the mistakes that thousands of other programmers have made before you, or you can read through this book and get your career moving in the right direction from day one.

If you're an experienced developer, this book will force you to take a more systematic look at the work you've been doing every day. Are you still forcing interview candidates to traverse binary trees on a whiteboard? Did you select the technology for your latest project based on a gut feeling or the latest hot trend online? Do you have a TODO somewhere that says "write documentation"? Do you feel like your company has gotten too big, too slow, and is no longer innovating? Then you'll find yourself nodding and smiling when you read the stories in this book. You'll be able to apply the advice to your existing job, or perhaps you'll figure out that it's time for a change.

If you are a manager, executive, or investor involved with high-tech companies, this book will help you understand why time estimates are off by an order of magnitude, or why your best developer just jumped ship to another company, or why the latest AgileExtremeScrumPairProgrammingDevOps™ process is not making your team any faster. Much of your success depends on understanding how programmers think, what they actually do all day, and how to motivate them. The stories in this book are the kind of honesty you won't get in a one-on-one with your direct reports.

If you're not already involved with but are interested in startups, this book is an insider's look at how the sausage is really made. Trying to understand a successful startup by studying only the end product (e.g., the website, the mobile app, the cool gizmo) is like trying to understand someone's college experience by studying only the diploma; it's an impressive looking piece of paper, sure, but it doesn't capture the years of classes, study sessions, exams, and homework—or the success and failures—that were necessary to get it. Companies like LinkedIn or Facebook might look simple from the outside, but they are not—and this book will reveal all the creativity, problem solving, and sleepless nights that occur within such companies to make that possible. In short, if you have any interest in startups, this book is for you.

## Conventions used in this book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

***Constant width***

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.


***Constant width bold***

Shows commands or other text that should be typed literally by the user.

***Constant width italic***

Shows text that should be replaced with user-supplied values or by values determined by context.

## Safari® Books Online

 **Safari**® *Safari Books Online* is an on-demand digital library that delivers expert *content* in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

## How to contact us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international or local)  
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at [http://bit.ly/Hello\\_Startup](http://bit.ly>Hello_Startup).

To comment or ask technical questions about this book, send email to [book-questions@oreilly.com](mailto:book-questions@oreilly.com).

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

## Acknowledgments

This book could not have happened without the help of a huge number of people. I decided to write a book in the first place thanks to the advice and help of Joe Adler, Adam Trachtenberg, Joshua Suereth, and Nilanjan Raychaudhuri. My fellow hackers, Florina Xhabija Grosskurth, Matthew Shoup, Prachi Gupta, and Bower Gai, helped me at every step of the way with ideas and feedback. The folks at O'Reilly (and beyond), especially Angela Rufino, Mary Treseler, Nicole Shelby, Gillian McGarvey, and Mike Loukides, helped this newbie writer create something that was worth publishing. And I had lots of help with intros from Peter Skomoroch, Sid Viswanathan, and Jiong Wang, and with legal questions from James Yeagle.

A number of brave volunteers read the early release of this book and fought through the content before it was fully polished and ready: Alistair Sloley, Joseph Born, Clarke Ching, Jay Kreps, Ara Matevossian, Prachi Gupta, Matthew Shoup, Martin Kleppmann, Dmitriy Yefremov, David J. Groom, Molly Pucci, Steve Pucci, Alla Brikman, and Mikhail Brikman. I'm grateful for you taking the time to review my work and for all the feedback and help you offered.