Running Lean

Second Edition

Iterate from Plan A to a Plan That Works

Ash Maurya

O'REILLY®
Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

Contents

Foreword . . . . . **XIII**

Preface . . . . . **XVII**

Introduction . . . . . **XXI**

PART 1: ROADMAP

Chapter 1
Meta-Principles **3**

Chapter 2
Running Lean Illustrated **15**

PART 2: DOCUMENT YOUR PLAN A

Chapter 3
Create Your Lean Canvas **23**

PART 3: IDENTIFY THE RISKIEST PARTS OF YOUR PLAN

Chapter 4
Prioritize Where to Start **49**

Chapter 5
Get Ready to Experiment **57**

PART 4: SYSTEMATICALLY TEST YOUR PLAN

STAGE ONE: Understand Problem

Chapter 6
Get Ready to Interview Customers 71

Chapter 7
The Problem Interview 81

STAGE TWO: Define Solution

Chapter 8
The Solution Interview 95

Chapter 9
Get to Release 1.0 111

STAGE THREE: Validate Qualitatively

Chapter 10
Get Ready to Measure 121

Chapter 11
The MVP Interview. 127

Chapter 12
Validate Customer Lifecycle. 135

STAGE FOUR: Verify Quantitatively

Chapter 13
Don't Be a Feature Pusher 145

Chapter 14
Measure Product/Market Fit 155

Chapter 15
Conclusion. 169

Appendix
Bonus Material 173

Index 197

Running Lean provides a step-by-step blueprint to put these ideas into action. A business plan rests on a series of leap-of-faith assumptions, each of which can be tested empirically. Will customers want the product we're building? Will they pay for it? Can we provide a service profitably? And once we find customers, can we grow? *Running Lean* lays out Ash's approach to breaking these assumptions down so that they can become the subjects of rigorous experiments.

Running Lean's simple, action-oriented templates provide tools that startups in all stages of development can use to help build breakthrough, disruptive new products and organizations.

It's been just about three years since I first wrote the phrase "lean startup" in a blog post that a few dozen people read. Since then, these ideas have grown into a movement, embraced by thousands of entrepreneurs around the world dedicated to making sure that new products and new startups succeed. As you read through *Running Lean*, I hope you will put these ideas into practice and join our community. Odds are there is a Lean Startup Meetup taking place in your city. A complete list of meetups and links to other resources can be found at the official Lean Startup homepage: *http://theleanstartup.com*.

Welcome to the cutting edge of entrepreneurial practice. I hope you'll share what you learn, what works and what doesn't. Thank you for being part of this experiment.

Eric Ries
January 20, 2012
San Francisco, CA

We'd Like to Hear from You

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://shop.oreilly.com/product/0636920020141.do

To comment or ask technical questions about this book, send email to:

bookquestions@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*

Attributions and Permissions

This book is here to help you get your job done. If you reference limited parts of it in your work or writings, we appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Running Lean, Second Edition* (O'Reilly). Copyright 2012 Ash Maurya, 978-1-449-30517-8."

If you feel your use of examples or quotations from this book falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Introduction

What Is Running Lean?

We live in an age of unparalleled opportunity for innovation. With the advent of the Internet, cloud computing, and open source software, the cost of building products is at an all-time low. Yet, the odds of building successful startups haven't improved much.

Most startups still fail.

But the more interesting fact is that, of those startups that succeed, two-thirds report having drastically changed their plans along the way.[1]

So, what separates successful startups from unsuccessful ones is not necessarily the fact that successful startups began with a better initial plan (or Plan A), but rather that they *find a plan that works* before *running out of resources*.

Up until now, finding this better Plan B or C or Z has been based more on gut, intuition, and luck. There has been no systematic process for rigorously stress-testing a Plan A.

That is what Running Lean is about.

Running Lean is a systematic process for iterating from Plan A to a plan that works, before running out of resources.

Why Are Startups Hard?

First, there is a misconception around how successful products get built. The media loves stories of visionaries who see the future and chart a perfect course to intersect it. The reality, however, rarely plays out quite as simply. Even the unveiling of the visionary computer, the iPad, in Steve Jobs' words

1 John Mullins and Randy Komisar, *Getting to Plan B* (Boston, MA: Harvard Business Review Press, 2009).

was years in the making, built on several incremental innovations (and failures) of software and hardware.

Second, the classic product-centric approach front-loads some customer involvement during the requirements-gathering phase but leaves most of the customer validation until after the software is released. There is a large "middle" when the startup disengages from customers for weeks or months while they build and test their solution. During this time, it's quite possible for the startup to either build too much or be led astray from building what customers want. This is the fundamental dilemma described by Steve Blank in *The Four Steps to the Epiphany*, in which he offers a process for building a continuous customer feedback loop throughout the product development cycle that he terms "Customer Development."

And finally, even though customers hold all the answers, you simply cannot ask them what they want.

> *If I had asked people what they wanted, they would have said faster horses.*
>
> *—Henry Ford*

A lot of people cite the preceding quote and declare it hopeless to talk to customers. But hidden in this quote is a customer problem statement: had customers said "faster horses," they would really have been asking for something *faster* than their existing alternative, which happened to be horses.

Given the right context, customers can clearly articulate their problems, but it's your job to come up with the solution.

> *It is not the customer's job to know what they want.*
>
> *—Steve Jobs*

Is There a Better Way?

Running Lean provides a better, faster way to vet new product ideas and build successful products:

- Running Lean is about speed, learning, and focus.
- Running Lean is about testing a vision by measuring how customers behave.
- Running Lean is about engaging customers throughout the product development cycle.
- Running Lean tackles both product and market validation in parallel using short iterations.
- Running Lean is a disciplined and rigorous process.

Running Lean references an array of methodologies and thinkers. Three of the most important follow.

Customer Development

Customer Development is a term coined by Steve Blank and is used to describe the parallel process of building a continuous feedback loop with customers throughout the product development cycle. It is defined in his book, *The Four Steps to the Epiphany*.

The key takeaway from Customer Development can best be summed up as:

> *Get out of the building.*
>
> —*Steve Blank*

Most of the answers lie outside the building—not on your computer, or in the lab. You have to get out and directly engage customers.

Lean Startup

Lean Startup is a term trademarked by Eric Ries and represents a synthesis of Customer Development, Agile Software Development methodologies, and Lean (as in the Toyota Production System) practices.

The term *Lean* is often misunderstood as "being cheap." While "being Lean" is fundamentally about eliminating waste or being efficient with resources, that interpretation is not completely misguided because money happens to be one of those resources.

However, in a Lean Startup, we strive to optimize utilization of our scarcest resource, which is time. Specifically, our objective is *maximizing learning (about customers) per unit time*.

The key takeaway from Lean Startup can best be summed up around the concept of using smaller, faster iterations for testing a vision.

> *Startups that succeed are those that manage to iterate enough times before running out of resources.*
>
> —*Eric Ries*

Bootstrapping

Bootstrapping is more commonly understood as a collection of techniques used to minimize the amount of external debt or funding needed from banks or investors. Too often, people confuse bootstrapping with self-funding. A stricter definition is funding with customer revenues.

However, I subscribe to a much more philosophical definition of bootstrapping put forward by Bijoy Goswami:

> *Right action, right time.*

Startups are inherently chaotic, but at any given point in time, there are only a few key actions that matter. You need to just focus on those and ignore the rest.

What Will This Book Teach You?

In this book, you'll learn:

- How to first find a problem worth solving, before defining a solution
- How to find early customers
- When is the ideal time to raise funding
- How to test pricing
- How to decide what goes into Release 1.0
- How to build and measure what customers want
- How to maximize for speed, learning, and focus
- What is product/market fit
- How to iterate to product/market fit

Is This Book for You?

If you are an entrepreneur considering building a new product, or if you already have a product and you want to raise your odds of making it successful, this book is for you.

Running Lean is for:

- Business managers
- CEOs
- Developers and programmers who are interested in becoming successful entrepreneurs
- Bloggers, cofounders, small-business people, writers, musicians—anyone who's creative and interested in starting a new business project
- Innovators
- Startup founders

How Is This Book Organized?

This book is organized into four parts. The parts are meant to be read in order, as they outline the chronological steps required to apply Running Lean to your product—from ideation to product/market fit. Even if you already have a product launched, I recommend starting from the beginning. You will not have to spend as much time going through the stages, and this exercise will help you baseline where you currently are and formulate your next actions.

Each part ends with gating criteria that will help you decide if you're ready to move on to the next one.

Part 1: Roadmap

Part 1 provides an overall roadmap of the Running Lean process. Specifically, it describes the three core meta-principles that capture the essence of Running Lean and ends with a short case study that helps illustrate these principles in action.

The rest of the book covers each of the following meta-principles in detail in three parts.

Part 2: Document Your Plan A

Part 2 walks through the process of documenting your initial vision (or Plan A) using a portable one-page format called Lean Canvas. Your Lean Canvas will serve as your product's tactical map and blueprint.

Part 3: Identify the Riskiest Parts of Your Plan

Part 3 helps you identify which aspects of your plan to focus on first. It lays some groundwork on the different types of risks startups face, shows you how to prioritize them, and prepares you to start the process of testing and experimentation.

Part 4: Systematically Test Your Plan

Part 4 outlines the four-stage process for systematically stress testing your initial plan and shows you how to iterate from your Plan A to a Plan That Works.

About Me

I bootstrapped my most recent company, WiredReach, in 2002, and sold it in late 2010. Throughout that time, I built products in stealth, attempted building a platform, dabbled with open sourcing, practiced "release early, release often," embraced "less is more,"[2] and even tried "more is more."

The first realization early on was that building in stealth is a really bad idea. There is a fear, especially common among first-time entrepreneurs, that their great idea will be stolen by someone else. The truth is twofold: first, most people are not able to visualize the potential of an idea at such an early stage, and second (and more importantly), *they won't care*.

The second realization was that startups can *consume years of your life*. I started WiredReach with just a spark of an idea, and before I knew it, years had passed. While I've had varying levels of success with the products I built, I realized that I needed a better, faster way to vet new product ideas.

Life's too short to build something nobody wants.

And finally, I learned that while listening to customers is important, *you have to know how to do it*. I used a "release early, release often" methodology for one of my products, BoxCloud, and launched a fairly minimal file-sharing product built on a new peer-to-web model we had developed in 2006. After we launched, we got covered by a few prominent blogs and dumped some serious cash into advertising on the DECK network (primarily targeted at designers and developers).

We started getting a lot of feedback from users, but it was all over the place. We didn't have a clear definition of our target customer and didn't know how to prioritize this feedback. We started listening to the most popular (vocal) requests and ended up with a bloated application and lots of one-time-use features.

Around that time, I ran into Steve Blank's lectures on Customer Development, from which I followed the trail to Eric Ries's early ideas of the Lean Startup. I had dreamt the big vision, rationalized it in my head, and built it and refined it the long, hard way. I knew customers held the answers but didn't know when or how to fully engage them. That's exactly what Customer Development and Lean Startup were attempting to address.

I was sold.

2 A product development philosophy popularized by 37signals.

Why This Book?

I was determined to test these techniques on my next product (CloudFire) but ran into many early challenges when trying to take these concepts to practice.

For one, Steve Blank's book was written for a specific type of business, enterprise software, which made it hard to carry over many of the tactics to my products. Also, while Eric Ries was sharing his retrospective lessons learned from working at IMVU, IMVU was no longer a startup. With a technical staff of 40 people and more than $40 million in revenue, what you saw was a fully realized Lean Startup machine, which was at times daunting.

I had more questions than answers, which prompted my two-year journey in search of a better methodology for building successful products. The product of that journey is *Running Lean*, which is based on my firsthand experiential learning building products and the pioneering work of Eric Ries, Steve Blank, Dave McClure, Sean Ellis, Sean Murphy, Jason Cohen, Alex Osterwalder, and many others who I reference throughout the book.

I am thankful to the thousands of readers who subscribed to my blog, left comments week after week, sent me notes of encouragement to keep on writing, and subjected their products to my testing. This book was really "pulled" out of me by them.

Field-Tested

As a way to test the content for this book, I started speaking and teaching Running Lean workshops. I have shared this methodology with hundreds of startups and worked closely with many of them to test and refine it.

Whereas my blog is a near-real-time account of my lessons learned, this book benefits from retrospective learning and from reordering and refining steps for a more optimal workflow.

I am applying this new workflow to my next startup, which is also a by-product of my blogging and learning over the past year. As of this writing, I have sold WiredReach and am in the process of building and launching a new startup, Spark59.

Disclaimers

Practice Trumps Theory

You get a gold star not for following a process, but for achieving results. One of the things that particularly drew me to the Lean Startup methodology is that it is a meta-process from which more specific processes and practices can be formulated. The same principles used to test your product can and should be applied to test your tactics when taking these principles to practice.[3]

Everything in this book is based on first-hand experiential learning and experimentation on my own products. I encourage you to rigorously test and adapt these principles for yourself. The legal, financial, and accounting aspects of launching a company are outside the scope of the book. When the time comes, it is important to get competent professional advice about financing and structuring your company and its intellectual property assets.

There Are No Silver Bullets

No methodology can guarantee success. But a good methodology can provide a feedback loop for continuous improvement and learning.

That is the promise of this book.

3 There is no room for faith in a Lean Startup: *http://www.ashmaurya.com/2011/02/do-you-have-faith-in-lean-startups/*.

PART 1:

ROADMAP

CHAPTER 1

Meta-Principles

The proper application of any methodology first requires a clear understanding and separation of principles from tactics.

Principles guide what you do. Tactics show you how.

The essence of Running Lean can be distilled into three steps:

1. Document your Plan A.
2. Identify the riskiest parts of your plan.
3. Systematically test your plan.

In this chapter, we'll cover these meta-principles. The rest of the book will focus on the reduction of these meta-principles to practice.

Step 1: Document Your Plan A

There Is an "I" in Vision

> *All men dream: but not equally. Those that dream by night in the dusty recesses of their minds wake in the day to find that it was vanity: but the dreamers of the day are dangerous men, for they may act their dreams with open eyes, to make it possible.*
>
> *—T.E. Lawrence, "Lawrence of Arabia"*

Everyone gets hit by ideas when they least expect them (in the shower, while driving, etc.). Most people ignore them. Entrepreneurs choose to act on them.

While passion and determination are attributes that are essential in order to drive a vision to its full potential, if they are left unchecked, they can also turn the journey into a faith-based one driven by dogma.

Reasonably smart people can rationalize anything, but entrepreneurs are especially gifted at this.

Most entrepreneurs start with a strong initial vision and a Plan A for realizing that vision. Unfortunately, *most Plan A's don't work*.

While a strong vision is required to create a mantra and make meaning, a Lean Startup strives to uphold a strong vision with facts, not faith. It is important to accept that your initial vision is built largely on untested assumptions (or hypotheses). Running Lean helps you systematically test and refine that initial vision.

Capture Your Business Model Hypotheses

Too many founders carry their hypotheses in their heads alone, which, though the fastest way to iterate, only helps to further support their own "reality distortion fields."

The first step is *writing down* your initial vision and then *sharing* it with at least one other person.

Traditionally, business plans have been used for this purpose. But, while writing a business plan is a good exercise for the entrepreneur, it falls short of its true purpose: Facilitating conversations with people other than yourself.

Additionally, since most Plan As are likely to be proven wrong anyway, you need something less static and rigid than a business plan. Taking several weeks or months to write a 60-page business plan largely built on untested hypotheses is a form of waste.

> *Waste is any human activity which absorbs resources but creates no value.*
> *—James P. Womak and Daniel T. Jones,* Lean Thinking *(Free Press)*

My format of choice is to use the one-page business model diagram (Lean Canvas) shown in Figure 1-1.

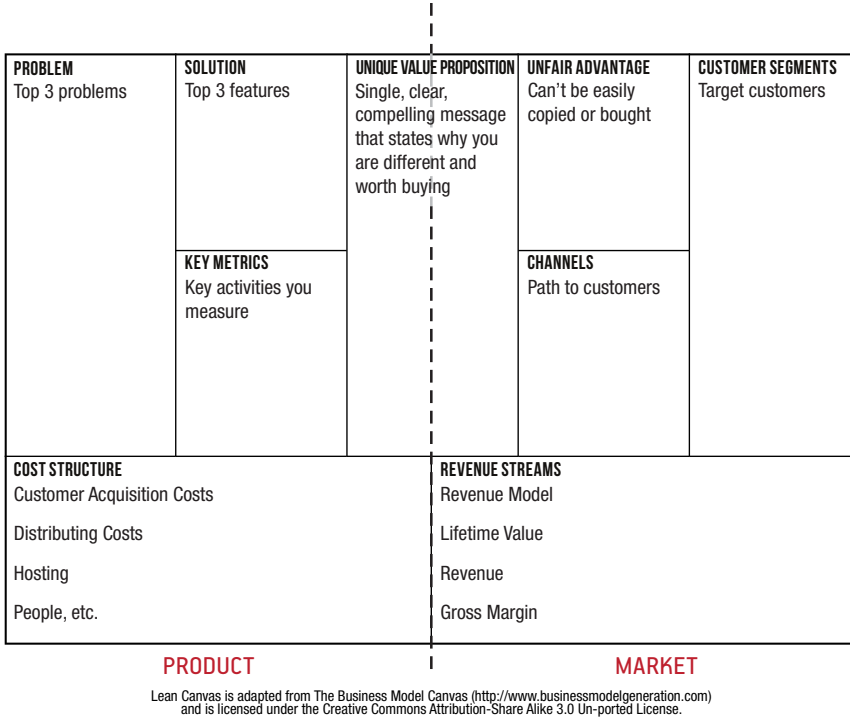


Figure 1-1. *Lean Canvas*

Lean Canvas is my adaptation of Alex Osterwalder's Business Model Canvas, which he describes in the book *Business Model Generation* (Wiley).[1]

I particularly like the one-page canvas format because it is:

Fast

Compared to writing a business plan, which can take several weeks or months, you can outline multiple business models on a canvas in one afternoon. Because creating these one-page business models takes so little time, I recommend spending a little additional time up front, brainstorming possible variations to your model and then prioritizing where to start.

Concise

The canvas forces you to pick your words carefully and get to the point. This is great practice for distilling the essence of your product. You have 30 seconds to grab the attention of an investor over a

1 To understand the differences between Lean Canvas and the original Business Model Canvas, see *http://www.ashmaurya.com/why-lean-canvas*.

hypothetical elevator ride, and eight seconds to grab the attention of a customer on your landing page.[2]

Portable

A single-page business model is much easier to share with others, which means it will be read by more people and probably will be more frequently updated.

If you have ever written a business plan or created a slide deck for investors, you'll immediately recognize most of the building blocks on the canvas. I won't spend time describing these blocks right now, as we'll cover them in great detail in Part 2 of the book.

A key point I would like you to take away for now, though, is that *your product is NOT "the product"* of your startup.

Your Product Is NOT "the Product"

I purposely made the solution box less than one-ninth of the entire canvas because, as entrepreneurs, we are most passionate about the solution box and what we are naturally good at (see Figure 1-2).

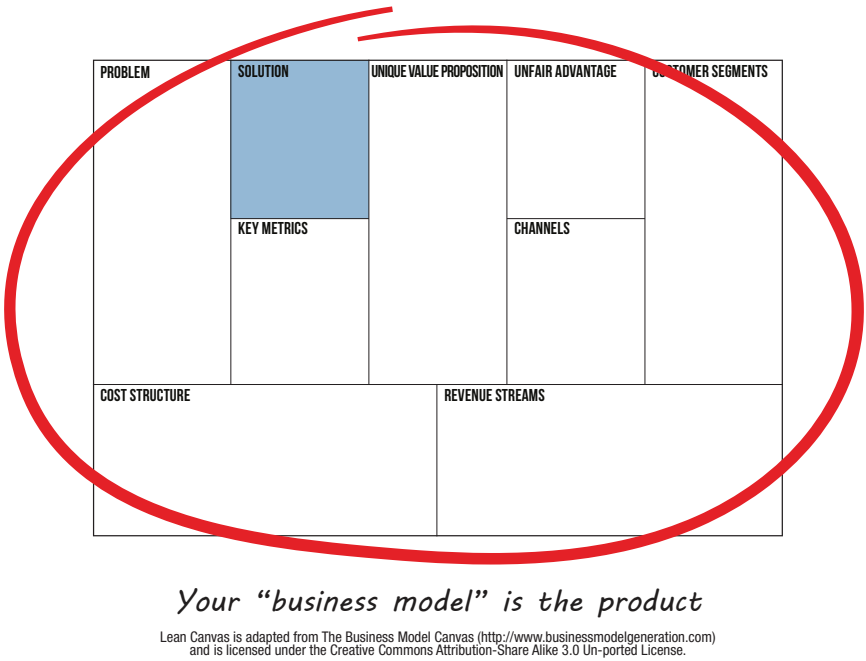


Figure 1-2. *Your product is NOT "the product"*

2 It is estimated that up to 50% of visitors to landing pages will bail in the first eight seconds. Source: MarketingSherpa's "Landing Page Handbook" (2005).

Dave McClure of 500 Startups has sat through hundreds of entrepreneur pitches and will probably sit through hundreds more. During these sessions, he has repeatedly called out entrepreneurs for spending a disproportionate amount of time talking about their solution and *not* enough time talking about the other components of the business model.

> *Customers don't care about your solution. They care about their problems.*
>
> *—Dave McClure, 500 Startups*

Investors, and more important, customers, identify with their problems and *don't care about your solution (yet)*. Entrepreneurs, on the other hand, are naturally wired to look for solutions. But chasing after solutions to problems no one cares enough about is a form of waste.

Your job isn't just building the best solution, but owning the entire business model and making all the pieces fit.

Recognizing your business model as a product is empowering. Not only does it let you own your business model, but it also allows you to apply well-known techniques from product development to building your company.

If you take a step back, you'll see that these meta-principles are nothing more than the *divide and conquer* technique applied to the process of starting up.

Lean Canvas helps deconstruct your business model into nine distinct subparts that are then systematically tested, in order of highest to lowest risk.

Step 2: Identify the Riskiest Parts of Your Plan

Building a successful product is fundamentally about risk mitigation.

Customers buy from you when they trust you can solve their problems. Investors bet on you when they trust you can build a scalable business model.

Startups are a risky business, and our real job as entrepreneurs is to systematically de-risk our startups over time.

Another technique taken from the Product Development playbook is that of "tackling the riskiest parts first." Not coincidentally, for most products, the solution isn't what's riskiest.

Unless you are trying to solve a particularly hard technical problem (like finding a cure for cancer, building the next big search algorithm, or splitting isotopes), chances are you will be able to build your product given enough time, money, and effort.

The bigger risk for most startups is building something nobody wants.

While what's riskiest varies across products, a lot of that risk is driven by the stage of your startup, which we'll cover next.

The Three Stages of a Startup

A startup goes through three distinct stages, as shown in Figure 1-3.

Figure 1-3. *Three stages of a startup*

Stage 1: Problem/Solution Fit

Key question: Do I have a problem worth solving?

The first stage is about determining whether you have a *problem worth solving* before investing months or years of effort into building a solution.

While ideas are cheap, acting on them is quite expensive.

A problem worth solving boils down to three questions:

- Is it something customers want? (must-have)
- Will they pay for it? If not, who will? (viable)
- Can it be solved? (feasible)

During this stage, we attempt to answer these questions using a combination of qualitative customer observation and interviewing techniques that we'll cover in great detail Chapters 5 and 6.[3]

From there you derive the minimum feature set to address the right set of problems, which is also known as the minimum viable product (MVP).

Stage 2: Product/Market Fit

Key question: Have I built something people want?

Once you have a problem worth solving and your MVP has been built, you then test how well your solution solves the problem. In other words, you measure whether you have built *something people want*.

3 In *The Four Steps to the Epiphany*, Steve Blank points out the importance of in-depth customer interviews, which he terms "Customer Discovery."

In Part 4 of this book, we'll cover both qualitative and quantitative metrics for measuring product/market fit.

Achieving traction or product/market fit is the first significant milestone for a startup. At this stage, you have a plan that is starting to work—you are signing up customers, retaining them, and getting paid.

Stage 3: Scale

Key question: How do I accelerate growth?

After product/market fit, some level of success is almost always guaranteed. Your focus at this stage shifts toward growth, or *scaling* your business model.

Pivot Before Product/Market Fit, Optimize After

Achieving product/market fit is the first significant milestone of a startup and greatly influences both strategy and tactics. For this reason, it is helpful to further delineate the stages of a startup as "before product/market fit" and "after product/market fit."

Before product/market fit, the focus of the startup centers on *learning and pivots*. After product/market fit, the focus shifts toward *growth and optimizations*. (See Figure 1-4.)

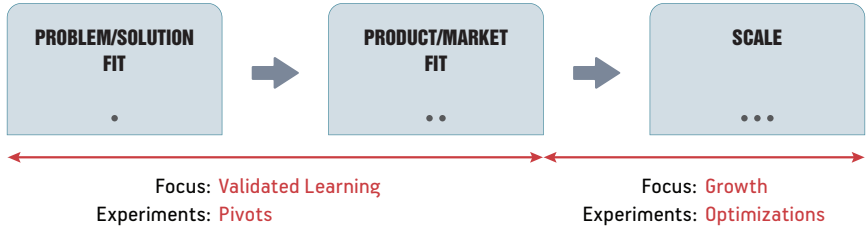


Figure 1-4. *Before and after product/market fit*

Pivot is a term used by Eric Ries to describe a change in direction of a startup while staying grounded in learning. The best way to differentiate pivots from optimizations is that pivots are about *finding a plan that works*, while optimizations are about *accelerating that plan*.

In a pivot experiment, you attempt to *validate* parts of the business model hypotheses in order to find a plan that works. In an optimization experiment, you attempt to *refine* parts of the business model hypotheses in order to accelerate a working plan. The goal of the first is a course correction (or a pivot). The goal of the second is efficiency (or scale).

This may sound like a subtle distinction, but it has a significant impact on both strategic and tactical execution. Before product/market fit, a startup needs to be architected to maximize learning.

You stand to learn the most when the probability of the expected outcome is 50%; that is, when you don't know what to expect.

In order to maximize learning, you have to *pick bold outcomes* instead of chasing incremental improvements. So, rather than changing the color of your call-to-action button, change your entire landing page. Rather than tweaking your unique value proposition (UVP) for a single customer segment, experiment with different UVPs for different customer segments.

Later in the book, we'll visit many other examples that explain how you purposely architect for learning over optimization.

Where Does Funding Fit into All This?

It's funny to note how the 37signals folks went from "Outside money is Plan B" to "Outside money is Plan Z" between their last two books: *Getting Real* and *Rework* (37signals.com). Once you're profitable, it's easy to make such a declaration, but some times are certainly better than others to consider external funding (see Figure 1-5).

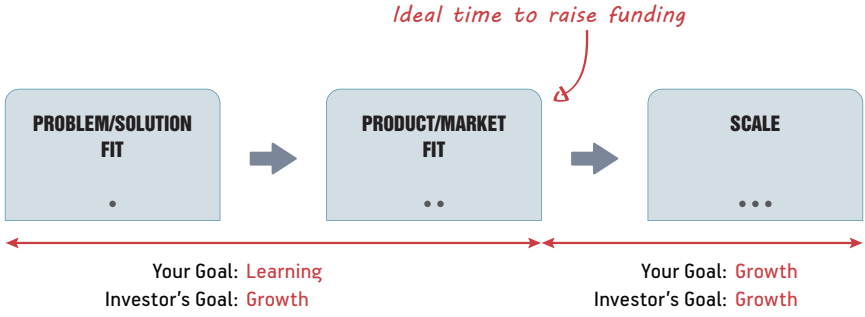


Figure 1-5. *Ideal time to raise funding*

Even though you may need to raise seed funding sooner, the ideal time to raise your big round of funding is *after product/market fit*, because at that time, both you and your investors have aligned goals: to scale the business.

> *Traction is a measure of your product's engagement with its market. Investors care about traction over everything else.*
>
> *—Nivi and Naval, Venture Hacks*

A lot of (especially first-time) entrepreneurs feel that Step 1 involves writing a business plan/building a slide deck and getting funded. Taking several months to write and pitch a business plan to investors is not the best use of time for a startup; especially since all you have at that point is a vision and a set of untested hypotheses. Selling this to investors without any level of validation is a form of waste.

Instead, your first goal should be to establish *just enough of a runway* to allow you to start testing and validating your business model with customers.

While not the same thing, bootstrapping and Lean Startups are quite complementary. Both cover techniques for building *low-burn startups* by eliminating waste through the *maximization of existing resources* before expending effort on the acquisition of new or external resources.

Bootstrapping + Lean Startup = Low-Burn Startup

(For more, see "How to Build a Low-Burn Startup" in the Appendix.)

Step 3: Systematically Test Your Plan

With your Plan A documented and your starting risks prioritized, you are now ready to systematically test your plan. In a Lean Startup, this is done by running a series of experiments.

The Lean Startup methodology is strongly rooted in the scientific method, and running experiments is a key activity. We'll cover steps for running effective experiments in Part 3 of the book, but for now, let's start by defining an experiment.

What Is an Experiment?

A cycle around the validated learning loop shown in Figure 1-6 is called an experiment.

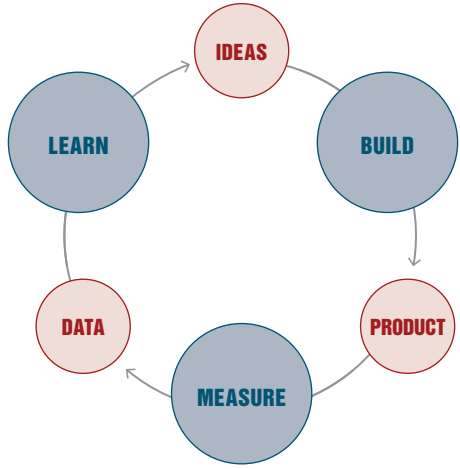


Figure 1-6. *Build-Measure-Learn loop*

The validated learning loop, or Build-Measure-Learn loop, was codified by Eric Ries and describes the customer feedback loop that drives learning in a Lean Startup.

It begins in the Build stage with a set of ideas or hypotheses that are used to create some artifact (mock-ups, code, landing page, etc.) for the purpose of testing a hypothesis. We put this artifact in front of customers and *"measure"* their response using a combination of qualitative and quantitative data. This data is used to derive specific *"learning"* that serves to validate or refute a hypothesis, which in turn drives the next set of actions.

The Iteration Meta-Pattern

While an experiment helps you validate or invalidate a specific business model hypothesis, an *iteration* strings multiple experiments together toward achieving a specific goal, such as getting to product/market fit.

Figure 1-7 shows the basic iteration meta-pattern we'll use throughout this book.

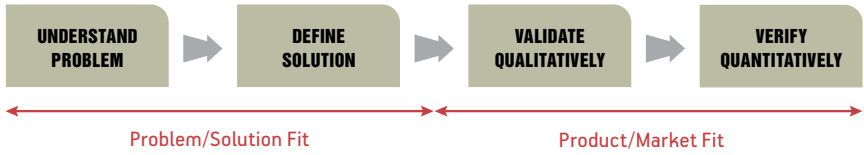


Figure 1-7. *Iteration meta-pattern*

The first two stages (Understand Problem and Define Solution) are about getting to problem/solution fit or finding a problem worth solving.

Then you iterate toward product/market fit by testing whether you've built something people want using a two-stage approach: first qualitative (micro-scale), then quantitative (macro-scale).

CHAPTER 2

Running Lean Illustrated

A great way to understand the meta-principles covered in Chapter 1 is to see them applied to a real product.

I wanted to pick a simple example that would be readily understood. So, rather than picking a software or hardware product, I decided to outline the process I used to write this book.

Even if you haven't written a book, you can probably appreciate the steps that go into writing a book, which, as you'll see, isn't unlike building a product.

Case Study: How I ~~Wrote~~ Iterated This Book

Writing a book was never in my plans. I was too busy running my company. I started my blog in October 2009 because I had more questions about Lean Startups than answers.

Along the way, a few of my blog readers started suggesting that I turn my blog posts into a book. I knew writing a book (even from blog posts) would be a massive undertaking, so while I was flattered by the requests, *I did nothing at first*. After about a dozen such requests, I decided to explore further.

What follows is how I applied the Running Lean process to writing this book.

Understand the Problem

I called these readers and asked them why they wanted me to write a book. Specifically, I asked *what would be different* about this book from what was already on my blog, or in other blogs and books that are already out there. In other words, I was trying to understand this book's *unique value proposition* in relation to *existing alternatives*.

From these interviews, I learned that, like me, my readers were also struggling with taking Customer Development and Lean Startup techniques to practice (*problem statement*) and viewed my blog posts as a "step-by-step" guide for applying these techniques from the ground up (*solution*). Many of them were also technical founders like me who were building web-based products (*early adopters*).

Define the Solution

With that knowledge, I spent a day *building a demo*. It was a teaser landing page with a table of contents, a title, and a stock book-cover image (see Figure 2-1).

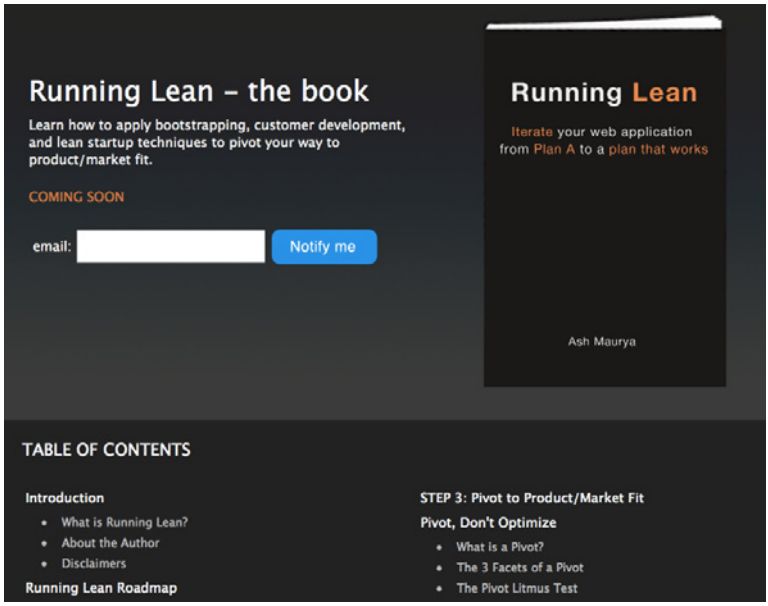


Figure 2-1. *Running Lean teaser page*

I knew the *riskiest part* about writing this book was nailing down the table of contents—not the title, book cover, or even price (since most business books have fairly established pricing).

I called the same readers and asked them: "If I were to write this book, would they buy it?" Their feedback helped me refine the table of contents (*define the solution*) and gave me a strong signal to move forward.

While encouraging, writing a book for just a dozen or so readers wasn't indicative of a *problem worth solving*. So I left the teaser page up and announced the book with a "coming this summer" launch time frame on my blog in March 2010. My readers helped spread the message (*channel testing*). Then I went back to running my company.

By June, I had collected 1,000 emails (*potential prospects*), which made writing the book a problem worth solving for me. My rationale for this was at least covering my costs using a simple back-of-the-envelope calculation.

Validate Qualitatively

Writing a whole book was still a massive undertaking. I tried writing the first chapter using "copy-and-paste" from my blog but wasn't happy with the results. I needed to build something even smaller that allowed me to start learning from customers (*a minimum viable product*).

I took the table of contents and turned it into a slide deck with the same outline and a few bullet points under each topic. I announced a free Running Lean workshop in Austin, Texas and got 30 people interested.

A local incubator, Tech Ranch, was gracious enough to provide a venue but only had room for 10 people. This was perfect, as it meant I could run at least two more workshops with the others (*iterate in small batches*).

Based on the success of the first workshop, not only did I run more workshops, but I also started charging for them (*getting paid is the first form of validation*). With each workshop, I continually tweaked the slide deck content for better flow and doubled pricing until I hit some resistance.

By the end of the summer, I understood the solution well enough and started writing. Here again, instead of writing the whole book in isolation, I contacted my potential prospects from the teaser page, many of whom were growing impatient as my initial launch date had come and gone. I apologized for not having finished the book and told them I'd be writing and releasing the book *iteratively*, much like software. Rather than waiting six more months to get the book, if they preordered the book, now they would get two chapters of the book every two weeks in PDF format.

About half of the people agreed to this arrangement. The others chose to wait for the "finished product," citing print, iPad, and/or Kindle as their preferred reading format. This further helped me distinguish *early adopters* from latter-stage customers. These early adopters were driven by the

content alone and didn't care how it was packaged. The content for me was still the riskiest part of the product to test.

Customer feedback during this two-week iteration cycle was immensely valuable. Entire chapters were rewritten for better flow, illustrations were improved,[1] and little typos and grammatical errors were nipped in the bud. Not only was I able to write a *better* book using this process, but I did so *faster*.

Verify Quantitatively

Only once the book was "content-complete" in January 2011 did I hire a designer for the book cover, start testing book subtitles, research print/ebook options, and build a marketing website (*right action, right time*).

While I'd always been prepared to self-publish this book, an interesting thing happened. I was contacted by a major publisher in December 2010 that got wind of the fact that I was writing this book. Not only had they already reviewed the latest version out at the time, but they were interested in publishing the book as-is.

I asked them if my model for writing and selling the book so far would be a deal breaker. On the contrary, they wished more authors wrote their books this way.

At first I was confused, but then it all made sense. The fact that I was able to sell 1,000 copies of the book on my own demonstrated *early traction*, which helped mitigate market risk for the publisher. This is not unlike how a latter-stage investor views a startup.

As with building a product, the ideal time to attract external resources is after product/market fit, which may or may not be the right action for you at that time.

In my case, I'm happy to say that additional conversations with other publishers, along with advice from Eric Ries, helped me determine that going the publisher route was the right action given my goals. I signed a contract with O'Reilly. Not only had the O'Reilly folks been early proponents of the Lean Startup movement, but they were also highly supportive of an official *Lean* series of books.

As of September 2011, I had sold just over 10,000 copies of *Running Lean* on my own and was writing a new and updated edition (the book you hold in your hands). This version was even further refined through countless interviews and workshops with entrepreneurs that spanned a wide

1 A reader and fellow Austin entrepreneur, Emiliano Villarreal, redid my illustrations and sent me the updated files. We started collaborating on other visuals and he now works with me at Spark59.

spectrum of products (*build a continuous feedback loop with customers*). The goal was to synthesize my learning over the past year and broaden the audience beyond my initial prototypical early adopters of web-based entrepreneurs.

The timeline shown in Figure 2-2 summarizes the process I used to write the first edition of *Running Lean*.

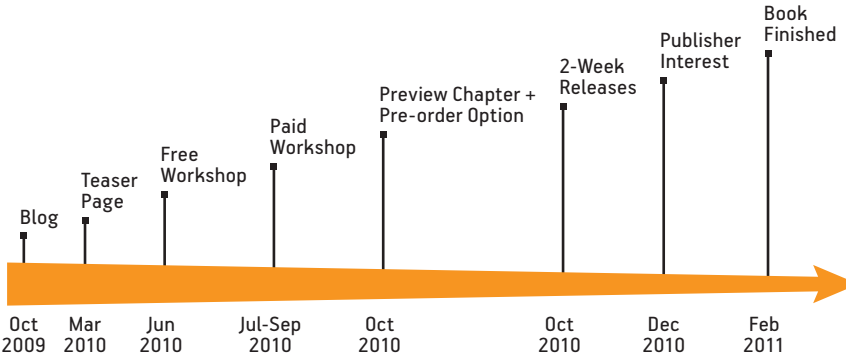


Figure 2-2. *Running Lean timeline*

Is the Book Finished?

A book, like large software, is never finished—only released.

Because I wrote this book iteratively about a topic that is still evolving, the book was just the beginning.

- I still share my raw learning on my blog.
- I write a biweekly "Running Lean Mastery" newsletter.
- Demand for my workshops has gone up.

While I love teaching these workshops, my true passion still lies in building products. Immersing myself in the world of hundreds of startups helped me identify a number of problems worth solving.

That is how Lean Canvas and USERcycle came to be.

Lean Canvas is a business model validation tool. It's a companion tool to this book that helps you document your business model, measure progress, and communicate learning with your internal and external stakeholders.

USERcycle is customer lifecycle management software that helps companies convert their users into passionate customers. Passionate customers come back and use your product, tell others about your product, and pay for your product (or get you paid).

PART 2:

DOCUMENT YOUR PLAN A

CHAPTER 3

Create Your Lean Canvas

Capture your business model in a portable, one-page diagram.

The Lean Canvas is the perfect format for brainstorming possible business models, prioritizing where to start, and tracking ongoing learning.

The best way to illustrate the use of the canvas is through an example. I'll describe the thought process that went into building my first product, CloudFire, using this methodology.

Brainstorm Possible Customers

When you first start out, all you have is an inkling of a problem, a solution, and maybe a customer segment. Just as rushing to build a solution can lead to waste, so can prematurely picking a customer segment or business model. The danger here is that this "selection bias" is untested and may result in a suboptimal business model or local maxima.

The Hill-Climbing Algorithm and the Problem of Local Maxima

In computer science, *hill climbing* is a mathematical optimization technique. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, and the process is repeated until no further improvements can be found.

Hill climbing is good for finding a local optimum (a solution that cannot be improved by considering a neighboring configuration), but it is not guaranteed to find the best possible solution (the global optimum) out of all possible solutions (the search space).

Source: *http://en.wikipedia.org/wiki/Hill_climbing*

While there is no way to completely avoid the local maxima problem, you raise your odds for finding a better solution when you are initially open to exploring and even testing multiple models in parallel.

Start by brainstorming the list of possible customers for your product:

Distinguish between customers and users.

If you have multiple user roles in your product, identify your customers.

A customer is someone who pays for your product. A user does not.

Split broad customer segments into smaller ones.

I've worked with startups that felt the problems they are solving are so universal, they apply to everyone.

You can't effectively build, design, and position a product for everyone.

While you might be aiming to build a mainstream product, you need to start with a specific customer in mind. Even Facebook, with its now 500 million+ users, started with a very specific user in mind: Harvard University students.

Put everyone on the same canvas at first.

If you are building a multisided business, you might find it necessary to outline different problems, channels, and value propositions for each side of the market. I recommend starting with a single canvas first and using a different color or tag to identify each customer segment. This helps you visualize everything on a single page. Then split if needed.

Sketch a Lean Canvas for each customer segment.

As you'll find shortly, the elements of your business model can and will vary greatly by customer segment. I recommend starting with the top two or three customer segments you feel you understand the best or find most promising.

CASE STUDY

CloudFire

Background:

Prior to CloudFire, I had launched a file-sharing application called BoxCloud that simplified the process of sharing large files, using a proprietary peer-to-web (p2web) framework we had built.

BoxCloud's unique value proposition (UVP) was that it allowed the sharer to share a file/folder directly from his computer without any uploading. Recipients accessed the shared file/folder directly from their browser without the need to install any additional software.

BoxCloud was primarily targeted at business users and was in use by graphic designers, attorneys, accountants, and other small-business owners.

I was interested in exploring other uses of the p2web framework, especially around media sharing (photos, videos, and music), which is how CloudFire came about.

Really broad category:

Anyone that shares lots of media content.

More specific possible customers:

- Photographers
- Videographers
- Media consumers (scratch my own itch)
- Parents

While I was initially drawn to building something for the consumer segment (with myself as the prototypical customer), I had recently become a parent and witnessed some pain points around photo and (especially) video sharing. That is the segment I decided to model first.

Sketching a Lean Canvas

In this section, I'll outline the process for sketching a Lean Canvas.

Sketch a canvas in one sitting.

> While it's tempting to iterate endlessly on the whiteboard, your initial canvas should be sketched quickly—in less than 15 minutes. The point of creating the canvas is to take a snapshot of what's in your head at this moment in time, then move on to identifying what's riskiest, and finally get out of the building and test your model with people other than yourself.

It's OK to leave sections blank.

> Rather than trying to research or debate the "right" answers, put something down or leave it blank. Leaving a section blank might be indicative of what's really riskiest about your model and the place to start your testing. Some other elements, like "Unfair Advantage," take time to figure out, and your best answer right now might be "I don't know," which is also OK. The canvas is meant to be an organic document that evolves with time.

Be concise.

> It's a lot easier to describe something in a paragraph than in a single sentence. The space constraints on the canvas are a great way to distill your business model down to its essence. Aim to fit your canvas on a single page.

Think in the present.

> Business plans try too hard to predict the future, which is impossible. Instead, write your canvas with a "getting things done" attitude. Based on your current stage and what you know right now, what are the next sets of hypotheses you need to test to move your product forward?

Use a customer-centric approach.

> Alex Osterwalder describes several techniques for approaching an initial business model canvas in his book. Since Running Lean is heavily customer-driven, I find it sufficient to start with just a customer-centric approach. As we'll see shortly, tweaking just the customer segment can completely change the business model.

When creating my canvases, I follow the prescribed order shown in Figure 3-1, which is the order the rest of the sections will follow.

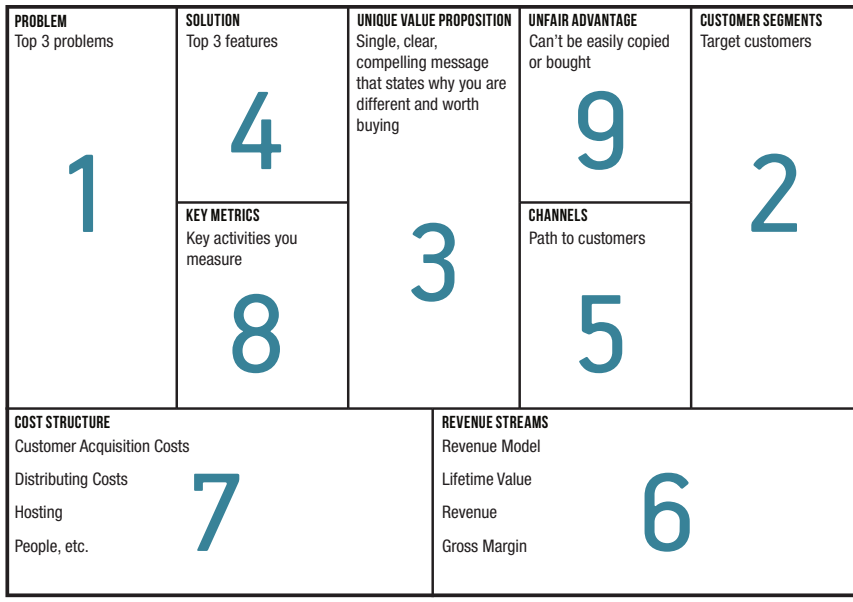


Figure 3-1. *Lean Canvas*

Problem and Customer Segments

I find that the "problem-customer segment" pair usually drives the rest of the canvas, which is why I tackle them together.

List the top one to three problems.

For the customer segment you are working with, describe the top one to three problems they need solved. Another way to think about problems is in terms of the jobs customers need done:

> *When people need to get a job done, they hire a product or service to do it for them. The marketer's task is to understand what jobs periodically arise in customers' lives for which they might hire products the company could make.*
>
> —*Clayton M. Christensen*

List existing alternatives.

Document how you think your early adopters address these problems today. Unless you are solving a brand new problem (unlikely), most problems have existing solutions. Many times these solutions may not be from an obvious competitor.

As an example, the biggest alternative to most online collaboration tools is not another collaboration tool, but email. Doing nothing could also be a viable alternative for a customer if the pain is not acute enough.

Identify other user roles.

Identify any other user roles that will interact with this customer. Examples:

- In a blogging platform, the customer is the blog author while the user is a reader.
- In a search engine, the customer is the advertiser while users are people running searches.

Hone in on possible early adopters.

With these problems in mind, get more specific on the customer segment. Narrow down the characteristics of your prototypical customer.

Your objective is to define an early adopter, not a mainstream customer.

CASE STUDY

CloudFire: Problem and Customer Segments

Having just become a parent, I observed the "perfect storm" of problems that I wanted to explore further, listed here and shown in Figure 3-2:

- The number of photos (and especially videos) we took increased significantly after the baby was born.
- We were sleep-deprived and found the existing solutions time-consuming and sometimes painful to use.
- The demand for this content from family (especially grandparents) and friends was high and often time-sensitive.

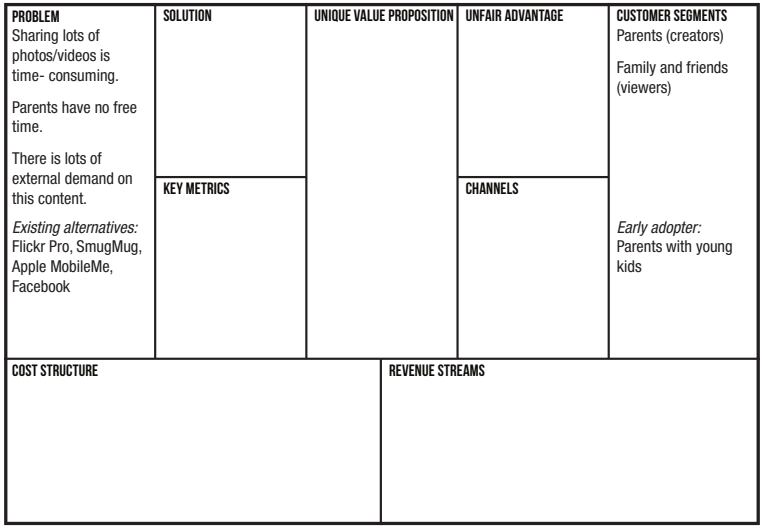


Figure 3-2. *CloudFire: problem and customer segments*

Unique Value Proposition

Dead center in the Lean Canvas is a box for your UVP. This is one of the most important boxes on the canvas and also the hardest to get right.

Since writing the first version of *Running Lean*, I have refined my definition of the UVP:

> *Unique Value Proposition: Why you are different and worth ~~buying~~ getting attention.*

"Selling" is a conversation, and I believe it's too hard to do that with a single statement. More important, the first battle isn't even selling; it's getting a prospect's attention.

> **NOTE**
>
> First-time visitors spend eight seconds on average on a landing page. Your UVP is their first interaction with your product. Craft a good UVP and they might stay and view the rest of your site. Otherwise, they'll simply leave.

Even with this revised definition, the UVP is still hard to get right because you have to distill the essence of your product in a few words that can fit in the headline of your landing page. Additionally, your UVP also needs to be different, and that difference needs to matter.

The good news is that you don't have to get this perfect right away. Like everything on the canvas, you start with a best guess and iterate from there.

How to craft a unique value proposition

First, I highly recommend getting a copy of the classic book on marketing by Al Ries and Jack Trout: *Positioning: The Battle for Your Mind* (McGraw-Hill). Ries and Trout are considered the fathers of modern advertising. This is an "easy read" and the best crash course on marketing I've ever come across.

Here are some of my tips on how to craft a UVP:

Be different, but make sure your difference matters.

> The key to unlocking what's different about your product is deriving your UVP directly from the number-one problem you are solving. If that problem is indeed worth solving, you're more than halfway there already.

Target early adopters.

Too many marketers try to target the "middle" in the hopes of reaching mainstream customers, and in the process they water down their message. Your product is *not* ready for mainstream customers yet. Your sole job should be to find and target early adopters, which requires bold, clear, and specific messaging.

Focus on finished story benefits.

You've probably heard about the importance of highlighting benefits over features. But benefits still require your customers to translate them to their worldview. A good UVP gets inside the head of your customers and focuses on the benefits your customers derive *after* using your product.

So, for instance, if you are creating a résumé-building service:

- A feature might be "professionally designed templates."
- The benefit would be an "eye-catching résumé that stands out."
- But the finished story benefit would be "landing your dream job."

A good formula for crafting an effective UVP (by way of Dane Maxwell) is:

> Instant Clarity Headline = End Result Customer Wants + Specific Period of Time + Address the Objections

NOTE

The second and third items in the preceding formula are great if you can use them, but they are not required.

A classic example that fits this formula is Domino's slogan:

> *Hot fresh pizza delivered to your door in 30 minutes or it's free.*

Pick your words carefully and own them.

Words are key to any great marketing and branding campaign. Look at how the top luxury car brands have used a single word to define themselves:

- *Performance:* BMW
- *Design:* Audi
- *Prestige:* Mercedes

Picking a few "key" words that you consistently use also drives your search engine optimization (SEO) ranking.

Answer: what, who, and why.

A good UVP needs to clearly answer the first two questions—what is your product and who is your customer. The "why" is sometimes hard to fit in the same statement, and I'll frequently use a subheading for that.

Here are example UVPs I have used in products:

Lean Canvas

Spend More Time Building Versus Planning Your Business.

The faster, more effective way to communicate your business model

USERcycle

Turn your users into passionate customers.

Customer Lifecycle Management Software

Study other good UVPs.

The best way to craft a good UVP is to study the UVPs of the brands you admire. Visit their landing pages and deconstruct how and why their messaging works.

Some of my best teachers have been Apple, 37signals, and FreshBooks.

Create a high-concept pitch.

Another useful exercise is creating a high-concept pitch. High-concept pitches are used heavily by Hollywood producers to distill the general plot of a movie to a memorable sound bite. The high-concept pitch was also popularized as an effective pitching tool by Venture Hacks in its ebook, *Pitching Hacks*.

Examples:

- YouTube: "Flickr for video"
- *Aliens* (movie): "Jaws in space"
- Dogster: "Friendster for dogs"

The high-concept pitch should not be confused with a UVP and is not intended to be used on your landing page. There is a danger that the concepts the pitch is based on might be unfamiliar to your audience. For this reason, the high-concept pitch is more effective when used to quickly get your idea across and make it easy to spread, such as after a customer interview. We'll cover this specific use of the high-concept pitch in Chapter 7.

CASE STUDY

CloudFire: Unique Value Proposition

Given the current list of existing alternatives, I decided to use speed as the "difference that would matter" for my UVP and "no uploading" as the key words to position around (see Figure 3-3).

Later, you'll see how this UVP evolved significantly after just a few customer interviews.

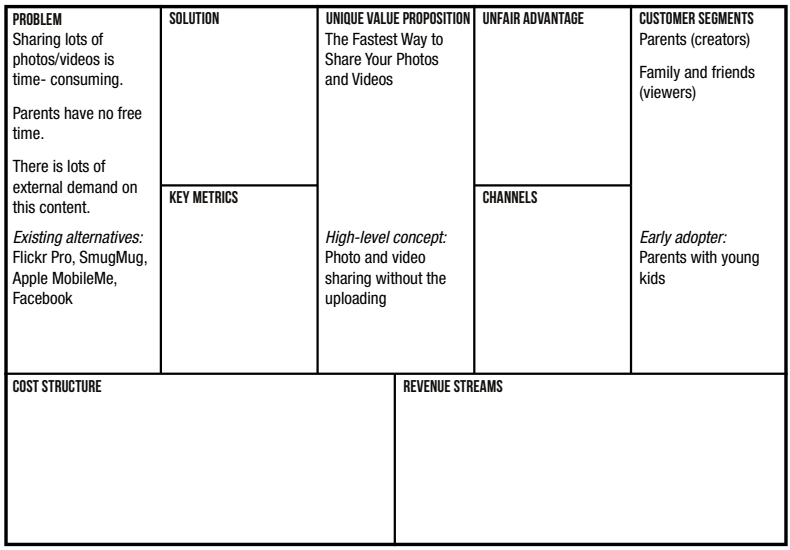


Figure 3-3. *CloudFire: UVP*

Solution

You are now ready to tackle solution possibilities.

Because all you have are untested problems, it is fairly common for them to get reprioritized or completely replaced with new ones after just a few customer interviews. For this reason, I recommend not getting carried away with fully defining your solution just yet. Rather, simply sketch out the simplest thing you could possibly build to address each problem.

Bind a solution to your problem as late as possible.

CASE STUDY

CloudFire: Solution

Based on my list of problems, I created a short list of top features I would include in the minimum viable product, or MVP (see Figure 3-4).

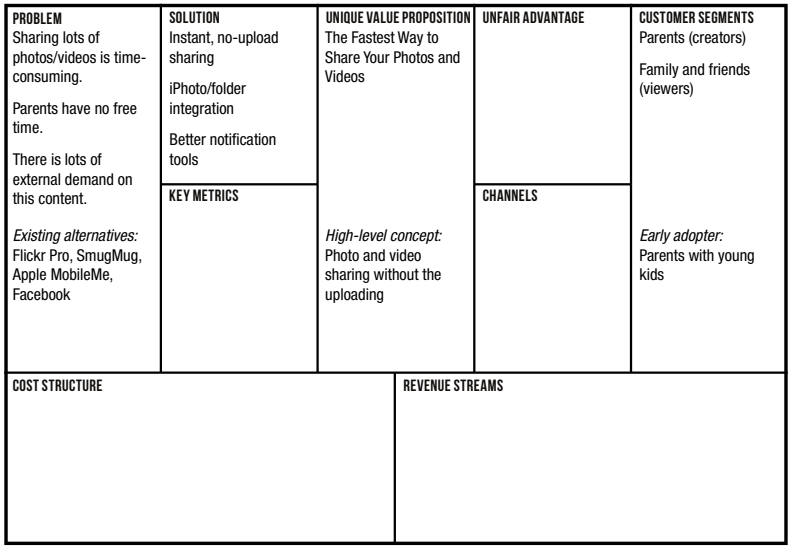


Figure 3-4. *CloudFire: solution*

Channels

Failing to build a significant path to customers is among the top reasons why startups fail.

The initial goal of a startup is to learn, not to scale. So, at first it's OK to rely on any channels that get you in front of potential customers.

The good news is that following a "customer discovery[1]/interview" process forces you to build a path to "enough" customers early. However, if your business model relies on acquiring large numbers of customers to work, that path may not scale beyond the initial stages, and it's quite possible you'll get stuck later.

1 The first step described by Steve Blank in his book *The Four Steps to the Epiphany* (*http://www.cafepress.com/kandsranch*).

For this reason, it's equally important to think about your scalable channels from day one so that you may start building and testing them early.

While there are a plethora of channel options available, some channels may be outright inapplicable to your startup, while others may be more viable during later stages of your startup.

I typically look for the following characteristics in my early channels.

Freer versus paid

First, there is no such thing as a free channel. Channels we normally associate as being free, like SEO, social media, and blogging, have a nonzero human capital cost associated with them. Calculating their ROI is complicated because, unlike a paid channel that is used up after you pay for it, these channels keep working for you over time.

A commonly cited paid channel is search engine marketing (SEM). Eric Ries has written about how he tested his early product on $5 a day using Google AdWords, driving 100 clicks at a cost-per-click of 5 cents. If you can pull this off today, by all means use it, but unfortunately those days are long gone for most products. Keyword competition is so fierce now that you need to either outspend or outwit your competition. Both of these activities are better suited to the after product/market fit time frame when your focus shifts to optimizing versus learning.

Inbound versus outbound

Inbound channels use "pull messaging" to let customers find you organically, while outbound channels rely on "push messaging" to reach customers.

Examples of inbound channels:

- Blogs
- SEO
- Ebooks
- White papers
- Webinars

Examples of outbound channels:

- SEM
- Print/TV ads
- Trade shows
- Cold calling

When you don't yet have a tested value proposition, it's hard to justify spending marketing dollars or effort on outbound messaging. Getting "tech-crunched" or seeking other forms of PR before then is a form of waste. Now might be the time to start building inroads to influencers, but you are not ready to "get covered."

Interviews are a form of outbound channel that are the exception. As we'll see with the next two points, the return on learning from an interview far exceeds the cost of running an interview.

Direct versus automated

As a scalable channel, direct sales only make sense in businesses where the aggregate lifetime value of the customers exceeds the total compensation of your direct sales people, such as in certain B2B and enterprise products.

But as a learning channel, direct selling is one of the most effective, since you interact face to face with the customer.

First sell manually, then automate.

Direct versus indirect

Another area where startups waste energy is prematurely trying to establish strategic partnerships. The idea is to partner with a larger company to leverage its channels and credibility. The problem is that until you have a proven product, you won't get the right level of attention from the bigger company's sales reps to make this work. Imagine you are a sales rep at the bigger company. Given the choice of selling what you know or selling an unproven product to make your quota, which would you choose?

The same principle applies to hiring external salespeople. While a salesperson can probably outsell you on the execution of a sales plan, she can't create that plan.

You have to first sell your product yourself, before letting others do it.

Retention before referral

Many startups are obsessed with building virality and referral/affiliate programs into their product from day one. While referral programs can be very effective in spreading the word about your product, *you need to have a product worth spreading first.*

> *Build a remark-able product.*
>
> —*Seth Godin,* Purple Cow *(Portfolio Hardcover)*

CASE STUDY

CloudFire: Channels

I planned to start with several outbound channels (friends and other parents at daycare) for interviews, and list a few possible, more scalable channels for later (see Figure 3-5).

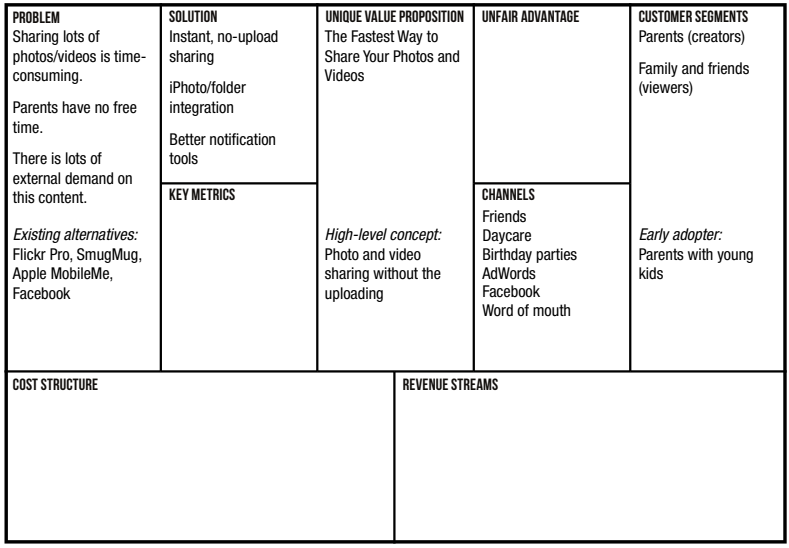


Figure 3-5. *CloudFire: channels*

Revenue Streams and Cost Structure

The bottom two boxes, labeled "Revenue Streams" and "Cost Structure," are used to model the viability of the business. Rather than thinking in terms of three- or five-year forecasts, take a more ground-up approach.

First, model the runway you will need to define, build, and launch your MVP. Then, revise after you get there.

Revenue streams

A lot of startups choose to defer the "pricing question" because they don't think their product is ready. Something I hear a lot is that an MVP is, by definition, embarrassingly minimal. How can you possibly charge for it?

First, an MVP is not synonymous with a half-baked or buggy product. Your MVP should address not only the *top problems* customers have identified as being important to them, but also the problems that are *worth solving*. By that definition, you should plan to deliver enough value to justify charging.

But there is another line of reasoning that is frequently cited for deferring pricing: to accelerate initial learning. The argument goes that pricing creates unnecessary friction that should be avoided early on.

The mindset most of us have when we're launching a new product is one of *lowering signup friction*. We want to make it as easy as possible for the customer to say yes and agree to take a chance on our product, hoping the value we deliver over time will earn us the privilege of his business.

Not only does this approach delay validation of one of the riskier parts of the model (because it's too easy for a user to say yes), but a lack of strong customer "commitment" can also be detrimental to optimal learning.

Furthermore, you don't need a lot of users to support learning—*just a few good customers*.

I believe that if you intend to charge for your product, you should charge from day one.

NOTE

A reasonable exception is when you're offering a value proposition that is built over time—for example, premium LinkedIn accounts.

Here's why:

Price is part of the product.

> Suppose I place two bottles of water in front of you and tell you that one costs 50 cents and the other costs 2 dollars. Despite the fact that you wouldn't be able to tell them apart in a blind taste test (the products are similar enough), you might be inclined to believe (or at least wonder) whether the more expensive water is of higher quality.
>
> Here, price has the power to change your perception of the product.

Price defines your customers.

> More interesting is the fact that the bottled water you pick determines your customer segment. From the existing market for bottled water, we know there is a viable business for bottled water at both price segments. What you charge signals your positioning on which customers you want to attract.

Getting paid is the first form of validation.

> Getting a customer to give you money is one of the hardest actions you can ask them to take and is an early form of product validation.

Although there is a lot of science around pricing, pricing is more art than science. For a great primer, I highly recommend getting a copy of Neil Davidson's free ebook on software pricing, *Don't Just Roll the Dice*.

One technique for setting initial pricing is pricing against the list of existing alternatives from the Problem box. These alternatives provide reference price anchors against which your offering will be measured.

(For more specific techniques for pricing Software as a Service [SaaS] products, including when to use freemium pricing, see "How to Set Pricing for a SaaS Product" in the Appendix.)

Cost structure

List the operational costs you will incur while taking your product to market. It's hard to accurately calculate these too far into the future. Instead, focus on the present:

- What will it cost you to interview 30 to 50 customers?
- What will it cost you to build and launch your MVP?
- What will your ongoing burn rate look like in terms of both fixed and variable costs?

Use the revenue streams and cost structure inputs to calculate a break-even point and estimate how much time, money, and effort you need to get there. You will use this later to prioritize which model you start with.

CASE STUDY

CloudFire: Revenue Streams and Cost Structure

Using the existing alternatives for price anchoring, which ranged from $24 to $39 per year for Flickr and SmugMug, to $99/year for Apple's MobileMe (a lot more than just photos/videos), I decided to start with $49/year pricing.

Prints (and other merchandise) were also revenue streams these companies used, but I wasn't sure if enough people still purchased prints anymore to make it worthwhile (a hypothesis that would need to be tested). More important, prints represented a potential secondary revenue stream that could only be realized once customers derived a core UVP. For this reason, I left out prints from both the MVP and initial canvas (see Figure 3-6).

The only initial costs to getting an MVP out were people costs, which I list in the next section.

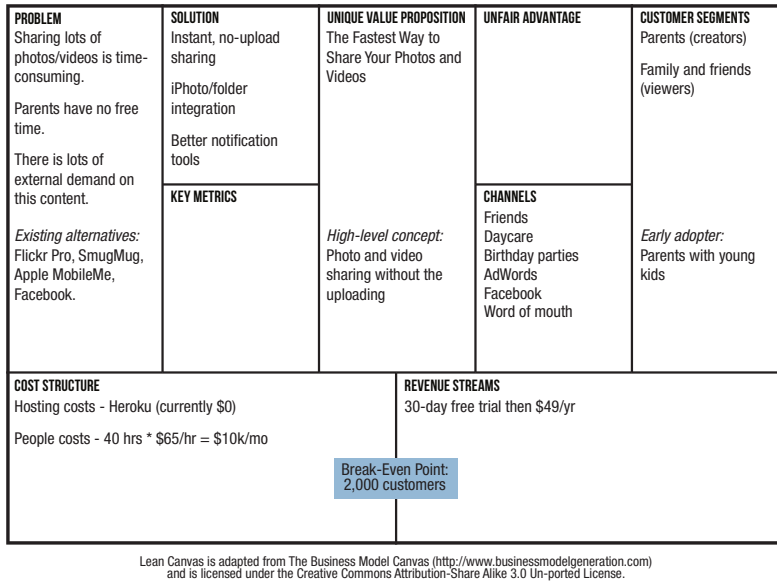


Figure 3-6. *CloudFire: revenue streams and cost structure*

Key Metrics

Find the key number that tells you how your business is doing in real time, before you get the sales report.

—Norm Brodsky and Bo Burlingham,
The Knack *(Portfolio Hardcover)*

Every business has a few key numbers that can be used to measure how well it is performing. These numbers are key for both measuring progress and identifying hot spots in your customer lifecycle.

A model I use heavily is Dave McClure's Pirate Metrics,[2] shown in Figure 3-7.

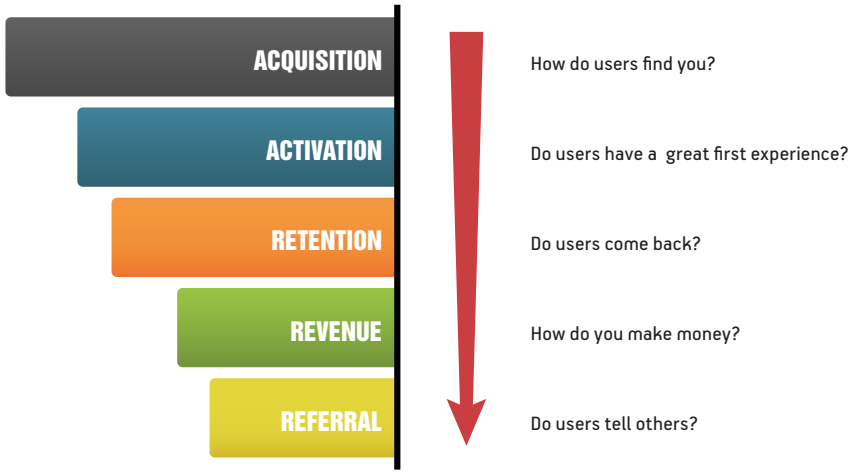


Figure 3-7. *CloudFire: Pirate Metrics*

Even though Pirate Metrics was built with software companies in mind, the model is applicable to many different types of businesses. Let's walk through each step using a flower shop and a software product as examples.

Acquisition

Acquisition describes the point when you turn an unaware visitor into an interested prospect.

In the case of the flower shop, getting someone walking by your window to stop and come in to your shop is an acquisition event.

On a product website, getting someone to do anything other than leave your website (abandon) is a measure of acquisition. I specifically measure successful acquisition as getting my visitors to view my signup page.

Activation

Activation describes the point when the interested customer has his first gratifying user experience.

2 Dave McClure called them Pirate Metrics because when you put the first letter in each funnel step together, they spell the word: AARRR.

In the case of the flower shop, if the prospect found the shop in disarray once he comes inside, there would be a disconnect with the promise made at the front of the store. That wouldn't be a gratifying first user experience.

On the product site, once the prospect signs up, you have to make sure you get the customer to a point where he can connect the promise you made on your landing page (your UVP) with your product.

Retention

Retention measures "repeated use" and/or engagement with your product.

So, in the case of the flower shop, the action of coming back to the store—and in the case of the product website, the act of logging back in to use the product again—would count toward retention.

As we'll see in Part 4 of the book, this is one of the key metrics to measuring product/market fit.

Revenue

Revenue measures the events that get you paid.

These could be buying flowers or buying a subscription for your product. These events may or may not occur on the first visit.

Referral

Referral is a more advanced form of a user acquisition channel where your happy customers refer or drive potential prospects into your conversion funnel.

In the case of the flower shop, this could be as simple as telling another friend about the store.

For the software product, this could range from implicit viral or social sharing features (like Share with a friend), to explicit affiliate referral programs or Net Promoter Score.

CASE STUDY

CloudFire: Key Metrics

In Figure 3-8 I map specific user actions that correspond to each of the key metrics discussed earlier.

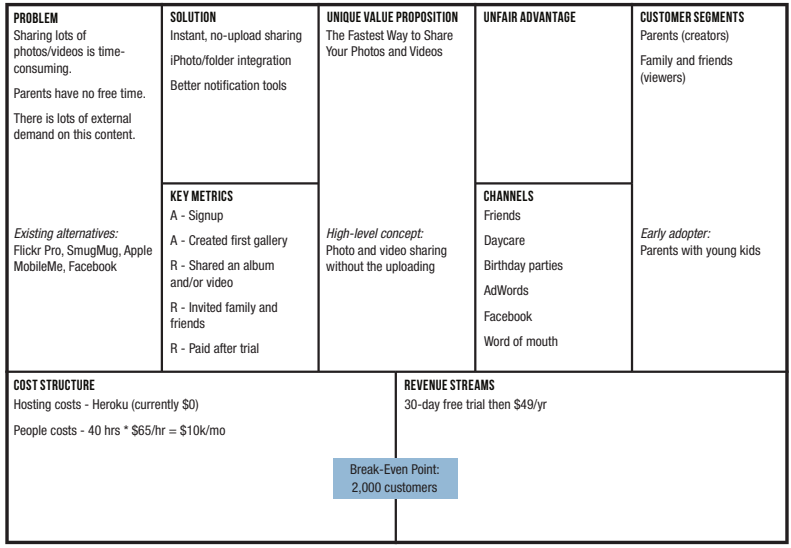


Figure 3-8. *CloudFire: key metrics*

Unfair Advantage

This is usually the hardest section to fill, which is why I leave it for last. Most founders list things as competitive advantages that really aren't—such as passion, lines of code, or features.

Another frequently cited advantage in business models is the "first-mover" advantage. However, it doesn't take much to see that being first can actually be a disadvantage, as most of the hard work of paving new ground (risk mitigation) falls on your shoulders, only to be potentially picked up later by fast-followers unless you're able to constantly outpace them with a real "unfair advantage." None of these companies were first movers: Ford, Toyota, Google, Microsoft, Apple, or Facebook.

An interesting perspective (via Jason Cohen) to keep in mind is that anything worth copying will be copied, especially once you start to demonstrate a viable business model.

Imagine a scenario where your cofounder steals your source code, sets up shop in Costa Rica, and slashes prices. Do you still have a business? How about if Google or Apple launches a competitive product and drops the price to $0?

You have to be able to build a successful business in spite of that, which led Jason Cohen to offer the following definition:[3]

> *A real unfair advantage is something that cannot be easily copied or bought.*
>
> —*Jason Cohen,* A Smart Bear *blog*

Here are some examples of real unfair advantages that fit this definition:

- Insider information
- The right "expert" endorsements
- A dream team
- Personal authority
- Large network effects
- Community
- Existing customers
- SEO ranking

Some unfair advantages can also start out as values that become differentiators over time.

For example, Zappos CEO Tony Hsieh believes strongly in creating happiness for his customers and employees. This manifested itself in many company policies that, on the surface, didn't make much business sense, such as allowing customer service representatives to spend as much time as was needed to make a customer happy and offering a 365-day return policy with two-way paid shipping. But these policies served to differentiate the Zappos brand and build a large, passionate, and vocal customer base that played a large role in the company's eventual $1.2 billion acquisition by Amazon in 2009.

You may have to leave this box blank when you first start out, but it's here to make you really think about how you can/will make yourself different and make your difference matter.

3 Jason Cohen. "No, that IS NOT a competitive advantage"; *http://blog.asmartbear.com/not-competitive-advantage.html.*

CASE STUDY

CloudFire: Unfair Advantage

Even though CloudFire is built on a proprietary p2web framework that might give us an early advantage, anything worth copying will be copied. So I decide to base my unfair advantage on something harder to replicate. In this case, community (see Figure 3-9).

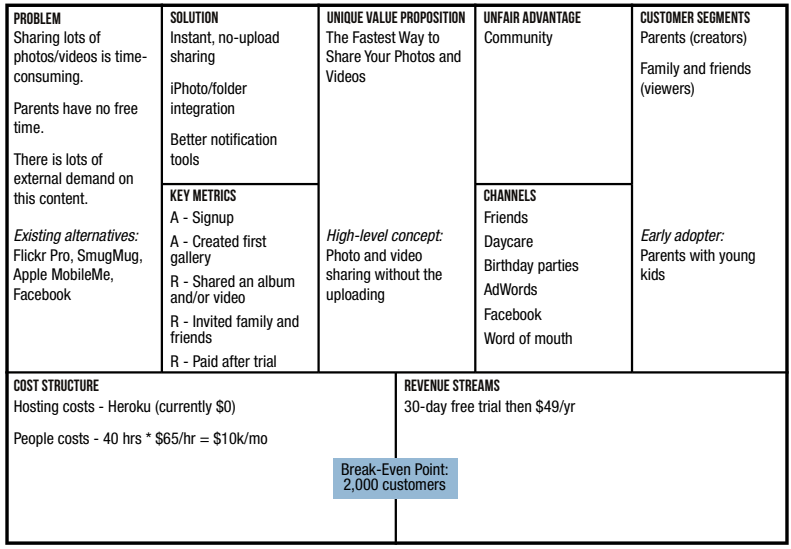


Figure 3-9. *CloudFire: unfair advantage*

Now It's Your Turn

Documenting your Plan A is a prerequisite for moving on. Too many founders carry their hypotheses in their heads alone, which makes it hard to systematically build and test a business.

You have to draw a line in the sand.

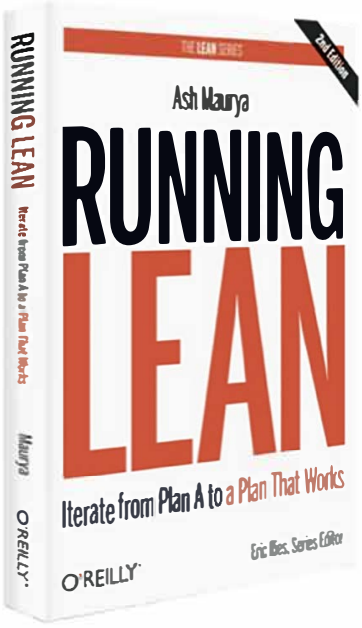
How you create your Lean Canvas is up to you.

You can:

- Visit *http://LeanCanvas.com* and create your online canvas there.
- Create a version in PowerPoint or Keynote.
- Sketch a canvas on paper.

The important thing is to share your Lean Canvas with at least one other person when you are done.

Enjoyed the excerpt?
Buy the full copy.



"★★★★★"

500+ 5-Star reviews

amazon

Available Everywhere, including:

amazon.com O'REILLY BARNES&NOBLE iTunes