



















<b>CHAPTER 5</b>	<b>Focusing on the Customer</b>	<b>157</b>
	Your Marketing Re-Education	157
	Marketing for Micro-ISVs	158
	Starting with SIMS	159
	Hand Me the MAP, Please	161
	Email: Retail, Wholesale, and You	163
	You Have Mail—Lots of It!	163
	You Can't Say That Anymore!	164
	Current Email Marketing Realities	168
	Technical Support Is Customer Support	171
	No Sympathy for the Devil	171
	Doing Tech Support Right	172
	Tech Support Is Like Beta Support, Only More So	173
	Discussion Boards: Listening to Your Customers	176
	What to Look For	178
	When to Do It	179
	Approach 1: Code It Yourself	179
	Approach 2: Open Source, Kind Of: phpBB	180
	Approach 3: Outsource to Invision	181
	Approach 4: By, for, and of Micro-ISVs	182
	Where Your Customers Start	185
	How to Do General Site Submission Right	186
	Google, Relevancy, and Your Micro-ISV	191
	Doing the Download Tango	194
	CNET Download.com	194
	Tucows.com	197
	Microsoft Office Marketplace	199
	All the Rest and Lessons Learned	203
	The Influencers	204
<b>CHAPTER 6</b>	<b>Welcome to Your Industry</b>	<b>209</b>
	What About Microsoft?	209
	The Microsoft Empower Program	211
	The Microsoft Buddy Program	212
	The Other Microsoft	215
	The Microsoft Digital Locker Program	218
	The Office Marketplace Program	222
	Microsoft Wants You! (Maybe.)	223
	Keeping an Eye on Microsoft	228

Business Intelligence Is Intelligent .....	231
The Initial Analysis .....	231
Strengths, Weaknesses, Opportunities, and Threats (SWOT) ....	232
Talk, Talk.....	233
Finding Others on the Road .....	238
Joel on Software.....	238
Association of Independent Software Industry Professionals (AISIP) .....	239
Association of Shareware Professionals (ASP) .....	240
Educational Software Cooperative (ESC) .....	242
Your Micro-ISV Industry Cheat Sheet .....	244
<b>CHAPTER 7   What Happens Next?</b> .....	245
Emerging Micro-ISVs .....	246
Successful Micro-ISVs .....	267
Very Successful Micro-ISVs .....	294
You’ve Reached the Bottom Line .....	313
<b>APPENDIX</b> .....	315
Chapter 1: “Having a Vision” .....	315
Books .....	315
Web Sites.....	315
Chapter 2: “Developing the Micro-ISV Way” .....	315
Books .....	315
Web Sites.....	316
Chapter 3: “Presenting the Product” .....	317
Books .....	317
Web Sites.....	318
Chapter 4: “Business Is Business” .....	319
Books .....	319
Web Sites.....	320
Chapter 5: “Focusing on the Customer” .....	321
Books .....	321
Web Sites.....	322
Chapter 6: “Welcome to Your Industry” .....	323
Books .....	323
Web Sites.....	323
<b>INDEX</b> .....	325

# Foreword

**H**ow the heck did I become the poster child for the micro-ISV movement?  
Of all people. Sheesh.

When I started Fog Creek Software, there was gonna be nothing “micro” about it. The plan was to build a big, multinational software company with offices in 120 countries and a skyscraper headquarters in Manhattan, complete with a heliport on the roof for quick access to the Hamptons. It might be a few decades—after all, we were going to be bootstrapped, and we always planned to grow slowly and carefully—but our ambitions were anything but small.

Heck, I don’t even *like* the term *micro-ISV*. The *ISV* part stands for Independent Software Vendor. It’s a made-up word, made up by Microsoft, to mean “software company that is not Microsoft,” or, more specifically, “software company that for some reason we have not yet bought or eliminated, probably because they are in some charming, twee line of business, like wedding table arrangements, the quaintness of which we are just *way* too cool to stoop down to, but you little people feel free to enjoy yourselves. Just remember to use .NET!”

It’s like that other term, *legacy*, that Microsoft uses to refer to all non-Microsoft software. So when they refer to Google, say, as a *legacy search engine*, they are trying to imply that Google is merely “an old, crappy search engine that you’re still using by historical accident, until you bow to the inevitable and switch to MSN.” *Whatever.*

I prefer *software company*, and there’s nothing wrong with being a start-up. *Start-up software company*, that’s how we describe ourselves, and we don’t see any need to define ourselves in relation to Microsoft.

I suppose you’re reading this book because you want to start a small software company, and it’s a good book to read for that purpose, so let me use my pulpit here to provide you with my personal checklist of three things you should have before you start your micro...*ahem*, start-up software company. You should also do some other things—Bob covers them pretty well in the rest of the book—but before you get started, here’s my contribution.

*Number One:* Don’t start a business if you can’t explain what pain it solves, for whom, why your product will eliminate this pain, and how the customer will pay to solve this pain. The other day I went to a presentation of six high-tech start-ups and not *one* of them had a clear idea for what pain they were proposing to solve. For example, I saw a start-up that was building a way to set a time to meet your friends for coffee, a start-up that wanted you to install a plug-in in your browser to track your every movement online in exchange for being able to delete things from that history, and a start-up that wanted you to be able to leave text messages for your friend who was tied to a particular location (so if they ever walked past the same bar they could get a message you had left for them there). What they all had in common was that none of them solved a problem, and all of them were as doomed as a long-tailed cat in a room full of rocking chairs.

*Number Two:* Don’t start a business by yourself. I know, there are lots of successful one-person start-ups, but there are even more failed one-person start-ups. If you can’t even convince *one friend* that your idea has merit...um...maybe it doesn’t. Besides, it’s lonely and depressing,

and you won't have anyone to bounce ideas off of. And when the going gets tough, which it will, as a one-person operation, you'll just fold up shop. With two people, you'll feel an obligation to your partner to push on through. (P.S. Cats do not count.)

*Number Three:* Don't expect much at first. People never know how much money they're going to make in the first month when their product goes on sale. I remember five years ago, when we started selling FogBugz, we had no idea if the first month of sales would be \$0 or \$50,000. Both figures seemed just as likely to me. I have talked to enough entrepreneurs and have enough data now to give you a definitive answer for *your* start-up.

That's right, I have a crystal ball and can now tell you the one fact you need to know more than anything else: exactly how much money you're going to make during the first month after your product goes live.

Ready?

OK.

In the first month, you are going to make...

*about...*

\$364, *if you do everything right*. If you charge too little, you're going to make \$40. If you charge too much, you're going to make \$0. If you expect to make any more than that, you're going to be really disappointed and you're going to give up and get a job working for The Man and referring to us people in start-up-land as *legacy micro-ISVs*.

That \$364 sounds depressing, but it's not, because you'll soon discover the one fatal flaw that's keeping 50 percent of your potential customers from whipping out their wallets, and then *tada!* you'll be making \$728 a month. And then you'll work really hard, and you'll get some publicity, and you'll figure out how to use AdWords effectively, and there will be a story about your company in the local wedding planner newsletter, and *tada!* You'll be making \$1,456 a month. And you'll ship version 2.0, with spam filtering and a Common Lisp interpreter built in, and your customers will chat amongst themselves, and *tada!* You'll be making \$2,912 a month. And you'll tweak the pricing, add support contracts, ship version 3.0, and get mentioned by Jon Stewart on *The Daily Show* and *tada!* \$5,824 a month.

Now we're cooking with fire. Project out a few years, and if you plug away at it, there's no reason you can't double your revenues every 12 to 18 months. So, no matter how small you start (detailed math formula omitted—*Ed.*), you'll soon be building your own skyscraper in Manhattan with a heliport so you can get to that 20-acre Southampton spread in 30 minutes flat.

And that, I think, is the real joy of starting a company: creating something all by yourself, nurturing it, working on it, investing in it, watching it grow, and watching the investments pay off. It's a hell of a journey, and I wouldn't miss it for the world.

Joel Spolsky  
*Cofounder, Fog Creek Software*

# About the Author

---



■ **BOB WALSH** has been a contract software developer in the San Francisco Bay Area for the past 22 years, specializing in desktop information systems. His company, Safari Software, has for the past decade amazingly focused on the same thing, albeit at a higher hourly rate.

In 2003, as outsourcing finished what the dot-com bust started, he developed MasterList Standard Version, an Excel-based project and task management application. Two years and 40,000 users later, Safari Software became a real, live, rootin'-tootin' micro-ISV by releasing MasterList Professional, a Windows personal project and task management application that, unlike traditional time management tools, gives you total control over your business and personal life while improving how you spend your time.

Before joining the ranks of the computer industry, Bob was a reporter for several news organizations, most worth bragging about being United Press International (UPI).



# About the Technical Reviewers

■ **CRAIG SNYDER** is currently the chief software architect for Includ, the publisher of a new RSS/ Web feed reader for Microsoft Outlook.

Craig has a bachelor’s degree in computer science from a local university in San Diego and has more than 25 years of experience in all facets of engineering and engineering management, including software development, quality assurance, and technical publications for start-ups and established organizations. Craig has a diverse background in several vertical markets spanning entertainment, financial, Internet security, homeland security, industrial controls, customer management, real estate, and communications.

■ **THOMAS RUSHTON** has been programming since his first computer, a Sinclair ZX80. He has since progressed through creating complex workflow and document management systems for financial and legal organizations and now works as the IT technical development manager for a U.K.-based law firm. He has a bachelor’s degree in computer science from Durham University and spent some research time in the field of software quality before moving into the more financially rewarding IT career roles of programmer, DBA, and consultant.

When not slaving away over a hot keyboard, he enjoys spending time with his wife, Sarah; their young son, William; and his double bass.





# Acknowledgments

**A**cknowledgement sections of books tend to get skipped by readers eager to get to the good stuff, and that's a shame because without these people this book would not have happened.

First off to my Apress editor, Jonathan Hassell, and project manager, Kylie Johnston: thanks guys for your help and support and for holding my feet to the fire when deadlines loomed! Also thanks to Kim Wimpsett for whacking my poor prose into proper copyedited shape, Kurt Krames for the cover, and to Susan Glinert and Lori Bring for getting everything to actually fit on a printed page.

Next off, Joel Spolsky, who helped sell Apress on the idea of this book, let me badger him with questions and has helped hundreds of developers with Joel on Software: thanks, Joel!

A great many people were interviewed for this book, and to each and every one of them I say, thanks for taking the time out of your busy lives to answer my questions about what you do.



# Introduction

In February 2005, after releasing my first commercial application, I went looking for all sorts of information that would help me market, support, and improve my product. I wasn't especially happy with what I found.

There were books aplenty on starting retail businesses, restaurants, inns—you name it—except a self-funded software company. There were a few pre-Internet books, now mostly out of print, about how to start a shareware company, and there were a few books out about how to write the killer business plan that would woo venture capitalists to fund your start-up but nothing about how to define a product, develop it, support it, market it, and do all this 100 percent on the Web.

I did find one really good Web site, the Business of Software forum at Joel on Software (<http://discuss.joelonsoftware.com/?biz>), where a whole bunch of developers starting or running companies would offer suggestions and advice to anyone politely asking.

As plentiful as the advice was at Joel on Software, it tended to be uneven and fragmentary. I decided that if there wasn't a single good book on how to start an Internet-based software company, then I should go out there and research and write one. This is that book.

A very long time ago, before becoming a programmer, then a developer, I was a reporter. I figured that if I dusted off my old journalism habits and went looking for the information I and lots of other developers needed, I could find people out there with the answers.

What I did not figure on when I started this book was that there is real *news* going on here: from Boise to Bulgaria, developers are starting their own companies to bring to market their own solutions in record numbers.

For every Internet software vendor you read about who just got funded by one or another venture capital funds, there are hundreds of micro-Internet software vendors successfully building desktop applications and Web-based products, distributing their software exclusively on the Net, and building companies that start with one person and often scale up to 20, 50, and 100 employees in a few short years.

## Who This Book Is For

This book is for that one developer who starts the whole thing off. One day, after yet another mind-numbing meeting at Big Company, Inc., when they've had a bellyful of working for clueless people, I want that developer to go searching with Google or browsing Amazon, find this book, and see how the pieces can come together for them to start their own, wildly successful micro-ISV.

I'm assuming you already know how to code: in fact, this is one of those rare Apress books without a single line of code! What I'm guessing you're looking for is really current, Internet-centric information about how to go from the desire to be your own boss, how to define what you want to work on, and exactly how best to code a solution through all the facets of running an online software business all the way to seeing the money roll in.

## How This Book Is Structured

In a lot of ways, this book is a process book. You start at the beginning with a desire and then work through in roughly chronological order all the moving parts you need to connect to get to the point where your micro-ISV is up, running, and making money.

Here's the chapter-by-chapter rundown:

*Chapter 1, "Having the Vision":* The two big take-aways from this chapter are how we got to a place where micro-ISVs can be successful and how you can find a problem worth solving as your micro-ISV's first product. I'll also cover who thought up this mouthful of a term, *micro-ISV*, and seven rules for avoiding much micro-ISV pain.

*Chapter 2, "Developing the Micro-ISV Way":* Once you've found the right idea, it's time to get into developing. But not so fast—developing your micro-ISV's product is unlike working at Big Company, Inc., or being a contract developer.

In this chapter, I cover those differences and look at designing your first commercial product, setting up a development environment that leverages your limited time and money to produce high-quality, customer-focused software, and finding and managing beta testers.

*Chapter 3, "Presenting the Product":* While you're developing away, it's time to look at your product. Your application is not your product. Your application plus your Web site, blog, documentation, installer, license, graphics, collaterals, payment processing, customer experience, and Unique Selling Proposition is your product.

Everything from finding the right domain name to how easy it is to buy your software is going to affect your sales, and in this chapter, I cover a slew of things that go into how potential customers experience your software.

You will especially take a look, bit by bit, at what makes a good micro-ISV Web site good. And I'll talk with Mena Trott, cofounder of Six Apart (makers of TypePad and Movable Type), about how to build a blog that makes friends, builds credibility, and influences potential customers.

*Chapter 4, "Business Is Business":* This chapter focuses on the business aspects of creating a micro-ISV business: finding the right legal structure for your fledgling firm (in the United States, in the United Kingdom, or in Australia). Once you get the paperwork out of the way, you need to focus on Getting Things Done (GTD), so I'll review the GTD approach many micro-ISVs use and talk with its creator, David Allen, about applying GTD to building and running a micro-ISV.

*Chapter 5, "Focusing on the Customer":* Now we get to the start of your micro-ISV show—your customers. In this chapter, I cover a systematic way of defining, finding, and marketing to your customers. I'll also cover some of the other ways you interface with your customers: email (wholesale and retail), customer support (a micro-ISV must get right), and how to set up and run a robust discussion forum about your company and its products.

You'll also look at how customers find you on the Net: Search Engine Optimization techniques, download sites, and Google AdWords. And you'll see how you can and should get the attention of reporters and editors in the mainstream media.

*Chapter 6, “Welcome to Your Industry”:* In this chapter you’ll broaden your micro-ISV horizons and take a look at what developer and ISV resources are out there that you can benefit from. Interestingly enough, several of those resources come from Microsoft, and whether you love or loath Microsoft, you can’t afford to ignore them.

*Chapter 7, “What Happens Next?”:* That’s going to be largely up to you. But in this chapter you’ll hear from 25 micro and not-so-micro ISVs about how their stories have turned out so far and what advice they’d like to pass on to you.

*Appendix:* Don’t look for 200 pages of error codes and API syntax in this appendix—you won’t find it. Instead, I’ll recap all the links you’ve seen in Chapters 1–7 and recommend books for those who want to dig deeper into specific aspects of business, law, productivity, developer best practices, and online marketing.

And yes, the links in this chapter are online! (See the next section.)

## Downloading the Code

You’ll find all the checklists, templates, and other files for this book, as well as a page of links chapter by chapter, at this book’s page at Apress (<http://www.apress.com>) and at my micro-ISV, Safari Software, at <http://safarisoftware.com>.

## Contacting the Author

Got a question, or want to learn more? Please visit my blog, <http://www.todoorelse.com>; stop by my micro-ISV’s Web site at <http://safarisoftware.com>; or drop me an email at [bobw@safarisoftware.com](mailto:bobw@safarisoftware.com).





# Having a Vision

*We are told that talent creates its opportunities. But it sometimes seems that intense desire creates not only its own opportunities, but its own talents.*

—Eric Hoffer, author and philosopher<sup>1</sup>

**T**his chapter introduces what this book is all about: building a successful micro-ISV. But before getting into all the plans and practices of creating an Internet-based, self-funded, start-up company to sell software, a service, or a product (and make a nice pile of change in the process), I'll discuss a few issues. For example, is it really possible you can—from scratch—build a real, live company in today's global, interconnected, multinational marketplace? And if you can in theory do that, how do you in practice decide on and define an application, a Web service, or a product for which people will pay good money?

The short answer to both questions is the Internet. Ten years ago, when Netscape blew away the collective wisdom of the financial establishment, people wondered, "Where is this Internet thing going?" Now, after a dot-com boom and a bust, as well as tens of thousands of new companies—large and small—selling applications, services, and products not possible ten years ago, you can see where at least some of this is heading:

- The Internet makes possible a different kind of business model than what has worked in the past.
- The Internet means even a one-person company can connect to the right people in a billion-person market instantaneously if that person has something of value to offer.

I could bore you to death right now by citing all the little one-person start-ups that are now 60-person businesses valued in the millions or by citing all the cool apps and products popping up all over the Net being produced by other little companies, but I won't. Well, I'll mention just one: Niklas Zennström and Janus Friis started a company called Skype in Europe late in 2003. It was their fourth start-up, so they had some experience and access to capital and had weathered some legal troubles, but the bottom line was that these two guys built and sold an Internet-based company that delivered a new service to 54 million people in 24 months for \$4.1 billion to eBay.

That's \$4.1 billion—with a *b*. In 24 months.

---

1. <http://quotations.about.com/cs/inspirationquotes/a/Ability12.htm>

Still doubt that conventional business wisdom isn't worth the paper it's not written on? Look at your Start menu or your favorite Web sites. Besides Microsoft (or Apple) products, how many of the apps or Web services you use are sold by obviously big companies, and how many are brought to you by inconspicuously small start-ups?

Game, set, and match.

In this chapter, I'll cover four topics: how we got here, what *here* looks like, how you can join the micro-ISV party, and what the cover charge is for getting in the door.

## How We Got Here

To understand just how you can make a bigger pile of money than you'd make in a hundred years of working in a corporation's cubicle, you'll have to jump in your handy time machine and go back in history—back before the Internet was public, back before Microsoft had more money than Norway, back when Osborne was a type of computer, back when laptops were bigger than suitcases, and back when coffee came in two flavors: Folgers and Maxwell House.

Let's say your time machine deposited you in San Francisco, California, in 1983. Ignoring the boring politics, economics, and all the rest, what was going on with software and personal computers?

- IBM had made the idea of Personal Computers (PCs) safe for businesses with its IBM PC two years before. PCs were springing up in offices all over the place.
- Hundreds of programs were available for PCs—either running CP/M or the newfangled MS-DOS operating system. But they weren't cheap; you had to buy them to try them, and most were saddled with copy protection schemes heavier than a 20-pound bicycle lock.
- Three programmers—one an IBMer in Bellevue, Washington; one an attorney and computer magazine editor based near San Francisco; and one a programmer who was one of the first programmers to leave Microsoft—each decided to buck this trend of expensive, shrink-wrapped software with a different approach. Jim Knopf (known as Jim Button), Andrew Fluegelman, and Bob Wallace were selling, respectively, a flat-file database, a modem application, and a word processor by giving the software away and requesting a small payment. All three programs, despite distributions limited to fledgling computer clubs, disk duplicators, and word of mouth, did extremely well financially. "I could not have predicted what would happen next," says Jim, in a piece he posted years later on the Internet.<sup>2</sup>

*My wife said I was "a foolish old man" if I thought even one person would voluntarily send me money for the program. I was more optimistic. I suspected that enough voluntary payments would come to help pay for expansions to my personal computer hobby—perhaps several hundred dollars. Maybe even a thousand dollars (in my wildest dreams!). But my tiny post office box was too small to receive the responses from a wildly enthusiastic public.*

*I had always said I would never consider leaving my secure job with IBM until I was receiving at least twice as much money from another source. I was wrong. By the summer of 1984 I was making ten times as much with my little software business.*

---

2. <http://www.freewarehof.org/sstory.html>



- [\*Cruelty and Desire in the Modern Theater: Antonin Artaud, Sarah Kane, and Samuel Beckett pdf, azw \(kindle\), epub, doc, mobi\*](#)
- [read online Draw Realistic Animals: Wildlife, Pets and More book](#)
- [The Party's Over: Oil, War and the Fate of Industrial Societies pdf, azw \(kindle\)](#)
- [read Pale Fallen Angel, Parts 1 & 2 \(Vampire Hunter D Novels, Book 11\) pdf, azw \(kindle\)](#)
  
- <http://bestarthritiscare.com/library/The-One-Year-Manual--Twelve-Steps-to-Spiritual-Enlightenment.pdf>
- <http://xn--d1aboelcb1f.xn--p1ai/lib/Draw-Realistic-Animals--Wildlife--Pets-and-More.pdf>
- <http://www.netc-bd.com/ebooks/The-Thousand-Emperors--Final-Days--Book-2-.pdf>
- <http://thermco.pl/library/Dominion--Warhammer--Von-Carstein-Trilogy-Book-2-.pdf>