

# ERIC SINK ON THE BUSINESS OF SOFTWARE

---

*Eric Sink*

Apress®

## **Eric Sink on the Business of Software**

**Copyright © 2006 by Eric Sink**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-623-4

ISBN-10 (pbk): 1-59059-623-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jim Sumser

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Matt Wade

Project Manager: Kylie Johnston

Copy Edit Manager: Nicole LeClerc

Copy Editor: Kim Wimpsett

Assistant Production Director: Kari Brooks-Copony

Compositor: Lynn L'Heureux

Proofreader: Liz Welch

Indexer: Carol Burbo

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

## Seven

---

# MAKE MORE MISTAKES

*Everybody seems to love this article. That is, everybody except me. ☺  
I do value the lessons I have learned, but sometimes it stings to  
think about some of these stories.*



FRIDAY, DECEMBER 19, 2003

Robert Scoble, weblogger extraordinaire, recently said, “I want to see more software companies, not fewer.”<sup>1</sup> I heartily agree.

At the risk of being too obvious, let us observe that every ISV is started by an entrepreneur who somehow overcomes the fear of failure. The genesis of a new company usually involves hundreds of hours of study, deliberation, and conversation, most of which is focused on a single question: Can I make this business work?

That process is healthy and necessary. It involves market research and number crunching and presentations and conjecture and coffee, all of which are critical elements of business success.

Invariably, the process also involves a great deal of self-examination. The core question isn’t just “Can this business work at all?” but, rather, “Can *I* make this business work?” This, too, is healthy and necessary. I’ve seen research studies that show that self-awareness is the number-one factor in success. There is no substitute for knowing your own abilities and limitations.

---

1. <http://radio.weblogs.com/0001011/2003/12/03.html#a5642>

But the self-examination stuff usually includes some basic worrying as well. Entrepreneurs tend to worry about the mistakes they might make. Unlike all the other research and study and deliberation that happens in the formation of a new company, this worrying actually isn't all that helpful.

To continue quoting only from the giants of the technology industry, let me now cite a remark made by Thomas J. Watson, Sr., founder and former CEO of IBM:

Would you like me to give you a formula for success? It's quite simple, really. Double your rate of failure. You are thinking of failure as the enemy of success. But it isn't at all. You can be discouraged by failure—or you can learn from it. So go ahead and make mistakes. Make all you can. Because, remember that's where you will find success.

I don't think Thomas J. Watson, Sr., advocated courage to the point of recklessness. After all, I'm sure IBM did plenty of prudent market research and planning during his tenure.

But he makes a very important point for new entrepreneurs: Mistakes just don't need to be all that scary.



## Endless Decision Making

Life in a small ISV feels like a never-ending stream of decisions, many of which you were never trained to make:

- Should we sign this five-year lease or negotiate for a shorter term?
- How should we incorporate?
- Should we seek outside funding?
- Should we write this code ourselves or buy a component?
- Should we host our own site or find co-location space?
- When should we spend money on advertising?
- If we build this product, will anybody buy it?
- Should we do consulting work to help our cash flow?

None of my college classes taught me how to make these choices, so I had to learn by doing. Actually it would be more accurate to say that I learned this stuff by doing it *badly*. In the seven years since I started SourceGear, I have made lots and lots of mistakes. Although the memories of my failures sometimes sting, the lessons I have learned have been so valuable.

Through all these mistakes, I have learned an important distinction. Whenever a decision yields bad results, we call it a *mistake*, but actually “clueless errors” are quite different from “bad bets.”

## Clueless Errors

In high school I once cost my math team a trophy by writing down “102/17” as my answer. I did all the calculations correctly, but I failed to realize that this fraction can be further reduced to “6.” Leaving my answer in the incorrect form was not a calculated risk—I simply did not see the alternative. My math team coach was quick to observe what a knucklehead I was, and he was right. This was a clueless error.

If any kind of mistake is worth worrying about, I suppose clueless errors qualify. These are the ones where you can’t see the bad result coming.

We can take some consolation in the fact that experience helps. Over time, we will tend to make fewer and fewer mistakes of this kind. However, that’s not helpful now, when you’re trying to decide whether your new company is going to make it.

To some extent, you can reduce the likelihood of clueless errors by reading. Almost every successful entrepreneur is a voracious reader. But take care: It is possible to learn about entrepreneurship by reading, but only if you approach it properly:

- It’s important to read skeptically. Think about the author’s context and compare it to your own. Does the advice really apply to your situation?
- Read a diversity of opinions. Don’t read one book on marketing—read ten.

Reading is not a place to find the answers to your problems. On the other hand, reading is a way to learn the background that can help you figure out your own answers.

## Bad Bets

Sometimes a so-called mistake is actually just a calculated risk that did not pay off. You were aware of the possibility of the bad outcome in advance. You placed a bet that the bad outcome wouldn't happen, and you were wrong.

This is basic risk taking. Study all the outcomes that might happen, analyze the probabilities, and make a choice. Remember to account for the fact that the status quo is a choice. If you don't want to become an expert at choosing your risks, then don't become an entrepreneur. A lot of your bets just aren't going to work out, but you still have to go back to the table and try again. There is no business without risk.

You also have to be prepared to deal with those who are waiting to gloat when your bets don't work out. When you take a risk and get burned, there will be a chorus of "I-told-you-so's" from people who would never have taken that risk in the first place. Sometimes their feedback is part of the learning experience, but not always. Perhaps their posture toward risk is simply different from yours. You are an entrepreneur, so by definition, you are somewhat more of a risk taker than most. In all likelihood, your personal sphere includes at least one person who would fold with four jacks when the opponent has a possible straight flush.

Sometimes you have to ignore the naysayers. If you know the odds are in your favor, you can go back and take the same risk again, even if people think you're crazy.



## My Stories

When I was interviewed after SourceGear won the *Inc.* 500 award<sup>2</sup> in 2002, they asked me to share the biggest surprise of my experience as an entrepreneur. I told them my biggest surprise was that someone could make as many mistakes as I have and still end up on the *Inc.* 500 list. ☺

---

2. <http://www.inc.com/app/inc500/viewCompany.jsp?cmpId=2002156>

I often share the following advice with other entrepreneurs: Make all the non-fatal mistakes that you can—don't make any of the fatal ones. Given that guideline, so far I'm rather happy with the mistakes I've chosen. As you are about to see, I've done some truly dumb things, but SourceGear is still in business today.

In the following sections I tell the stories behind some of my biggest failures at SourceGear. Some of these tales are really embarrassing, but they're still an important part of my history. I learned an important lesson from every one of them, and I have no regrets.

## A Failed Project

SourceGear's first year of business was 1997. We started out by doing custom software development projects for companies bigger than us. In fact, one of our first deals was with one of the largest and most well-known accounting firms in the world. This project was a big opportunity for us to get a “banner-name” client.

The project was to build a custom application to replace a very complex Microsoft Excel spreadsheet. The company we contracted with was stretching the limitations of Excel rather badly. We decided to build the application in Java. After determining the required features, we agreed to build the app for a fixed cost of \$28,000 (USD). The project was a disaster.

We knew our bid was a little low, but we realized in hindsight that we might have underbid by an order of magnitude. When the project finally got killed, we had expended well over \$100,000 in effort.

---

Lesson learned: Be careful about fixed-bid projects.

---

Circa 1998, betting on Java for a graphical user interface (GUI) application was suicidal. The platform simply didn't have the maturity necessary for building quality user interfaces. I chose Java because I was “head over heels” in love with it. I adored the concept of a cross-platform C-like language with garbage collection. We were hoping to build our Java expertise and make this exciting new technology our specialty.

But Java turned out to be a terrible frustration. The ScrollPane widget did a lousy job of scrolling. Printing support routinely crashed. The memory usage was unbelievably high.

I should have gotten over my religious devotion to semicolons and done this app in Visual Basic.

---

Lesson learned: Be careful about using bleeding-edge technologies.

---

## Buying an Office Building

When I first started the company, I rented one room in an office building. As the company grew, we simply rented more rooms. Eventually we started talking about moving. After considering several alternatives, I ignored the advice of a wise friend and bought an office building. Within a year we outgrew the building, and I sold it at a loss.

---

Lesson learned: Small ISVs should do software and stay out of real estate.

---

## AbiWord

In early 1998, Netscape announced that the source code for its browser would be made publicly available. As a former participant in the browser wars,<sup>3</sup> I found this news to be incredibly exciting. Somehow I knew that “open source” was going to become an important trend. I jumped on this bandwagon with both feet by launching AbiSource, an effort to build an open source office suite, starting with a word processor.

I had a big vision. I predicted that a wave of open source and Linux IPOs would happen and that AbiSource would be one of them. I began writing business plans and preparing to get funding. We wrote code like mad and unveiled the first public release of AbiWord at the O'Reilly Open Source Developer Day in August 1998.

AbiWord turned out to be a great way to get buzz. People really liked the story. Our tradeshow booths were packed. The press considered our idea a novelty and liked to write stories about us. In perhaps the strangest piece of attention we received, a Microsoft exec mentioned AbiWord in

---

3. [http://software.ericssink.com/Browser\\_Wars.html](http://software.ericssink.com/Browser_Wars.html)



the antitrust trials (referred to on page 16 of the transcript) as evidence that Microsoft Office was facing viable competition.

But although AbiWord was fun, it was never much of a business. Our funding search didn't go very well. The buzz of AbiWord got us in the door, but we always walked out with no money. Tim O'Reilly said "no." Bob Young said "no." Frank Batten said "no." Bill Kaiser said "no." Hindsight confirmed these gentlemen to be as smart as we all know them to be.

Hadar Pedhazur<sup>4</sup> also said "no," but he said more than that. Hadar is probably the most "clueful" venture capital guy I've ever known. He spent a lot of time talking with me and eventually convinced me that it would be extremely difficult to make the AbiSource business model work. The research and development costs of a word processor are simply too high to give it away. After my discussions with Hadar, I made the decision to abandon AbiWord as a business. (At the time of this writing, the AbiWord<sup>5</sup> project continues to move forward as a community project.)

---

Lesson learned: Investors don't like low-margin business models.

---

## RADish

After the death of AbiWord, the prevailing question in our company was "What now?" We still had a moderately successful developer tool product, but we wanted to do more, so we continued looking for new business ideas. We weren't ready to give up on the open source world, so we decided to get into developer tools with an open source twist. We designed a product that we code-named "RADish," intended to be "the Visual Basic of Linux." Suffice it to say that this idea didn't last long. Visual Basic is mostly about rapid development of desktop apps for the corporate environment. Linux didn't (and still doesn't) have any substantial penetration in that space. RADish was designed to solve a problem that almost nobody has.

---

Lesson learned: A market with no competition ain't.

---



---

4. <http://www.opticality.com/>

5. <http://www.abisource.com/>

## A Weasel

It seems like ancient history now, but during my AbiWord and RADish days I *really* wanted to get some outside capital funding. I was watching the dotcom bubble produce ridiculous valuations, and I wanted my company to play, too. I pitched my business ideas to all kinds of investors. When that didn't work, I decided to hire someone to do the fund raising. This turned out to be a big mistake.

I should have heeded the advice of Doug Colbeth. Prior to starting my own company, I worked as a programmer for Spyglass, where Doug was the CEO. Around the time of the Spyglass IPO, Doug humorously explained to us the hierarchy of the world's true scumbags:

- At the top of the scumbag pile, as the lesser of four evils, are the lawyers.
- Just below that you've got the people who cheat senior citizens with scams.
- A bit further down, you find the pimps.
- Finally, at the very bottom of the scumbag hierarchy, you find the investment bankers.

Of course, not all investment bankers are bad eggs. Still, there is some irony in the fact that Doug's joke turned out to have a grain of truth in it in my experience. When I was in my "funded company" funk, a guy whose firm specialized in raising private placement capital for small companies contacted me. I ended up signing an investment banking agreement with this guy who promised to connect us to some investors who had a whole bunch of capital. As you can already guess, this guy's help gained us no investors.

---

Lesson learned: The negative connotations of the word *middleman* are often deserved.

---

Believe it or not, this tale gets even worse. I failed to have an attorney review this agreement, and it had a big problem in it. We ended up paying this guy \$40,000 cash to cancel the agreement.

---

Lesson learned: All contracts must be reviewed by an attorney. No exceptions.

---

## Investing Outside Our Field

I wish working with the investment-banking weasel was my worst financial decision. Unfortunately, it doesn't even come close.

In the fall of 2000, we were hoping to get more involved in building high-end Web sites. Somebody introduced us to a newly formed print magazine that had big plans for expansion. The people producing the magazine were looking for seed capital as well as for a technology partner to help them get their magazine onto the Web. We decided to jump in with both feet:

- We invested \$150,000 cash for an equity stake in their company.
- We agreed to set up a basic content management system, for which they would pay us \$100,000 of the money they just got from us.
- For the purpose of building their Web site, we bought some rather expensive content management tools out of our own pocket.
- They agreed to pay us several hundred thousand dollars more for "phase 2" of the content management system development.

In the end, this deal turned out to be a financial nightmare. They never found any other investors. They never paid us for phase 1. Phase 2 obviously never happened.

---

Lesson learned: Cash is supposed to flow from your customers to you, never the other way around.

---

## Building the Platform Under the Application

Not all of my boneheaded moves have been financial. I've somehow found time to work in a few truly ridiculous technology choices as well.

After failing at word processors and magazine publishing, we decided perhaps it was time to refocus on our core competency: developer tools.

Through it all, SourceOffSite had continued to be our flagship product. In late 2000 we decided to build another edition of SourceOffSite with some more collaborative features built in. This product (which we do still sell) is called SourceOffSite Collaborative Edition, or “Collab” for short.

Actually, Collab has thousands of customers, and they generally like it very much. It’s not a bad product at all. The biggest problem with Collab is that we spent far too long building it. We built too much of the underlying platform instead of building on the platforms that were already available:

- Collab includes a Web-based bug-tracking system. The sensible thing to do would have been to simply build it using ASP and IIS. Unfortunately, I felt some crazy desire to make the problem ten times harder, so we decided to build our own Web server and our own ASP-like templating engine.
- Aside from the bug tracking, all the other features of Collab are built on a separate server that uses a message system based on XML. The sensible thing to do would have been to use XML-RPC or SOAP. Here again, we found those existing technologies to be just slightly wrong for our needs, so we built our own.

Everything crystallized for us at the 2001 Professional Developers Conference (PDC). Listening to all the presentations about the new Microsoft .NET, we realized that we had created from scratch our own implementation of Web services, completely incompatible with any other.

---

Lesson learned: Small ISVs should build apps, not platforms.

---



## Summary

As I write this, it has been just over two years since the 2001 PDC. In hindsight, that event was the time when many of the lessons I had been learning really started to get put to use. Just after that PDC, we made a lot of tough decisions about our company direction. We got ourselves

out of the contracting business, which allowed us to focus exclusively on our own products. We decided to build Vault, which has been very popular. Since then, SourceGear has hit its stride and found its identity. In the future, we will face new challenges, and once again we'll try to figure out which are the best risks for us to take on.

As you can see, I've made some clueless errors, and I've taken some risks that didn't work out at all. These mistakes dovetail with my successes and form my history. Truth be told, *every* CEO has similar tales to tell. If everybody told their stories, we would almost certainly see that success and failure are as strongly correlated as Thomas J. Watson, Sr., said.

I hope you have read these stories and said to yourself, "I can do better than that!" These stories are my contribution to stamping out the fear of failure. As Scoble said, we need more small ISVs, not fewer.

