On Startups: Patterns and Practices Of Contemporary Software Entrepreneurs

By
Dharmesh Shah

B.S. Computer Science, University of Alabama Birmingham

Submitted to the Alfred P. Sloan School of Management in Partial Fulfillment of the Requirements for the degree of

Master of Science
In Management of Technology

at the
Massachusetts Institute of Technology
June 2006

Signature of Author: ______________________________
Alfred P. Sloan School of Management
May 16, 2006

Certified by: ______________________________
Professor Michael A. Cusumano
Thesis Advisor

Certified by: ______________________________
Professor Edward B. Roberts
Thesis Advisor

Accepted by: ______________________________
Stephen J. Sacca
Director, MIT Sloan Fellows Program in Innovation and Global Leadership

On Startups: Patterns and Practices Of Contemporary Software Entrepreneurs

By
Dharmesh Shah

Submitted to the Alfred P. Sloan School of Management on May 12, 2006 in partial fulfillment of the requirements for the Degree of Master of Science in Management of Technology

Abstract

> *"When you write a book, you need to have more than an interesting story. You need to have a desire to tell the story. You need to be personally invested in some way. You need to care about it."*
>
> Malcolm Gladwell
> Author, "The Tipping Point"

"I have never thought of writing for reputation and honor. What I have in my heart must come out; that is the reason why I compose."

Ludwig van Beethoven

The above quotes answer the basic question that many people have asked me (and I have asked myself): Why are you writing a thesis? The answer is simple: I believe there's a story to be told, and I have a personal passion for the subject.

This thesis will analyze the patterns and practices of contemporary software startups. Though much has been written about software companies and much has been written on startups, I'm primarily interested in the intersection: software startups. More specifically, I will explore contemporary software startups that were started after the Internet bubble burst.

About The Website: onstartups.com

As part of the thesis writing experience, I started a blog called "OnStartups" (http://onstartups.com) where I have been posting articles on software startups. More information on the onstartups.com site is included at the end of the thesis.

My passion for learning more about software startups, the opportunity to work with two exceptional individuals as co-advisors and the overwhelming response to my onstartups.com website are my primary motivations for this thesis.

Onward!

Thesis Advisor: Michael A. Cusumano
Title: Professor of Management

Thesis Advisor: Edward B. Roberts
Title: Professor of the Management Of Technology
Chair, MIT Entrepreneurship Center

Acknowledgements

First, and foremost, I want to thank my wife, soul-mate and soon to be world-wide traveling partner, Kirsten Waerstad. Her staunch support while I toiled away in "my cave" made this thesis possible. Without her love and understanding I could not have pulled it off. I'm sure she'll be glad to have her husband back.

I would also like to thank my co-advisors, Professor Michael Cusumano and Professor Ed Roberts. I could not have hoped for a better pair of individuals with whom to work with on this project. MIT is a great place to learn about (and start) software companies and they know more about software and new ventures than anyone. Their accomplishments continue to inspire me and I hope someday that I will succeed in such a way that they will be able to say: "We knew him when he was just a stressed-out graduate student of ours..."

I'd like to thank the MIT Sloan community and the MIT Sloan Fellows Program. It has been a great two years and I am forever indebted to the faculty, staff and my fellow classmates that made these years at MIT transformative along so many dimensions.

The following individuals all agreed to be interviewed as part of my research: Ray Deck, Don Dodge, Jason Fried, Brian Halligan, Doug Levine, Michael McDerment, Barry Moltz, and Bob Walsh. I recognize how busy you all are, thanks for taking the time.

Last, but not least, I would also like to thank the thousands of software entrepreneurs around the world that read my OnStartups.com articles. They have provided moral support, encouragement and unique insight regarding the fascinating topic of software startups. Knowing that in some small way I am now connected to all of them has made the thesis work that much more meaningful.

Table of Contents

Chapter 1: Introduction

The focus of this thesis is to look at the intersection of software and new ventures by exploring contemporary software startups and the people that found them. I'll be looking at software startups from a variety of perspectives ranging from finance and capitalization to sales and marketing. As a software developer and entrepreneur myself, my larger goal for this thesis is the same as I would have for a software product: I want to create something *interesting* and *useful.*

The thesis will concentrate on software *products* companies and not software *services* companies. Admittedly, the line between product and services companies is becoming increasingly blurry as both products and services companies trend towards being "hybrid" companies.[1] However, I will limit the scope of this thesis to companies that primarily generate revenue from a "write once, sell many" business model. Even companies that sell software as a service (like Salesforce.com) still fit this definition as they make a majority of sales on a product that was written once and generally speaking, each customer sold is not labor intensive.

Software Industry Overview

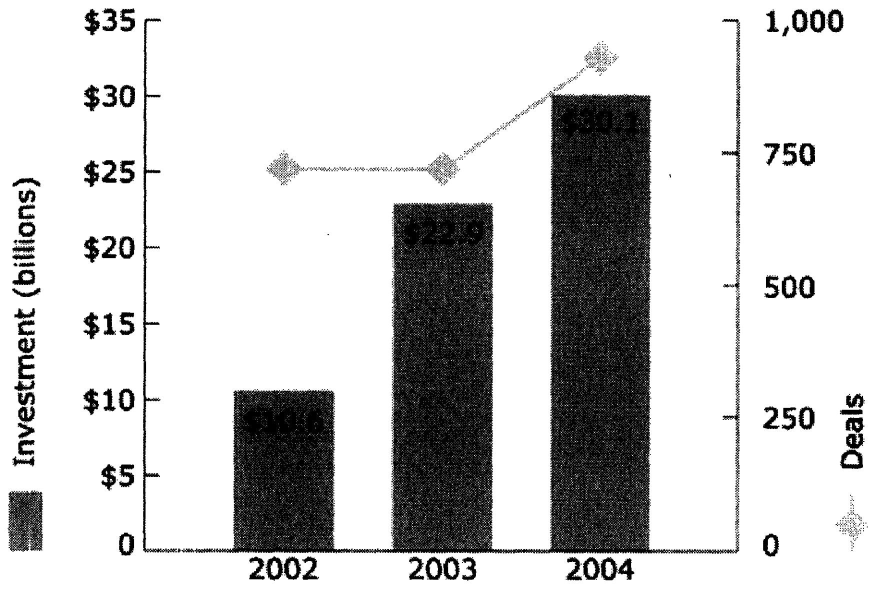
The software industry continues to be exciting and vibrant. The worldwide software industry for all platforms was estimated by IDC at $179 billion in 2004 – a 5% growth over 2003 revenues. The U.S. is estimated to represent approximately 50% of the world market.[2] The U.S. software market grew by 3.9% in 2005 and reached a value of $75.6 billion.[3] According to the National Venture Capital Association, software continued to be the single largest investment category for venture capitalists in Q3 2005, with approximately 185 companies funded and raising over $1 billion. Software companies continue to be one of the most common types of hi-tech startups in the U.S.[4]

[1] Cusumano, "The Business Of Software"
[2] Software and Industry Association, "Packaged Software Industry Revenue and Growth"
[3] DataMonitor, "Software In The United States"
[4] Rangaswami, "Seismic Shifts for the Software Market"

Industry Consolidation

Over the past several decades, the software industry has gone through significant consolidation and matured. In 2004, there were 19 public software companies acquired which reduced the net number of public software companies to 237[5]. This consolidation has a significant impact on the industry – especially as it relates to software startups. Consolidation results in a fewer number of companies with much greater market power. Almost 80 percent of the profits in the industry are concentrated in three companies: SAP, Oracle and Microsoft.[6]

Software M&A Trend



Source: Thomson Financial

Exhibit 1. Source: Thomson Financial

Large companies such as Microsoft, Oracle and SAP are now dominant in the market making it exceedingly challenging for new software startups to capture meaningful market-share. As a result, the industry is now divided into two segments: Large players focusing on broad market opportunities and smaller ones that are generally relegated to targeting niche markets with limited growth opportunities.

[5] Burnham, Bill "Software's Top 10 2005 Trends"

[6] Pallatto, "U.S. Software Faces Loss Of Global Domination"

We will explore the challenges presented by today's software landscape and how contemporary software startups are addressing these challenges.

Personal Experience

I have been involved in the software industry in some capacity for almost 20 years. For over twelve years I have been leading software startups that I have founded and managed. As this thesis is being written, I am working on HubSpot, my third and most recent software startup. In addition to my own companies, I have recently been an angel investor and/or advisor to approximately five other software companies. These eight startups in which I've been involved represent a diverse mix of companies from high-end enterprise software to small business and consumer products. Some have raised external capital while others are bootstrapped. Some were reasonably successful, others were not.

Where possible, I will attempt to draw from my own personal experiences and offer first-hand examples and insights where appropriate.

Document Structure

The following are the chapters included in the thesis with a brief summary of the topics they address.

The Entrepreneurial Leap: An look at the risks and rewards that impact software entrepreneurs and a determination of why individuals start software companies.introduction.

Financing and Capitalization: An exploration of the various financing options available to software entrepreneurs including angel investment and venture capital including the advantages and disadvantages of each. Also a look at bootstrapping whereby companies are founded with little or no outside capital.

Market Positioning and Strategy: An analysis of how software startups position themselves within the market including a look at horizontal vs. vertical market strategies. Also looks at the differences between enterprise and consumer software markets.

Product Development : Analysis of what causes the high failure rate in software development projects and how to address them. Also looks at development models and practices and the factors influencing product development success.

Acquiring Customers: Outlines how contemporary software startups are addressing the challenge of customer acquisition. Explores recent trends in online marketing and advertising as well as business blogging

Software Business Models: Contrasts traditional software business models such as perpetual licensing with more recent approaches such as Software As A Service (SaaS) and identifies the advantages and disadvantages of both..

Case Studies, Interviews and Survey: Startup case studies and interviews with current software entrepreneurs and industry experts that advise them. Also includes a summary of the results of an online survey of software entrepreneurs conducted for this thesis.

Tips for Entrepreneurs: Provides practical guidance and advice for software entrepreneurs based on readings and discussions with successful entrepreneurs.

Conclusion: Summarizes the material in the thesis and attempts to look forward at how the industry will progress and provides a position on opportunities for software entrepreneurs in the future.

Chapter 2: On The Entrepreneurial Leap

Questions:

1. What are the common circumstances under which software entrepreneurs take the leap and start companies?
2. How do software entrepreneurs come up with the ideas for their startups?
3. Why do entrepreneurs start companies in the first place?

Over half (56%) of all Americans dream of starting their own business. This is not surprising given that in a 2003 Conference Board poll on job satisfaction, U.S. job satisfaction hit a record low. More and more individuals feel like they are "disengaged" from their jobs and are seeking the entrepreneurial experience.[7]

Having started three software companies myself, and being involved in an advisory capacity to about half a dozen more, I've become very interested in the software entrepreneurial process.

Software vs. Other Types Of Startups

One of the factors that has made software startups (vs. other types of startups) so attractive for aspiring entrepreneurs is the potential rewards. In 1998, during the economic boom years, 22 of the world's 40 richest people came from the software firms of Microsoft, SAP and Oracle.[8] This type of financial reward can be a strong driver for attracting some of the best entrepreneurial talent.

Software entrepreneurs have a certain set of "patterns" that make them different from other types of entrepreneurs. Many of the software entrepreneurs I have encountered and advised have a software development background. They have worked as programmers in

[7] Judson, "Go It Alone"
[8] Forbes Magazine, June 1998

larger, more established organizations and then decided to take the leap and start companies of their own.

Further, there continues to be rapid and significant change in the software industry creating ongoing opportunities for software entrepreneurs. Large shifts such as that from mainframe to mini-computers, mini-computers to PCs and PCs to browser-based software have created a continuous stream of business needs from the software marketplace. This type of significant change generally does not occur in other industries with the same degree of frequency.

Motivation for Starting

One of the most common reasons cited for leaving their existing positions and launching software companies is that the established company was not a software company and as such did not value the software development process. In this context, the process of creating software was viewed as a necessary evil (i.e. overhead). Exceptionally talented programmers working for non-software companies tend to get frustrated with this lack of recognition and are motivated to either change companies (to one that does value software development) or start their own.

A major factor for why programmers start software companies is the simple fact that it's relatively *easy* to do so. The capital costs are minimal (a computer and a compiler). There are very few other types of product companies which provide the potential for high rewards that can be effectively started by just one person.

Though there are certainly a variety of opinions on why individuals should start businesses, there are some surprising reasons why some experts posit should *not* be the primary motivation for taking the entrepreneurial leap[9]:

1. Making More Money: Though it is certainly possible for individuals to create exceptional personal wealth for themselves – much more than they would likely

[9] Moltz, "You Need To Be A Little Crazy"

do working for an established firm, this opportunity comes at substantial risk. Many people that would make great entrepreneurs would also have likely made great executives at large companies – and also likely have made more money in most cases, once risk is factored into the equation.

2. Having More Power: Though it would seem that entrepreneurs can have more "control" over their destiny by starting their own company, the startup path is also replete with its own set of constraints and controls – just different ones. Startup founders often have to focus acutely on the needs of a small set of customers, investors and last but not least, their management team.

Of course, the fact that these two motivations (more money and more control) should not be the primary drivers does not change the fact that many first-time entrepreneurs get started precisely for these reasons. Entrepreneurially-minded individuals crave the independence of going out on their own and the potential financial reward that building a successful startup can bring.

> "You need three things to create a successful startup: to start with good people, to make something customers actually want, and to spend as little money as possible. Most startups that fail do it because they fail at one of these. A startup that does all three will probably succeed."
>
> Paul Graham
> Founder, ViaWeb

The Big Idea or The Big Escape

It is not uncommon for aspiring entrepreneurs to believe that in order to get started, they need to have “the big idea”. However, most startups, especially the successful ones, end up pursuing businesses that are often significantly different from the initial idea. Paul Graham, a software entrepreneur and angel investor states, “The fact is, most startups end up nothing like the initial idea. It would be closer to the truth to say the main value of your initial idea is that, in the process of discovering it’s broken, you’ll come up with your real idea.”[10] This theme is common across the software entrepreneurs interviewed for this thesis. In most cases, the ultimate idea that the business pursued was different from what was originally conceived. The business was launched with one idea, but ended up pursuing another.

Further, there is evidence from the research of Professor Ed Roberts from MIT to indicate that many startup entrepreneurs don’t start businesses because they are motivated by some big idea, but because they are looking to escape from the instability or frustrations of their current job.[11] These views are echoed by Mukti Khaire, in assistant professor at the Harvard Business School who studies new ventures. States Khaire, “Many entrepreneurs strike off on their own because they get tired of the way things work in their previous organizations and are determined to do things very differently.”[12]

Understanding Startup Risks

It is common wisdom that most startups are likely to fail. Taking the entrepreneurial leap has considerable risks. Despite the fact that most startups fail and that this is widely known by most entrepreneurs, it is interesting that just about all entrepreneurs believe that they will magically defy the odds and succeed with their own ventures. It seems that this “suspension of disbelief” is a necessary ingredient for the entrepreneurial process.

[10] Graham, “Ideas For Startups”
[11] Roberts, “Entrepreneurs In High Technology”
[12] Gilbert, “How Can Startups Grow”

In a paper titled "Entrepreneurial Risk and Market Entry", Brian Wu and Anne Marie Knott make a number of significant observations:[13]

- To date, the empirical literature has consistently found that entrepreneurs' risk profiles seem to be indistinguishable from regular wage earners.
- There are actually *two* forms of uncertainty in entrepreneurial ventures: market demand uncertainty and entrepreneurial ability uncertainty.
- Entrepreneurs display risk aversion when it comes to market demand uncertainty but exhibit over confidence or "risk-seeking" tendencies with regards to ability uncertainty.
- The above means that entrepreneurs are willing to bear economic risk when overconfidence compensates for their risk-aversion regarding the market.

So, the above report may partly explain why so many entrepreneurs take the leap of faith when in fact it may not be in their best financial interests to do so. The primary driver may simply be that their confidence in their abilities may more than compensate for their lack of clarity on market demands and opportunity.

Entrepreneur's Advice: Get Going!

Despite the risks and the widely discussed trials and tribulations of entrepreneurship, people continue to want to start companies. At the 2005 Entrepreneurship Conference held at the Harvard Business School, the common advice from the entrepreneurs speaking on the panels was: "Jump in, take risks and challenge yourself."[14] From both these and the entrepreneurs I've spoken to as part of this thesis, the message is consistent: Get going!

Software entrepreneurs looking to start companies should not spend a majority of their time and energy in the early stages writing business plans, crafting investor pitches or for that matter, creating any documentation at all. They should *implement*. One of the

[13] Wu and Knott, "Entrepreneurial Risk and Market Entry"
[14] Silverthorne, "Lessons of Successful Entrepreneurs"

biggest advantages of developing software (compared to other kinds of products) is that it's *virtual*. Software entrepreneurs can create a prototype in days or weeks and have customers review the early product. There is no clearer articulation of how an entrepreneur will solve a customer problem than by building a partially working product.

Chapter 3: On Financing and Capitalization

At a presentation I gave recently, the audience's questions were all along the same lines: "How do I get in touch with Venture Capitalists? What percentage of the equity do I have to give them?" No one asked me how to build a business!

Arthur Rock
Venture Capitalist

Questions:

1. After the dot-com bust, are VCs still interested in software startups?
2. What are the pros and cons of the major types of external financing?
3. How do software entrepreneurs go about "bootstrapping" their companies to success – with so little cash?

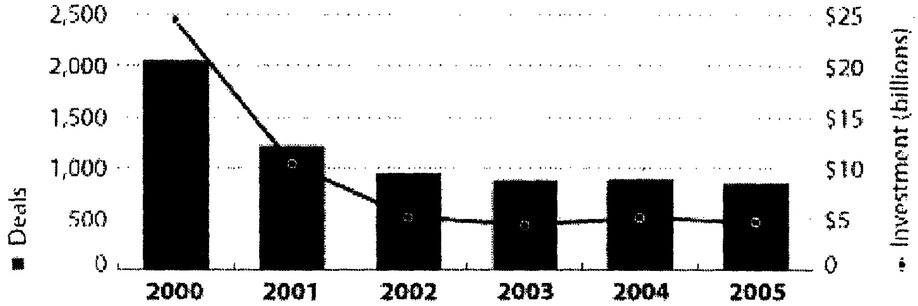
Venture Capital Financing

The venture capital industry is a key component of the entrepreneurial landscape. A recent study by Global Insight states that venture capital backed companies employed more than 10 million American workers and generated $1.8 trillion in sales in 2003.[15] Though the volume of venture capital investment in software decreased by 10% in 2005 to $4.7 billion in 840 deals, it is still the single largest industry category accounting for 22% of the total dollars and 29% of all deals.[16]

VC investment in Internet related companies, which is an area of active interest to many software startups, represented approximately 48% of the companies financed in 2005.[17]

[15] Global Insight. "Venture Impact 2004"
[16] National Venture Capital Association, Press Release
[17] Dow Jones, "Venture Capital Industry Report – 2006"

U.S. Software VC Investment Trend



	2000	2001	2002	2003	2004	2005
Deals	2,047	1,215	954	880	887	849
Investment (billions)	$24.4	$10.4	$5.2	$4.4	$5.1	$4.7
Deal average (millions)	$11.9	$8.6	$5.4	$5.0	$5.8	$5.6

Source: PricewaterhouseCoopers/Thomson Venture Economics/National Venture Capital Association MoneyTree Survey

Exhibit 2. Source: National Venture Capital Association

From a software startup's perspective, there are several benefits derived from raising venture capital. The following chart identifies some of these benefits.

Finance	The venture capitalists injects long-term equity finance, which provides a solid capital base for future growth.
Future Funding	VCs can provide future funding to their portfolio companies based on needs
Business Partner	The VC can become a business partner, sharing the risks and rewards. The VC is rewarded by business success.
Mentoring	The VC can often provide strategic, operational and financial advice to the entrepreneur based on past experiences in similar situations.
Broad Network	The VC has a network of contacts in many areas that can add value to the startup. Assistance can be provided in areas such as recruiting key personnel, providing sales contacts, locating strategic partners, and attracting other investors.

Exit Assistance	The VC is experienced in the process of preparing a company for an eventual exit – such as an IPO or an acquisition/merger.

Exhibit 3. Source: National Venture Capital Association

Based on the immense amount of venture capital flowing into startups – especially software startups, many entrepreneurs are inclined to pursue the VC path. To some degree, this is driven by the immense number of resources that are all aligned with helping entrepreneurs navigate the venture capital raising process. There are books, courses, advisors, brokers and lawyers that all specialize in helping entrepreneurs understand the venture capital raising process and help raise money.[18]

However, from a founder's perspective, raising venture capital is not always the optimal path. Venture capital is one of the most expensive forms of capital available to an entrepreneur. The chances of a venture capital financed company succeeding and generating a return for entrepreneurs is relatively low. About 60% of the companies that are venture funded eventually go bankrupt. Another 30% are sold for less than the capital invested – which generally means that the founders and management team receive minimal funds from the sale. Even if an entrepreneur is able to beat the odds and successfully exit the business and achieve liquidity, the average expected value for the founder in a venture backed startup over five years is about $6.5 million.[19]

Many serial entrepreneurs, industry experts and other members of the startup community advise against venture financing for software startups. The following chart provides some of the most relevant reasons why software startups may want to consider alternatives to venture capital financing.[20]

[18] Bhide, "Bootstrap Finance: The Art of Startups"

[19] Nesheim, "High Tech Startup"

[20] Ireland, "10 Reasons To Shy Away From Venture Capital

Tempting Distraction	The decision to pursue venture capital is often a tempting distraction from the much more complex and important tasks of creating something to sell and persuading someone to buy it.
Flavor Of The Month	Venture capitalists often behave like sheep, investing only in whatever industries are the flavor of the month. Everyone else need not apply.
Brain Sucking	Entrepreneurs are often shown interest by VCs simply to help VCs learn more about a given industry and sector.
Misalignment	VC terms can frequently work to erode and ultimately destroy the founding team's motivation and commitment by misaligning interests.
Loss of Control	To protect their investment, VCs often exert power and control – even though they may know little about the business.
Time Allocation	Precious founder time can be spent on serving the needs of investors:creating reports, attending meetings, and comforting impatient or meddle-some investors.
Race To Liquidity	VCs need to achieve liquidity for their limited partners, even when it may make no sense for the founders.
Firing The Founder	In times of distress, often the only available recourse for VCs is to fire the founders. As such, they may exercise this option more often than necessary.

Exhibit 4. Source: "Ten Reasons To Shy Away From Venture Capital"

Angel Funding

An angel investor is an individual that provides capital to a startup venture. Angel investors are often perceived as sitting somewhere between the capital that can be raised from friends and family and full-fledged capital investors. Though the level of capital invested by an angel investor is usually much lower than that of a VC, multiple angel investors can group together and invest larger amounts of capital.

U.S. angel investments rose 2.7 percent in 2005 compared with 2004. The total angel investments reached $21.3 billion in 2005 and the number of ventures backed by angels also increased by 3.1% to 49,500. By way of comparison, $21.7 billion was invested by venture capitalists across 2,939 deals. Of further interest to entrepreneurs should be the "yield rate". In 2005, the percentage of deals presented to angel investors that actually got invested in rose to 23 percent from 10.3 percent in the prior year.[21] This increase in yield is causing some concern within the investment community as it may be an indication that weaker companies are landing investments. For women entrepreneurs, the yield was even higher. 33 percent of women entrepreneurs that sought investment capital from angels received it.

The following are some of the advantages of angel investment as compared to other sources of capital.[22]

1. **Smaller Investment Sizes:** Angels often prefer smaller investments as compared to VCs. For software entrepreneurs, this is often advantageous as lower amounts of capital are often desired.
2. **Invest in Early-Stage Startups:** These are generally the types of firms that have the most difficult time raising capital.
3. **More Flexible Financial Decisions:** Compared to VCs, angels often have different investment criteria, longer investment horizons ("patient money"), shorter investment processes, and lower target rates of return.

[21] Calnan, "Report: Angel Investments Up, Yield Raises Flag"
[22] Van Osnabrugge and Robinson, "Angel Investing"

4. **Lower Fees:** Raising angel funds often does not incur the high fees associated with raising funds from institutional investors like VCs.
5. **Value-Added Investors:** Angels can often contribute their business skills to helping young businesses. This free assistance and advice can often be invaluable.
6. **Facilitates Future Investment:** Obtaining money from angels has a leveraging effect in that it makes the startup more attractive to other sources of capital. Angel investments can heighten VC interest in such ventures.
7. **Loan Guarantees:** In addition to the capital invested, angels can often help the startups raise additional funds through loans by possibly guaranteeing these loans.
8. **Not Averse To High-Risk Technology Investing:** Angels are willing to take on the risk of investing in high-tech firms (such as software companies). In the U.S., 60 percent of all young technology firms looking for $1 million or less are funded by angels.

In contrast to the potential advantages of angel investing, there are also some disadvantages:

1. **Less Follow-On Investments:** Angels are generally less likely to make follow-on investments than VCs – which spend around two-thirds of their funds on expansion funds for their existing portfolio companies.
2. **Deeper Involvement:** Angels may want to be more closely involved in the running of the firm, but may not have the direct experience necessary to do so.
3. **Conflicting Motives:** On rare occasions, angel investors may be "devils" who have self-serving motives for their investment (beyond just increasing the value of their investment by helping the company).
4. **Lower Reputation Value:** Unlike VCs, angels often do not have the national reputation and prestige of a big-name institutions.

Bootstrapping

Bootstrapping is a term used to describe businesses that are launched with little or no external funding.

" Someone once told me that the probability of an entrepreneur getting venture capital is the same as getting struck by lightning while standing at the bottom of a swimming pool on a sunny day. This may be too optimistic. "

Guy Kawasaki

Raising external funding (through VCs or angels) is not as common as one might suspect. Although VC funded companies get most of the press and visibility, fewer than 1 in 10,000 companies in the U.S. receive their initial funding from venture funds.[23]

Though many entrepreneurs may consciously decide to not raise significant outside capital from external sources, most are simply unable to raise such financing because they lack the critical set of ingredients that investors look for.[24] In fact, successful software entrepreneurs like Greg Gianforte, the CEO of RightNow technologies, advocate that venture capital is not a requirement for software companies to get started. In support of this position, Mr. Gianforte states that less than 1% of software companies get VC funding – and most are launched with less than $20,000.[25]

An alternative to raising external financing is bootstrapping. Bootstrapping can be done by a combination of two things:

1. Using creative ways of acquiring financing without having to raise it from traditional sources.
2. Using strategies to minimize the need for financing by acquiring resources at little or no cost.

In the bootstrap model, founders often rely on personal savings and ultimately customer revenues to fund growth. One of the major advantages of bootstrapping is that founders can focus a majority of their energies on building their desired product instead of writing

[23] Cook, John "Venture Capital: Success without VCs abounds in business"
[24] Harrison, "Financial bootstrapping and Venture Development in the Software Industry"
[25] Gianforte, "Bootstrapping: The Secret to Entrepreneurial Success"

a business plan and pitching VCs. The following table summarizes some of the advantages of bootstrapping. [26]

Benefit	Rationale
Immediate contact with customers	Founders are forced to deal with customers and address their needs from day one
Scarcity of cash increases sales	Excess cash allows startups to avoid having to learn about sales – which is often one of the hardest things to do.
Minimal Waste	The more money a startup has, the more likely it is to spend it – often inappropriately.
Optimized Use Of Time	Instead of spending time (and money) raising funds, founders spend it on solving actual problems
Maximize Earnings	Bootstrap founders will often own a majority of their companies at the time of exit , far more than compared to VC backed companies
Independence and Freedom	Founders retain control of their company and can pursue their original dream
Forced Creativity	Having no money forces entrepreneurs to be creative and seek alternatives

Exhibit 5. Source: "Financial Bootstrapping and Venture Development in the Software Industry"

In a study of software startups conducted by Richard Harrison, et al., it was found that over 95% of the companies used one or more bootstrapping techniques.[27]

[26] Harrison, "Financial Bootstrapping and Venture Development in the Software Industry"
[27] Harrison, "Financial bootstrapping and Venture Development in the Software Industry"

The following table provides some useful ideas from Guy Kawasaki on tactics for entrepreneurs that want to take the bootstrap path.[28]

Idea	Rationale
Focus on cash flow, not profitability	The theory is that profits are the key to survival. If you could pay bills with theories, this would be fine. The reality is that you pay bills with cash, so focus on cash flow.
Forecast from the bottom up	Instead of using a "small percentage of a large market opportunity" or "top-down" approach, a bottoms-approach whereby the capabilities of the startup sales process are used to predict sales yield much more accurate predictions.
Ship, then test	Startups should beware the dangers of trying to ship the "perfect" product. Cash flows start when you ship, so ship early.
Forget the "proven" team	Bootstrappers have to hire who they can afford and turn them into great employees. Trying to hire the "proven" team is just too expensive.
Start as a service business	Starting has a service business provides immediate revenue and real customers to test the software.
Pick your battles	Startups can not afford to fight on all fronts simultaneously – they don't have the resources to do so. Instead, they should pick the appropriate battles and remain

[28] Kawasaki, "The Art Of Boostrapping"

	focused.
Go direct	The optimal number of people in between a startup and its customers is zero. This gives the ability to learn about customer needs, frustrations and future demands.

Exhibit 6. Source: "The Art Of Bootstrapping"

However, bootstrapping does have its challenges. Though on the one hand, bootstrapping can instill a very important culture of discipline in the early stages of a startup, it may also serve to limit firms from growing as fast as they might have otherwise done. In a survey conducted by McKinsey and Company, results showed that on average, successful software companies used on average, seven times more startup capital than the less successful ones.[29]

Bootstrapping Techniques
Product Development
Special deals for access to hardware
Pre-paid licenses, royalties, or advances from customers
Development of product at nights and weekends
Research grants
Customer funded R&D
Commercializing university-based research
Turning a consulting project into a commercial product
Using open source development tools
Business Development
Delay payments
Barter arrangements
Personal Credit Cards
Home equity/mortgage loans

[29] Hock, "Secrets Of Successful Software"

Discounted advanced payments from customers
Below market or very low rent space
Deals with professional service providers at below competitive rates
Leasing vs. purchasing assets
Working out of home
Gifts or interest-free loans from relatives
Reduced compensation
Outsourcing key parts of the business

Exhibit 7. Source: Harrison, "Financial bootstrapping and Venture Development in the Software Industry"

Capital Efficiency In Software Startups

Software companies have generally known to be capital efficient. There is no requirement to make a massive upfront investment in infrastructure, heavy machinery or real-estate. This is one aspect that makes them appealing to both entrepreneurs and venture capital investors. Ann Winblad, of Hummer Winblad Venture Partners started the first venture fund focused exclusively on software companies in 1989. When Ms. Winblad was asked for the reason for the focus on software, she responded: "Software is very capital efficient."

Case In Point

Doug Carlston and his brother bootstrapped a software company that sold computer games. The founders did not have sufficient capital to afford paying printers to print the help guides for their games. But, they were able to convince a local printer to defer payment until they could pay him. The founders spent their savings and maxed out their credit cards. This, and about $8,000 in loans from relatives helped them grow the company slowly. They did not accept venture capital until after the company turned profitable. The company, Broderbund, eventually went public and hit its peak in 1995 with a market capitalization of over one billion dollars.[30]

In recent years, software companies have become even *more* capital efficient. Before, it may have cost about $7-$10 million to build a product and get it to market.[31] Resources were spent on tools, middleware, databases, compilers – and of course, people. Today, software products can be created using open source tools, free "infrastructure" software (databases, operating systems, etc.) and outsourced development resources around the globe. As such, it is much less expensive to build a comparable product today, than ever before.

"Excite.com took $3 million to get from idea to launch. JotSpot took $100,000"

Joe Kraus
Founder, Excite
Founder, JotSpot

[30] Carlston, "Software People"
[31] Miller, "Software VC Outlook: A Flight To Quality"

The following chart illustrates the primary reasons that software startups today can be created as a much lower cost than before.

Change	Description
Hardware is Cheaper	Moore's Law continues to drive the price of computing power down. Commoditization and standardization (such as Intel processors) continues to push prices downward.
Infrastructure Software Is Free	Linux, MySQL and other open-source projects provide free (as in "beer") software for many of the common infrastructure needs of startups.
Access To Global Labor Markets	Startups today have unprecedented access to global markets. There are a number of online services and exchanges that make it possible for startups to tap into the global talent pool with reasonable efficiency.
Online Marketing and Advertising	Ten years ago, startups often has to forge expensive distribution deals to access markets. Or, they may have had to spend money on expensive advertising (TV, print, etc.) to gain market visibility. Today, startups can affordably reach their markets over the Internet both in terms of online marketing and online advertising.

Exhibit 8. Source: "It's a Great Time To Be An Entrepreneur"

Wrap-up

Software entrepreneurs continue to have access to external financing. Venture Capitalists continue to invest in software companies and angel investments are addressing some of the "financing gap" for companies requiring less capital.

However, there are clear advantages to the entrepreneur for delaying external financing as long as possible. By bootstrapping during the early stages of the company, it is possible to mitigate some of the risk for investors and thereby retain a larger portion of the company equity.

Chapter 4: On Market Positioning and Strategy

Questions:

1. How should startups go about selecting the market that they enter?
2. What are the benefits and disadvantages of focusing on big customers vs. small customers.
3. What's a Web 2.0 company and is this a good way to start an Internet software company?

Enterprise Software Markets

In the early years of the software industry, the primary consumers of complex software systems were large enterprises which could leverage technology for strategic advantage. This need within large enterprises gave rise to many of the well recognized categories for "enterprise software" such as customer relationship management (CRM), enterprise resource planning (ERP), business intelligence (BI) and content management systems (CMS).

Each of these categories created significant shareholder wealth for the companies that dominated them. Examples include SAP, Oracle and Siebel (now owned by Oracle).

As the industry evolved and matured, the market for enterprise software became increasingly challenging. Enterprise customers became dissatisfied with the terms being offered by enterprise software vendors – especially those involving high upfront fees. Because of the pressure to meet quarterly forecasts, high-pressure sales tactics came into play.[32] Ultimately, as both competition increased and the return on investment (ROI) of these large-scale enterprise software products came into question, power shifted to the customer.

[32] "Power Shift", Information Week, April 19, 2004

At the recent "Software 2006" conference, Ray Lane outlined some of the challenges with enterprise software, many of which are particularly acute for software startups looking to address the enterprise market. The following are some of the most critical challenges identified[33]:

1. Startups may find it difficult to access large, enterprise buyers
2. Enterprise customers often have long evaluation cycles and committee-based purchasing. These extended sales cycles are difficult for cash-strapped startups to manage.
3. Enterprise customers often have large customization requirements before the software product can be useful to them. Though these customizations can often be a source of revenue, startups often lack the resources to take on this effort.

I experienced this shift in customer power in my first company, Pyramid Digital Solutions. Pyramid developed and sold high-end enterprise software to leading institutions in the financial services industry. Over the years, particularly after the bursting of the Internet bubble, we experienced a number of challenges. Sales cycles became longer. Customers demanded increasing price discounts, bundled services (like customizations and implementations) and free pilot programs so that they could ensure that the software would do what they needed. All of this happened despite the fact that we had *decreased* competition (many of our competitors went out of business during that time). The company had to migrate to selling *solutions*, consisting of both products and services, instead of just "shrink-wrapped" products. Though this reduced profit margins, due to an increased dependence on services, services grew the top-line revenues of the company consistently.

Enterprise software generally has a much smaller potential pool of customers than that for consumers and small businesses. For example, SAP, a leading provider of ERP software installed its flagship R/3 software at about 16,500 sites worldwide over a period of about 5 years. Microsoft, on the other hand, sold more than 60 million copies of

[33] Price, "Enterprise Start-Up Strategy"

Windows95 in 1997 alone.[34] One outcome of the reduced pool of potential customers in the enterprise market is that a startup may be faced with the issue of "revenue concentration" whereby a small number of large customers represent a significant portion of the total revenue of the startup. This increases risk for a startup as the loss of any one of these customers can often be fatal. In my personal experience, this revenue concentration problem can be a major challenge. At Pyramid (my enterprise software company), our top clients not only had a fair amount of leverage when it came to negotiating pricing and terms – but often had significant influence over the product roadmap and the allocation of our R&D resources.[35]

Consumer and Small Business Software Markets

At the other end of the continuum from enterprise software, in terms of customer size, there is the consumer and small business market. Selling software into these markets avoids some of the challenges of the enterprise segment.

Sales cycles in these markets are usually shorter. The primary reason for this is that there are fewer individuals involved in the decision – often just one. Further, the price point being paid often doesn't warrant a deep analysis by the customer, so more purchasing decisions are completed with minimal consideration.

Further, since each customer usually represents only a fraction of the startup's revenue, the loss of any particular customer does not present a grave threat. As a result, each individual customer usually has less power and leverage than seen in the enterprise software segment..

However, the consumer and small business markets present their own sets of challenges:

1. These markets are often highly fragmented and difficult to reach efficiently.

[34] Hoch, "Secrets of Software Success"
[35] O'Neill, Notes From Discussion

2. Price points for software products sold into these markets are often significantly lower than those in the enterprise sector. This often precludes investing significantly in any individual sales opportunity.
3. There are often several alternatives in any given product category creating competitive pressures.

Horizontal vs. Vertical Markets

One of the fundamental decisions software companies need to make is whether to focus on a horizontal (broad) or vertical (narrow) market strategy. Horizontal markets have the appeal of a big market opportunity as the end product has wide applicability to a large pool of potential customers.

However, for a software startup, tackling a horizontal market in the initial stages can be risky. Horizontal strategies present a number of challenges for software startups:[36] Horizontal market strategies often require extensive resources for product development, sales and marketing – resources that many startups simply don't have and cannot afford. Large market opportunities attract a larger number of potential competitors – including large, established companies like Microsoft, Oracle, Google and Yahoo!

Case In Point

In 1999, I founded my second software company, Captivo. Captivo was targeting a broad, horizontal market with a web-based software solution for information management. This was similar in many aspects to Salesforce.com. After about two years of product development and over a million dollars of capital invested, mostly my own, Captivo still could not gain any significant traction in the market-place. Ultimately, Captivo was sold and the product was targeted at a more vertical market. The product, under it's new owner and with a narrower focus, ultimately *did* succeed.

[36] Cusumano, "The Business Of Software"

Geoffrey Moore, in his book "Crossing The Chasm" makes a strong case for why companies, especially startups should focus on niche markets. Moore posits that a key driver of initial market success for startups is establishing a strong "word of mouth" reputation among buyers. For this word of mouth to be effective, there needs to be a sufficient number of informed individuals who exchange views. In most cases, these buyers communicate within certain domains. "Chemists talk to other chemists, lawyers to other lawyers, insurance executives to other insurance executives, and so on." So, in order for a startup to be able to effectively and efficiently generate crucial word-of-mouth referrals, it must focus on a niche market where potential buyers communicate with each other and convey information.

> "...a new software venture needs to start somewhere and prove itself before tackling the world, and a narrow but attractive vertical market seems like a good place to start."
>
> Michael Cusumano
> "The Business Of Software"

Market Blending: Individual vs. Business Customers

One of the current trends in software is a blending of some of the attributes of the consumer and enterprise markets. Increasingly, the adoption of technology by business users does not stop at the office – and not all products chosen by businesses are done by the central IT department.

Chris Shipley, director of the DEMO conference which showcases new technology products commented in a recent article: "The distinction between consumer and business if fading, and fading quickly. We adopt the tools that allow us to be most productive, no matter how we describe productivity or where we use these tools."[37]

One way that might startups might leverage this trend is to create products that are selected by consumers for their individual use – and then leverage these customers for expansion into the enterprise. Examples include the RSS (Really Simple Syndication) client software market which started as a tools that individuals used to keep track of interesting news and stories in their personal lives – but then started increasingly penetrating the enterprise market where knowledge workers wanted to efficiently track information about markets, competitors and customers.

Wrap-Up

Startups must make key decisions regarding how they will position their product in an ever-more competitive marketplace and with an increasingly sophisticated set of customers. Each decision comes with a set of tradeoffs that will likely influence other aspects of the business.

[37] Shipley, "Consumer, business technologies face same challenges"

Chapter 5: On Product Development

Questions:

1. How can a software company tell if it's good at developing a successful software product?
2. Is software now more or less expensive to develop than before?
3. How are startups tackling the product development challenge today?
4. What are the advantages and disadvantages of outsourcing product development for a software startup?

For software startups, one of the key influencers of success is an ability to build a working product that solves some meaningful customer problem. If developing working software is not a core competency of the startup, it's unlikely that the startup will succeed. Further, the success rate for software development projects can be extremely low. A Standish Group study which evaluated 8,000 software projects in 1995 found that 84 percent of all software projects do not finish on time, on budget, and will all features installed. The same report found that 30 percent of all projects were cancelled before completion.[38]

Causes For Product Development Failures

Mark Crowne, a development consultant who has considerable experience working with development projects within startups and is a leading industry speaker, analyzed the root cause for why product development projects fail within startups.[39] The following are the key reasons:

1. **Developers Are Inexperienced:** A lack of cash often leads startups to hire a development team that has not shipped a product before. This leads to making mistakes that more experienced development teams have likely already made

[38] Hoch, "Secrets of Software Success"

[39] Crowne, "Why Software Startup Products Fail"

elsewhere. The solution is to ensure that the lead developers have extensive experience in all aspects of building and shipping a commercial software product.

2. **Product Isn't Really A Product:** Each customer is essentially running a modified or customized version of the product. This usually arises when the initial product development is done for a specific customer (or set of customers) with specific needs. The solution is to budget an investment to *productize* the custom solution by creating a single version of the product for all customers that can be independently tailored for customers at lower cost by not requiring changing of the core product source code.
3. **Product Has No Owner:** No one knows who has the authority to decide what does or doesn't go into the product. As such, decisions are delayed or conflicting. The solution is to designate a single individual or at most, a small but coordinated group to make product decisions.
4. **No Strategic Plan For Product Development:** The startup's business plan does not establish objectives and goals for the product and as a result the product owner makes decisions on an ad-hoc basis. The solution is to ensure that there is a clear product plan for the company that identifies objectives for the short term and medium term.
5. **Product Platform Is Unrecognized:** The importance of technologies and components which make the product are not understood, discussed or managed. Selection of these components is left entirely to the product development team. The solution is to review all major technology choices and ensure that they are aligned with the strategic plan of the company and make both business and technical sense.

Development Methods

The historical approach to thinking about software development (often known as the "waterfall" approach) was that software development was linear. Each project went through a certain set of predefined "stages" (as shown in the figure below). The waterfall model emerged 30 years ago in an effort to address the challenges of managing large-scale, custom-development projects as were evident in the U.S. military.[40]

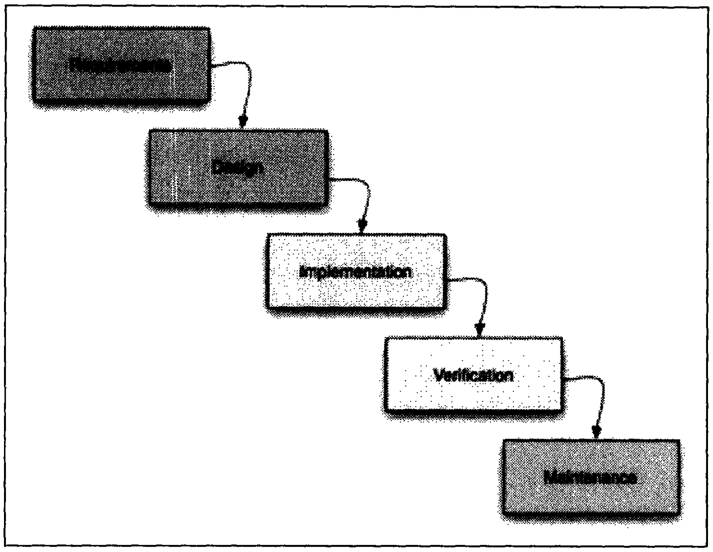


Exhibit 9.: The Waterfall Method (Source: Wikipedia.com)

The motivation for this approach was centered around the notion that each team member in a project should be doing a highly structured task and minimizing the degree of "coordination" required across various members of the team. This concept originally sprung from the implementation of the assembly line in the early 20th century and hit its peak with the time-and-motion research done in the 1950s.[41] The idea was that if each individual did a job repeatedly, they eventually became an "expert". It was also unnecessary for individuals to see the big picture. Everyone focuses on his or her own tasks and by keeping interactions to a minimum, the end result is the sum of the tasks.

However, the waterfall approach software development may be inappropriate for contemporary software development projects. The approach assumes, at some level, that it's possible to capture all requirements and complete analysis before design starts. However, in many development projects, by the time all the requirements have been

[40] MacCormack, "Product-Development Practices That Work: How Internet Companies Build Software"
[41] Cantor, "Software Leadership"

written down, they will have changed.[42] In the early days of the software industry, when applications were commonly "stand-alone" with minimal needs for integration across them, the waterfall method may have been appropriate. However, it does not often suit the problems of the complex, risky and integrated projects that need to be developed today.[43]

> "A complex system that works is invariably found to have evolved from a simple system that worked…A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over, beginning with a simple working system."
>
> J. Gall

So, if the waterfall method is not the right approach for contemporary software development projects – then what is? The answer varies, based on who you ask, but many of the currently practiced models of software development center around a set of concepts collectively labeled "agile methods". Agile methods use an *iterative* approach to software development whereby the team iterates through the various stages of the development cycle. By breaking the project up into multiple iterations, it is possible to receive early feedback from customers and factor this feedback into future iterations.

In "The Agile Manifesto"[44], a group of leading software practitioners identified what they believed were better ways of developing software. In this widely circulated work, the group stated that they have come to value:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

[42] McBreen, "Software Development: Dismantling the Waterfall"
[43] Langham, "Waterfall Development Model Considered Harmful"
[44] http://www.agilemanifesto.org

In a two-year empirical study led by Alan MacCormack, 29 completed software projects were analyzed and the characteristics most associated with the best outcomes were identified. The following chart outlines these key findings.[45]

Four Software Development Practices That Spell Success

1. An early release of the evolving product design to customers
2. Daily incorporation of new software code and rapid feedback on design changes
3. A team with broad-based experience of shipping multiple projects
4. Major investments in the design of the product architecture

In the report MacCormack states "The most remarkable finding was that getting a low-functionality version of the product into customers' hands at the earliest opportunity improves quality dramatically."

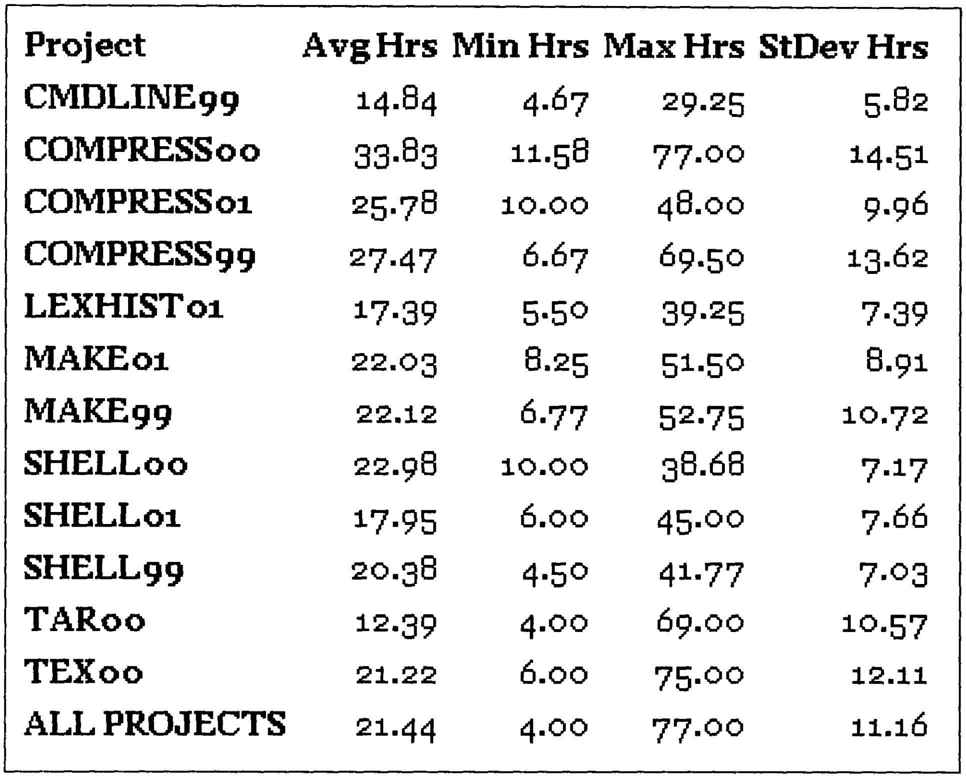
Interestingly, this is consistent with the approached used by Microsoft, the world's largest software company. Microsoft uses the "synch and stabilize" approach which divides large products into manageable chunks so that large teams can work like small teams and evolve features incrementally with direct input from customers during the development process.[46]

Hiring The Best Developers

Those with extensive experience in software development, either as managers or programmers, have learned that there is a wide gap between the productivity of an "average" programmer and that of an *exceptional* programmer. Often, the best programmers will out-perform the average programmer by 10 times.[47] One of the challenges to actually empirically determining whether this productivity difference is that large is that rarely are multiple individuals asked to do *precisely* the same work on which studies can subsequently be done. However, Joel Spolsky, in his article "Hitting The

[45] MacCormack, "Product Development Practices That Work: How Internet Companies Build Software"
[46] Cusumano, "Microsoft Secrets"
[47] Spolsky, "Hitting the High Notes"

High Notes" discusses an informal study he conducted from an intensive computer programming course at Yale. The course has a standard set of assignments given to each student and data were kept on how long each student took to complete the assignment – and to what degree the results met expectations (by running a battery of tests on the developed program). The results of this informal study are included in the figure below.



Project	Avg Hrs	Min Hrs	Max Hrs	StDev Hrs
CMDLINE99	14.84	4.67	29.25	5.82
COMPRESS00	33.83	11.58	77.00	14.51
COMPRESS01	25.78	10.00	48.00	9.96
COMPRESS99	27.47	6.67	69.50	13.62
LEXHIST01	17.39	5.50	39.25	7.39
MAKE01	22.03	8.25	51.50	8.91
MAKE99	22.12	6.77	52.75	10.72
SHELL00	22.98	10.00	38.68	7.17
SHELL01	17.95	6.00	45.00	7.66
SHELL99	20.38	4.50	41.77	7.03
TAR00	12.39	4.00	69.00	10.57
TEX00	21.22	6.00	75.00	12.11
ALL PROJECTS	21.44	4.00	77.00	11.16

Exhibit 10. Source: "Hitting The High Notes"

From the chart, we see that across all the assignments, the average student took 21.44 hours to complete the assignment whereas the *best* student took only four hours. Further, Spolsky explains that further investigation indicates that the *quality* of the results (i.e. the degree to which the programs did what they were supposed to do) were not correlated to the amount of time taken. So, not only did the best programmers complete the assignment in a fraction of the time, the quality of the result was comparable to those that took longer.

Though the data are by no means conclusive, it does provide indication that a gap between average and exceptional does likely exist.

The fact that such a large differential can exist between great programmers and average programmers creates an added incentive for software startups to find and recruit these great programmers. In fact, the overall quality of the initial development team is a large predictor of future success. Having one or two "10X" programmers on the team reduces the time it takes to deliver a product to the market-place. Further, in my experience, the great programmers do not cost orders of magnitude more than the average programmer. Or, said differently, the productivity/cost ratio for these great programmers is *much* better than for the average programmer. Even if a programmer is ten times more productive, it is likely that her salary will be only 50%-100% higher than the average programmer.

Another significant challenge for software companies is that the best and most productive developers become most proficient after significant exposure to the product's design and direction. As such, replacing a developer lost through attrition or other factors can be very expensive. This issue is exacerbated by the reality that the software industry generally faces higher attrition rates than other industries. A 1997 study that analyzed 19 different industries ranging from aerospace to utilities found that the software industry (products and services) had the highest attrition rate in the U.S. followed closely by consumer goods and banking.[48]

[48] Saratoga Institute, "Human Resource Financial Report

Case In Point

Pyramid Digital Solutions, the first company I founded, and which has been discussed earlier benefited from an *exceptionally* low attrition rate – especially amongst the development team. In the first seven years of operations, the company did not lose *any* of its key developers. Vipul Shah, who led operations and finance for the company in the early years attributes a large part of the company's success to this relative stability in the development team. He states: "Not only does a low attrition rate save on the costs of hiring new staff, the end product is more elegant, more maintainable and generates better profits over the long-term."

Outsourcing Software Development

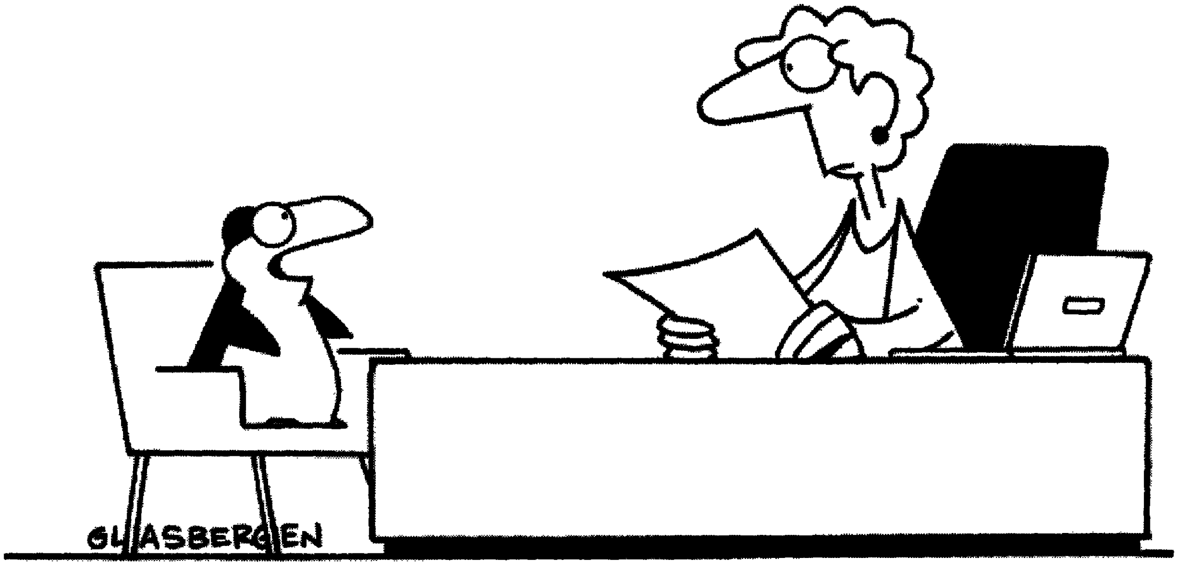
The concept of outsourcing in general, can be positioned as follows:[49]

1. Concentrate the firm's own resources on a set of "core competencies" where it can achieve definable preeminence and provide unique value for customers.
2. Strategically outsource other activities - including many traditionally considered integral to any company - for which the firm has neither a critical strategic need nor special capabilities.

Overall, outsourcing is a significant trend. 2006 is expected to be a record year for outsourcing contracts as companies continue to leverage the price and availability of a global talent pool. However, despite the upward trend in the use of outsourcing, TPI, Inc. an outsourcing advisor firm recently reported that outsourcing deals are likely to save businesses only about 15% compared to the 60% that managers typically expect.[50] Despite the overall potential benefits of outsourcing from a cost and availability of talent perspective, I would posit that for the early stages of a software startup, it often does not make sense to outsource the core development effort. Software projects continue to be

[49] Quinn, "Strategic Outsourcing"
[50] Schelmetic, "TPI Publishes Global Outsourcing Trend Report"

difficult to manage and prone to cost and time over-runs. Development projects are difficult even when members of the development team are in close proximity to each other – and the customer. By outsourcing development, coordination becomes even more difficult and the odds of succeeding are even lower. I would argue that the software products industry is in some regards like the book publishing or movie industry. The idea is to create the runaway bestseller or the big box office blockbuster. In each of these industries, it takes a unique set of ingredients that are often not completely understood to create something that will create exceptional stakeholder value. Unlike manufacturing, software development is not a mechanical process and often defies systemization (despite many attempts to do so). Though outsourcing can certainly provide the opportunity to reduce development expenses, this comes with the added risk that the resulting product has a lower likelihood of being that "blockbuster hit".



"I'm here to talk to you about outsourcing. I have 15,000 friends who are willing to work for fish and blankets!"

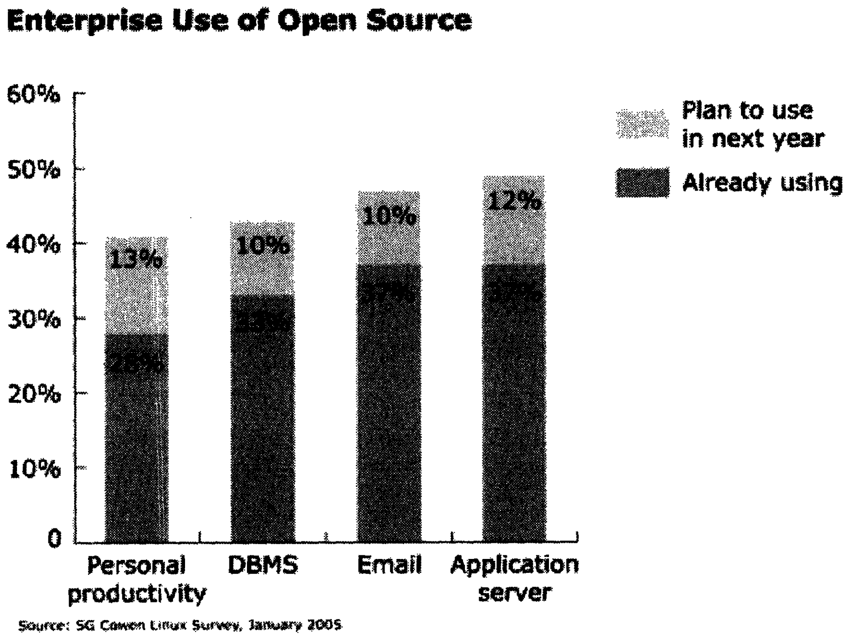
Given the earlier discussion on the "great vs. average" programmer, it is important to note that finding and retaining *great* programmers on a project that is outsourced is highly unlikely. This should not be surprising. Most outsourcing projects are contracted on a "time and materials" basis. There is insufficient incentive for the outsourcing company to recruit the best programming talent to place on client projects. In most cases, the outsourcing company is evaluated based primarily on its ability to deliver on near-term objectives at a lower cost. Further, given that the results of a project cannot truly be measured until well after the project is complete, it is naïve to expect outsourcers to take on the risk and cost of attempting to find and retain the best talent as they will likely not receive proportionate rewards for this effort.

Given this situation, outsourcing would not seem like an effective strategy for the early-stages of product development for a startup. My thoughts on this topic are shared by *all* of the software entrepreneurs I interviewed for this thesis. Not a single one of the companies interviewed outsourced all or most of their software development, despite the potential cost savings. It is also notable that these companies are not "cash rich" and hence the potential for cost savings is of high interest. When asked why they did not outsource, the response was relatively consistent and can be summarized as: "Because we're a software company, and developing software is what we're *supposed* to be good at."

Open Source Software

Open source software (OSS) refers generally to programs or projects whose source code is made available for use or modification .

The open source movement is impacting the software industry in several ways. First, software companies are increasingly using open source software in their product development. Though not conclusive, in an online survey I conducted of software entrepreneurs, 65% of the respondents indicated that they were using some open source code as part of their product development. Second, enterprise acceptance of open source within certain parts of the software stack is already significant. Startups need to be mindful of which categories already have widely adopted open source alternatives as these categories may not be attractive markets to pursue for a software products company. For example, if a startup decided to pursue the market for relational database management software, not only would it be competing with industry heavy-weights like Microsoft, Oracle and IBM – it would also be competing with popular open source alternatives like MySQL.



Enterprise Use of Open Source

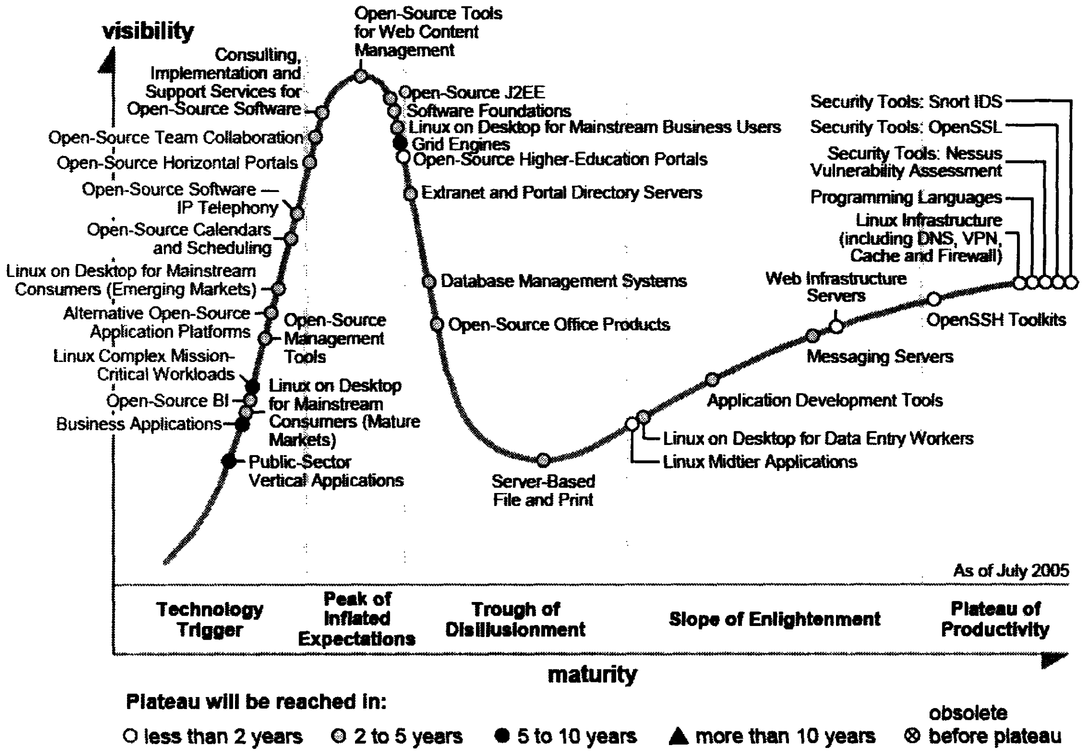
Source: SG Cowen Linux Survey, January 2005

Open source software has made a significant impact on the software industry, but with varying degrees of magnitude in the various layers of the software stack. Where it has had the most impact has been in the infrastructure layer of the stack consisting of operating systems, databases and development tools. Gartner predicts that by 2008, OSS solutions will directly compete with closed-source products in all software infrastructure markets.[51]

[51] Driver, "Open-Source Solutions Will Restructure the Software Industry"

In a recent Gartner report, various parts of the open source landscape are plotted on the Gartner "hype-cycle curve", as illustrated in the figure below

Figure 1. Hype Cycle for Open-Source Software, 2005



Acronym key:

BI	business intelligence	SSH	Secure Shell
DNS	Domain Name System	SSL	Secure Sockets Layer
IDS	intrusion detection system	VPN	virtual private network
J2EE	Java 2 Platform, Enterprise Edition		

Source: Gartner (July 2005)

Exhibit 11. Source: Gartner

From a software startup's perspective, the cost of developing a product can be significantly reduced by leveraging many of the open and free (as in beer) OSS products available. A well recognized example is the "LAMP" software stack which consists of Linux (operating system), Apache (web server), MySQL (database) and PHP (programming language). By using these products, software startups can avoid the costs of many of the key tools and technologies they need in their product development.

Case In Point

My current startup, HubSpot, is built on top of a popular portal framework that is distributed as open source. This framework provides some of the core functionality of our product offering. I estimate that the use of this open source project has saved my startup approximately $250,000 in development costs and will reduce our time-to-market by about 6-9 months. By way of comparison, my prior startup (started in 1999) had to build something very similar but was not able to find a viable open source component and as such, had to develop proprietary code.

Wrap-Up

Having talked to a number of software startups (both successful and not so successful), I have a set of questions I use to determine what the likelihood of success might be in developing a working product and surviving in the market place:

1. Does at least one of the founding team members have a strong software development background? Does her experience include having shipped a commercial software product? Without this experience on the "core" team, it is unlikely that the startup will succeed. Building a software product takes a rare combination of skills. Non-technical founders that don't have such an individual on the team already will generally have a difficult time hiring for this role as they often don't know what kind of person they need to hire.
2. Is the development team focused on getting an "early release" out? Startups that have a development plan that goes out beyond a year before the first version of the product can be put in customer's hands are less likely to succeed than those that are focused on shipping something – even if it is not perfect. The value of

getting software into customer's hands early should not be underestimated. This early feedback is critical.

3. Does the development team understand and use basic tools for efficient development including: source control, automated builds,, bug/feature tracking database and automated testing?
4. Does the development lead know how to find and recruit *exceptional* developers? Despite the global talent pool of developers, the proverbial "10X-20X" developer (i.e. the developer that is orders of magnitude more productive than the average developer) remains hard to find and even harder to recruit and retain.
5. Does the development manager have sufficient knowledge of available open source software and can she make rational decisions on which such components might benefit the company? In today's marketplace, startups that are not competent at making wise and prudent use of existing code are at a competitive disadvantage.

Chapter 6: On Acquiring Customers

Questions:

1. What creative methods are today's software startups using to reach customers?
2. What is working and what's not in terms of online marketing techniques for startups?

Acquiring customers and generating sales is one of the most critical challenges that all software startups must address. Finding the first customer is an exercise that Steve Papa, CEO and co-founder of Endeca calls "the relentless pursuit of credibility".

Hiring sales people to acquire customers is often not feasible in the early stages of a startup when capital is scarce. As such, one of the most important skill sets for software founders is the ability to close deals. In my experience, the probability of success of a startup is significantly influenced by whether or not one or more members of the founding team are able to generate sales.

For a startup to survive and thrive, there is evidence to indicate that founders need to place an increasing emphasis on sales. In a research study involving 114 technology-based firms within the Greater Boston area conducted by Professor Ed Roberts of MIT the results indicated that over the first several years after founding, founders show an increased orientation towards sales and marketing with a lessened emphasis on engineering.[52] This paper also cites a study of 12 educational software companies. Though the sample size is not sufficient to draw meaningful conclusions, the information is still useful. The mean percentile time allocation of the principals toward sales and marketing increased from 34.1% to 48.6% as the startup evolved from early-stage to late-stage firms.

[52] Roberts, "Evolving Toward Product and Market-Orientation: The Early Years Of Technology-Based Firms"

Online Software Delivery

Given that software products are *digital*, it is one of the few products than can be delivered to customers online. This ability to avoid the costs of shipping physical media does more than simply avoid the cost of "shipping and handling", it can give customers the ability to try the software immediately before committing to a purchase.

Case In Point

Jeremy Allaire, the co-founder of Allaire Corporation (now owned by Adobe), used the Internet to drive early sales of his server-based development product. Customers could download his $500 product and try it before buying. I was one of the thousands of developers that downloaded the flagship Allaire product "Cold Fusion". I ultimately chose the product for the company I was running at the time. I was evidently not alone. Even with no outside capital, delivering software trials online allowed Allaire to break-even in its first year, achieve significant growth and ultimately be acquired by Macromedia.[53]

Online Marketing

Online marketing is a relatively recent phenomenon for software startups. Startups can now leverage the large and growing base of Internet users through relatively targeted and efficient online marketing techniques. These techniques include search engine optimization (SEO), online advertising and blogs.

Given that most software startups can actually *deliver* the product online (either as a download or a hosted service), marketing through this channel can be very effective. With larger, more established companies, marketing dollars are often spent to promote the company "brand" instead of the company's technology or product. For successful

[53] Silverthorne, "How To Sell In Start-Ups"

software companies, as much as 78% of the advertising dollars are spent marketing the company name.[54] The rationale for this is that these large companies are looking to gain dominant market share in their categories. This goal is often best served by cementing the company's brand as the "leading" provider of software in the mind of the customer.

For startups, the marketing strategy is quite different – especially when it comes to online marketing. In contrast to other channels (like print or television), online marketing as evidenced in vehicles like Google's AdWords is focused less on creating "impressions" of the company's brand and more on fulfilling an immediate need for certain functionality or capabilities. For example, whereas a company like Microsoft might advertise on television to create additional visibility for its brand, a software startup might purchase Google AdWords for the search term "plumbing service management" in order to attract customers looking for a specific type of product.

A New Type Of Shareware: Freemium Marketing

Many contemporary software startups, including 37signals which was interviewed for this thesis employ a marketing technique which can be described as "The Freemium Business Model". The term was originally coined by venture capitalist Fred Wilson of Union Square Ventures in New York City. Though Mr. Wilson calls this a "business model", I believe it is better characterized as a marketing technique. Here is how Wison describes this approach:[55]

> *Give your service away for free, possibly ad supported but maybe not, acquire a lot of customers very efficiently through word of mouth, referral networks, organic search marketing, etc, then offer premium priced value added services or an enhanced version of your service to your customer base.*

[54] Hoch, "Secrets of Software Success"

[55] Wilson, "The Freemium Business Model"

Basically, this model is centered around giving away software for free in the hopes of using word of mouth marketing to amass a large user-base. This user-base of "free" customers is then targeted for upgrades to a premium product or services. This model is somewhat similar to the shareware software distribution model . With the shareware model, software developers would make their products easily available with a request for payment from those users that wanted to receive upgrades and support. With the wide reach of the Internet, this "freemium" type model of marketing has become more prevalent. Since distribution costs over the Internet are effectively zero, certain products can gain thousands of users for the free product. All of these users are then candidates to become paying customers.

Case In Point

John McAfee recognized early that computer viruses were becoming an increasing problem. So, he drove across the country to collect samples of computer viruses. After studying and analyzing them, McAfee developed software that would detect these malicious programs and delete or disable them. He distributed the program through bulletin boards (this was before the Internet became prevalent). The program proved useful and spread quickly. Corporations soon discovered that their employees were installing McAfee's software on corporate computers. These corporations, being risk averse, *demanded* to be able to pay money for the program in order to receive upgrades and support. McAfee was happy to oblige and grew the company to $40 million in annual revenue based on this early marketing model.[56]

Search Engine Optimization

Search Engine Optimization (SEO) is a set of strategies and tactics used to ensure users that are searching on the major search engines (Google, Yahoo!, MSN, etc.) for relevant

[56] Fitzsimmons, "Lone Hackers, Power Suits, and Gold Rushes"

keywords will easily find the startup's website. The ultimate goal is to rank "high" on the search results (ideally being in the top 10 list of results shown for the search).

For startups looking to reach a wide audience of Internet users, and particularly for those startups selling web applications, search engine optimization can be a highly effective tactic.

There are two major advantages to using SEO vs. online advertising (such as Google's AdWords):

1. Users often believe the results of search engines to be more credible than purchased ads. The rationale here is understandable. Companies can buy their way onto search site results by advertising (in the case of Google, these ads are generally shown on the right-hand side of the page or at the top of the page). Showing up first on the standard results page, on the other hand, is seen to be a more "objective" selection criteria.

2. Search Engine Optimization can often be much *cheaper* than purchasing online AdWords. By some relatively simple optimization techniques, companies can rank "higher" for certain key phrases. SEO is often a "fixed" monthly cost (its often sold as a service) whereas online advertising through vehicles like AdWords are done on a "pay per click" basis.

As a mini experiment (and to learn a bit more about SEO), I've researched some of the most basic SEO techniques and embedded them in the platform for my latest software startup, which intends to make SEO "accessible" for its small business customers. My new blog site, OnStartups.com runs on this platform and leverages this SEO feature. As of May 1, 2006, searching on the generic phrase "software startups" ranks the OnStartups.com site in the top position on both Yahoo! and MSN Search (two of the top three search engines).

Note: Google is believed to have a "sandbox" approach to its algorithm which disallows new sites (started less than 6 months ago) from ranking high on search results. This is to avoid the new "splog" (spam + blog) sites that are formed to attempt to exploit the search engines and drive Internet traffic to themselves.

Business Blogging

Blogs (short for "web logs") have grown significantly in popularity in recent years. It is estimated that 7 million of the approximately 120 million U.S. adults who use the Internet have created one and 27 percent of Internet readers say they read blogs.[57]

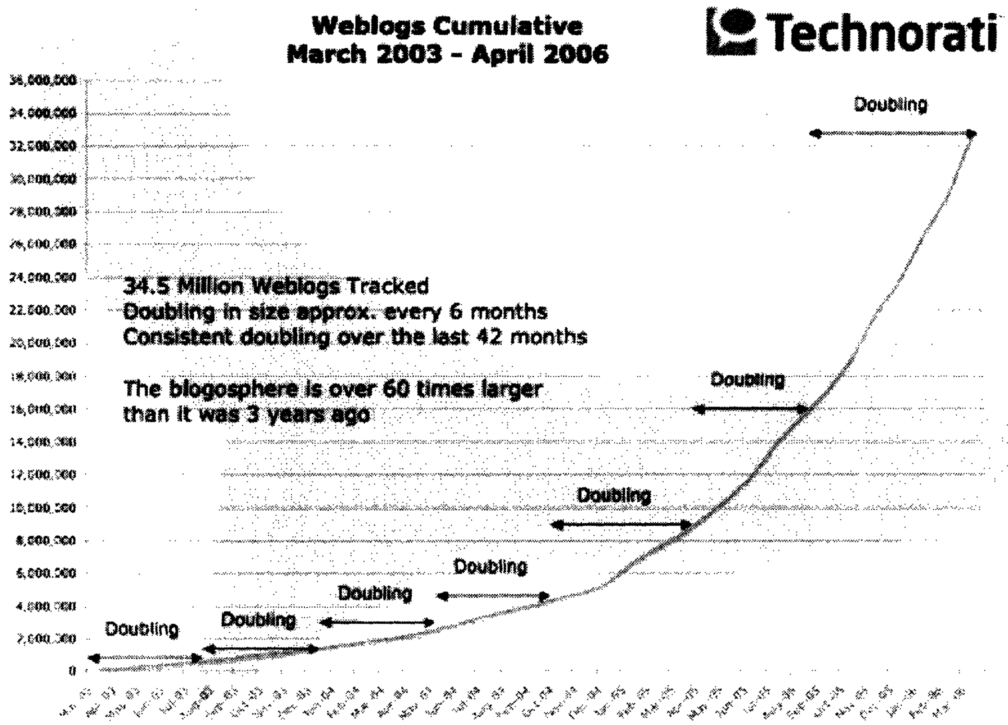


Exhibit 12. Source: Technorati

[57] PewInternet.org, "The State Of Blogging"

Startups can benefit from the creation of a blog as it provides a channel to communicate with interested parties (customers, prospective customers, employees, etc.) on both the vision of the company and information about the products the company offers. Blogs can provide a highly efficient and non-intrusive way to build a relationship with customers by sharing practical tips for using the product and gaining the most benefit from it. As a personal example, I have received over two dozen customer leads for my current startup, HubSpot, from my blog site. What is interesting is that the blog itself is not about HubSpot, but about startups generally. However, enough visitors are curious about my background, find the link to HubSpot and ultimately send an email requesting more information. I would expect that once HubSpot has an independent blog focused directly on its customers and products, the response-rate should be even higher.

Blogs are also a quick and inexpensive vehicle for a startup to promote itself. Blogs can often be started for free (or minimal cost) using any number of online blog service providers. Writing a blog with these systems requires little technical knowledge.

In an interview with Mena Trott, the co-founder of Six Apart, a developer of a leading blog software package, outlined the following potential benefits of blogging for startups[58]:

1. It helps to show that there are real people behind the startup.
2. Blog readers are often searching for interesting new software and tend to favor the "underdog". As such, they often make excellent early-adopter customers for software companies.
3. Happy customers that are also bloggers will often write about the startup and its products, thereby increasing the startup's overall visibility in a crowded market.

Ms. Trott further provides some tips for startups when creating a blog:

[58] Walsh, "Micro-ISV: From Vision To Reality"

1. Ensure that those writing the blog recognize that they are speaking on behalf of the company.
2. Startup bloggers should have a "personality" that engages customers and makes the writing interesting. Simply taking dry marketing materials and putting them on a blog is ineffective.
3. There should be a relatively consistent frequency of content updates so that customers feel that the startup is making an investment in the blog.
4. Writers should have the liberty to write about topics that will be of interest to customers – but are not necessarily about the startup or its products. This makes the blog a more useful destination, and thereby increases traffic.

Wrap-Up

Contemporary software startups now have the opportunity to reach a wider audience more efficiently than ever before. Tools like blogging and Internet advertising can be major drivers of sales leads.

However, not all startups can benefit from these kinds of approaches to distribution equally. Startups selling complex software to large enterprises may still require the classic approaches to selling – including sales people. These markets continue to be challenging and it is less likely that these customers will discover a startup's products through an online search. Many of these customers still find new products to meet their needs through the traditional channels: tradeshows and word of mouth. These customers will still likely require a level of effort to be "sold".

Chapter 7: On Software Business Models

Questions:

1. How have the business models and revenue models changed for software startups?
2. What are the implications for startups using "subscription" oriented revenues vs. the traditional "license" revenue model?
3. Is advertising a viable revenue model for software startups as an alternative to charging customers directly?

At some point in the operations of a startup, customers have to be charged for the startup's software product and the software has to be delivered. Without a business model that defines how this will happen, it is unlikely that the startup will ever become a sustainable business.

The traditional methods for delivering the software have been to use physical media (originally diskettes, then CD/DVD) or by some type of electronic download. In either case, once customers receive the software, they install it on computers within their environment. Increasingly, software companies are not *delivering* their software to customers at all – but making the benefits of the software available to customers via a hosted service. In this chapter, we'll like at the various licensing models and ways that startups can provide software to their customers.

Traditional Software Licensing

Historically, one of the most common models for software companies has been to license software for use by their customers. In this model, software is delivered to the customer (either on physical media like CD or online). The customer then installs the software on computers within their environment. Customers pay for use of the software in a variety of ways (fixed fee, per user, per computer, etc.).

"The sales cycle leverage is on the side of the enterprise. The enterprise buys less, takes longer and works vendors harder on price then ever before. Most of these enterprises have a stated goal to reduce the number of vendors they buy software from. Figuring out a sustainable, profitable, predictable selling model is the biggest challenge."

John M. Jack,
Fortify Software

The software licensing model has a number of advantages for software companies[59]:

1. **Cash Upfront**: Customers generally pay a licensing fee for use of the software. These fees are often significant (especially when licensing enterprise software). Startups can often benefit significantly from this upfront payment as cash is often a limited resource.
2. **High Margins**: A majority of the costs for software are "fixed" and embedded in the cost of development. Actually copying the software (or making it available for download) and getting it into the hands of customers is insignificant. As such, profit margins for licensed software are generally very high.

In addition to the advantages, the traditional licensing model also has some disadvantages, which have pushed the software industry to explore other alternatives. These disadvantages include:

1. **Support Costs**: It can often be very expensive for software companies to provide the necessary support to customers that are running software within their facilities. One of the primary reasons for this increased cost is the need to support multiple

[59] LeClaire, "The Right Price"

versions of the software in the field as it is usually infeasible to mandate that all customers upgrade immediately to new versions.

2. **Low Customer Loyalty**: Once a software license deal is made, it requires investment on the part of the vendor to maintain loyalty and ensure a steady stream of cash-flows. It is very difficult for the vendor to determine which customers are actually using the software and to what degree.
3. **Revenue Recognition**: Given that license fees are paid upfront, there is the question of when the company should actually recognize the revenue from an accounting perspective. Also, unless some fees are being charged for providing updates and support, it is possible that the company has already recognized all of the revenue from a software license sale, but continues to incur costs for providing support or upgrades.

Software As A Service

Software As A Service (SaaS) involves making software available to customers via a hosted service. Instead of software companies delivering software for installation on the customer premises, SaaS providers provide the infrastructure for running their software – and customers simply access the software using standard PCs and an Internet connection. Generally, this means the customer does not have to invest in any additional hardware, all they need is a computer with Internet access and a browser). SaaS addresses many of the historical challenges of the traditional licensed model approach such as high upfront fees and long implementation times. In a recent IDC study, 73% of the software companies surveyed "strongly believe" that a majority of their revenue will flow from SaaS offerings by 2010.[60]

Similar to the memo Bill Gates from Microsoft sent to his team in 1995 identifying the "Internet Tidal Wave", a similar memo was issued in 2005 – a decade later, signaling the coming "services wave" of applications available instantly over the Internet.[61] Indeed,

[60] Hadley, "How To Get Started With SaaS"

[61] Knorr, "Software As A Service: The Next Big Thing"

many software companies are looking at leveraging the benefits of the SaaS model to bring their software to market. One striking example of the potential of SaaS is salesforce.com, which began in 1999. Despite entering the customer relationship management (CRM) software market which was dominated by large, established companies like Siebel and PeopleSoft, Salesforce.com experienced significant growth since its launch and now claims over 350,000 users on its hosted software service.

The trend towards SaaS is partly driven by the benefits it provides to customers. Customers might prefer SaaS for one or more of the following reasons:[62]

1. Provides them access to software which might otherwise be too costly or complex to implement or support.
2. Provides near immediate "Time To Value" instead of waiting for implementation.
3. Often avoids the capital expenditures and internal support costs necessary.
4. Provides convenient access to "outside" users (remote workers, customers, partners, etc.) since it sits outside the corporate firewall.

Startups have responded to the customer interest in SaaS by quickly leveraging this new model of software delivery. By reducing the risk for the customer, startups can use SaaS as a way to address the credibility gap that often stands in the way of closing deals. Customers have an opportunity to try the software (often at no cost) with minimal risk. VCs have also demonstrated a keen interest in funding startups that take advantage of SaaS. Jason Green, a general partner at Emergence Capital, a venture firm that invests solely in SaaS companies is ready to invest in SaaS startups. "The money is there and the interest is there. We feel that there is such a fundamental shift in the opportunity that we decided to devote 100% of our funds to it", stated Mr. Green at a recent VC panel.

SaaS Challenges

Despite some of the advantages of offering SaaS from a startup's perspective, there are also some challenges. Unlike licensing software for installation on customer premises, SaaS requires startups to establish an infrastructure (servers, connectivity, backup) in

[62] Wohl, "Tracking Software As A Service"

order to make their software acceptable for customer use. This infrastructure can often come at considerable expense if it is to meet the requirements of sophisticated customers. Customers have demonstrated an increasing comfort with hosting their software (and the associated data) externally as evidenced by the success of companies like Salesforce.com. However, even a successful company like Salesforce.com, experienced a large and widely publicized outage in December, 2005 which caused its customers to lose critical access to its systems for several hours.[63] In order to ensure greater reliability in its service, Salesforce.com is scheduled to invest over $50 million in its infrastructure and data center. This type of event should cause startups to take notice as offering critical software using SaaS requires a degree of operational excellence. It's no longer sufficient for companies to simply *build* an excellent software product that meets customers needs, they have to ensure it is available too.

Michael Skok, of Northbridge Venture Partners (NBVP) says it takes 70% to 100% more capital to fund a SaaS company to break-even vs. a traditional perpetual license company. Further, it takes 2-3 times longer to reach break-even than the traditional company. NBVP has funded 8 software companies using the SaaS model, and shares the following observations regarding these investments[64]:

1. SaaS companies need an average of $35 million in VC capital vs. $20 million for a similar perpetual license company.
2. It takes 6-7 years to get to break-even.
3. Public equity markets pay a 10% to 20% premium for predictable revenue streams.
4. SaaS requires an architecture that supports end user customization.
5. Steady state business models require 15-18% for engineering and 30-35% for sales and marketing.

Further, as noted in #4 above, since SaaS software is not running at customer facilities there needs to be an architecture that allows customers to customize the product and integrate it. This causes the engineering costs of SaaS products for startups (and

[63] Cowley, "Salesfore.com cuts users off"

[64] Dodge, "SaaS: New Software Model, New Problems"

established companies) to be higher than products designed for installation at customer facilities. In the traditional model, software companies often sell consulting services to customize/integrate the product within the customer's environment – with SaaS, this is less common.

Finally, SaaS can also present challenges for managing the sales effort of the startup. Examples include the need to adjust incentives and commissions since SaaS deals often don't have the large upfront license fees and associated commissions of the traditional licensing deals. The following are some of the ways to deal with challenges of SaaS from a sales perspective:[65]

1. **Equalize Compensation:** Replace the large upfront commissions with smoother commissions that map to the subscription-based revenues of the new model.
2. **Eliminate End Of Year and End Of Quarter Discounting:** These types of tactics to meet quotas are no longer as relevant given the nature of SaaS subscription pricing.
3. **Develop A Farming, Not Hunting Sales Force:** SaaS sales require a longer term commitment to the customer from the sales force. As such, it is important to hire sales people that have the patience to work with customers over the long-term in order to reap the benefit of their commissions.

Product Pricing

There are a number of different ways that software startups can price their products. The most common pricing model is a per-seat or per-user type model, where the customer pays a fee for each user that will utilize the software.

In the current generation of web startups, a common tactic is to have at least two "tiers" of pricing. The first tier, often termed the "light" version, is offered at a deep discount – or free, in order to build an initial customer base. There are also one or more "premium"

[65] Graham, "How Moving To SaaS Affects Software Sales And Marketing People"

tiers where customers pay a higher price for access to more features and capabilities. The motivation for this particular model is that a large customer-base can be established using the first tier pricing and then the company tries to "upgrade" as many of these customers to premium pricing as possible.

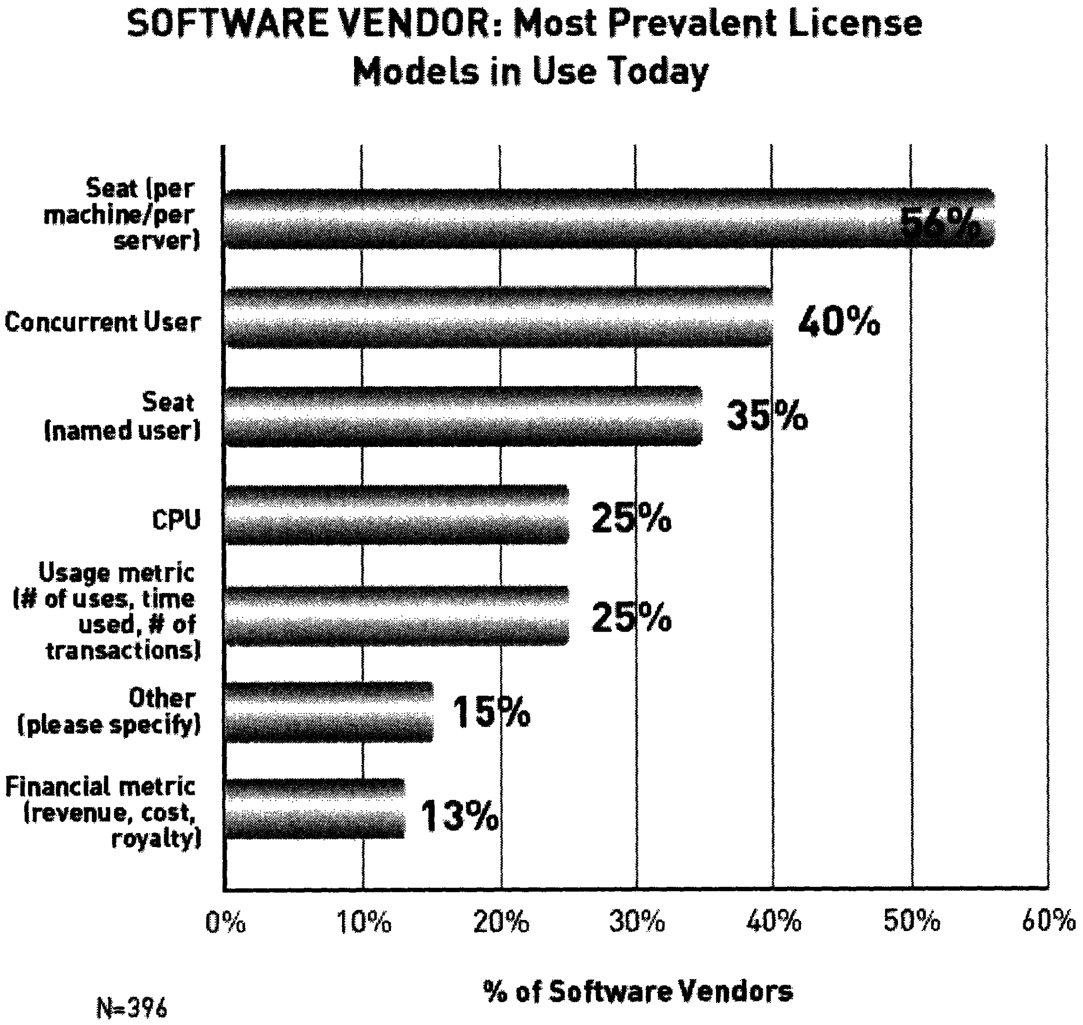


Exhibit 13. Source: Macrovision, "Key Trends In Software Pricing and Licensing"

Subscription Pricing

In addition to the emergence of SaaS is a related shift towards subscription-based pricing models from the traditional model which usually involved a perpetual license. There is increasing evidence that customers are demanding the flexibility of a subscription-based pricing model and thereby shifting power to the customer. The research firm IDC predicts that there will be dramatic shifts in software business models as a growing number of customers shop for more flexible licensing options. Vendors that can't accommodate these models will be at a competitive disadvantage.[66] In 2004, IDC estimated that already over 25% of all software sales were tied to subscription licenses.

This trends towards a subscription-based model is driven by a number of factors:

1. Customers are dissatisfied with the traditional deals offered by vendors. The classic "perpetual license" often involves large up-front costs.
2. Often, a vendor's need for these large upfront fees causes high-pressure sales tactics
3. With a perpetual license model, vendors often "lose interest" after the deal is closed as their primary incentive is to collect the large upfront fees.
4. There is often a large volume of "unused" software for which customers have already paid, but have not recouped their investment.

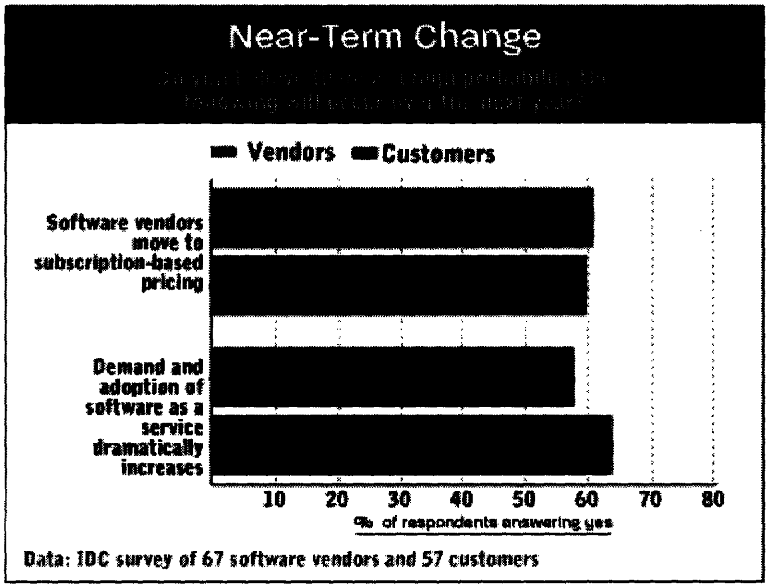


Exhibit 14. Source: IDC

[66] Foley, "Power Shift"

The shift to subscription pricing addresses these customer issues. Subscription models generally do not involve large upfront fees – thereby reducing the software company's temptation to employ high-pressure sales tactics. Also, since payments are made in a smooth, consistent manner, the software provider generally must maintain continual interest in the customer relationship and provide competitive service to maintain the stream of revenue. From a customer's perspective, the chances of "unused but paid for" software are also reduced, as agreements involving software that is not delivering value can be terminated.

For a software startup, the shift to subscription-based pricing creates new challenges. Whereas establish software companies don't have to rely on the cash generated from large-upfront fees, these can be critical for a software startup. Without the upfront cash that is typical of the class perpetual license deals, startups have to find a way to manage their cash flows. Often, this means increased financing to fund the company and compensate for the lack of large cash infusion from customer sales.

Advertising Revenue

Online advertising provides another potential revenue for software startups that are providing web-based applications. This is exemplified by the current generation of "Web 2.0" companies, discussed later in this chapter. The concept here is that the startup focuses on attracting a large user-base by providing free or subsidized access to their software application over the Internet. The traffic generated by this user activity can then potentially be monetized by displaying advertising to those users. In most cases, this advertising is provided through one of the larger advertising service providers, the most popular being Google with its AdSense™ program.

The following are some of the advantages of advertising revenue as a business model for software startups:

1. Startups can partner with any of the online advertising engines and within days can begin generating revenues from their site traffic.

2. There is usually no custom software development required as the primary technology required is provided by the online advertising platform. This allows a startup to continue to focus on building its product and user-base instead of writing advertisement management software.
3. By using an online advertising engine, like Google, the startup does not have to concern itself with collecting payment from many customers. Instead, all payment comes through a limited set of "partners".

Despite the advantages above, which have caused a multitude of web startups to pursue the advertising revenue path, there are some significant challenges with using advertising as the primary revenue driver for a software company.

1. Advertising revenues are generally a function of website traffic. However, it is difficult to predict site traffic over extended periods of time. This can create significant variations in revenue for the startup.
2. Advertising that is based on a "per click" model (the primary model used by Google and others), depends on creating "context-sensitive" ads to garner high click-through rates. However, in many web-based applications, the data that would be necessary to generate relevant ads is not publicly accessible. For example, if salesforce.com were trying to generate context-sensitive ads, it would somehow need to expose sensitive customer data to Google to create the right ads. This can be problematic.
3. To achieve significant revenues through advertising, an *immense* amount of traffic is necessary – well into the millions of hits per day. This type of traffic is difficult to achieve for a startup unless it can create a "sticky" web application that users are willing to adopt.
4. The number of advertising "suppliers" that act as an intermediary between advertisers and startups is very limited, with Google currently having the largest presence. Lack of meaningful competition puts power into the hands of these established players.

5. There is the potential for significant abuse in the online advertising industry which may put revenue models at risk if advertisers feel that they are paying more for these ads than the value that they are deriving.

Web 2.0 Business Models

As part of "15.358: The Software Business" class at MIT, I was involved in a team that studied "Web 2.0" businesses and wrote a paper on the topic. Our objective was to attempt to determine what being a Web 2.0 company meant and what kind of business strategy drove these companies. This section is a summarization of some of our findings and thoughts from that paper.[67]

"Web 2.0" is a relatively new term that really represents a combination of new Internet technologies that deliver a richer user experience over the web with a combination of business models and theories. A common attribute of Web 2.0 companies is that they attempt to leverage the "social" nature of the web to achieve network effects and drive rapid user adoption.

Many of these Web 2.0 startups deal with the same challenge as that of their predecessors in the 1990s: lack of defined revenue models. At a recent Web 2.0 networking event in Cambridge, Massachusetts three startups presented their offerings and not a single one made any mention of how they were planning to generate revenue.

Another challenge related to revenue generation for Web 2.0 startups is that they often derive their value from integrating existing data and services. Examples include sites such as chicagocrime.org which combines data from the Chicago Police department with Google Maps. These kinds of product offerings are often called "mashups" as they combine information and capabilities from multiple existing providers. Though many of these applications are very useful to users, it is unclear how revenue will be divided between the various players that participate in the total solution. This may be why

[67] Allen, Davidi, Shah, Shapira, "Web 2.0: Myth, Reality or the Future?"

Google, Yahoo! and others that provide API access to their services put a constraint on the volume of queries that will be accepted from a single organization and further stipulate that the data cannot be used for commercial purposes.

A number of critics are comparing the Web 2.0 phenomenon with the Internet bubble of the 1990s. Peter Rip, Managing Director of Leapfrog Ventures summarizes this phenomenon nicely. He presents what he calls a "new Web 2.0 equation:"[68]

> Lots of talented developers wanting to start companies
> + Cheap sources of computing resources and availability of personal time
> + Relaxed financial criteria (minimal business model screening)
> + Shared knowledge of the *big tech trends* posing as technical innovation
> + Relative difficulty of knowing what everyone else is doing
> = An abundance of startups pursuing the same general markets

The net result is that this large number of companies all competing with each other in the same category raises the bar for capital requirements, believing that those that can "outspend the competition" will break out. This is causing VC investments in certain Web 2.0 categories already. Lastly, there is the ultimate shakeout when it is finally discovered that only a small fraction of the companies—both funded and non-funded—will either reach sustainable status or be acquired along the way.

[68] Rip, "The Web 2.0 Entrepreneur Bubble"

Web 3.0 (Web 2.0 For The Enterprise)

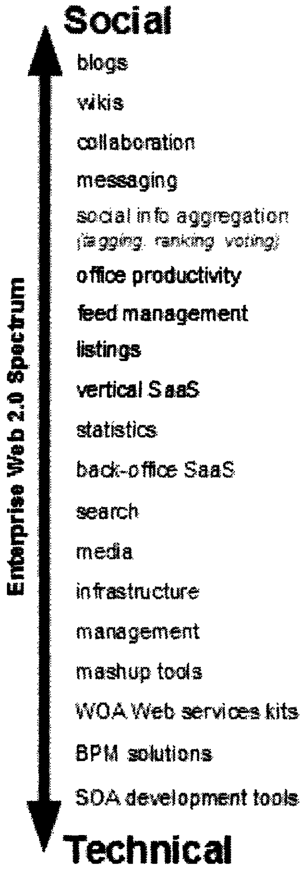


Exhibit 15. Hinchcliffe, "Running A Business On Web-Based Software"

The discussion on Web 2.0 businesses earlier was based mostly on the consumer market (as most of today's web 2.0 companies are offering a consumer product). However, some of the elements that make Web 2.0 companies attractive to consumers also have value for the enterprise. For example, creating a rich user experience within a standard Internet browser can improve the usability of products – and thereby enhance productivity. In the diagram above, we see a number of concepts and features plotted on a continuum from social aspects of software (which allow people to work better together) to the technical aspects of software which can provide increased utility to the enterprise. Some early industry luminaries have begun labeling this new phenomenon of Web 2.0 in the enterprise as "Web 3.0". Putting ambiguous labels aside, there does seem to be a distinct opportunity to take some of the lessons from the consumer "Web 2.0" world

Wrap-Up Comments

Software startups today have a variety of possible business models and revenue strategies they can use. In addition to the traditional licensing models prevalent throughout the 1980s and 1990s, the introduction of SaaS and success stories such as salesforce.com provide potentially replicable models.

Marketing driven models as are manifest in the "Web 2.0" companies also provide a new, but not completely proven mechanism to build new businesses with minimal capital . Several acquisitions of these Web 2.0 companies such as Flickr, upcoming.org and MySpace have demonstrated that though these companies lack revenue models, they may represent substantial value to large acquirers. For example, MySpace was acquired by News Corp. for $580 million However, veteran venture capitalist Bill Burnham faults what he calls the "Triple A" approach to startups. Burnham states: "You can't take AJAX, AdSense and Arrogance and call it a company. It's so easy to create a company, that you have many companies in the same space."[69]

[69] Sen, "Silicon Valley Faces Web Startup Glut"

Chapter 8: On Startups and Their Strategies

So far, we have considered several aspects of contemporary software startups. To get a better sense of what contemporary software entrepreneurs are *doing* and what lessons might be learned from their experience, I interviewed several of them. My intent was to see how what I have learned from reading, research and personal experience coincides with the experience and situations of other entrepreneurs and industry experts.

This chapter includes some startup case studies, a few interviews and finally a discussion about the OnStartups.com website, which was started in conjunction with this thesis. The chapter concludes with the results of an online survey I conducted to learn more about some of the practices of today's software entrepreneurs.

37signals – Simplifying to Succeed

37signals is a small team that creates simple, focused software. 37signals products help small businesses collaborate and get organized. More than 500,000 users use 37signal's web-apps to get things done.

Within the "Web 2.0" startup circles, 37Signals is somewhat of a legend. The company has received many accolades and has a fanatical following with its "less is more" approach to software development. [70] 37signals started as a manifesto in 1999. The founders wanted to launch a web design firm that was focused on clean, fast, usable designs, and the manifesto was a series of statements covering how they felt about web design. In 1999, while everyone else was elbowing for the loudest, brightest, most colorful sites, 37 signals went in the opposite direction."

Though they started as a web design shop, the focus shifted to BaseCamp and Ta-Da List and the company transformed into a product development company. The founders spent

[70] Hedlund, "The Builders Of BaseCamp"

about 80 percent of their time on products and 10 percent on client work. The remaining 10 percent was reserved for whatever needed extra time.

Basecamp was built, because the founders needed the tool. "I'm a big believer in investing in what you know and what you need. We invested our time, energy, and focus into building a product that we knew we needed to run our own business. When you build what you know, and when you use what you build, you've got a head start on delivering a breakout product", stated Jason Fried, founder of 37signals.

Since Basecamp was built initially only for internal use, not much thought was put into making it a product until it was demonstrated to others outside the company. The founders recognized that thousands of other companies had similar needs. Just like 37signals, these companies consisted of three or four employees who never really thought much about project management but knew project management was something they really should be taking seriously. 37signals wanted to build a project management application for companies without dedicated project managers. So they built Basecamp into a simple, hosted, subscription-funded, web-based application. Basecamp launched on February 1, 2004, after about four months of design and development.

37Signals and Ruby On Rails

Ruby On Rails (RoR), a web development framework that has gotten significant visibility within certain circles of the web development community was actually born within 37signals. RoR was released as an open source framework to the development community.

When Basecamp was built, the company didn't know they were building Rails at the same time. Basecamp came first; Rails was born from Basecamp. As Fried notes, "Basecamp was the divine chicken, Rails was the egg."

When asked why 37signals chose Ruby, a relatively obscure language, instead of more popular development languages like Java or PHP, Fried replied: "The way I look at it is this: I want developers to be comfortable with their development environment." Given that Fried was primarily a business person, and not a developer he elected not to push the team to use a more mainstream language such as PHP or Java. Instead, he deferred to David Heinemeier Hansson (lead software developer at 37signals). "It's all a matter of trust," continues Fried. "If you don't trust your developer to choose the right environment, then how can you trust him to build the best application?" Further, it was considered risky to impact the morale of the development team by forcing a technology or language on them. The rationale is that the best work comes from people who enjoy doing their work. According to Fried, he would rather take a happy average programmer over a disgruntled, frustrated master programmer any day.

Measuring Growth and Success

Many software startups like 37signals that distribute their products online have a variety of metrics that could be used to measure how the business is doing. These could include the number of trial users, conversion rate, qualified leads per channel, click-through rates for online advertising campaigns, etc. 37signals does not spend a lot of time "watching the numbers", but instead focuses on some key indicators such as the number of new signups per day and the number of users that are "upgrading" from the free version to one of the paid plans. Unless there is some trend whereby there is a reduced number of users or upgrades for a few consecutive days, the team does not generally dig deeper.

One observation from the 37signals experience is that it is much easier to sell software to businesses than it is to consumers. This is not surprising given that there is still the expectation in the minds of many consumers that Internet software should be free (i.e. subsidized by something else).

Acquiring Customers Efficiently

One of the most difficult aspects for many startups is acquiring customers efficiently. Given the level of competition in the market in most software categories and the limited resources available to bootstrapped startups like 37signals, this can be a particularly large challenge.

Interestingly, 37signals has succeeded tremendously in the face of this challenge. Though the company is private held and does not disclose financial information, their website indicates that over 500,000 users are registered for their web-based applications. This achievement is particularly surprising given that 37signals spends no money on traditional marketing, advertising or promotional plans. Their primary vehicle for finding paying customers is to provide free trials for premium products and *free forever* versions of some of their simpler products. The goal is to then attempt to "upgrade" these free users and convert them into paying customers. Given that the product is offered on a hosted basis, the cost of providing trials or free editions is relatively low.

Another strategy that has been successful for 37signals in getting wide reach for their offering is to leverage their blog and shares lots of information about the company and its founders with their customers and the media – more so than most other companies. Jason quotes Kath Sierra,a well known technical author: "You can either out-spend or out-teach [your competition] and we've decided to out-teach." 37signals teaches its customers and others in the online community via its blog, workshops and its newly self-published book "Getting Real". The book, which has also become a source of revenue for the company, explains the 37signals approach to building products and creating a great company – while staying small and nimble

Last, as noted earlier, 37signals makes available its increasingly popular "Ruby On Rails" framework as open source software. Ruby On Rails (RoR) is a web development platform based on the Ruby language that allows unparalleled productivity when building database-driven web applications. By sharing RoR with the development community, 37signals further raises visibility for the company and its products.

Avoiding The Software "Cold War"

One of the primary challenges that 37signals deals with as a startup is the number of competitors offering similar products. The company's approach to addressing this challenge and differentiating itself amongst the existing alternatives is to build a *simpler* solution. Jason comments, “We’re not players in the software cold-war – the constant one-upmanship of features over features over features. We stay away from that and just give people simple products. Products that do less. That’s our angle – our products do less because less is often all you need.”

37signal’s main competition is email, IM, post it notes, and phone calls. The way people are currently used to doing things. That's why the company strives to keep its products as simple as possible – given that it’s competing with the simplest of tools.

Controlling Support Costs

As a hosted software company, 37signals must maintain a stable infrastructure and support its growing pool of customers (both paying and non-paying). In order to keep support costs low, all technical support is handled via email. The company gets about 60-70 emails a day, which Fried answers personally. This effort takes about 90 minutes a day – and it’s the only real support costs that the company has. When asked why email was the only method by which the company provides technical support, Fried replied: “If we took support requests over the phone, I couldn’t do it myself. So, we’d have to hire one or two more people. We keep that cost out of the equation by sticking to email”.

Providing technical support to thousands of customers can kill a startup if it is not careful. The key is to build simple products that require less support effort. Less software requires less support. The company also invests considerably in making the product as easy to use as possible and providing online assistance to avoid potential confusion. .

Key Lessons

On the topic of startup mistakes, Jason commented: “We've made plenty of mistakes, but I think those mistakes are essential. It's important to make them so you feel them. But at the end of the day, we try to keep our decisions small so any mistake can easily be rectified. Our cost of change is low so we're not afraid of getting it wrong.”

Some key lessons and advice from Jason Fried: Don't borrow money, don't go into debt, don't hire people until you have absolutely no choice, hire the best people no matter where they are, don't waste time in meetings, excessive planning, and writing documents -- just build the thing and learn along the way."

Element55: Planning For Change

Element55 was founded by Ray Deck, who I interviewed personally for this case. Element55 makes software to automatically capture the billable time of attorneys.

Ray originally started the company, which was then known as eGlean, in 2000. The original business model was different, and he went through three more business models before arriving at the current one. Element55 entered the time-tracking business in 2002 and had its first customer in the beginning of 2003. Ray focused on the products business full-time beginning in early 2005. The company was funded mostly by Ray with some additional funds generated through "friends and family". The company has subsequently been grown through customer revenue.

Product Idea

Ray arrived upon the ultimate product idea for Element55 through an unrelated services project with a law firm. The potential customer came to know of an internally developed tool that Element55 was using to track its own time on the project. Ultimately, the customer was more interested in using this tool than the original project, and so the idea to "productize" the internally developed tool was born. Ray decided to focus on the legal market because the legal services industry used billable hours as a central part of its business model.

When devising the product, the company decided to focus on the upper end of the market (law firms with over 50 attorneys). The rationale for this was that they needed to ensure that each sale would generate enough revenue to support the provisioning of a network appliance – the vehicle by which the application was installed on customer premises. A further segmenting was done to focus initial sales on those that would likely serve as strong references to future clients. Mr. Deck calls this the "Skaaden Arps Rule". The rule basically posits that every firm wants to be like Skaaden Arps (a large, prominent law firm that is a leader in the industry). A large national law firm will have a stronger influence on potential mid-tier clients than other mid-size regional firms in the same peer group.

Initial sales efforts were focused on the Boston area, where the company was based, so as to keep sales costs to a minimum. However, the company only had modest sales success with this approach. The company later invested in establishing an alliance with the Legal Technology Association, which proved to be a more meaningful source of new sales.

Pricing

On the issue of pricing, the company used the following approaches:

1. Tie pricing to the value proposition: per timekeeper, on a recurring basis.
2. Offer a discount based on scale (so that after licensing the first 50 timekeepers, the next 50 timekeepers is less expensive).
3. Offer a significant discount to customers that wanted to purchase an "enterprise-wide" license which simplifies deployment for the customer and increases product adoption.

Promotion and Distribution

Element55 uses a variety of methods to reach customers and close sales. Ray offered the following comments on what has worked (and what hasn't) for his company. The following methods are provided in *descending* order of effectiveness.

1. Authoring and distributing white papers of relevance to his product offering
2. Customer referrals whereby existing customers contact prospective customers directly (i.e. "word of mouth")

3. Affiliations with the International Legal Technology Association (ILTA) and Blackberry (a wireless product of Research In Motion that is used by many lawyers)
4. TechnoLawyer.com: Sending out sponsored, targeted emails to their user base.
5. AdWords and Other Online Marketing. These have proven to generate a very limited number of qualified leads.
6. Direct-Mail: Targeting the IT and financial staff at law firms.
7. Cold-calling prospective customers directly.

Growth Plans

Ray plans to continue to grow within his current target market of large law firms. Staffing will be done primarily via customer revenue. Ray states that this is viable given the sales cycle is 6-9 months and he'll usually be able to hire staff as needed. No external financing is currently planned, though if there are opportunities to more aggressively expand sales/marketing and there is evidence capital can be used efficiently, he will consider it.

2ndSite: The Myth Of The Marketing "Silver Bullet"

This case is based on an interview with Michael McDerment, the founder and CEO of 2ndSite.

Company Overview

2ndSite is an online, web-based invoicing service that helps small businesses save time and stay organized. The product provides time tracking, support ticketing and document sharing functionality. The application is built on top of an open source platform – but is itself a closed end proprietary product.

In 2003 Michael was running a web design and online marketing consulting firm named Anicon and was working on several client projects. He was constantly struggling with managing the company's receivables and decided to build a simple online web application that his clients could use. One of Michael's colleagues, Dr. Joe Sawada was teaching part time at the University of Toronto and assisted with enhancing the application to support invoicing. Ultimately, the product was launched to the public in May, 2004. With the hiring of an engineer (Levi Cooperman) and a designer (Jeff Sarmiento) the company further expanded the application and began marketing it on a very modest budget.

The company currently has three full-time and three part-time employees, with revenues growing consistently. There are now over 50,000 users on the service.

Financing

Financing for the company to date has been based on a combination of angel investors and debt financing. In the very early stages of the business, the founders began discussions with VCs and Angel investors. Although this required a significant investment of time, and resulted in only a minimal amount of capital being raised, it was an immensely beneficial process. The founders learned a lot about how to grow and manage the business. During the capital raising process, the founders made a concerted effort to continue to grow the business with their own personal funds.

Target Market

The company had a number of "revelations" with the service's target market. In the beginning, the target was web designers and search engine marketing professionals. There were a number of reasons for this: First, all of Michael's prior experience was with web design and marketing consulting and as such, he had specific domain knowledge. He also believed that web designers would be good "early adopters" for online applications. These web designers also provided access to a lot of other small businesses that could benefit from using 2ndSite's service. So the hope was that the word would spread organically via customer referrals.

After further meetings with investors and additional research, the company decided to focus on two market niches: Web designers and PC Technicians. More recently, with the growth and visibility of "Web 2.0" and the discovery of effective online channels to market the product, the company has returned full circle to targeting web designers.

The Myth Of The "Silver Bullet"

A lot of the "surprises" the founders experienced occurred over an extended period of time. One example of a key insight that took some time to develop was the realization that there was no silver bullet for marketing the 2ndSite product. The team spent a lot of energy and time trying to find the "perfect" marketing idea that would bring millions of people to the company's website. This was then expected to catapult sales revenues. However, after a lot of iterations, the team realized that this "silver bullet", as they liked to call it, perhaps just didn't exist for the market they were in. The company is now more realistic and realizes that it can grow more effectively by staying focused and growing organically through referrals than by searching for the proverbial "silver bullet".

Pricing

Determining an effective and appropriate pricing model has been a challenge. The company has gone through several iterations of pricing. Initially, pricing was based on a complicated pay per usage pricing chart that may have confused the target market. Based on feedback from their advisors, the team ultimately settled on three "buckets" of pricing. They then went on to test the price elasticity of the product. The advisors also advised that 2ndSite had fallen into the classic trap of under-pricing the offering. Adjustments were made to more accurately reflect the service's value to the customer. Interestingly, with each increase in the price, the volume of customers actually *increased.*

The company now has what it believes to be a reasonable price for the service. However, Michael stated that it was likely that pricing would continue to be refined as they learned more about the market. In order to maintain customer loyalty, existing customers are generally "grandfathered" in to their original price.

Product Development

The core product and architecture is the responsibility of a local computer science professor. This is done on a part-time basis (in addition to teaching classes at the university level). There are two other developers that also work part time on development and spend the rest of their time doing business development and providing customer support (via email and phone). The company does not outsource any of their software development work.

Marketing and Distribution

2ndSite markets its products through several different channels:

1. **Niche Advertising:** Providing a limited budget for paid advertising on niche sites. Though this has worked reasonably well, it has not yet created a positive return on investment. The company continues to invest in this channel to help continue to build visibility for the itself and its offering.
2. **Pay-per-click on Google and Yahoo:** There has been limited success with this channel and it remains uncertain as to whether the company will continue to invest in this channel.
3. **Organic Search:** Leveraging the team's experience in search engine marketing, the company has been able to successfully attract customers that are searching for key terms related to the product and target market.
4. **Blogging:** The company maintains an online blog (web log). This is still a relatively new initiative, but the results so far have been very promising. The biggest benefit of the company's blog has been the contacts generated and relationships established. One example is the Toronto Web 2.0 Mesh Conference (www.meshconference.com) that Michael is organizing with a number of prominent Toronto based bloggers. It is unlikely that Michael would have had the opportunity to be an integral part of this conference without the relationships that came through the company's blog.

Brian Halligan: On Software Sales

Brian Halligan is a 15+ year veteran of the software industry and has worked for both large, successful companies like Parametric Technologies Corporation (PTC) and smaller, more entrepreneurial firms like Groove Networks. He has extensive experience in developing and executing sales strategy for software companies.

One of the key insights from the discussion was Brian's views on how sales has changed for software companies in the past decade. Traditional methods of software sales involving large upfront fees and trailing "maintenance" fees are starting to break down. I discussed this earlier in the thesis in the chapter on sales and distribution. Brian believes that this shift is due to a number of fundamental industry changes.

First, the advent of the Internet has enabled software companies to communicate directly with customers in detail about the product, configurations and pricing. This has reduced the asymmetry in information between the sales team and the customer and therefore has reduced the sales team's leverage.

Second, in general the software industry has become more competitive with more overlap between product categories, more "substitutes" (in the way Michael Porter would describe them), and more new entrants.

Third, buyers of software weigh the trade offs of being early adopters and seem to be opting for being late adopters of new technology, rather than going through the pain of early adoption.

These fundamental changes have shifted the leverage from the sales team to the buyers of these technologies. Most software sales organizations like PTC, Oracle and SAP, have either resisted or denied the increasingly symmetric information and increasingly competitive environment and have stayed with their traditional selling models. Companies like Salesforce.com on the other hand have embraced the shift in leverage to the customer by offering a free online trial of their software over the Internet and pricing the software such that the customer pays by the month which lowers the customer's risk and shifts it to the vendor.

Software startups would be best served by recognizing the market realities and the new relationship with customers. By being transparent about pricing, taking on a majority of the adoption risk and leveraging more of the "direct" sales opportunity made available through the Internet, software startups can gain an advantage over their competitors.

Don Doge: On Microsoft and Software Startups

Don has been in the software business for more than 20 years. He is a veteran of five start-ups including Forte Software, AltaVista, Napster, Bowstreet, and Groove Networks. Don is currently Director of Business Development for Microsoft's Emerging Business Team. Don also maintains a very popular blog called "The Next Big Thing" (http://dondodge.typepad.com).

Microsoft and The Software Startup

The Emerging Business Team at Microsoft has two major roles; VC relations and business development with venture backed start-ups. The group share its opportunity maps with VCs which details where Microsoft is investing, details of its product roadmaps, and where Microsoft is not investing but looking for partners. This results in VCs knowing if a potential investment is likely to compete with Microsoft or it lets the VCs know if Microsoft is seeking a partner in that area and if introductions might be meaningful.

Don looks for companies that are building great software applications on Microsoft platforms, or filling potential holes in Microsoft's product offerings. Startups then receive help with development questions, introductions to members of the Microsoft product groups, and invitations to enroll in Microsoft partner programs. As appropriate, Microsoft may also provide assistance with sales leads.

Startup Capitalization

In his role working with venture capitalists, Don has also observed that the cash required to launch a start-up has reduced substantially over the past five years. However, SaaS applications can be more expensive to develop because of the end user customization features and the support of remote development. Again, enterprise applications are much more complex, require more integration capabilities, and must be customizable. Consumer applications are less demanding in this area. As a data point, Salesforce.com, the poster child for SaaS companies, raised $175M in VC funds, and took 4 or 5 years to break even.

Web 2.0 consumer companies can be built very inexpensively. They are typically simple, very specific applications with limited features, little or no integration requirements, and limited customization. These companies can self fund, or take some Angel funding to get started.

The Venture Funding Gap

VCs have enormous funds that they need to invest. It takes as much time and effort to manage a small investment as it does a big investment. The small investments don't tend to pay out big returns so they don't "move the needle" of the fund. There are still lots of big investment opportunities for VCs. When those dry up they will tend to move up market, not down. Meaning, they will do later stage investments or leveraged buyouts. Angel investors are covering the smaller deals all the way up to several million dollars.

There are some VCs who are focusing on seed stage investments, making many more investments, but spending less time with each of them. In Don's opinion, it is too early to tell if this approach will be successful.

Startup Decision: Independent Platform or Platform Complement?

Microsoft has had a demonstrated history of creating strong relationships in the development community (i.e. establishing an ecosystem that allows ISVs to make money by building on top of Microsoft platforms). However, many startups are concerned that Microsoft will "compete" with startups using offerings like Microsoft Live.

Don's response to this startup concern is that Microsoft is a platform provider. Even applications like Office and Outlook are evolving into platforms and servers. Many startups build applications on top of Office and Outlook. More than 80% of Microsoft's revenue comes from partners (however, this includes Dell, HP, and all the PC vendors as partners). Regardless, System Integrators and ISVs (Independent Software Vendors) still generate an enormous amount of pull-through revenue for Microsoft.

Windows and Office Live is really an attempt to showcase what can be done with AJAX, hosted applications, and the SaaS software model. These services are targeted at small businesses and home users. Microsoft doesn't have a big consumer presence so this is an experiment. Microsoft generates approximately $45B in revenue so small consumer applications and services are not going to move the needle or get much attention from management.

So, a startup is faced with a fundamental decision as to whether it seeks to build an independent platform of its own, or build a complement to an existing platform like one of those offered by Microsoft. Large companies like Microsoft provide a variety of platform choices that startups can potentially leverage. These include the highly successful Windows desktop operating system family (Windows 95, Windows XP, etc.) to newer platforms like Microsoft Windows Mobile and Xbox 360. Though is it is possible for startups to be a platform providers, most will either choose to be, or ultimately become, complementors.[71]

New Opportunities

When asked where the new opportunities are in software, Don replies: "Consumer services are really hot now. Cell phones are the platform of the future. Games are still a big opportunity. In the enterprise space security and compliance are hot." However, as always, entrepreneurs need to focus on solving a real customer problem. Startups are cautioned from pursuing a market opportunity just because it is "hot". Software entrepreneurs should leverage what they know, in markets that they understand, and attempt to solve a customer problem in a unique way. Customers should be involved early in the conceptualization phase and consulted often to ensure that the product is not getting off track. Don summaries: "Stay focused on the customer and things will probably work out fine."

[71] Cusumano, "The Business Of Software"

Bob Walsh: Writing Software to Writing Books

Bob Walsh is the founder/CEO of Safari Software, a software company founded in 1994. The company originally started as a contract programming/consulting company. Subsequently, the company developed a product called MasterList Professional, which is a task management system for individuals. As a developer, Bob personally experienced the challenges of proper time/task management. There was too much to do, and not enough time to do it. Bob ultimately decided to solve this problem as he couldn't really find an application that met his needs. He wanted to "make a fundamental difference in individual's lives".

The company operates as a single-person software company and has elected not to raise any outside capital.

Product Development

Bob has not yet used any outsourced development resources. Given that he has a software development background, this is not surprising. He said he might consider outsourcing some of the maintenance for his product at a later date, but at this time, it just doesn't make sense.

An area that Bob is investing time in is leveraging the advantages of test driven development (TDD). This is a process by which the tests for a given product or piece of functionality are written *before* the actual code is written. The motivation for this is to produce higher quality software that does not have unnecessary complexity. The goal of TDD is to only develop the software such that it "passes the test". This is often coupled with refactoring (the process of improving the design of existing code without modifying its functionality). TDD is part of a larger body of development methods called "agile methods" (discussed earlier in the thesis).

Sales and Distribution

Bob is making minimal investments in online advertising, through Google's AdWords program. So far, the results have been positive. For every dollar invested in online advertising, there is a return of approximately $2-$3 in sales.

One of the most successful initiatives for Bob in raising awareness for his company and product has been blogging. Bob maintains a regular blog at http://www.todoorelse.com. The blog is very relevant to the target audience of Safari Software – those interested in time/task management.

The Business, The Book

One of the things that surprised Bob was the sheer number of disparate issues and challenges that had to be addressed to operate a small software company. As part of the process of learning all the "tricks of the trade" of running a small software company, Bob sought out books that addressed many of the questions that he had as a software entrepreneur. Failing to find just the right book, he decided to author one that was focused on small software companies. The result is "Micro-ISV: From Vision To Reality", which addresses the needs of small software startups (or Micro-ISVs). The book has been successful in entrepreneurial circles and has sold well. Bob has been attempting to integrate many of the things he learned in researching and writing his book into his own software business.

Thoughts On "Web 2.0 Startups"

Given that Bob has talked to many software entrepreneurs as part of the research for his book, I asked what he thought of the new crop of "Web 2.0" companies. Bob replied: "Half of them will be gone in 18 months after they burn the rest of their money – and that will be far better than past cycles. Maybe 10% will be acquired as various companies try to stitch together a revenue model." However, Bob believes that several companies that started very small such as Six Apart and 37signals will do well as they have a product that people want.

Software Startup Success and Future Outlook

Bob believes that in order for a startup to succeed in today's challenging market, it has to dig very deeply into a problem domain and develop a product that makes sense for the given market.

However, he believes that the outlook and prospects for new software startups is exceptionally good: "The costs of production, distribution and marketing approach zero, while the number of people who might be interested in software relevant to them is skyrocketing."

OnStartups.com: A Community for Software Entrepreneurs

In November, 2005 as part of a small "experiment" to coincide with my thesis, I started a website focused on the topic of software startups. Appropriately the site was named "OnStartups.com". The initial motivation for the site was to write a blog and provide a vehicle for me to share some of my own personal experiences and opinions regarding software startups and get feedback and insights from other software entrepreneurs.

To initially promote the new blog, I first made friends, family and colleagues aware that I had started a blog. This generated some minimal traffic. I also posted comments to the blogs of others that were on related topics. The result was that the readers of these other, more established and popular blogs, would often read my comments and then click on my name to visit my website. This is a common practice in the blogosphere and a great way to build some initial traction. The biggest increase in traffic resulted from a few of my readers posting links to some of my popular articles to the social book-marking website reddit.com. These sites allow user submissions of interesting content and the community votes on which content they like. The result is that the popular articles and content rise to the top of the listings and garner more and more traffic from the community.

In the early stages of the blog, the average traffic was approximately 10-20 unique visitors per day. As of May, 2006 I've written over 50 articles on a variety of startup topics including marketing, strategy, hiring and capitalization and others. I recently spent some time determining just how much content was written for the site. The result surprised me. The articles on the site represent approximately 250,000 words of content – almost eight times the level of content in the thesis itself!

The blog now has a readership spanning over 80 countries and on average is visited by over a thousand visitors a day. Technorati, which ranks millions of blogs on the Internet, now ranks OnStartups.com in the top 1% of *all* blog sites on the Internet. Given this success, my intent is to continue to work on the site even after this thesis is complete and submitted.

The following are some of the key lesions learned from the onstartups.com blogging experience. Though the OnStartups.com site is not a software *business*, I believe that most of these lessons are also applicable to startup blogs. In fact, I have recently launched a new blog for my software startup (HubSpot) and am using some of these ideas.

- To maximize traffic and repeat visitors, it is important to have a site that is *focused* on a given topic. This causes increased referrals and inbound links.
- It is difficult to predict which specific articles will be popular with the reading community, it is often best to write consistently and frequently and let the readership determine what it wants to read or not read.
- It is important to offer RSS (Really Simple Syndication) feeds from the website. RSS is a technical protocol that defines how RSS client software can interact with web servers and determine if there is new content available. This allows users to easily monitor the site for new content without having to visit the site periodically. OnStartups.com currently has over 600 subscribers to its RSS feed. This means that over 600 individuals are notified anytime a new article is posted to the site, without any intervention on my part or on the part of my visitors.

- Social book-marking sites like del.icio.us, digg and reddit are major drivers of blog traffic. Many web users now use these sites to locate content that is popular and interesting. Content is posted to these sites by the community. This can be the original authors of the content or readers that found a particle article interesting enough to share with others.

OnStartups.com: The Startup Survey

As part of the thesis, an online survey of software entrepreneurs was conducted. This survey was made available to readers of the OnStartups.blog.

The purpose of the survey was to determine answers to the following questions:

1. Are these firms using open source as a *component* of their product offering?
2. Are these firms using open source as a *tool* in the building of their product?
3. To what degree are firms outsourcing software development?
4. What technical platforms are they using to build/deliver their product?
5. Are they targeting a horizontal (broad) market or a vertical (narrow) market?

Method

A link to the survey was made available to visitors of the onstartups.com website. Traffic to the onstartups.com survey site is driven by the following channels:

- Placement of links to articles on key social book-marking sites (like reddit.com and digg.com)
- Links to the site from third-party sites that have found the content of interest
- Users searching on the Internet through one of the major search engines (Google, Yahoo! and MSN)

In the early days of the launch of the onstartups.com site, friends and colleagues of mine were also notified of the existence of the site, so some early respondents of the survey may have been driven through this exposure.

Caveat: there is nothing scientific about the sample of responses to the questions posed. Those who responded were merely those who found the site, thought it interesting, and were motivated somehow to express their opinions. However, the responses are at least illustrative of the views and experiences of some software entrepreneurs, and are presented here to share those perspectives with others. Obviously, one might attempt to carry out a more scientifically valid study of the same questions.

Key Observations and Insights

The following are some of the hi-lights from the survey results that I found interesting and notable:

- Seems Microsoft's suite of .Net technologies are popular amongst software startups with a majority of survey respondents picking this platform.
- Only a limited number of respondents (11%) attest to outsourcing all or a significant portion of their software development.
- Over half (58%) of the respondents claim to be investing in Google AdWords™ to help promote their product.
- Most respondents (95%) have founder investment in the companies. This is not surprising. However, 6% of the respondents had received venture funding in their companies and 15% claim to have received some type of angel funding.

Details of the survey results can be found in the appendix at the end of this document.

Chapter 9: Tips For Software Entrepreneurs

As part of writing this thesis, I have interacted with dozens of software entrepreneurs. Many of these discussions occurred over late night email, within online forums or as comments on the OnStartups.com site. In this chapter, I'll attempt to capture some of the most insightful "lessons" I have learned from current software entrepreneurs as they shared their experiences and wisdom.

Get Started!

A consistent theme across all of the entrepreneurs that I spoke to for this thesis was that aspiring entrepreneurs should get started *now.* Further substantiation of the importance of getting started can be found in a study of almost 400 hi-tech companies led by Professor Ed Roberts of MIT. This study found that individuals that had not yet started a company by age forty were almost certainly not likely to do so later.[72]

Co-Founders Are Critical

Finding a co-founder is both critical and challenging for many entrepreneurs. Often, for "technology oriented" founders that have a software development background, they are lacking in the necessary skill-set and experience to fulfill the "business needs" of the startup. Co-founders are an excellent way to fill this gap. This impact of co-founders on the likelihood of the success of entrepreneurial has been extensively researched and documented by Professor Ed Roberts of MIT. In his research, Professor Roberts observes that in studies of new enterprise performance a larger number of co-founders is better.[73]

However, many first-time software entrepreneurs struggling with finding an appropriate co-founder. This is partly because they're unsure of what type of individual would be the best "fit". The decision is often between someone of a similar skill-set that can help with

[72] Roberts, "Business Planning in the Start-Up High-Technology Enterprise"
[73] Roberts, "Business Planning in the Start-Up High Technology Enterprise"

much of the product development in the early stages of the company, or someone with business skills to complement the original technical founder. Another challenge is that startups often don't have the financial resources to pay large salaries or bonuses and must instead rely on some form of equity as compensation. Though the upside of this kind of equity arrangement is attractive to some, there is often a limit to how much risk individuals are willing to take.

The Value of Self-Serving Products

Several of the software startups I researched and interviewed launched products that were originally built for their own internal use. For example, Element55 started as an IT consulting company. To better track and bill time on their projects, the team created a simple tool that allowed better time-tracking. Ultimately, customers expressed in the time-tracking tool and the company was created around it. Similar, 37signals created a simple project-management tool to help with their web design engagements with clients. Ultimately, the company launched this tool as a web-based software product , called BaseCamp.

Release Early, Release Often

Successful software entrepreneurs have learned that the most reliable way to build a product that customers want is to involve them in the process as early as possible. One way to do this is adhere to the "release early, release often" model of development. By making a product available to customers early, startups can gather critical information about what the market *wants*. Startups can then "continually iterate" and improve the product based on customer feedback. This approach may feel counter-intuitive to some, with the rationale being that it is better to "get it right" than to ship a product prematurely. Though there is clearly a balance to be found, the guidance from successful entrepreneurs is to err towards the side of releasing a product too early, rather than too late.

Focus, Focus, Focus

One of the most common mistakes that entrepreneurs make is attempting to do too much at once. This is evidenced in research done by Professor Ed Roberts of MIT who studied business plans submitted to venture capitalists for financing. He found that over half of the company plans suggested that the startups were thinking too broadly or attempting to do too many different things at once. These plans identified multiple product lines that the startup would pursue at the outset and outlined several large markets that the companies would "instantly attempt to enter and conquer". However, these companies failed to communicate in their plans a rational allocation of resources or priorities.[74]

This need to focus is echoed by experienced entrepreneurs like Doug Levin from Black Duck Software who has a single word of advice for software startup founders: "Focus!" Given the limited set of resources available to startups, it is imperative that the founders remain focused on their goals and not attempt to "boil the ocean". The probability of success of a startup is based largely on the degree of reality of what the team decides to accomplish and how focused and remains on that goal.

Sell, Sell, Sell

Outside of building a product that customers want, the most important task for most startup founders is to actively *sell* the product. The goal is to begin generating revenue as early in the process as possible. The approaches to selling can vary based on the market being targeted, size of potential customers and the nature of the business. Regardless, founding teams need to be able to sell to succeed.

"Word Of Mouth" Can Work Wonders

One common pattern was the success of customer referrals or "word of mouth" as a mechanism for acquiring customers. One example is 37signals, a small software startup

[74] Roberts, "Business Planning in the Start-Up High-Technology Enterprise"

that offers web-based project management tools. 37signals has managed to acquire over 500,000 users on its platform without spending *any* money on traditional marketing and advertising. 37signals has done this primarily through customer referrals Element55, another software startup interviewed for this thesis also relies heavily on customer referrals to generate new business. This particular pattern is not surprising. One of the biggest challenges for software startups is establishing credibility in the market place. One way this can be done is by encouraging customers to refer other customers. This type of endorsement and promotion can work much better than traditional marketing and advertising as customers are likely to trust and believe other customers (especially others they know) than a paid advertisement for the offering. Further, many startups cannot afford the cost of traditional advertising so customer referrals are even more attractive.

Plan To Change The Plan

A recurring theme across most of the entrepreneurs I talked to was that their companies did not ultimately end up with the same plan that they started with. So, the advice for entrepreneurs is to not overt-think the planning phase and "just do it". Many of the lessons needed in order to make the startup successful cannot be learned until *after* the company has already launched and is building a product and serving customers.

Chapter 10: Conclusion

As the thesis comes to a close, I'd like to summarize by taking a look back at some of the primary findings and insights from my overall exploration of the topic and then spend some time looking forward.

Looking Back: Summary Of Material

Entrepreneurs start software companies for a variety of reasons, not the least of which is the fact that software businesses provide the opportunity for exceptional growth and profitability. With examples like Microsoft, Oracle and others, entrepreneurs have clearly witnessed the potential reward for those that are able to establish dominant market positions for their software products. Software companies are also relatively easy to start – especially for computer programmers. Capital expenditures are low, and recent trends have pushed them even lower.

In the chapter on finances and capitalization we saw that software startups have a variety of paths they can choose from "bootstrapping" (i.e. raising little or no outside capital) to raising funds from angel investors or venture capitalists. There are many entrepreneurs I have talked to that favor a more "capital efficient" model and advise deferring raising any outside capital – or ideally, not raising any at all. Their claim is that software startups are better served by focusing their energies on getting customers and revenues – instead of raising funding, which tends to distract. I am personally in agreement with this line of thinking. However, it is clear from the data that the venture capital industry continues to invest heavily in software companies and there are many entrepreneurs whose plans involve raising some type of outside funding, be it from VCs or angel investors.

Startups need to make some important early decisions as to how they will position themselves in the market. They can focus on large, "horizontal" markets – which are risky, but can offer great rewards to founders and investors. Or, they can tackle a narrower, "vertical" market which has less risk but often less upside. Startups also need to decide what size customer they will address. Options range from large enterprises to small businesses and consumers. Each of these markets has tradeoffs. Enterprise markets often have long sales cycles – but can provide high average revenues per client that are often attractive to startups. On the other hand, small businesses and consumers often have less complex needs and require shorter sales cycles. However, these markets are generally very "fragmented" and can be difficult for startups to address efficiently.

At some point, software startups need to be able to *develop working software*. As it turns out, this is not a trivial feat. Despite the improvement in development tools, the availability of a global talent pool and new methods of development (like agile processes), software projects are still prone to failure. The sheer complexity of a reasonably large body of software code makes developing quality software challenging. Changing or ambiguous customer needs present additional challenges. This is why so many successful software companies have been started by software developers. These are individuals that understand the inherent challenges of software development and know how to address these challenges. It is rare to find a successful software startup that did not have a technically astute individual on the founding team. Though there is considerable discussion about the benefits of outsourcing for startups, it is unclear as to whether this makes sense for software startups as it can be argued that software development should be a core competency of these firms and as such should not be outsourced.

Once a product is developed, startups have to find a way to acquire customers and generate sales. This is often one of the most difficult aspects for software entrepreneurs. Not only does the company need to *find* potential customers for whom the product offering is a good fit, they must *convince* these customers to purchase the product. In large markets such as consumer and SMB (small-medium business), the availability of relatively new online marketing tools such as search advertising and blogging has provided startups with a new vehicle for reaching customers efficiently. In an online survey of software entrepreneurs conducted for this thesis, over half (58%) of the respondents claimed to be using Google AdWords to promote their business or product.

One way startups are tackling some of the inherent issues with the traditional software business is using new business models like Software As A Service (SaaS). The SaaS model allows for simpler deployment of a software product and generally creates a shorter "time to value" for the customer. Startups can benefit from this model as it avoids the resource-intensive "software update" process that plagues many traditional software companies. By providing their software on a hosted basis, startups can incrementally update their product on a frequent schedule and have all customers essentially upgraded *at once*. Companies like Salesforce.com and NetSuite have demonstrated the viability of SaaS by acquiring tens of thousands of customers. However, SaaS creates some distinct challenges for startups. One such challenge is the removal of the large, upfront license fees historically associated with enterprise software sales. Since SaaS products are generally sold on a "subscription" basis, startups using this model do not benefit from the upfront cash, but instead receive payments over time.

Looking Forward: Where Do We Go From Here?

As I look forward in the industry, I continue to see the opportunity for significant innovation. Despite the Internet bubble bursting, and the ensuing frustration of investors and management, it is important to remember that this still resulted in the creation of companies like Amazon and eBay. Though many of the "dot coms" did not have viable business models, some of the underlying foundations of the Internet in terms of technology did not go away. The number of users worldwide on the Internet continues to rise and I expect that software companies will continue to find ways to tap the potential of this large market.

Examples like Google demonstrate that even in software sectors like search, where the battles were thought to have been fought and won, there is frequently the opportunity for new innovations to disrupt and create opportunities for exceptional growth. Another example is Intuit. Scott Cook, the founder, entered the personal financial applications market using loans from friends and credit card purchases. Further, Quicken the flagship product, was the *forty-third* such application in its category. Despite all of this existing competition, Intuit went on to become a software powerhouse with a current market capitalization of over $9 billion.

My point here is simple. There are few other industries where an entrepreneur has the opportunity to create such immense value with such minimal capital. The "soft" nature of software continues to drive innovations in both established and new market categories.

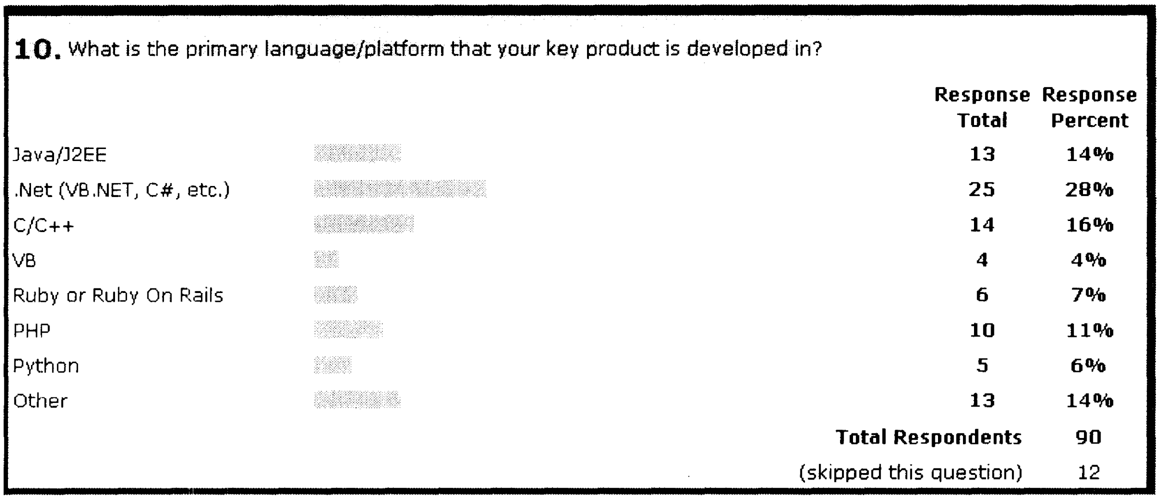
I firmly believe that the software industry will continue to be a vibrant area displaying exceptional growth and innovation. If there was previously any doubt in my mind that entrepreneurs would continue to start exciting new software companies, that doubt has been dispelled by the process of researching this thesis. Around every corner I found new ideas, new companies and passionate entrepreneurs looking to write the next great software success story. I am convinced that the story of the software industry is not coming to an end as it "matures", but that it will continue to reinvent itself as new opportunities emerge to apply human creativity in what is likely its purest intellectual form.

Ten years from now, I expect many new companies to either replace or stand beside the big names in software today like Microsoft, Oracle and Google. As we speak, the next big software success story is likely being written as an enterprising entrepreneur sits at her computer late at night, somewhere in the world. All it takes is a computer, a compiler, some creativity and some commitment to start the next billion dollar software company.

In any case, I plan to be around to see this happen – and with any luck, be part of the software entrepreneurial adventure for years to come. Onward!

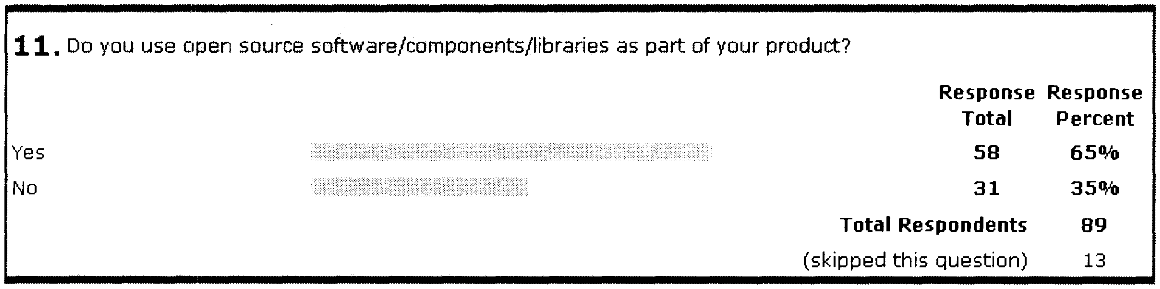
Appendix: Survey of Software Entrepreneurs

The following is a summary of the survey for some of the key questions in the survey:



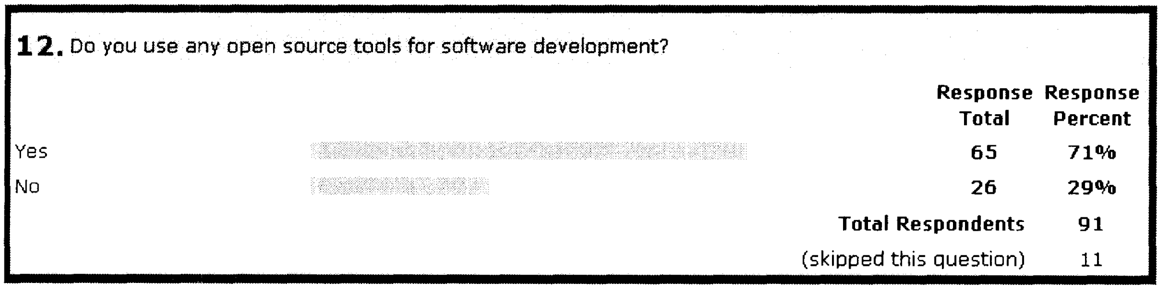
10. What is the primary language/platform that your key product is developed in?

	Response Total	Response Percent
Java/J2EE	**13**	**14%**
.Net (VB.NET, C#, etc.)	**25**	**28%**
C/C++	**14**	**16%**
VB	**4**	**4%**
Ruby or Ruby On Rails	**6**	**7%**
PHP	**10**	**11%**
Python	**5**	**6%**
Other	**13**	**14%**
	Total Respondents	**90**
	(skipped this question)	12



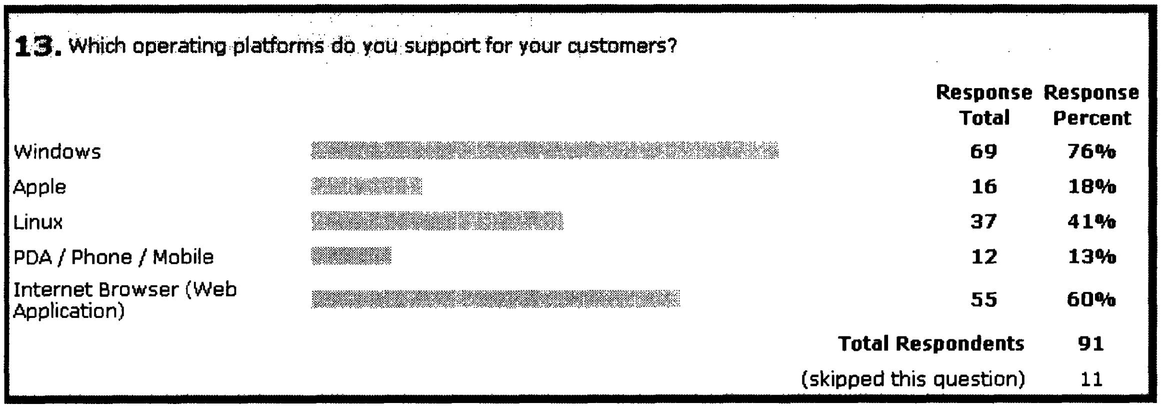
11. Do you use open source software/components/libraries as part of your product?

	Response Total	Response Percent
Yes	**58**	**65%**
No	**31**	**35%**
	Total Respondents	**89**
	(skipped this question)	13



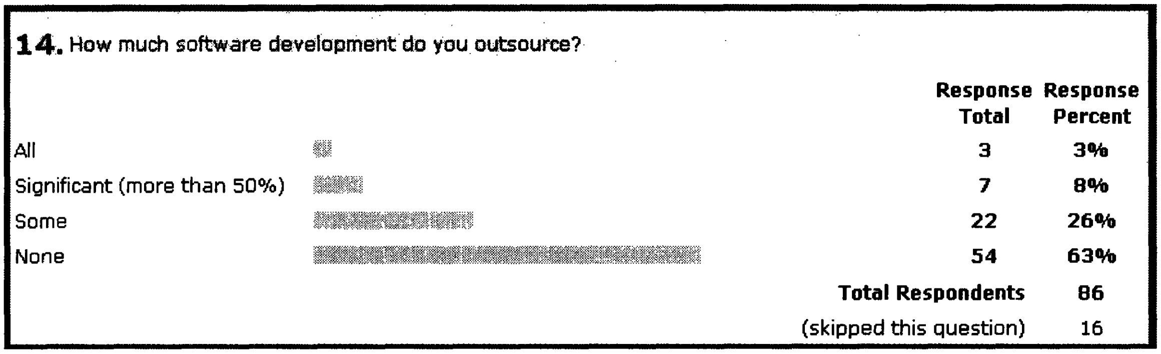
12. Do you use any open source tools for software development?

	Response Total	Response Percent
Yes	**65**	**71%**
No	**26**	**29%**
	Total Respondents	**91**
	(skipped this question)	11



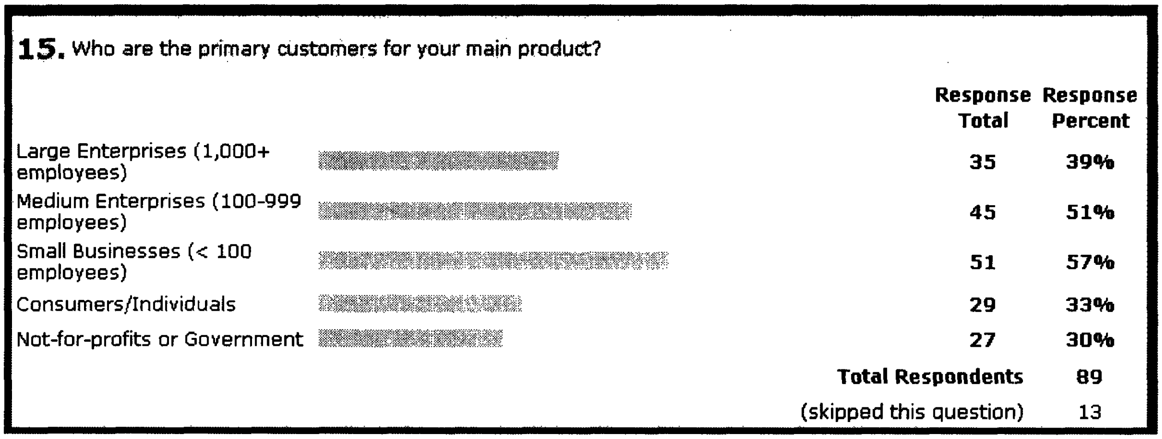
13. Which operating platforms do you support for your customers?

	Response Total	Response Percent
Windows	69	76%
Apple	16	18%
Linux	37	41%
PDA / Phone / Mobile	12	13%
Internet Browser (Web Application)	55	60%
Total Respondents		91
(skipped this question)		11



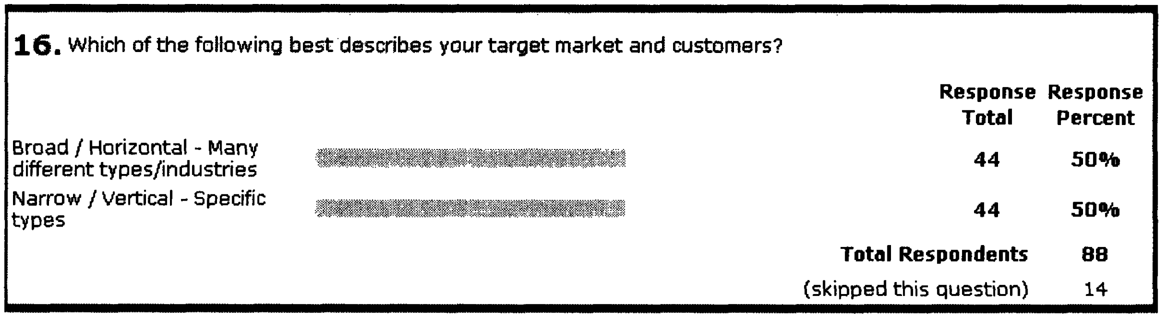
14. How much software development do you outsource?

	Response Total	Response Percent
All	3	3%
Significant (more than 50%)	7	8%
Some	22	26%
None	54	63%
Total Respondents		86
(skipped this question)		16



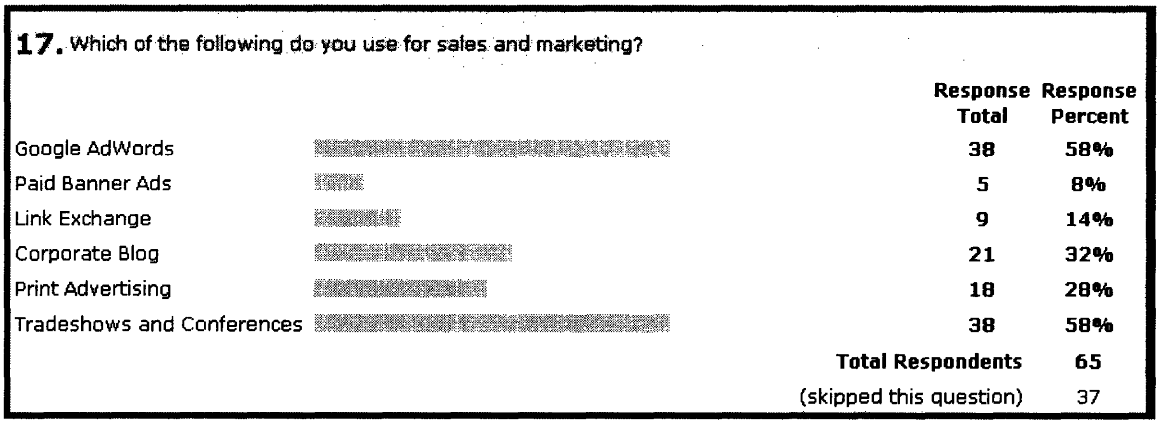
15. Who are the primary customers for your main product?

	Response Total	Response Percent
Large Enterprises (1,000+ employees)	35	39%
Medium Enterprises (100-999 employees)	45	51%
Small Businesses (< 100 employees)	51	57%
Consumers/Individuals	29	33%
Not-for-profits or Government	27	30%
Total Respondents		89
(skipped this question)		13



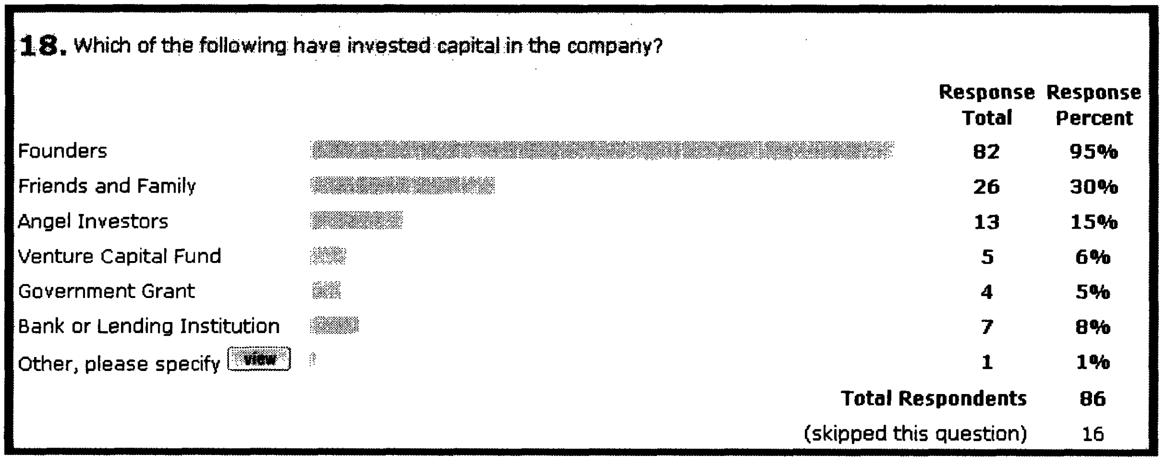
16. Which of the following best describes your target market and customers?

	Response Total	Response Percent
Broad / Horizontal - Many different types/industries	44	50%
Narrow / Vertical - Specific types	44	50%
Total Respondents		88
(skipped this question)		14



17. Which of the following do you use for sales and marketing?

	Response Total	Response Percent
Google AdWords	38	58%
Paid Banner Ads	5	8%
Link Exchange	9	14%
Corporate Blog	21	32%
Print Advertising	18	28%
Tradeshows and Conferences	38	58%
Total Respondents		65
(skipped this question)		37



18. Which of the following have invested capital in the company?

	Response Total	Response Percent
Founders	82	95%
Friends and Family	26	30%
Angel Investors	13	15%
Venture Capital Fund	5	6%
Government Grant	4	5%
Bank or Lending Institution	7	8%
Other, please specify	1	1%
Total Respondents		86
(skipped this question)		16

Bibliography

1. Allen, Davidi, Shah, Shapira. *"Web 2.0: Hype, Reality, or the Future?", 2005*
2. Ashbrook, Tom. *"Risk in Entrepreneurship".* Kauffman Foundation, 2002
3. Brodsky, Norm. *"Who Are The Real Entrepreneurs?".* Inc. Magazine, 1996
4. Calnan, Christopher. *"Report: Angel investments up".* Mass. High Tech, 2006
5. Cantor, Murray. *"Software Leadership".* Addison-Wesley, 2002.
6. Carlston, Douglas. *"Software People".* Simon & Schuster, 1985.
7. Christensen, Clayton. *"The Innovator's Dilemma".* Harper Business, 1997
8. Cook, John. *"Success Without VCs abounds in business",* Seattle PI Reporter, 2005
9. Cowley, Stacy. *"Salesforce.com outage cuts users off",* IDG News Service, 2005
10. Cristensen, Clayton. *Series of Presentations to MIT Sloan Fellows,* 2006.
11. Crowne, Mark. *"Why Software Product Startups Fail",* International Engineering Management Conference, 2004

12. Cusumano, Michael and Selby, Richard. *"Microsoft Secrets",* Simon & Schuster, 1998
13. Cusumano, Michael. *"The Business Of Software".* Free Press, 2004
14. DataMonitor. *"Software In The United States".* DataMonitor, 2005
15. Dodge, Don. *"Saas: New Software Model, New Problems".* Dondodge.typepad.com, 2006
16. Driver, Mark. *"Open Source Solutions Will Restructure the Software Industry"*, Gartner 2005.
17. Fitzsimmons, Patrick. *"Lone Hackers, Power Suits, and Gold Rushes: A History of Software Entrepreneurship"* Unpublished Working Paper, 2006
18. Foley, John. *"Power Shift".* Information Week, 2004
19. Fried, Jason. *"Getting Real".* Electronic Distribution, 2006
20. Gall, J. *"Systemantics: How Systems Really Work and How They Fail".* The General Systemantics Press, 1986
21. Gartner, *"Hype Cycle for Open-Source Software".* Gartner Research, 2005
22. Gianforte, Greg. *"Bootstrapping Your Business".* Adams Media, 2005
23. Gianforte, Greg. *"Bootstrapping: The Secret to Entrepreneurial Success".* 2005
24. Gilbert, Sarah Jane. *"How Can Startups Grow?".* HBS Working Knowledge, 2005
25. Global Insight. *"Venture Impact 2004".* National Venture Capital Association, 2005

26. Gordon, Graham. *"Blogmania: Should your software firm jump on the blog bandwagon?"*, Electronic Distribution, 2005

27. Graham, Paul. *"How To Start A Startup"*. http://paulgraham.com, 2005

28. Graham, Paul. *"Ideas For Startups"*. http://paulgraham.com, 2005

29. Hadley, Bruce. *"How To Get Started With SaaS"*. Electronic Distribution, 2005.

30. Harrsion, Mason, Girling. *"Financial Bootstrapping and venture development in the Software industry"*. Entrepreneurship and Regional Development, 2004

31. Hinchclliffe, Dion. *"Running a Business On Web-Based Software"*. ZDNet.com, 2006

32. Hock, Roeding, Purkert, Lindner, *"Secrets Of Software Success"*. Harvard Business School Press, 1999

33. Hedlund, Marc, *"The Builders Of BaseCamp"*. http://oreilly.net, 2005

34. Ireland, Peter. *"The Smart Startup"*. http://antiventurecapital.com, 2004

35. Judson, Bruce. *"Go It Alone"*. Collins, 2005

36. Kawasai, Guy. *"The Art Of The Start"*. Penguin Books, 2004

37. Klein, Karen. *"The Bottom Line On Startup Failures"*. Business Week, 2003

38. Knorr, Eric. *"Software As A Service: The Next Big Thing"*. InfoWorld, 2006

39. Kraus, Joe. *"It's a Great Time To Be An Entrepreneur"*. Electronic Distribution, 2005

40. LaMonica, Martin. *"Small Is Beautiful for Web 2.0 Startups"*. Cnet.com, 2006

41. Langham, Martin. *"Waterfall development model considered harmful"*. It-director.com, 2005

42. LeClaire, Jennifer. *"The Right Price: How Some of the World's Leading Software Firms Are Looking At Pricing Strategies Today – and Tomorrow"*. SoftwareCEO.com, 2005

43. Levine, Doug. *Personal Interview,* 2006

44. MacCormack, Alan. *"Product Development Practices That Work: How Internet Companies Build Software.* Sloan Management Review, Winter 2001

45. Marasco, Joe. *"Unraveling the Mystery of Software Development Success"*. SandHill.com, 2006

46. McBreen, Pete. *"Software Development: Dismantling the Waterfall"*. Addison Wesley, 2002

47. McGrath, Vance, Gray. *"With A Little Help From Their Friends: Exploring the Advice Networks of Software Entrepreneurs"*. Creativity and Innovation Management, 2003

48. Miller, Matt. *"Software VC Outlook: A Flight To Quality"*. Sandhill.com, 2006

49. Moltz, Barry. *"You Need To Be A Little Crazy"*. Dearborn Trade Publishing, 2003

50. Nesheim, John L. *"High Tech Startup".* Free Press, 1997

51. O'Neill, Jim. Personal Interviews Via Phone and Email. 2005-2006

52. Pallatto, John. *"U.S. Software Faces Loss Of Global Dominance".* 2006

53. Price, Will. *"Enterprise Start-Up Strategy".* http://willprice.blogspot.com, 2006

54. Quinn, James and Hillmer, Frederick. *"Strategic Outsourcing".* The McKinsey Quarterly, 1995

55. Rakham, Neil. *"SPIN Selling".* McGraw-Hill, 1988

56. Rangaswami, M.R. *"Seismic Shifts For The Software Market".* SandHill.com, 2005

57. Rip, Peter. *"The Web 2.0 Entrepreneur Bubble".* EarlyStageVC.TypePad.com, 2005

58. Roberts, Allen. *"Angel Funding Dollars Rise, Deals Decline".* Inc. Online, 2005

59. Roberts, Edward B. *"Business Planning in the Start-Up High-Technology Enterprise",* Keynote Address to the MIT Enterprise Forum, 1982

60. Roberts, Edward B. *"Evolving Toward Product and Market-Orientation: The Early Years of Technology-Based Firms",* Journal of Product Innovation Management, 1990

61. Roberts, Edward B. *"Entrepreneurs In High Technology".* Oxford University Press, 1991

62. Saratoga Institute, *"Human Resource Financial Report",* Saratoga Institute, 1997

63. Schelmetic, Tracey. *"TPI Publishes Global Outsourcing Trend Report".* Customer Interaction, 2005

64. Schroter, Wil. *"Go Big or Go Home".* Go Big Media, 2005

65. Sen, Snigdha. *"Silicon Valley Faces Web Startup Glut".* Business 2.0, 2006

66. Shah, Vipul. *Personal interviews via phone and email.* 2006

67. Shipley, Chris. *"Consumer, business technologies face same challenges".* Wisconsin Technology Network, 2006

68. Silverthorne, Sean. *"Lessons of Successful Entrepreneurs".* HBS Working Knowledge, 2005

69. Singh, Shruti. *"The New Breed Of Internet Startup Doesn't Want Your Money".* Crain's Chicago Business, 2006

70. Software Industry and Information Association (SIIA), *"Key Trends in Software Pricing and Licensing".* SIIA.com, 2004

71. Spolsky, Joel. *"Joel On Software".* Apress, 2004.

72. von Hippel, Eric. *"Democratizing Innovation".* MIT Press, 2005

73. Wainewright, Phil. *"Ownership and On Demand"*. ZDNet, 2006

74. Walsh, Bob. *"Micro-ISV: From Vision To Reality"*. Apress, 2006

75. Wilson, Fred. *"The Freemium Business Model"*. http://avc.blogs.com , 2006

76. Wohl, Amy. *"Tracking Software As A Service"*. *http://amys.typepad.com*, 2005

77. Wu, Brian. *"Entrepreneurial Risk and Market Entry"*. Small Business Administration, 2005