December 2003

Software Development in Small IT Firms: The Whitewater Method

Kris Aebischer
University of South Florida

Michael Harris
University of South Florida

SOFTWARE DEVELOPMENT IN SMALL IT FIRMS: THE WHITEWATER METHOD

Kris Aebischer
College of Business Administration
University of South Florida
Kaebisch@coba.usf.edu

Michael Harris[1]
College of Business Administration
University of South Florida
Mharris@coba.usf.edu

Abstract

Nearly 99% of all U.S. firms are small businesses and roughly half the private sector employees work for a small business. Yet surprisingly little research attention is given to IT and its role in small firms; neither from the small consumer nor producer perspective. This area is of increasing importance, as small firms struggle to grow and gain competitive advantage against the backdrop of rapid technological advancements. This paper looks at some of the more salient components of software usage and development gained from preliminary interviews with executives and IT workers at 3 small IT firms with the purpose of identifying how small firms address technology decisions related to the software they use and develop. Preliminary interviews have revealed that the development processes is distinctly different for small firms and the perspective from which they address key decisions in the development process is also different. In contrast to the traditional waterfall software development methodology we introduce the whitewater approach used by small IT developers.

Keywords: Small business IT, software development processes, small business software processes

Introduction

In nearly any firm today, information technology plays a significant role in the way business is done. The use of technology ranges from automation of tasks for the purpose of gains in efficiency to a more sophisticated use of IT in an effort to create a competitive advantage. According to the U.S. Small Business Administration, small businesses (firms with less than 500 employees) account for two thirds of the expected growth in jobs (Anonymous 2003). In the same report, a listing of expected growth in employees by industry listed "Computer and Data Services" as the fastest growing industry. But perhaps the most interesting point in the report, an estimated one in seven IT professionals work for a small business.

Adding further interest to an investigation of IT in small businesses is the belief by many that small businesses rely on creativity and bring innovative processes to the marketplace. Furthermore, the IS research community has shown a great interest in software development methodologies and processes. But in spite of these beliefs, the small business IT research has not received a justifiable level of attention.

This research examines small businesses in targeted interviews in order to determine how they shape their products and companies in order to compete. We examine three different companies: one that has gone through an initial growth phase and subsequently had to downsize to meet market conditions, another that has established an ongoing market position in serving a targeted industry area, and a third company that is just establishing itself in its market. In section 2 we describe the companies participating in the interviews as well as the plans for this research. In the following two sections we discuss some of the interesting findings uncovered thus far. Finally, in section five we summarize the findings to date.

[1] Authors names are in alphabetical order.

Background/Methodology

To begin to understand the small business software development life cycle (SDLC) a series of interviews will be conducted with employees in three organizations. The organizations are described in Table 1.

Table 1. Description of Interviewed Companies

Client Industry(ies)	Description
Skateparks	Point of sale system for managing skateparks.
Legal, Legal Services	Develops both packaged and custom software solutions for law firms and legal service organizations including knowledge management, case hearing management and other legal services.
Small businesses and organizations	Content management system for web sites that allows non-technical users to create and manage dynamic information and commerce web portals for either intranet or internet purposes.

This analysis is the first step in an eventual research stream. Our goal will be to use a grounded approach to uncover key issues affecting product decisions in these small IT businesses. In a later study we can conduct a wider survey across a larger sample size to determine the applicability of these factors in a larger sample, and thus to develop a more general product development model of small IT businesses.

Business Environment

IT professionals work in small IT businesses that are shaped by the twin forces of their environment and their customers (Figure 1). These developers and engineers find themselves in a unique environment when compared to their counterparts at larger companies.

Many of these small businesses are resource constrained with a short-term need to generate cash for survival. Their technical skill depth is based on the skills of one or two key individuals, and employee-training programs may be minimal. Payoff on human and other investments must be quick, resulting in rapid tumultuous transitions as custom projects for key customers are converted quickly to products for multiple sales.

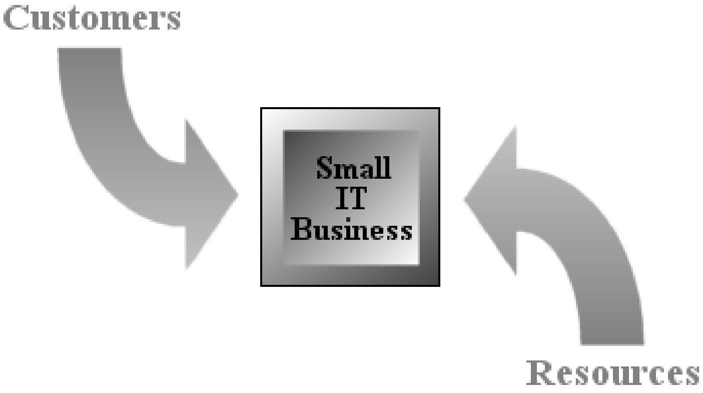

Figure 1. Small Business Environment

A small company's customer base likewise represents a unique market niche. Small IT businesses look for the hidden nuggets in the marketplace. Many (but not all) of these small businesses target customers who are too small to have substantial IT shops of their own.

The potential competition not only comes from other small IT companies, but also from large consulting firms and commodity IT products. The small company can face severe product pressures if the market grows too large, or if commodity products gain the features to enter into the specialized space. The small company competes by providing specialized features and high service into these product niches.

Finally, technology is often dictated to small firms by both the market and the standards processes, particularly when the small firm has a much larger client. For example, one of the interviewed companies worked with a fishing product distributor who was a Wal-Mart supplier. As a requirement of doing business with Wal-Mart, the supplier had to adopt Wal-Mart's data interchange standards. As a result, the small business found itself matching resources with the hundreds of developers in Wal-Mart's IT shop.

Likewise, while many large businesses assign personnel to influence technology standards to their best advantage, smaller companies are often resource constrained and they find themselves observers not participants in the standards setting process.

Existing Software Development Methodologies

The Waterfall Approach

Although numerous variations exist, the most well known software development methodology is the waterfall method (McConnell 1996). Figure 2 offers a simplified version of a typical waterfall methodology

There are three major steps, each with a set of associated tools and processes. The "Design" stage involves requirements generation as well as developing the documentation. Many of the technical decisions at this stage are made according to existing corporate preferences for platforms, processes, documentation formats and more. The "Code" stage, involves choices about programming languages. Finally, the QA stage involves the testing and review of the product.

The 'waterfall' label illustrates the cascade of the development process down the metaphorical steps of the development process. Each step is completed sequentially before the process cascades to the next step. Although everyone recognizes this as the waterfall approach, a more evocative metaphor of this traditional process may have been a 'canal'. Picture a huge ocean liner guided down a slow moving canal that uses locks to transition in a controlled fashion between stages.

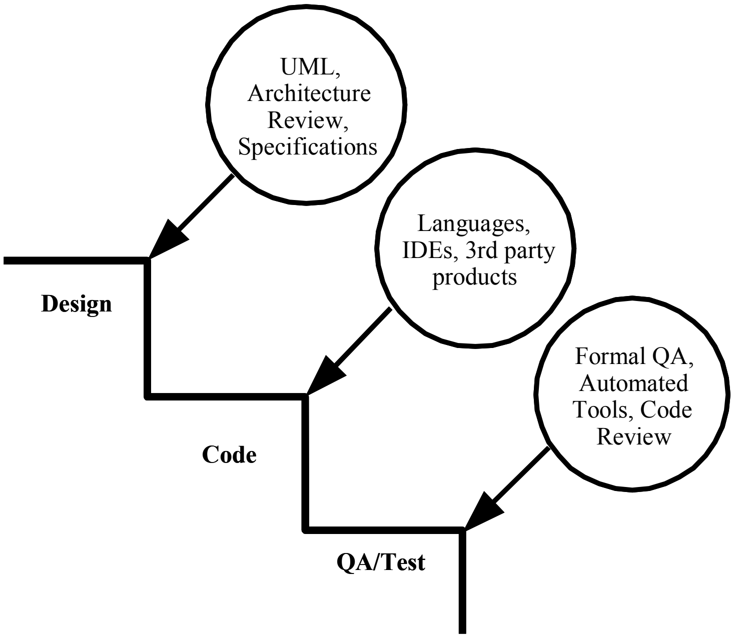
UML, Architecture Review, Specifications
Languages, IDEs, 3rd party products
Formal QA, Automated Tools, Code Review
Design
Code
QA/Test

Figure 2. Waterfall Process

Dynamic Development Methodologies

The staged approach of the waterfall method is best applied to a mature environment where the end-game can be clearly envisioned at the beginning of the project. In a turbulent situation, no amount of planning will deliver more than a blurred approximation of the final goal. Instead, the most effective approach is an experiential process that gradually brings the final goal into focus during the development lifecycle (Eisenhardt and Tabrizi 1995). Several studies have suggested alternative approaches for flexible product development.

- One recommendation is the creation of parallel market probes to better map the market space (Brown and Eisenhardt, 1997; Dahan and Mendelson, 2001). Each parallel market probe attempted will require four to six developers.

- An alternative approach is the synchronize and stabilize methodology (Cusumano and Yoffie 1999) used by Netscape and Microsoft to develop internet browser software. This fluid process divides a large development team (200 – 400 developers) into component groups (four to six developers). Although these small groups are given flexibility, they are bounded by their need to fit into the overall product solution.

- Product teams can also coordinate with custom development teams in order to blend the latest customer thinking with the latest product strategy (Hevner, et al. 2003). Coordination of a multi-tiered organization can lead to a better overall solution.

- A third development model of 'negotiated quality' (Baskerville et al. 2001) allows the use of quick prototypes for market introduction. Features that prove interesting to the market can be subsequently redeveloped. Customers who appreciate this approach must be technology friendly with a high appetite for new features and a high tolerance for rough edges.

Each of these processes have implicit boundaries that constrain potential development drift. Consider once again our cruise ship metaphor. Individual guests (developers) on the ship have freedom to choose from a plethora of events and dining alternatives. However, whatever the guests choose, they are contained within the same ship and the ship's course is controlled by the captain. Unfortunately, these control features are not as useful for small IT businesses. Each small business development team may have less than ten individuals. This leaves the team without the resources for multiple pre-development market probes, and without the organizational inertia to retain focus on overall product direction. Finally, the target customers are often small business owners who are IT pragmatists: They don't know how it works, but they expect it to work the first time and they may be intolerant of 'negotiated quality'.

The Whitewater Process

In contrast to the stately progress of an ocean liner, a small IT business resembles a kayak negotiating a whitewater river. Our kayaking developer does not have the options available to the big budget 'cruise ship' developer. However, the small developer controls their own course – they can explore tributaries or even portage overland. With this freedom of choice also comes the risk of losing focus and paddling off-course. Furthermore, no amount of quick paddling will save you if you pass over a 50 foot waterfall. The key is to develop an intimate knowledge of your eventual goals, and a foreknowledge of the problem spots ahead.

Rather than reliance on processes, smaller IT companies depend on tight customer relationships to help them map the upcoming territory and stay focused. Products are constantly exposed to current and potential customers from the earliest stages of the development lifecycle. Even the most junior developers receive direct customer guidance on a daily or even hourly basis. In comparison, developers at larger companies are often insulated from direct customer contact by intervening layers of team leaders, managers, and product marketing specialists.

In fact, many small IT businesses have a 'captive' customer that offers them special market insights. For example, one interviewed company develops products for law firms. This company began as an internal IT division of a law firm. The principals at the law firm saw broader value in the products they had developed for their own use. As a result they created a new company selling those products to other law firms. In fact, all three of the companies in our sample first developed custom solutions for a parent company and later offered that solution to a broader market base.

One risk of concentrating on a captive customer is that the product may become too customized and may miss the needs of the overall market. However, a captive customer relationship can also provide a doorway into the larger marketplace. The content management company we interviewed began with an internal content solution for an Internet-based magazine. The evolving software solution was demonstrated to every partner, vendor, and investor who visited the magazine. This constant interaction with outsiders kept the project focused on the market needs, and kept it from becoming too tailored to the needs of the original customer. In comparison, larger companies may treat product development as a 'secretive' activity that is only revealed to customers during the final development stages.

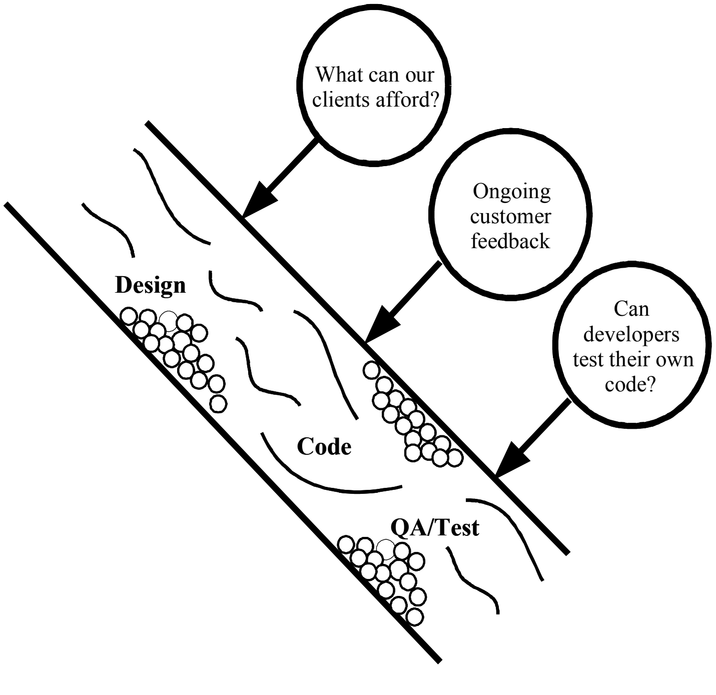
What can our clients afford?
Ongoing customer feedback
Can developers test their own code?
Design
Code
QA/Test

Figure 3. Whitewater Process

Throughout this process the smaller IT developer may also be constrained in terms of cost and resources. Some technology choices are cost prohibitive given the target market. For example, a database technology that costs $10,000 is too expensive to include in a product whose price point is only $15,000. Furthermore, the small business faces a resource issue. Each developer must be a generalist who can handle a wide range of issues. For example, a large developer may have an entire team of database specialists where a small developer may assign database responsibilities as one portion of one team member's job. Consequently, the small developer's depth of investigation can be more limited, and they often bring less specialized knowledge to specific issues.

Conclusion

Despite accounting for a significant and rapidly growing segment of business, little research in small firm IT use has been conducted to date. This research has uncovered some interesting findings in the area of software development process. By conducting preliminary interviews at three small IT firms we have seen how the viewpoints of the small firm are different and have begun to explore how the processes and methodologies are different. Small IT firms do not typically follow many of the IT community's "best practices", and yet are still able to enjoy success in the software development business. Further interviews with executives, project managers, and developers will offer more detailed explanations of these differences and will assist us in establishing a framework to describe steps to success in small business software development.

References

Anonymous (2003). "Small Business Economic Indicators for 2001." Office of Advocacy United States Small Business Administration, 2003.

Baskerville, R., Levine, L., Pries-Heje, J., and Slaughter, S. "How Internet Software Companies Negotiate Quality." IEEE Computer (34:5), 2001, pp. 51-57.

Brown, S. L. and K. M. Eisenhardt "The Art of Continuous Change: Linking Complexity Theory and Time-paced Evolution in Relentlessly Shifting Organizations." Administrative Science Quarterly (42:1), 1997, pp. 1-34.

Cusumano, M. A. and D. B. Yoffie "Software Development on Internet Time." IEEE Computer (32:10), 1999, pp. 60-69.

Dahan, E. and H. Mendelson "An Extreme-Value Model of Concept Testing." Management Science (47:1), 2001, pp. 102-116.

Eisenhardt, K. M. and B. N. Tabrizi "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry." Administrative Science Quarterly (40:1), 1995, pp. 84-110.

Hevner, A. R., R. W. Collins, Garfield, M "Product and Project Challenges in Electronic Commerce Software Development." The DATA BASE for Advances in Information Systems (33), 2003.

Huchzermeier, A. and C. H. Lock "Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D." Management Science (47:1), 2001, pp. 85-101

McConnell, S. "Rapid Development: Taming Wild Software Schedules", Microsoft Press, 1997.