

# What Differentiates Chilean Niche Software Companies

## Business Knowledge and Reputation

Sergio F. Ochoa, Romain Robbes, Maíra Marques, Luis Silvestre, and Alcides Quispe, University of Chile

// The extensive knowledge gained by specializing in a niche increases a small software company's odds of success. Consequently, the company's improved reputation lets it better negotiate contracts and improve its financial situation. //



**SMALL SOFTWARE COMPANIES** (SSCs, which have fewer than 50 people) represent approximately 70 percent of software companies in Brazil,<sup>1</sup> 95 percent in the US,<sup>2</sup> and

over 80 percent in Canada<sup>3</sup> and Chile.<sup>4,5</sup> In Chile, SSCs usually focus on developing and maintaining software and eventually providing consulting services.<sup>5</sup>

Unfortunately, these companies are fragile. Over seven years, in Chile, 47.2 percent of micro companies (fewer than 10 people) died and 5.7 percent became inactive, and 24.0 percent of SSCs died and 6.2 percent became inactive.<sup>6</sup> This tendency is also representative of most Chilean business sectors,<sup>6</sup> including software. So, this supports the hypothesis that the Chilean software industry is like a revolving door, with very small and small companies surviving only a short time. The situation in developed countries is similar: 60 percent of start-ups don't survive the first five years.<sup>7</sup>

To understand this situation, we conducted focus groups with people who were both experienced entrepreneurs and project managers at SSCs. A key survival factor was whether an SSC specialized. Seventy percent of the start-ups in our study began as generalist SSCs (GSSCs), and only five percent of them survived after five years. In contrast, the survival rate of the niche SSCs (NSSCs) grew to 78 percent. The key factors for the GSSCs' failure and NSSCs' success were the companies' business knowledge and reputation. Here we look in detail at the focus group results, examining those two factors and others related to SSC success and failure.

Researchers have documented other success factors, particularly the better technical and management practices in some companies' software processes.<sup>8-10</sup> However, no previous research we know of has explored niche knowledge.

### Study Characterization

We held four focus group sessions, each involving from two to six project managers, a moderator (one of the authors), and two transcribers (two of the other authors).

Of the 20 participants, 19 were software entrepreneurs early in their career. One participant had succeeded on the first attempt, six on the second, and two on the third. The rest joined an existing company after a failure (on average 1.05 failures per participant; standard deviation: 0.43). The participants currently were project managers in either the company they founded or another company, owing to its small size. They had an average of 10.5 years' experience in project management (SD: 5.6) and 5.8 years at the current company (SD: 5.2). Sixty-five percent of them had experience in both GSSCs and NSSCs. The companies were on average 11 years old (SD: 9.6); 70 percent of them were well established, focused on a niche, and were created after a GSSC failed.

The focus group sessions lasted from 60 to 90 minutes and were filmed. The moderator started with questions that served as starting points for open discussions between the participants. We later transcribed the opinions and analyzed them using an iterative coding scheme. Two of us independently categorized each sentence in the transcript according to an emerging classification. They then reviewed the classifications and reconciled the differences under the supervision of two more of us. This led to an exhaustive set of 45 fine-grained codes. We then employed axial coding to consolidate closely related codes into a set of 20 codes, and classified them as to whether they concerned GSSCs, NSSCs, or both.

Finally, we performed a card sort to infer broader categories (emerging themes) and the relationship between them. The final result was two sets of interconnected factors describing the situation in GSSCs and NSSCs. Some of our conclusions are supported by

two previous surveys conducted with Chilean SSCs<sup>11,12</sup> and an interview with the CEO of a successful NSSC in the judicial sector.<sup>13</sup>

### Generalized Companies: A Vicious Circle

According to the participants, the GSSCs started working on small, risky contracts while they were building a reputation. Their projects were short (less than six months),<sup>11,12</sup> focused on addressing the clients' needs, and usually started with an unclear problem that was clarified during product development.<sup>11</sup> These SSCs were financially fragile and always close to ceasing operations. This situation was similar to what Carmine Giardino and his colleagues reported.<sup>7</sup>

Figure 1 shows a high-level graph of the factors affecting the GSSCs, according to our focus group participants; we describe each factor next.

#### Low Knowledge of the Business Niche

The GSSCs usually had low knowledge of the customers' business domains. By definition, GSSCs run projects in different niches, preventing consolidation of the business knowledge. Because most projects started with an unclear goal and scope,<sup>11</sup> they became risky because the GSSCs had insufficient business knowledge to envision the projects' size and complexity.

#### Low Client Commitment

The clients had a low commitment in projects. They usually had little or no willingness to be the development team's counterpart. Typically, they outsourced the target problem<sup>11</sup> because they assumed SSCs knew enough of the business niche to develop products without the external support of clients or users, which wasn't true for

the GSSCs. Two focus group participants made these comments:

*In my experience, [the client's commitment] is rather low, and they think that we will read their minds.*

*In general, [clients] participate very, very little, and they are like, "See it yourself!" That is, the clients let the developers figure out the ambiguities by themselves.*

#### Risky Projects

The GSSCs took on risky projects. The GSSCs' lack of business knowledge, the lack of clear project goals and scope, and the clients' low involvement increased the projects' risks. In this scenario, requirements became prone to change, and developers couldn't clarify them alone. This situation tightened project schedules and added pressure.

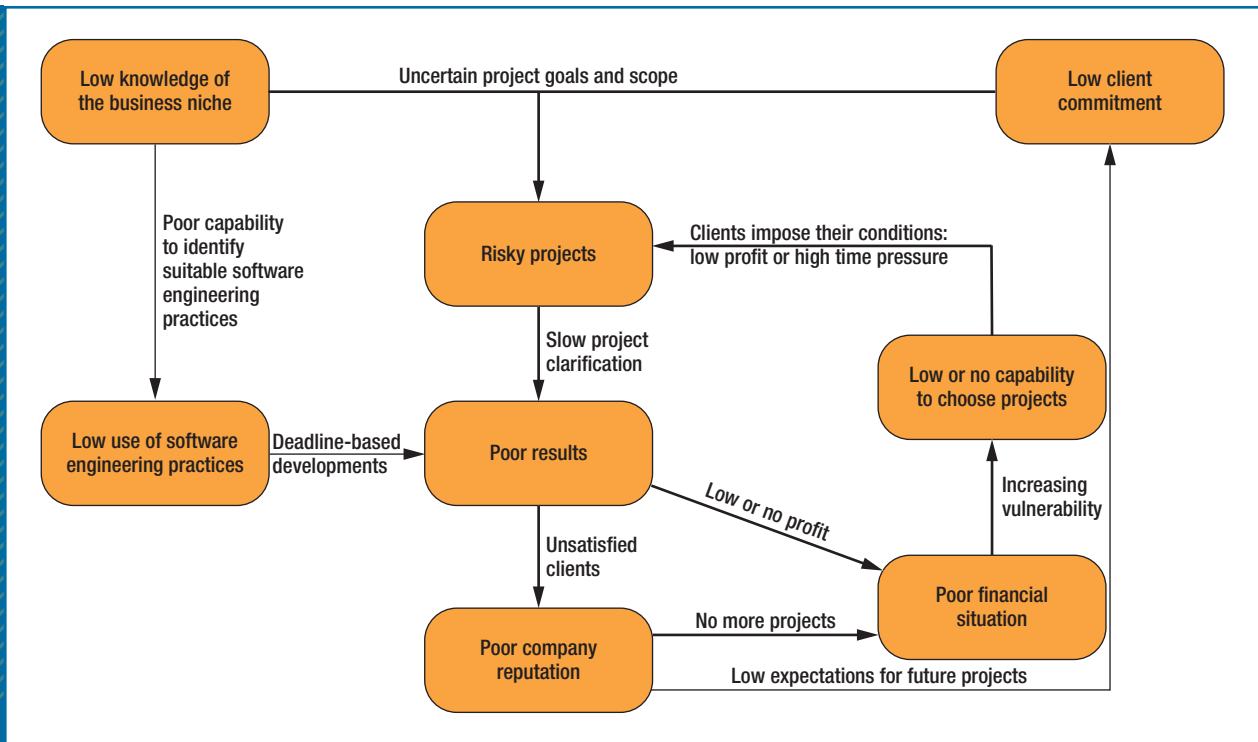
#### Poor Results

The results were usually poor. Typically, the obtained solutions weren't satisfactory for clients or had a low impact even if they met the specifications.<sup>11</sup> Even if projects didn't fail, they were rarely a complete success. Time pressure and the urge to minimize delays jeopardized product quality. This also reduced morale and self-esteem in the GSSCs. As one participant said,

*And so in the end you deliver a project ... to comply with a deadline. And you deliver whatever you have ... because the client put a date and you don't have the capacity to [do better] in this reduced time frame.*

#### A Poor Reputation

Poor products made the GSSCs lack success stories to bolster their



**FIGURE 1.** An influence graph of the factors affecting the generalist small software companies (GSSCs) in our study. Most of the GSSCs operated in survival mode throughout their lives.

credentials. Without success stories, the GSSCs’ reputation decreased. This produced a feedback loop: poor results and a poor reputation lowered clients’ expectations, perpetuating their low involvement in future projects.

**A Poor Financial Situation**

The GSSCs’ financial situation was precarious. Focus group members stated that staying in the market was a challenge: most GSSCs were often one paycheck from bankruptcy. Moreover, the projects’ profit margins were usually low because the clients imposed the contract conditions (see the next section). Delayed projects and growth in scope reduced the margin further because the budget didn’t grow in proportion, if at all.<sup>11</sup> So, the GSSCs usually operated in

survival mode. Connecting this factor with the one in the next section, one participant said,

*The companies, for financial reasons, do not have the liberty to pass on a client; they take bad clients to survive. Taking bad clients only brings them “bread for today, hunger for tomorrow,” and it quickly becomes a vicious circle. I have seen it many times.*

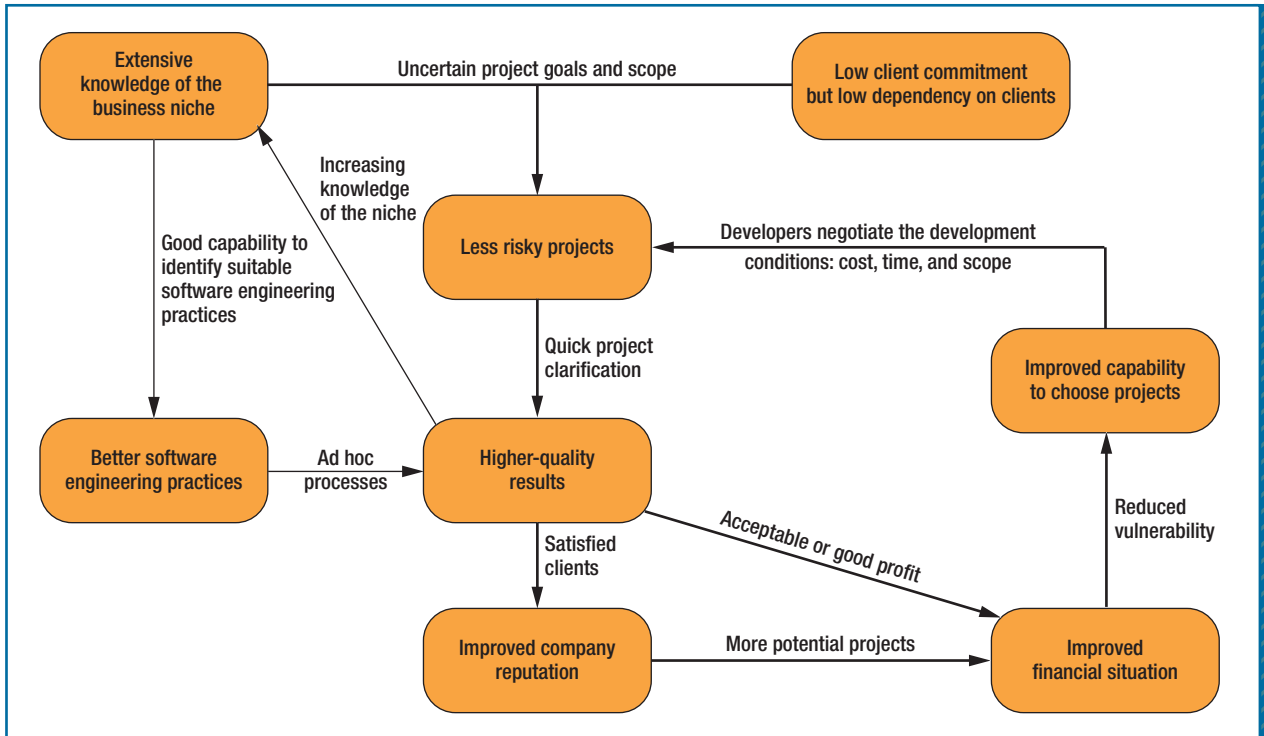
**Low or No Capability to Choose Projects**

The GSSCs had low or no capability to choose projects. Financially vulnerable GSSCs couldn’t afford to pass on a project. This gave clients a tremendous advantage during contract negotiations. The GSSCs were far more likely to accept harsh conditions (low margins or a tight

time schedule) than they would have otherwise, just to get a new project. The GSSCs’ low reputation and self-esteem also gave clients the advantage during negotiations. The resulting low profits and high time pressure increased the projects’ risk, thus closing the vicious circle.

**Low Use of Software Engineering Practices**

Another factor increasing project risk was the GSSCs’ low use of technical and management software engineering (SE) practices, such as the practices proposed by the ISO/IEC 29110 standard’s Entry Profile.<sup>14</sup> For instance, requirements management was often ad hoc.<sup>11</sup> Furthermore, the projects’ short time frame and variety made identifying suitable SE practices difficult and were incentives



**FIGURE 2.** An influence graph of the niche small software companies (NSSCs) in our study. Unlike the GSSCs, the NSSCs could drastically improve their situation and earn their way to financial stability and long-term growth.

to start coding early to minimize delays. This urgency made the projects deadline oriented and pushed developers to perform “heroics” to meet deadlines,<sup>13</sup> a risky strategy.

### Niche Companies: A Virtuous Circle?

From the focus group members’ observations, we gathered the situation in NSSCs was markedly different. Figure 2 shows the principal factors.

#### Extensive Knowledge of the Business Niche

The NSSCs usually had extensive knowledge of the business niche. Operating in a niche let them accumulate considerable knowledge about it. This included familiarity with the business domain concepts, processes, actors, challenges, and user or

client needs. This knowledge was extremely valuable for various reasons, as we explain later. Two of the participants made these comments:

*Development is much more effective when the developers “get” the business in which the project is inserted.*

*It’s easier when you are in a niche insofar as you can collect more history, history of the cases you had. ... You know the kinds of solutions that come to mind.*

#### Low Client Commitment but Low Dependency on Clients

Clients might still have had a low commitment in projects, but the NSSCs’ dependence on them was far lower. Because the NSSCs could use their extensive knowledge of the

niche, they depended far less on clients to make decisions, clarify project goals and scope, and envision product size and complexity. When the NSSCs needed knowledge from clients, they could formulate questions more effectively (for example, asking clients to choose between two alternatives). The NSSCs could also remind clients of potentially missing requirements, on the basis of the NSSCs’ previous experience in similar projects.

#### Less Risky Projects

The NSSCs took less risky projects that were more likely to succeed. By having extensive knowledge of the niche, the NSSCs reduced the sources of risks, the requirements volatility, and the time required to clarify uncertainties. The niche

knowledge was inherently more reusable and let the NSSCs identify and pass on risky projects or negotiate better conditions (we discuss this in more detail later). These factors greatly improved working conditions and significantly reduced risks and delays.

### Higher-Quality Results

The results were of a higher quality. The NSSCs' projects, being better planned, were less likely to experience delays. So, the time constraints were relaxed, making it far less likely the NSSCs would cut corners and harm quality. The niche knowledge also let the NSSCs make better decisions based on their experience, which usually resulted in the projects having a greater impact on clients.

### Improved Reputation

Over time, the NSSCs' reputation grew. Unlike with GSSCs, concluding a project much more likely resulted in a positive outcome. This produced satisfied clients and increased the NSSCs' reputation and self-respect. The greater the NSSCs'

*I know all the critical factors of the niche; I solve them and exploit them.*

### Improved Financial Situation

The NSSCs improved their financial situation. The increasing reputation let them charge higher fees while guaranteeing higher-quality results. They also delivered their projects on time. This too had a financial impact because delivering projects late was far more likely to eat into the NSSCs' profit margins than it was to cause clients to increase payments.<sup>11</sup>

### Improved Capability to Choose Projects

The NSSCs could choose projects and negotiate conditions. Their good reputation and financial strength opened a positive feedback loop because they

- received more project proposals owing to their reputation and
- could afford to wait owing to their financial strength.

If the NSSCs could grow their project portfolio and, in turn, their client

*these conditions]. I did this for this other client, and it's not like that. ... If one has accumulated data, ... one also has more arguments.*

Finally, the NSSCs might have had little competition in their niche because niche knowledge is hard to accumulate. This made it unlikely that several companies competed in the same niche. So, clients might have had to choose between an NSSC or a GSSC; in that case, the NSSCs' reputation was a decisive advantage.

### Better Software Engineering Practices

The NSSCs usually had better SE practices. They could use suitable SE practices for several reasons. Over time, the NSSCs could define a process, measure it, and tailor it to their niche, increasing project visibility and effectiveness. The SE practices embedded in the process could be selected according to the project context and international recommendations for the NSSCs' specific size and structure.

A side effect was that developers had a higher investment in the NSSCs. Their niche knowledge was much more useful there. Also, their work conditions were much less stressful because the projects were less risky and better managed; this reduced developer turnover.

### Additional Factors

Over time, as the SSCs delivered more projects, additional factors differentiated GSSCs and NSSCs.

### Room for Improvement

The GSSCs had little room for improvement. If they delivered projects, the low margins meant that little opportunity existed to reflect about opportunities to improve. This made it likely that the cycle would perpetuate

Gain experience with and knowledge of the market before engaging in entrepreneurship.

reputation, the higher their visibility was in the market. This brought new clients, retained current clients, and increased client participation in future projects. As one participant said,

*I am the best because I know the niche, or because they trust me. I have many years of experience;*

list, they could more easily pass on projects that imposed bad conditions on them. Their knowledge and reputation let them negotiate the conditions that clients proposed, thus reducing project risk. One participant said,

*With high certainty one can say no, no I can't [take a project under*

until the GSSCs closed down. Likewise, the GSSCs' precarious financial condition inhibited growth and prevented them from tackling large projects, which would have offered opportunities for larger margins.

### The Consequences of Failure

For the GSSCs, project failure could be fatal. The financial situation at a project's start was often alarming. Unsurprisingly, failing to deliver a project could have dire consequences, up to threatening company survival. Companies that couldn't afford to pass on a project could hardly afford to not get paid for it. Unfortunately, as we mentioned before, GSSCs often took on risky projects, which were inherently more likely to fail. This factor helps explain why Chilean SSCs have such high mortality.

### Client Involvement

Over time, the NSSCs could push clients to be highly invested in the projects. Clients, trusting that the NSSCs would deliver successfully on the basis of either their reputation or previous experience, were more involved in the projects. So, they were more likely to respond to requests for additional information, further reducing the project risk. This also reinforced the position of the NSSCs, which could make clients "pay with time"—that is, require a client representative to be available to resolve ambiguities and reduce requirements volatility.

### Growth and Self-Investment

With financial stability, the NSSCs could grow and invest resources in themselves. They could improve their processes and infrastructure to conduct more predictable development. They also could hire additional or

more experienced personnel, thus growing and becoming able to tackle larger projects. Finally, the NSSCs were better prepared to develop commercial products, which command a much higher profit margin.<sup>10</sup>

### Entrepreneurship and Survival

Counter to our expectations, part-time entrepreneurship increased the SSCs' survival rate. For cases of part-time entrepreneurship, the success rates were 14 percent for the GSSCs and 100 percent for the NSSCs. For cases of full-time entrepreneurship, the success rates were 0 percent for GSSCs and 30 percent for NSSCs. Moreover, the entrepreneurs who failed the first time but retried and succeeded, achieved their success through part-time entrepreneurship. Several participants mentioned that this approach helped them overcome the "valley of death." A key reason for this was the greatly decreased financial pressure.

## Discussion

Given the advantages we just described, it seems advisable that GSSCs focus on a niche market. Our focus group participants offered the following advice about how to become a successful NSSC.

First, gain experience with and knowledge of the market before engaging in entrepreneurship. Use that time to identify an interesting business domain and acquire knowledge of it. This knowledge will help you reduce project uncertainty and dependence on clients.

Second, start with part-time (15 to 20 hours per week) entrepreneurship involving only the company owners and focusing on developing a particular product or providing a specific service for that niche. This approach will help reduce the

financial pressure on your company and give you time to build success stories. It will also reduce the urgency to get new projects just to survive. Although consolidating a company takes more time with this approach, the risk is less than with full-time entrepreneurship.

Third, focus on extending and improving your products or services, because your company will have the projects under control: it will know the products and niche. Moreover, your company will keep acquiring knowledge of the niche, thereby building success stories and reputation, which will bring new clients and projects. Every new project will open opportunities to build a new product or improve an existing one, thus helping your GSSC become an NSSC.

Fourth, grow or change to full-time work when you have enough clients to survive. Keeping the risk under control should be a daily activity until you've consolidated the company.

Finally, enhance and improve your software engineering practices and make them explicit.<sup>13</sup> Product-centric companies can exploit practices such as software product lines (which let them quickly adapt an existing product to new needs) and those practices recommended by guides that consider the current and future maturity of the company software process—for example, the ISO/IEC 29110 Management and Engineering Guide.<sup>3</sup> This guide includes a set of processes and practices explicitly for very small entities, which comprise up to 25 people.

**T**he study is based on a limited number of focus groups. However, as we mentioned before, we triangulated some of our



**SERGIO F. OCHOA** is an associate professor in the University of Chile's Computer Science Department and an IT consultant for several Chilean public organizations. His research interests include computer-supported collaborative work, software engineering (including education), and mobile and ubiquitous computing. Ochoa received a PhD in computer science from the Catholic University of Chile. He is a member of IEEE, ACM, and the Chilean Computer Society and sits on the steering committee of the Latin American and Caribbean Collaborative ITC Research Initiative. Contact him at [sochoa@dcc.uchile.cl](mailto:sochoa@dcc.uchile.cl).



**LUIS SILVESTRE** is a PhD student in computer science at the University of Chile. His research interests include software engineering, domain-specific languages, transformation models, and software process lines. Silvestre received an MSc in computer science from the University of Chile. Contact him at [lsilvest@dcc.uchile.cl](mailto:lsilvest@dcc.uchile.cl).



**ROMAIN ROBRES** is an associate professor in the University of Chile's Computer Science Department and works in the university's PLEIAD research lab. His research interests involve empirical software engineering, including mining software repositories. Robbes received a PhD in computer science from the University of Lugano. He received best paper awards at the 2009 Working Conference on Reverse Engineering and the 2011 Working Conference on Mining Software Repositories and received a Microsoft Software Engineering Innovation Foundation award in 2011. He's on the editorial board of *Empirical Software Engineering*. Contact him at [rrobbes@dcc.uchile.cl](mailto:rrobbes@dcc.uchile.cl).



**ALCIDES QUISPE** is an assistant professor in the University of Santiago's Department of Informatics Engineering. He's also a consulting software process engineer for several small software companies. His research interests are in software engineering, particularly requirements engineering and software processes in small software companies. Quispe received a PhD in computer science from the University of Chile. Contact him at [alcides.quispe@usach.cl](mailto:alcides.quispe@usach.cl).



**MAÍRA MARQUES** is pursuing a PhD in computer science at the University of Chile, where she works in software engineering education. Her research interests include software processes, collaborative work, and software engineering education. Marques received an MS in computer science from the University of Chile. Contact her at [mmarques@dcc.uchile.cl](mailto:mmarques@dcc.uchile.cl).

findings with two surveys and an interview. Also, our participants' extensive and varied experience as entrepreneurs and project managers, in both generalist and niche companies—including successes and failures—increases our confidence in the results. Nevertheless, we also plan to conduct a large study of Chilean SSCs to determine additional causes of their success and failure.

SSCs can fail for reasons other than being generalist or niche. For instance, some Chilean SSCs don't follow standard software engineering practices.<sup>11</sup> Certainly, following better practices and appropriate standards<sup>3</sup> would improve the situation, and our participants were aware of that. However, they found that the niche aspect was critical to survival, hence this article's focus. We don't discount the impact of software engineering practices. Indeed, our study results show that the NSSCs, being less deadline driven, could use better practices to their advantage.

Issues might exist with the transcription and coding of the focus group transcripts because these are human activities. We mitigated this by adopting the iterative coding and review that we mentioned earlier.

Finally, the situation we observed might be specific to the Chilean market. Conducting similar studies in other markets would help assess whether the phenomenon is more general. 🌀

### Acknowledgments

The Project Fondef IDEa IT13I20010 partially supported this research. The work of Maíra Marques and Luis Silvestre was supported by the PhD scholarship program of CONICYT Chile, grants 2012-21120544 and 2013-63130130, respectively.

### References

1. C.G. von Wangenheim, A. Anacleto, and C.F. Salviano, "Helping Small Companies Assess Software Processes," *IEEE Software*, vol. 23, no. 1, 2006, pp. 91–98.
2. J. Aranda, "Playing to the Strengths of Small Organizations," *Proc. 1st Workshop Requirements Eng. in Small Companies* (RESC 10), 2010; [www.cs.toronto.edu/~jaranda/pubs/SmallStrengths-RESC2010.pdf](http://www.cs.toronto.edu/~jaranda/pubs/SmallStrengths-RESC2010.pdf).
3. C.Y. Laporte, S. Alexandre, and A. Renault, "Developing International Standards for Very Small Enterprises," *Computer*, vol. 41, no. 3, 2008, pp. 98–101.
4. M. Varela, *Diagnóstico de la Industria de TI en Chile* [Diagnosis of the IT Industry in Chile], white paper, Chile Innova Program, Chilean government, 2003 (in Spanish).
5. M. Otero, *Software en Chile* [Software in Chile], white paper, IFEMA-COCIM, 2005 (in Spanish).
6. P. Arellano and C. Carrasco, "Dinámica Empresarial: Brechas regionales y sectoriales de las pymes en Chile, Periodo 2005–2012" [Business Dynamics: Regional and Sectoral Gaps of SMEs in Chile, Period 2005–2012], Chilean government, 2014; [bit.ly/1RTVYAu](http://bit.ly/1RTVYAu) (in Spanish).
7. C. Giardino et al., "What Do We Know about Software Development in Startups?," *IEEE Software*, vol. 31, no. 5, 2014, pp. 28–32.
8. X. Larrucea et al., "Software Process Improvement in Very Small Organizations," *IEEE Software*, vol. 33, no. 2, 2016, pp. 85–89.
9. M. Crowne, "Why Software Product Startups Fail and What to Do about It: Evolution of Software Product Development in Startup Companies," *Proc. IEEE Int'l Eng. Management Conf.* (IEMC 02), vol. 1, 2002, pp. 338–343.
10. J. Aranda, S. Easterbrook, and G. Wilson, "Requirements in the Wild: How Small Companies Do It," *Proc. 15th IEEE Requirements Eng. Conf.* (RE 07), 2007, pp. 39–48.
11. A. Quispe et al., "Requirements Engineering Practices in Very Small Software Enterprises: A Diagnostic Study," *Proc. 29th Int'l Conf. Chilean Computer Science Soc.* (SCC 10), 2010, pp. 81–87.
12. *Análisis estadístico de la encuesta de caracterización de las PyMEs de software chilenas* [Statistical Analysis of the Survey That Characterizes the Small and Medium-Sized Software Companies in Chile], Computer Science Dept., Univ. of Chile, 2015; [bit.ly/1VE7Op7](http://bit.ly/1VE7Op7) (in Spanish).
13. R. Robbes, R. Vidal, and M.C. Bastarrica, "Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft," *IEEE Software*, vol. 30, no. 5, 2013, pp. 46–53.
14. *ISO/IEC TR 29110-5-1-1:2012—Software Engineering—Lifecycle Profiles for Very Small Entities (VSEs)—Part 5-1-1: Management and Engineering Guide: Generic Profile Group: Entry Profile*, Int'l Org. for Standardization, 2012.

myCS Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>