In August 2006, I interviewed two officers of a small software company based in Evanston, Illinois. This company is trying something interesting: combining offshore development with an agile development approach managed onshore. I wanted to find out what it takes to make this combination workable: how would you set up such a company? How do you manage an agile business across continents? How do you deal with communication and cultural issues? What development challenges does this approach present, and how do you deal with them? And what does this mean for the future of software development in the U.S.?

But first we talked about an earlier company out of whose ashes this concept arose—a company that sank.

A Rogue Wave
Sometimes you do what you think you should, what has worked before, but the situation just overwhelms you. “Did you know that waves are somewhat chaotic?” Doug Grimsted asked, “and about every seventh wave is significantly bigger than the other six? Then sometimes conditions conspire to create a monster wave, much bigger than anything else. There was one in the North Sea back in 1995; it was well over 80 feet high and came out of nowhere. You might have ridden out earlier storms, but there’s not much you can do when you get hit with a rogue wave.”

Grimsted shrugged, “That’s what happened to Geneer.”

2002: A Perfect Storm
“It was a perfect storm” Grimsted continued, with the kind of wry smile that said he’d told the story before. Many times. “A perfect storm, and a bad bet.”

Grimsted had been founder and CEO of Geneer Inc. Geneer was one of those few software companies that really seemed to get it right. With 185 employees, not too big; at $27 million in revenues not too small. The company had been in business in the Chicago area for 17 years when the storm broke. And they really seemed to get it right: agile, enthusiastic, committed. Enough
process to deliver solid value, but no process police. Smart, enthusiastic people working hard, learning daily, and having fun.

Things were bad in the business of software in 2001 and 2002, but Geneer had been there before.

“We’d always grown during recessions,” said Grimsted. “I figured if we downsized we couldn’t catch the wave on the inevitable upswing. So we bet that we could do it again.” What Grimsted didn’t say was that the people in Geneer who would be downsized were also his friends as much as they were his employees. But business dried up and remained dry for a lot longer that they expected. Existing customers felt they now had the freedom to pressure margins like they’d never done before. Cash flow got tight and then tighter and then negative. Even though the company had previously been debt free, they took out a credit line to tide them over. It became more difficult make payroll. But before there were any staff reductions, before any employees faced any salary cuts, before they even took the free food and beverages out of the kitchen, Geneer’s company officers took a voluntary 33% pay cut. It was that kind of company.

But they didn’t cut people. They took their lumps and hung on. And they nearly made it too. People were working hard, the loan repayments were being made, business was looking up. And then their bank decided to merge with a larger bank and felt it needed to clean up its books to make things look good. So they called in the loan.

Ironically, on the very day the bank closed Geneer down, they landed a big contract with a major aircraft equipment manufacturer. Too late.

OFFSHORE
“The ideas and culture that were behind Geneer were too good to die: the people focus, obsessive attention to customer needs and understanding the customer’s business, smart people working in smart ways, having fun, learning new stuff every day, pushing the envelope, and always, always agile.” Grimsted, now CEO of Aginity LLC, continued, “We are trying to reproduce them with this company. But of course the business has changed since 2002, the climate is different.”

“There is clearly an imperative to move certain kinds of work offshore, and we felt we absolutely had to incorporate this new aspect when we set up Aginity,” added Dan Kuhn, Aginity’s CTO. “The question was, what kind of offshore?”

“Outsourcing, especially overseas, is being touted as a cure-all,” said Kuhn. “It isn’t. Software is always looking for some magic pill or silver bullet that will solve all our problems without us having to think much or work hard. But offshore does require different ways of working.”

AGINITY
Aginity is a small, agile business intelligence application development company. They also build metrics dashboard applications called RapidDash™. They use a smart metadata approach to extracting information from business systems, synthesizing it into usable business knowledge, and displaying it in real time on the desktops of the people running the business. The systems can be built from scratch in as little as 60 days. They also use offshore development.

Much has been written about the movement to offshore software development away from the U.S. We know the per-hour rate of programmers overseas is often less, sometimes much less, than it is in the U.S, but how can we manage across this gulf? Grimsted and Kuhn listed and described the critical success factors that make Aginity work.

CULTURE
DG: When we looked at different cultures, we found some of them did not mesh well with what we needed. We have three groups working in the Ukraine now. We had worked with other groups in other countries but we found that they would often tell us only what they thought we wanted to hear. We felt we had to have a culture that is closer to the U.S. We must have people who will tell us what we need to know, not what they think we want to hear. The Ukrainians will do that.

1Aginity LLC, Evanston, IL: www.aginity.com.
“When the project team is co-located, there are all kinds of social communication methods, norms, and expectations. You don’t get these with offshore, so everything must be must more visible in the toolset.”

A PERSONAL CONTACT
DG: I decided to look to the Ukraine, because I had a Ukrainian friend who was returning to Kiev. Now he is there, he is able to act as an agent between us. Frankly, I don’t know how you could make this work without some level of personal contact between people who trust each other.

THE RIGHT PEOPLE
DK: We interview people for creativity and intelligence, rather than application knowledge. And we use the first iterations of development as a shakedown, to see if they really are the right people.

DG: We find that when you have good people with the right tools, you don’t need to overspecify things. So we make sure that the offshore development staff really understand the kind of things we need, and then we give them the freedom to create the solution. It’s a matter of trust.

ARCHITECT THE PROJECT
DK: You can’t just throw a project offshore. It has to be built for offshore. This means architecting the system—the old ‘minimally coupled components’ idea with carefully managed interfaces—but it also means architecting the project by carefully managing the knowledge resources.

COMMON VOCABULARY AND PATTERNS
DK: Architecting the project means using proven patterns and good frameworks. It also means that everyone uses a universal development language, a common vocabulary with methods that really work. We’ve found this only comes when teams work together over and over. That means iterative development.

STORIES
DG: We’ve found it is better to write compelling stories that allow people to understand the business need rather than churn out the traditional dry functional requirements spec. This allows our developers to intelligently interpret what is needed for the business and to interpolate or extrapolate any details that may not be in the spec. We collect a subset of the stories for an iteration, enough to get something done, but also small enough to be controllable. We use the iterations to flush out the details some more as well as to build content.

DK: We may be a bit more relaxed on the description and specification side, but we do establish strong acceptance criteria against which we estimate and track carefully.

MANAGE BY ITERATIONS
DK: We have to manage by these iterations and manage by expectations. Screwups cost less and are more controllable this way, plus by doing this we limit and focus the work, which always makes it more manageable.

A LEVEL OF ‘FAILURE’
DG: We assume out of the box that we’ll get a certain amount of what you might call ‘failure’ in the early couple of iterations. We refer to this as the ‘head scratching’ phase. Its purpose is to generate the ‘Omgosh!’ realizations, but to do it early and in a controlled way. Then we don’t get the omigosh late and in the final product where we can’t manage the consequences.

MANAGE BY DATA
DK: Going offshore, we have a much greater need to manage by data and facts. When the project
team is co-located, there are all kinds of social communication methods, norms, and expectations. You don’t get these with offshore, so everything must be much more visible in the toolset. That includes builds, tests, test results, and so forth.

**DG:** We do miss the day-to-day social contact you get when the whole team is in the same space. But we have found we can get pretty good at communicating offshore and you can send and receive quite subtle messages through IM with practice.

**Open Source Groupware**

**DK:** An indispensable factor is the availability of good groupware. We use a combination of mostly open source (and therefore free) groupware. This is what allows us to manage the development space.

**DG:** Even five years ago we simply could not do what we are doing now just because the tools were not available, they didn’t have the horsepower, or they were just too costly.

**Development Space**

**DK:** We manage to a set of project home pages that give us the necessary visibility into the project progress and status. We use a tool called Confluence\(^2\) to manage three development spaces: Customer, Frameworks, and Standards. We use a modified Mantis\(^3\) system for task tracking, issue control, and time tracking. Our CM system is the open source SubVersion.\(^4\)

**DG:** We also use Cruise Control\(^5\) from Martin Fowler’s Thoughtworks and SourceForge. They have a lot of tools that allow us to manage unit testing, timings, coding structures, class structures, and code packaging rules. We have the NCover\(^6\) code coverage tool, and the Simian\(^7\) reporting system from Redhill Consulting in Australia that give us further code management capability. These tools allow us to really see what is happening as it is happening.

**Convergence**

“This toolset gives us the capability of managing what is going on in Evanston and in Kiev at a level that allows this whole thing to work. We couldn’t do it without these tools, and we couldn’t afford them if they weren’t either very reasonably priced or open source. But the tools don’t do the work, and the key factor is still the people,” Grimsted summarized.

“The convergence of these high-grade open source support tools with the availability and accessibility of smart people in other countries, is changing the way we do business,” added Kuhn.

**A Bigger Pie**

“The issue has always been smart people, working well” said Grimsted. “It’s probably been true throughout history. It was certainly true of Geneer and it is true today with Aginity. I sometimes think the negative reaction in the U.S. to work moving overseas may be misplaced. People view it as a zero-sum situation—there’s only so much pie to go around, so if the people overseas get more, we must get less. But if we manage it the right way, if we make the pie bigger, we can all get more.”

---


\(^3\)Mantis bug tracking system: www.mantisbt.org/.

\(^4\)Subversion version control system: subversion.tigris.org/.

\(^5\)Cruise Control.NET: cruisecontrol.sourceforge.net/.

\(^6\)NCover code coverage tool: www.ncover.org.

\(^7\)Simian code similarity analyzer: redhillconsulting.com.au.

**Phillip G. Armour** (armour@corvis-intl.com) is a senior consultant at Corvus International Inc., Deer Park, IL.

© 2007 ACM 0001-0782/07/0100 $5.00