

# GRAND RAPIDS BUSINESS JOURNAL



WWW.GRBJ.COM VOL. 21, NO. 9 THE BUSINESS NEWSPAPER OF METRO GRAND RAPIDS - HOLLAND - MUSKEGON & WEST MICHIGAN MAY 27, 2003 \$1.50 a copy; \$52 a year

## “Extreme Programming” Really Isn’t So Extreme

*Just A Different  
Way To Build  
Custom Software*

**By Scott Payne**

GRAND RAPIDS — One fabled frustration of the computer age is the beta test revelation that a software package doesn’t address several issues that the client assumed it would.

And that also is when it often comes to light that the client must live with the problem, because — after months in development — making major changes in the software architecture would send the project far beyond its budget.

Actually, Carl Erickson says this sort of thing isn’t a fable at all, but an unpleasant reality that constitutes a permanent crisis in the software industry.

Erickson says the word “crisis” is not his. It was first employed 35 years ago by a frustrated multi-national client named NATO.

“The status quo for the development of software right now is really bad,” he told the Business Journal, “with lots of awful statistics about overrunning costs and schedules.”

Erickson comes at the subject from an academic background and with his own personal business interest in mind.

He served 10 years as a computer science professor at Grand Valley State University before founding Atomic Object, a 3-year-old custom software partnership that is a practitioner of something called “extreme programming.”

Erickson said both the company’s name, Atomic Object, and the title of the methodology it uses may create unconventional pictures in readers’ minds.

He says he regrets the term “extreme programming” — but not Atomic Object.

“The Atomic Object name came from brainstorming,” he said, referring to sessions with his partner, Bill Bereza, who currently is consulting in South America.

“Obviously our Web site is quite



BUSINESS JOURNAL PHOTO/JEFF DYKEHOUSE

*Instead of pacing at the front of a university class, Carl Erickson now runs a local custom software firm that does business throughout the country and overseas as well.*

important to us and it was a bit tricky to find something sort of memorable for use on the Web. Atomic Object is what we came up with, and it’s working well for us.”

“The trouble with ‘extreme program-

ming,” he added, “is that it’s a silly name. When some managers hear it, they think of cowboy hacker types working late into the night alone.

“And it couldn’t be more the opposite,” he said. “The ironic thing is that the

name actually refers to taking successful, well-known software development practices and turning them up to their maximum — or the extreme, if you wish.

“I’m trying to raise awareness locally of extreme programming,” he said, “because we’ve had an awful lot of success with it and I know some other people in town have had success with it.”

He contrasts extreme programming with what he terms the traditional means of creating software architecture.

“In the traditional method,” he said, “you analyze what the client wants — or what the client thinks he wants. You sit down and identify needs, then do a design, then a quotation, then you actually implement the source code, and then you test it at the end.

“And that’s the one that hasn’t worked out.”

The reason it hasn’t worked, he explained, is that ultra-high expectations have made creating software an incredibly complex process. “We ask an awful lot of software,” Erickson said.

“We try to build very, very complicated systems in software — and one of the complicated things about software is that you can do anything with it. It’s not like concrete or steel or some other material where you know its properties and you know you can use it in only so many ways.

“Software can do anything your imagination can come up with, and that makes it difficult in that you’re getting very ambitious, working months and months and months toward some big deadline, breaking new ground, making new inventions. And when you’re trying to do so much, it’s pretty easy to make things that don’t hold together all the way.”

Trouble arises, he indicated, in that the programmer is trying to accomplish so much in isolation from the client. “And when you’re doing so much, and you’re behind schedule and over your budget, testing gets left to the end — and that’s the thing that gets chopped off.”

By contrast, he said that extreme programming entails working very closely with the client in very small, short iterations — typically two-week periods — which include testing.

“We get a very small simple system working very quickly,” Erickson said, “and then we grow that system to actually do everything that needs to be done.

“The ideal way that we operate is that we have one of our clients on our site, or get someone here who knows what needs to be done and they can answer questions.

“What we need from our clients are what we call ‘user stories’: what they want their software to do, written from the user’s perspective.

“So then we take those user stories,” he said, “and decompose them into tasks that pairs of programmers work on — which we manage through an extremely high-tech system known as 3-by-5 note cards.”

He said extreme programming has a number of advantages, but that constant client feedback is one of the most critical. Too often, he said, the client is so mired in detail that it can’t grasp the scope of the problem it’s asking the software to solve.

Only constant contact, Erickson said, keeps clients and programmers on the same page, eliciting the exact dimensions of the clients’ problems.

“If I come to you out of context in isolation and I say, ‘Okay, what do you need the tool to do?’ — you’re unlikely to be able to answer that question very well,” Erickson said.

“And it’s not because the person who’s getting the question asked of them is dumb or doesn’t really know, it’s just that he can’t know. As humans we can’t react to something that is out of context,” he said.

“It’s like the difference between editing and starting a fresh page. It’s much easier to react to something — to edit — than it is to start from scratch.”

Testing during creation, Erickson added, also helps avert the industry’s chronic problem of “getting to the end of your six months of work and budget and finding out that you didn’t know something ahead of time — that you didn’t get enough feedback from users so you never knew what their real problem was.

“Maybe you didn’t discover what the real problem was,” he added, “because nobody really knew what the real problem was. Again, it’s a question of context.”

Another advantage to extreme programming, he said, lies in the team approach to limited goals rather than what he called the “big bang approach.”

“It’s easier to start with a solar system,”

he said, “and, working from there, to create a perfect galaxy, rather than trying to create a galaxy and forever tweaking it and never getting it to work in the first place.”

Erickson said the software industry has produced tremendous vision over the past 35 years, but that the process of crafting it has lagged.

“Extreme programming practices are very craftsman-like,” he said, “in the sense that these are things not necessarily learned in school, but learned in practice from people who are better at it than you are.”

For its own part, he said, Atomic Object may not be traditional in its programming methods but it is completely traditional in its training. “Internally we use craftsmen terms for our developers. We call them ‘novice’ and ‘apprentice’ and ‘journeyman’ and ‘craftsman’ and ‘master craftsman.’

“Right now we have a very active internship program with four interns — one apprentice and three novices — and one craftsman, the rest journeymen.

“I try to be the businessman,” he added.

He said he founded the firm because he wanted to practice what he had been preaching in front of the blackboards at GVSU.

Atomic Object has a wide-open 2,000-square-foot office in the old trolley car barn in the East Building in Eastown.

“We have desks with one computer and two people. That way it’s not only easy to lean back and ask somebody else a question, but it’s also easy to overhear conversation from another desk that you can contribute to. Back and forth communication like that is one of the underlying values in extreme programming.”

He said Atomic Object’s first contact with a client usually is a visit from him in which he tries to determine whether there’s an actual need.

“No one should write any piece of software that doesn’t pay for itself. So that’s the first thing: looking for that return on investment and seeing whether it’s worth building custom software.

“And then if it is,” he said, “then we bring out the high-tech card system and start asking questions.”

He said Atomic Object practices in most segments of industry in Michigan.

**BJ**