



Jan Steinman



Barbara Yates

The Demo Trap

“**C**’MON, GUYS, WE don’t have time for testing, we’ve gotta get the demo ready for the department heads meeting!” announced Aaron, the project leader, as he stared at the ceiling in exasperation. Things had been going well—too well, it seemed. MegaCorp had been having some difficult times with development schedules and after many studies, meetings, and much political maneuvering, management went along with Aaron Blake’s plan to transfer their traditional development environment into Smalltalk with a pilot project.

Aaron had been in this game awhile; he knew this project was a career maker or breaker, so he had planned the transition in careful detail. His team had gotten the training they needed, he had budgeted for the best tools available, and in a real coup for MegaCorp, he had even convinced them to bring in some experienced mentors, so that his team wouldn’t repeat someone else’s mistakes.

Most important, MegaCorp MIS Director, Andrea Saunders, had personally approved Aaron’s proposed development process, which was unlike anything anyone had ever seen within MegaCorp. MegaCorp needed rapid turnaround on various software projects and their traditional waterfall process had been running an average of 250% off original schedule. Their users required six-week changes; IS was quoting six-month changes but actually delivering in a year and a half!

It had been a hard sell. Andrea knew there was a problem, but she wasn’t ready to simply swap a new problem for an old one. “Whenever one of my guys comes crying about all the hoops they’re jumping through, I ask them, ‘ya got something better?’”

Luckily, Aaron was in his usual state of preparedness and had come to the meeting with an impressive presentation citing Barry Boehm and other development process scholars. He showed Andrea and her department heads how simultaneous design, implementation, and

testing could result in an evolving, incremental product that could deliver quick changes to users without sacrificing quality. He had no idea that all his careful planning could result in disaster!

EARLY GAINS

“Trigger” Larsen was Aaron’s best developer and the fastest squirt-gunner in the company. Trigger had gone off for a month of “Smalltalk immersion therapy” in an apprenticeship program, and had returned with stars in his eyes. “Hey Aaron, look what I got working today!” he’d often exclaim, which was a bit tiring, but it was the kind of nuisance Aaron could easily live with—much preferred over the usual complaining about short schedules and missed deadlines.

Trigger quickly assembled a GUI that was years ahead of anything MegaCorp had ever put together. True, the database wasn’t connected yet, and the legacy systems were not interfaced, but hey—it looked sexy and actually seemed to be doing something, unlike the “paint and draw” prototypes that other departments had been putting together for years.

Enthusiasm has its drawbacks and when combined with a little boastful competitiveness, it can have bad side effects. Trigger played racquetball with Denny Hicks, a developer on a “traditional development” team, and could contain his enthusiasm no longer. “Hey, you should see the neat stuff *we’ve* been doing!” Trigger shouted over the sound of ricocheting balls and pounding sneakers. It seemed an innocent enough boast at the time.

PROCEDURAL DISSONANCE STIRRING

It was kind of nice being ignored. Aaron knew this couldn’t last forever, but neither did he suspect it was the calm before the storm. The pace of a traditional waterfall project meant that the first third or so of the schedule was dominated by documents—but no one wanted to see the documents, they just wanted assurance that the documents were being produced.

Aaron was cheating a bit here on the cyclic development process. He followed the letter of the MegaCorp Software Procedures and Standards Manual by naming

Jan Steinman and Barbara Yates are cofounders of Bytesmiths, a technical services company that has been helping corporations adopt Smalltalk since 1987. Between them, they have more than 20 years Smalltalk experience. They can be reached at Barbara@Bytesmiths.com or Jan@Bytesmiths.com, or on the web at <http://www.bytesmiths.com>.

and listing all the required documents, and checking them off as produced, reviewed, and delivered at the appropriate intervals.

The “cheating” part was that they were “incomplete” by MegaCorp standards—he kept them in their Smalltalk development environment, so they could be updated readily, instead of “checking them in” to the mainframe, which generally meant no one ever changed them again, because rigorous procedures were involved with check-in and check-out.

They were also much smaller than usual because instead of an extensive boilerplate for every conceivable situation (half of which were marked “N/A”), Aaron insisted on plain English descriptions of the behaviors and interactions of the objects in the system.

A greater heresy that wouldn’t have passed muster if he hadn’t bribed the bored QA guys (by promising to train them in Smalltalk and actually let them write some test code) was the fact that none of the documents his group produced said *anything* about the internal structure of their objects! He knew he could get shot down for that, but he also suspected he’d need that flexibility later during performance tuning. “Isn’t it enough for now to say that an Employee can answer its Department without stating that it *stores* its Department somewhere?” Aaron mused to himself.

The net result of this was that after several months, MegaCorp treated Aaron as though he were “waterfalling” through analysis into design—it ignored him—while in reality, his team had Pretty Neat Stuff running. No one at MegaCorp had ever seen Pretty Neat Stuff running before at least 110% of the initial schedule had been spent!

“Hey Aaron!” shouted Jake Sather across the lunch room, “Denny tells me you’ve got a GUI going—can I get a look sometime soon?” Jake was Aaron’s peer, managing the “traditional development” group Denny Hicks worked in. “Sure, c’mon up this afternoon!” replied Aaron. (He had been catching some of Trigger’s enthusiasm as of late.)

WHAT IS OUR PRODUCT?

Jake’s demo went well—too well, it seemed. It would have been better if the demo had gone worse, say, if it had crashed a few times. It would have been better if Aaron had stalled Jake to prepare, because Aaron’s team had been practicing continuous integration and continuous testing, and consequently this was the best spur-of-the-moment demo Jake had ever seen at MegaCorp. Aaron looked at it and saw a half-finished prototype; Jake looked at it and saw a product.

The Friday staff meeting was not a pretty sight for Aaron. “You guys should see what Aaron’s been hiding from us!” Jake started out, “We all should be looking at this stuff!” They scheduled several demos that week. Marketing should get a look; Sales was on a boondoggle, so they would need their own demo; of course the tech writers would need a working copy; and users—what about the users? “We’ll have to find some; Aaron, can you look into that?”

None of Aaron’s careful planning accounted for the

coming weeks. Oh, he had planned for demos all right, but he didn’t anticipate the magnitude of interest that the “early GUI” generated.

But the interest in Aaron’s stuff was not passive. Everyone thinks they’re a “GUI guru,” when really they are more likely seduced by “neat stuff” rather than what actually is useful to an end user. Marketing immediately wanted “more color and icons,” while various VPs who drifted in and out of Aaron’s office put in their own personal GUI order. Meanwhile, Sales wanted to immediately ship everything they saw. Aaron was averaging at least two demos per week, each one ending with some “recommendation” of some kind. His schedule was slipping as his developers thrashed away implementing conflicting, unplanned changes.

Aaron fantasized about the want ad he would soon be placing: “Product Demo Manager for hire. Expert at jumping through upper management hoops with constantly changing but meaningless user interface stuff. If your product is demos, I’m your man!” He thought hard about what he was trying to accomplish, mentally crumpled up the half-composed ad, and picked up the phone. “Andrea, can we have a talk?” he asked.

RETURN OF SANITY

After hearing him out, Andrea Saunders agreed that Aaron’s project needed a shield. Demos would be limited to the period immediately following “cycle end,” which averaged every six weeks. “Must do” demos in between the scheduled ones would only be on the software that had recently been formally presented—no more pulling people off “real work” to prepare a special demo.

Trigger had completed the architecture work and much of the design and had given the implementation a good start. His outgoing, enthusiastic, and somewhat boastful personality earned him the new unofficial title of “appeasement engineer.”

“Half your job is to see that no one else on the team is impacted by demo activity,” Aaron told him, “and the other half is to do everything else you were doing with all your time before!” he joked. He knew that Trigger was up to the challenge, and could carry on limited peer review and coding while carefully tracking all the little tasks that producing a good demo requires, and still have time for manic squirt-gun fights.

The Demo Trap had slipped the schedule, and they weren’t going to ship all the functions they had originally planned, but their “continuous development” regime was paying off regardless. They had taken on “productization” early, and all they had to do between cycles was add functionality. Sales wanted what they had *now*, and Marketing was actually considering a “MegaSoftLite” offering to “selected beta partners.”

“If this keeps up,” Aaron mused, “it could be the first time in history that MegaCorp has shipped a product—albeit not the one they set out to build—ahead of schedule!” He sighed a contented sigh and went back to planning his next Smalltalk project. ☞