Interview with Victor Basili

FINDING AN EXPERIMENTAL BASIS FOR SOFTWARE ENGINEERING

NINETEEN YEARS AGO, A CONsortium was formed between the University of Maryland, NASA Goddard Space Flight Center, and Computer Sciences Corporation. Since then, the Software Engineering Laboratory has been studying the entire software-development process. It is one of a very few places that treats software engineering as an experimental science. Last year, the SEL was the winner of the first Software Process Achievement Award, given by the IEEE Computer Society and the Software Engineering Institute. Early this year, one of its founders, Victor Basili, talked with Managing Editor Angela Burgess.

Q: *Why do so few organizations approach software engineering as an experimental science?*

A: I think it has to do with the relative immaturity of the field and the lack of respect for the complexity of the software business. Software development is an engineering process, not a manufacturing one. Everything we do is human-based. Most companies assume that if a method sounds good or feels good, it should work. They don't take into account how well it fits with whatever else they do, whether it is appropriate for the product and people characteristics of the organization, and so on. They don't learn as an organization from one project to the next in an organized way. That's what any business must do, including the software business.

Q: *Many of these companies are very profitable, despite the fact that they have little understanding of the development process. Are you saying they've just been lucky?*

A: In a way they have been lucky, because they've been running their business based on intuition and on good people working hard. We have been lucky to attract some of the very best technical people to our business. But our intuition isn't always right. In many of our experiments, we have shown that our intuition about software is wrong. But one intuition that the very best companies have used is to focus on the right goals — business goals. By adopting a focus that fits their own business, they adapt processes to fit those goals. One important lesson we've learned is that there isn't just one right process if you want to produce a quality product. There are many ways to do this. The key is to identify and tailor a process to fit your goals. Profitable companies have been more effective in doing this.

Q: *And is it now cost-effective to conduct experiments?*

A: Yes, although it's not free. It does take an up-front investment to set up an organization that can learn from experience and apply that experience to improve the process. But attitudes about experimentation are changing. Companies are beginning to realize that investing in this kind of knowledge is better than the alternative, which is to institute a change or adopt a new tool without any idea of what effect it will have. In other words, it's relatively less expensive and less risky to conduct experiments first. Companies have spent a lot of money on technologies that did not have the desired effect.

COMPANIES HAVE BEEN RUNNING ON INTUITION, BUT INTUITION ISN'T ALWAYS RIGHT.

Q: *How did you get into this area?*

A: I did my dissertation on semantic models of programming languages. There's a natural progression from modeling languages to designing them to implementing them to evaluating what you implemented. I was involved in the design of several programming languages: the graph algorithmic language, Graal; SL/1, an array language for the CDC Star; and the Simpl family of languages and compilers. At each step, I'd read the literature and try to contribute something. But that last step was a killer. There wasn't much literature on evaluating anything in the software arena. I couldn't believe it because it seemed like an obvious thing to do.

Q: *How did the SEL go about setting up a laboratory?*

A: There was a desire at NASA Goddard to evaluate and improve their software-development process and they knew that I had been working in the area of measurement and evaluation, so Frank McGarry and I got together to try to measure and evaluate their products and processes in order to improve them. CSC was the contractor developing software for NASA — Jerry Page was the CSC manager — and so the consortium was formed.

I spent my sabbatical in 1976 setting up data-collection forms and mechanisms. At first we just collected data — actually too much data — but we learned over time. We also began trying to evaluate before we understood the environment. So we began building baselines and models and organizing what data was collected. That's how the Goal/Question/Metric paradigm developed. It helped us focus on our goals so that we could decide what data needed to be collected.

We tried to apply other people's models but found they didn't work for us — they were based upon their data, their goals, their environment, and their way of doing business. Then we began to evolve the models based upon what we learned and how things changed. We began designing experiments, first at the university, and then at NASA itself, using professional programmers. We actually run different types of experiments from case studies, sometimes repeated over time to measure the effect of a change over a series of projects, to controlled experiments, where we isolate and control for the variables of interest. Once we have shown statistical significance in a controlled experiment, we try a case study. If we are successful, we apply it to multiple projects.

Q: *And how do you use the results?*

A: We use the results to understand, assess, and package models and best practices related to the problem domain. It's part of an overall approach called the Quality Improvement Paradigm, which is based on the scientific method, like the

Shewart-Deming model, except it's aimed at engineering rather than manufacturing. It's steeped in the idea that measurement is an excellent abstraction mechanism for learning what works and what doesn't. It treats each project as an experiment from which we can learn. The organization structure associated with the paradigm is the Experience Factory.

TECHNOLOGY TRANSFER IS BUILT INTO THE EXPERIENCE FACTORY.

Q: *What is an Experience Factory?*

A: It is an organizational structure that supports the analysis and synthesis of experience gathered from projects. One thing we learned was that our business does very little learning. It's nearly impossible to have a learning perspective when you are working on a real project, with real deadlines and real budgets. You don't have the time to learn. So we have two separate organizations. The project organization delivers the system, aided by whatever support the Experience Factory can give it; the Experience Factory analyzes and synthesizes experience and supplies it on demand. Of course, the same people can populate both organizations, spending some time in each. The key is that they are considered to be separate activities.

Q: *What exactly does the Experience Factory supply on demand? To whom?*

A: We've learned that you can't just hand developers a tool or a method and walk away. You have to be there for them, as a resource. That means supplying cost models, information about what types of defects they might anticipate in this class of problem and what approach has been most effective in uncovering that class of defect — whatever they need. We start with the premise that everyone wants to do a good job. Developers will tell you what works and what doesn't. They will tell you what they need and when. The Experience Factory supplies what the organization has learned about the problem at hand.

Q: *How do you deal with technology transfer?*

A: That's the best part: Technology transfer is built right in; as you learn, you modify and try again. You are transferring as you learn. Everyone is part of the learning and transfer process. A process is only put in place if it's based upon experience.

Q: *It sounds like management has as much to learn about learning as developers.*

A: Absolutely. Management needs to understand better the software business and that means such things as understanding the relationship between process and product. That is, if you want a product to have a certain set of characteristics, you have to define them and learn what processes you need to put in place to get them. They have to understand that process is a variable to be manipulated. Managers must support learning, and that includes off-line learning in the form of controlled experiments.

Q: *Now that you have this framework, where does the SEL want to go from here?*

A: One goal is to package the whole Experience Factory organization and move it to other groups inside and outside NASA. Currently it is only used to package experiences for ground-support software for satellites at Goddard. We would like to see something like it at Johnson, Langley, Jet Propulsion Laboratory, Marshall, and at a broader level at Goddard.

The SEL has probably had more of an impact outside of NASA. Several organizations have adapted and applied various aspects. The GQM has been quite widely disseminated, although in most cases it is an earlier version. Some organizations have applied the QIP and the Experience Factory is popping up at various organizations.

From a research perspective, we continue to evolve and formalize the steps of the QIP. We try to answer such questions as: How do you tailor technologies to the needs of the organization? How can you learn from another organization's experience based upon the context in which it was successful? We are trying to formalize and automate the GQM.

We have a group of organizations, universities, and companies, all over the world, who have teamed up to share knowledge and replicate each other's experiments. The group is called ISERN [International Software-Engineering Research Network] and we meet once a year, but constantly exchange information. Many of the founding members spent some time at the University of Maryland.

Another thing we want to do is form a consortium of companies, each with their own experience factory, aimed at sharing experiences so that we can learn in parallel. If we can understand how to take what has been effective in one organization and adapt and tailor it to another, we can share and speed up the learning process. ◆