Editors: Nancy Eickelmann ■ Motorola ■ nancy.eickelmann@motorola.com
Jane Huffman Hayes ■ University of Kentucky ■ hayes@cs.uky.edu

Victor R. Basili's Contributions to Software Quality

Forrest Shull, Carolyn Seaman, and Marvin Zelkowitz

On 16 May 2005, the worldwide software engineering community honored Professor Victor R. Basili of the University of Maryland and Fraunhofer Center for Experimental Software Engineering, Maryland, for his achievements in empirical software engineering. At a one-day symposium at the 2005 International Conference on Software Engineering in St. Louis, several speakers gave presentations highlighting Basili's influence on their research. A commemorative book, *Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili* (Springer, 2005), reprinted 20 of Basili's more influential papers.

Basili's contributions cover three broad areas:

- research in the 1970s and early 1980s on software measurement and the Goal Question Metric (GQM) model,
- research in the 1980s and 1990s on these measurement ideas' maturation into a software engineering model of empirical studies, including the development of the Quality Improvement Paradigm (QIP) and the influence of the NASA Goddard Space Flight Center Software Engineering Laboratory, and
- research since 1990 in the Experience Factory as a model for creating learning organizations for continuous software process improvement.

Measurement and the Goal Question Metric Approach

Some of Basili's most important contributions are in measuring software development processes and products. A tireless advocate of software measurement's benefits in industry, Basili gifted the community with an invaluable tool: the GQM approach. GQM made software measurement achievable. It embodies Basili's measurement philosophy, which espouses deriving measures from goals, limiting data collection to what's needed to answer relevant questions, stating assumptions explicitly, and using a clear model for interpreting measurement results.

The GQM approach is based on the assumption that for an organization to measure its products and processes usefully, it must first specify goals for itself and its projects. Once the goals are explicit, the organization must be able to trace them, using a hierarchy of related questions, to the data intended to evaluate those goals operationally. Finally, GQM allows the organization to derive a framework (again using the questions) for interpreting the

 0740-7459/06/$20.00 © 2006 IEEE

data with respect to the stated goals. The result of applying the GQM approach is the specification of a measurement system targeting a particular set of issues and a set of rules for interpreting the measurement data.

This approach has helped many organizations (such as Schlumberger, Motorola, and NASA, to name a few) establish and benefit from rational, effective measurement programs. Practitioners find GQM useful in both guiding improvement initiatives and developing metrics for managing software projects day-to-day. It's also the theoretical basis for the design of many constructs and instruments used in software engineering empirical studies. GQM has become an essential foundational topic in advanced courses in software engineering, research methodology, and software project management.

Empirical studies and the Software Engineering Laboratory

Extending his earlier work on measurement, Basili also developed the QIP, which applies the principles of continuous improvement approaches such as Total Quality Management to software development. Rather than viewing software measurement as an activity of interest to only a single project, the QIP encourages practitioners to understand how measures across projects contribute to larger organizational goals.

The QIP is a cyclical process for planning organizational improvement on the basis of understanding the current baseline, setting quantifiable goals for improvement, choosing development processes to meet those goals, measuring whether the goals are met, and abstracting lessons learned about the conditions that led to success or failure. This process leads to understanding the new environment, which can lead to establishing a new set of goals, and so on. The repeating cycle makes clear that goals can (and should) evolve over time. There's no static baseline; the act of measuring and improving process introduces new contexts that become the basis for future improvement efforts.

Victor R. Basili

The principles of cyclical measurement and improvement and evolutionary goal-setting also provide researchers with a model of how results from different contexts or types of studies can build on one another. For example, pilot studies and academic experiments might be appropriate for an immature technology when a feasibility demonstration is a main concern. Later measurement goals might be more concerned with showing, in more representative contexts, a quantitative improvement due to the technology. Basili's career provides several examples of how empirical studies of many types contribute to (and evolve into) measurable benefits in industrial applications.

The QIP encourages practitioners to understand how measures across projects contribute to larger organizational goals.

Perhaps the most notable example is the Software Engineering Laboratory, a collaboration among university, industry, and government at NASA's Goddard Space Flight Center from 1976 until 2002. The SEL used studies in university courses to test promising technologies in controlled conditions, reducing the risk when introduced in real development laboratories.

The Experience Factory as a learning organization

By the late 1980s, the SEL was making great progress in both improved productivity and decreased error rates. The GQM and QIP concepts first developed in the SEL environment started to coalesce into a model for software process improvement based on a double-feedback loop. The model became known as the Experience Factory (EF). The simple concept, although new to the software development world, is that

- the current project wants to learn about its own development and improve its activities later in the development cycle, and
- the organization itself wants to learn from this experience so that later projects will have the benefits of this earlier experience.

The EF then depends on two organizations: the EF and the Project Organization. The PO (similar to a traditional development organization) designs and develops software, calling on the EF's resources for information, help, and guidance. On the other hand, the EF extracts information the PO provides and installs it into a long-lived experience base. The EF uses this experience base to pass information back from previous projects to the PO and saves the new information for future projects. The process succeeds because each group optimizes its own success criteria—the PO wants to build the

best project and isn't as concerned about future developments, whereas the EF isn't as concerned about the current project but wants to extract relevant information to make its job easier in the future. This enables higher degrees of reuse, better prediction models (of costs, schedules, and defects), and better integration of tools and techniques into the development process. Several companies, such as Motorola and others in Germany and Japan, have applied the EF successfully.

The software business's significant impact on today's economy generates considerable interest in making software development more cost effective and producing higher quality software. Models such as the CMM and the CMM Integrated were developed to measure development process quality. Models such as Basili's EF form an important approach for implementing a plan for better software development performance.

Perhaps the most significant of Basili's contributions to the field is the character of his work. Although a recognition of the importance of continuous improvement has permeated his approach to software, it has also guided his approach to research. Openness to extension, revision, updating, and tailoring has always characterized Basili's approach to his own body of work. His students have always had to accept the evolution of ideas—no great idea is ever finished. Through Basili's continuing work (as well as that of his colleagues, both current and former, spread all over the globe), the GQM, the QIP, and the EF are still evolving to be evermore effective mechanisms for the software industry's advancement.

Forrest Shull is a scientist at the Fraunhofer Center for Experimental Software Engineering, Maryland. Contact him at fshull@fc-md.umd.edu.

Carolyn Seaman is a scientist at the Fraunhofer Center for Experimental Software Engineering, Maryland, and an associate professor of information systems at the University of Maryland, Baltimore County. Contact her at cseaman@umbc.edu.

Marvin Zelkowitz is a chief scientist at the Fraunhofer Center for Experimental Software Engineering, Maryland, and a professor of computer science at the University of Maryland. Contact him at mvz@fc-md.umd.edu.