

Process Diversity and a Computing Old Wives'/Husbands' Tale

Robert L. Glass

Computing, like any other field, has a number of “Old Wives’ (or, more likely, Husbands’) Tales”—stories told so often that the common perception is that they are true, whether they are or not. One of those Old Tales says that one best solution exists for all software problems. Whether that solution is a particular programming language, toolset, operating system, life-cycle model, methodology, or process model, the Old Tale says it is guaranteed to be better than all its alternatives, no matter what the problem is.

That Old Tale was best exemplified by computing’s persistent belief that the worst approach you could possibly take to solving a problem was an ad hoc one. By “ad hoc,” most computing people have meant something that is chaotic and undisciplined; it became a sort of computing dirty word. But a closer look at the dictionary says that “ad hoc” really means “tailored to the problem at hand.” Ad hoc approaches might or might not be chaotic, the dictionary is telling us, but what they are really about is using a best approach determined not by some kind of project-independent thinking, but by some very project-focused thinking.

This issue of *IEEE Software* focuses on process diversity. Although opinions differ about what that might mean, one interpretation is, “All software is not the same, and all software cannot be developed in the same way.” Using that interpretation, this issue flies in the face of that Old Tale.

There’s a Madness to Your Method

One facet of process diversity that has always fascinated me is methodology. For decades now, the popular position has been that a company

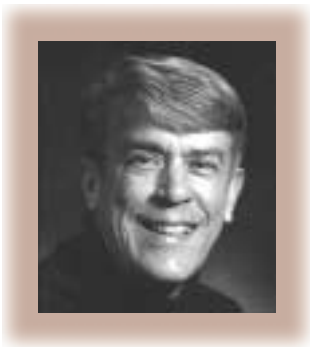
would choose one (best) methodology to develop all its software. Gurus and vendors have spent hours and big bucks defining and promoting their favorite version of such a methodology. Probably the big-name methodologies have been as well known in the software field as the big-name programming languages and the big-name operating systems.

But there is a mounting drumbeat of opposition to the notion of one best methodology. It began with some fairly simple but provocative statements:

- P.J. Plauger, in his Programming on Purpose column in the now-defunct *Computer Language* magazine, said, “If you believe that one size fits all, you are living in a pantyhose commercial.”
- Tom DeMarco, in *Peopleware*, carefully distinguished between capital-M Methodology (a prescriptive way of doing things) and small-m methodology (a collection of methods helpful for the problem at hand).
- Peter DeGrace, in *The Olduvai Imperative*, distinguished between Roman (management-driven, authoritarian) and Greek (technologist-driven, flexible) solution approaches, coming down strongly on the technology side.
- Ed Yourdon, in his journal formerly known as *The American Programmer*, lamented that “One of the sad, sobering consequences of reaching middle age is the discovery that, by and large, it’s all bull—; the best methodologies don’t guarantee the technical success of most systems development projects.”

Provocative editorializing is one thing; research

Continued on p. 127



Continued from p. 128

findings are another. Soon, more substantiated views and studies appeared. Colin Hardy, J. Barrie Thompson, and Helen Edwards reported that 88% of the organizations they examined were tailoring the structured methods to suit the problems they faced.¹ Gerjan Vlasbom, Daan Rijsenbrij, and Matthijs Gastra actually encouraged this methodology tailoring, suggesting some ground rules for doing it.² Joe Nandhakumar and David Avison claimed that the use of a standard methodology was a “necessary fiction” that was maintained to present an “image of control.”³ In fact, the “too mechanistic” methodology could not be—and was not—used in the manner intended. And from P. Middleton, we learned that—at least at the one company studied—the use of the famous, traditional SSADM (Structured Systems Analysis and Design Method) was “flawed” and that there was “little evidence found to support the ... claims made for it.”⁴

Method Engineering Takes Hold

Note that this collection of studies was industry-based. That is, those who were disdaining Methodologies were not the methodologists or the computing theorists, but software practitioners. That of course raised the question, are capital-M Methodologies being resisted and tailored and ignored in practice because the practitioners are stubborn and backward, or because the practitioners know something that the methodologists and theorists don't? That question was partially answered in the April 1996 *Information and Software Technology*, where “method engineering” was the theme, and a new theoretical discipline sprang forth—one devoted to the notion that tailoring methodology to suit the project at hand was a good thing!⁵ “Ad hoc,” it began to appear, was no longer necessarily a dirty word!

This method engineering community gradually grew. Conferences were held, and the notion began to acquire academic credibility. It is now not unusual to find both practitioners and academics singing from the same page of music, the one that says that the “single best methodology” Old Tale is no longer valid.

Dimensions of Diversity

Process diversity, in a variety of

guises—especially this methodology-focused one—has always been a favorite theme of mine. I believe, with my colleague Iris Vessey, that a deep need exists in our field for a taxonomy of application domains. We need this taxonomy because, among other reasons, it could form the basis for a mapping of methodologies to the domains for which they are best suited.

I also believe that software project needs vary dramatically along several dimensions, not just the domain dimension:

- **Size:** Large projects require much more methodological discipline and formality than small ones.
- **Domain:** Business applications are very different from systems programming applications, scientific applications, real-time applications, and edutainment applications.
- **Criticality:** Projects where lives or huge amounts of money are at stake must be treated differently from projects that are less critical.
- **Innovativeness:** Projects that tax the programmer's mind and creativity inevitably will result in methodological restraints being lifted.

Let's Be Real about This

Does anyone still believe that each new methodology is somehow universally better than its predecessors? Is there any evidence, for example, that the object-oriented approaches are vast improvements over their antecedents, the information- and process-focused approaches? Or that event-driven approaches, which largely are newer than the object approaches, are even better still?

Isn't it much more likely that each new methodology comes along because it is better than its predecessors on some kinds

of projects? And shouldn't we—both practitioners and researchers—be spending a great deal of time trying to determine what those kinds of projects are?

Only a little over 10 years have passed since one computing theorist, writing about the use of a variety of techniques on the same diverse project, spoke of an “eclectic” development approach. The leading academic journal that published his work, apparently having no idea what he was trying to say, changed his title on its table of contents to “electric” approaches.

Let me hail the people who have put together this issue's focus on process diversity. Perhaps it will soon be possible to return “ad hoc” to respectability, to add “eclectic” to every computing person's vocabulary, and—most of all—to obliterate that Old Wives'/Husbands' Tale with which I began this column! ☞

References

1. C.J. Hardy, J.B. Thompson, and H.M. Edwards, “The Use, Limitations, and Customization of Structured Systems Development Methods in the United Kingdom,” *Information and Software Technology*, Vol. 37, No. 9, Sept. 1995, pp. 467–477.
2. G. Vlasbom, D. Rijsenbrij, and M. Gastra, “Flexibilization of the Methodology of System Development,” *Information and Software Technology*, Vol. 37, No. 11, Nov. 1995, pp. 595–607.
3. J. Nandhakumar and D. Avison, “The Fiction of Methodological Development: A Field Study of Information Systems Development,” *Information Technology and People*, Vol. 12, No. 2, 1999, pp. 176–191.
4. P. Middleton, “Managing Information System Development in Bureaucracies,” *Information and Software Technology*, Vol. 41, No. 8, 15 June 1999, pp. 473–492.
5. S. Brinkkemper and S. Joosten, eds., special issue on method engineering and metamodeling, *Information and Software Technology*, Vol. 38, No. 4, Apr. 1996.

Copyright and reprint permission: Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Dr., Danvers, MA 01923. For copying, reprint, or republication permission, write to Copyright and Permissions Dept., IEEE Publications Admin., 445 Hoes Ln., Piscataway, NJ 08855-1331.

Circulation: *IEEE Software* (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., PO Box 3014, Los Alamitos, CA 90720-1314; (714) 821-8380; fax (714) 821-4010. IEEE Computer Society headquarters: 1730 Massachusetts Ave. NW, Washington, DC 20036-1903. Annual electronic/paper/combo subscription rates for 2000: \$30/37/48 in addition to any IEEE Computer Society dues, \$57 in addition to any IEEE dues; \$105 for members of other technical organizations. Nonmember subscription rates available on request. Back issues: \$10 for members, \$20 for nonmembers. This magazine is available on microfiche.

Postmaster: Send undelivered copies and address changes to Circulation Dept., *IEEE Software*, PO Box 3014, Los Alamitos, CA 90720-1314. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Product (Canadian Distribution) Sales Agreement Number 0487805. Printed in the USA.