loyal opposition

Editor: Robert L. Glass ■ Computing Trends ■ rglass@indiana.edu

Why Software Process Innovations Are Not Adopted

Stan Rifkin

What do these modern, buzzword software methods have in common: eXtreme Programming, Crystal, lean programming, scrum, feature-driven development, adaptive software development, "good enough" software, personal software process/team software process, Rational Unified Process, rapid development, code complete, ...?

They all offer something in exchange for a change in work habits, and sometimes in exchange for a "culture" change, too. An important reason organizations do not adopt such methods—that is, do not use them in daily software development—is that what the methods offer is not as important (read "cost-effective") as staying with the status quo. This has nothing to do with inertia or resistance to change. Put more strongly, what these methods offer is irrelevant. Put more diplomatically, what they offer is not strategic for the enterprise.

All the buzzword methods offer to reduce cost or effort, duration or time to market, or defects, and most offer to reduce all of these. How can we object? Don't we all agree that those reductions are desirable, even necessary, especially considering our profession's sorry state?

My perspective comes from two disparate directions: I have read nearly everything on adopting process innovations, and I have seen adoption succeed and fail in many software development organizations. What explains the difference between success and failure? The standard answers are about upper-management commitment and sponsorship, the ability or persuasiveness of change agents, the divisibility of the innovation, how disruptive the innovation is, and whether the change is planned and managed. All those are important, but I have seen something that is more powerful and more dominant: alignment with strategy. When the method is aligned with the organization's strategy, the adoption is much more often successful.

What's your strategy?

Sometimes strategy is called *value proposition*, which Jamie Fabian defines as

> *a set of facts, assumptions, and perceptions underlying assessments of what is "valuable" to someone. It's a theory, put forth to see if it is valid. The value proposition is what the seller suggests as the reason(s) why the buyer should select a particular product, service, or process.*

Continued on p. 110

Continued from p. 112

> *On the flip side, what makes someone pick a specific option is the value that a particular individual perceives he/she will derive from that choice.*

(For an annotated list of this and other sources of inspiration for this article, see the sidebar.)

The Discipline of Market Leaders, by Michael Treacy and Fred Wiersema, is a study of 80 high-performance firms. It seeks the secret to such high performance. Treacy and Wiersema find only three sustainable value propositions: *operational excellence*, *product innovativeness*, and *customer intimacy*. To succeed, all organizations must meet a threshold in all three and then excel at only one. The selection and implementation of that one value proposition is the strategy, and everything in the organization must align to it to achieve high performance. If an organization's operations and its strategy do not align, a more efficient, aligned competitor can fill the gap, consequently taking away the first organization's customers.

Finding out what works

The Discipline of Market Leaders can serve as a filter or screen to see if a method will work for your organization. Take, for example, a new method that promises to improve product quality. This aligns with operational excellence, which usually involves being the lowest-cost provider because you offer the highest-quality product in the marketplace. (Highest quality implies lowest price because the producer is not introducing errors and therefore does not have to incur the cost to detect, remove, or rework them.) If your organization does not seek operational excellence, a method that tries to achieve quality by, for example, instilling more discipline will appear antistrategic, and your software practitioners will not adopt it.

Imagine a method that emphasizes creativity. This aligns with product innovativeness, the proposition that your organization alone offers a certain combination of features. Innovativeness is inherently unpredictable; it cannot be usefully planned. So, risk management is the central skill required, and plans per se are not as important. (And we consumers and end users are well aware that when we are working with something new, we are much more tolerant of quality problems. In fact, we exchange our quality standards for innovation.) If your organization seeks product innovativeness, it is a prime candidate for the methods Martin Fowler describes in "Put Your Process on a Diet: Lightweight Methods."

If your organization seeks customer intimacy, its strategy is probably "one-stop shopping"—having an infinite menu. (Many information systems departments and providers fit this category.) Therefore, your organization offers a total solution, everything from a single source, all integrated and harmonious. You focus on the main capabilities you provide: flexibility and the ability to change quickly. So, for example, you need to be able to add a feature, field, or process at a moment's notice and not take long to implement it and not have a ripple effect. Accordingly, your organization should be driven toward methods that support an architecture, an overall picture of the total system that contains high barriers between functions to avoid side-effects.

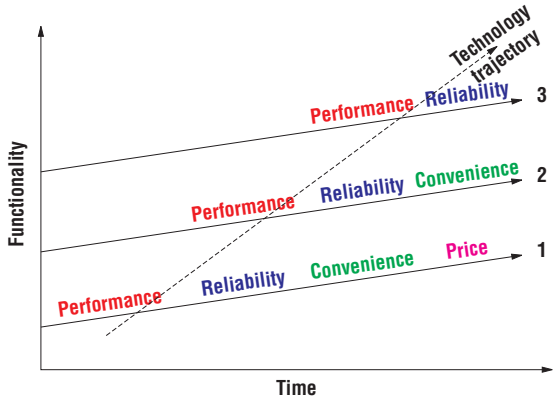

Figure 1. Each parallel line marks a different market segment for the same set of products. Line 3 is the upscale line of products, 2 the midrange, and 1 the downscale. The fast-rising technology trajectory line is the subject of *The Innovator's Dilemma*, by Clayton Christensen. Organizations forming the technology trajectory are surprise competitors to those along the parallel lines. (This figure is based on Figure 8.4 in *The Innovator's Dilemma*.)

Finding out what doesn't

The Discipline of Market Leaders can also help you determine which methods will not work for your organization. If you need to be architecture-centric, you are unlikely to be attracted to a method that permits architecture violations or is even architecture-neutral. If you are innovation-driven, quality needs to be only at the threshold set by the marketplace, which you can find out by benchmarking. Your quality only needs to be good enough. Making it better than the threshold provides no payoff because it is not a differentiator. If you are selling features, your development discipline needs to be only as good as the marketplace tells you, no higher. You would not likely

Information Sources

- Clayton Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston, Mass., 1997. This breakthrough book by an economist illustrates that products, even software ones, have a life cycle and that different value propositions matter during different phases.
- Jamie Fabian provides a good discussion of value proposition at www.jobcircle.com/career/coach/jf_1999_12.html.
- Martin Fowler, "Put Your Process on a Diet: Lightweight Methods," *Software Development*, vol. 8, no. 12, Dec. 2000, pp. 32–36. This article inspired this Loyal Opposition column.
- Mary Poppendieck, "Lean Programming: Assembly-Line Techniques Apply to Software, Too," *Software Development*, vol. 9, no. 5, May 2001, pp. 67–72. An advocate of so-called lean or lightweight methods, as learned from Toyota manufacturing.
- Michael Treacy and Fred Wiersema, *The Discipline of Market Leaders: Choose Your Customers, Narrow Your Focus, Dominate Your Market*, Perseus Books, Reading, Mass., 1995. The clearest explanation of strategy for us lay readers. For a comprehensive discussion of this book, see my article "What Makes Measuring Software So Hard?" in the May/June 2001 *IEEE Software*.

be attracted to a method that imposed a great deal of discipline, because that would put you out of the feature business and into the predictability and quality business.

Triangulation

There is evidence in addition to *The Discipline of Market Leaders* that strategy alignment is critical for implementation success. In *The Innovator's Dilemma*, Clayton Christensen points out that during the typical life cycle of products and services, different factors (performance, reliability, convenience, and price) become differentiators. (For example, see Figure 1, in which each parallel line marks a different market segment for the same set of products. Line 3 is the upscale line of products, 2 the midrange, and 1 the downscale.) When all the competitors offer those factors, they become commodities. So, depending on where you are in the cycle, you will select the method that optimizes the factor that is most important to your customers. You would select first a method that optimizes product performance, then later one that optimizes reliability, then later one that optimizes convenience, then finally one that optimizes cost. Of course, now other considerations creep in: Which software development or management methods provide a path from one factor to the next one in the series? Which methods preclude evolving to another method or tool? Which methods lock you into them so that you cannot easily progress as the cycle demands?

I screen each new buzzword method by evaluating the value proposition it offers. If the method supports my strategy, implementation will likely be easy, all other things equal. If it doesn't, I don't consider it—the cure would be worse than the disease.

Stan Rifkin is a principal with Master Systems, an advisory services firm that specializes in helping organizations for whom computing is strategic. He worked at the Software Engineering Institute on implementing software process improvement and is cochair of the 2002 Software Engineering Process Group Conference. He has a BS in business administration, an MS in computer science, and is completing a doctorate in education. He is a member of the IEEE Computer Society, the ACM, the Foundation for the Empirical Study of Programmers, the Academy of Management, the Institute for Operations Research and Management Science, the Project Management Institute, and the Future Search Network. He is on the editorial board of Empirical Software Engineering. Contact him at sr@master-systems.com.

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. Subscription rates: IEEE Computer Society members get the lowest rates and choice of media option—$40/32/52 US print/electronic/combination; go to http://computer.org/subscribe to order and for more information on other subscription prices. 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.