The Addison-Wesley Signature Series

A KENT BECK SIGNATURE BOOK

IMPLEMENTING LEAN SOFTWARE DEVELOPMENT

FROM CONCEPT TO CASH

MARY AND TOM POPPENDIECK

Forewords by Jeff Sutherland and Kent Beck

Principles of Lean Software Development

Eliminate Waste

The three biggest wastes in software development are:

Extra Features
We need a process that allows us to develop just those 20 percent of the features that give 80 percent of the value.

Churn
If you have requirements churn, you are specifying too early.
If you have test and fix cycles, you are testing too late.

Crossing Boundaries
Organizational boundaries can increase costs by 25 percent or more. They create buffers that slow down response time and interfere with communication.

Build Quality In

If you routinely find defects in your verification process, your process is defective.

Mistake-Proof Code with Test-Driven Development
Write executable specifications instead of requirements.

Stop Building Legacy Code
Legacy code is code that lacks automated unit and acceptance tests.

The Big Bang Is Obsolete
Use continuous integration and nested synchronization.

Create Knowledge

Planning is useful. Learning is essential.

Use the Scientific Method
Teach teams to establish hypotheses, conduct many rapid experiments, create concise documentation, and implement the best alternative.

Standards Exist to Be Challenged and Improved
Embody the current best known practices in standards that are always followed while actively encouraging everyone to challenge and change the standards.

Predictable Performance Is Driven by Feedback
A predictable organization does not guess about the future and call it a plan; it develops the capacity to rapidly respond to the future as it unfolds.

system design, software companies devise all manner of advisory panels and focus groups to discover firsthand what customers really need. Too often IT departments skip the marketing step assuming that someone else has done the market research and are surprised when their products don't meet the business needs.

2. Software companies sell to highly competitive markets, so they have to keep their costs in line. They design products that are simple enough that they can develop and maintain them cost-effectively, while making sure the feature sets provide their customers a compelling reason to buy the software. Too often IT departments assume that a list of business requirements are all essential features of the system, even if very expensive features are involved. They have little incentive to perform the same cost/benefit balancing act that software companies must play to stay in business.

3. Software companies realize that if customers are not successful with their products the company will not have a sustainable business, so it looks for every opportunity to help customers be successful, creating a broader revenue base for themselves while ensuring their products are successful in the marketplace over the long run. Too often IT departments feel that success with the systems they deliver is the responsibility of the business unit. While it is true, as we discuss below, that business units ought to be accountable for the success of software development efforts commissioned by their business, it is also true that the IT department is not successful unless its products contribute to the success of the business.

Many IT departments use the project model for software development, but the project model comes from the contracting industry, where trust is not part of the contract. In order to create healthy IT–business collaborations, we suggest that a product model be adopted, because the incentives built into this model are much more likely to produce a collaborative relationship. When IT is inside a company, there is no reason, really, to set up the we-they model for doing business that projects were designed to deal with. In particular, you do not need to fix scope at the beginning, you do not need customer sign-offs, you do not need to monitor detailed scope to schedule, and you do not need to do everything that your customers say is important. Instead, you need to work in collaboration with your business partner to deliver the most business value in the shortest amount of time for the lowest cost, help them to use the system effectively, and continue to deliver more and better features over time.

Everything Else Failed

I struck up a conversation with the chief information officer of a large financial holding company at a conference. He was an electrical engineer by training, who had been asked to take over IT and "fix it."

"That was well over three years ago, and you have to understand, I failed." He said. "I tried the CMM thing and I tried the PMI thing, and after two years I got frustrated and went back to my old job.

"But after another year they asked me to try again, so this time I did things differently. I split the organization in half: operations and development. Then I divided the development group into six product teams. Each product team has to sell its products to the businesses, and they are measured on how much profit they generate compared to cost.

"It's only been six months, but so far it has been immensely successful. Don't know why I didn't think of it earlier."

—Mary Poppendieck

Accountability

IT departments inside large companies have traditionally been separate organizations from the businesses they support. This is usually because the company desires a standard information infrastructure and because technical expertise can be more easily developed when experts are in the same organization. However, it has led to lack of clarity about who is responsible for the results of software development activities and, quite often, a lack of clarity about how to measure those results.

The problem of accountability is not unique to IT organizations. It occurs any time people on the same team work for different organizations with different ways of measuring performance. For example, an organization developing embedded software might be managed separately from the hardware development department. A consulting firm working for a business clearly has a separate management structure from its client.

It is not particularly effective to subdivide the effort along organizational lines and then have one part of the development team give "requirements" to the other part of the team. This approach has a tendency to obscure the overall business objective of the joint venture, and has a long history of generating suboptimal results. It is far more effective for members of a complete cross-functional team to share responsibility for achieving the business results that justified the funding of their work.

But in the end, there should be a single point of responsibility for the overall business results of an IT investment. We believe that this accountability should rest with the business funding the effort. When business leaders manage IT investments with the same rigor they bring to running their business, these investments will be more likely to produce significant business results.[25]

Try This

1. Use the Kano model to analyze a current development effort. Make a list of features and capabilities—at a high level—and put them on an index card or Post-it Note. Use a large sheet to make the Kano model, and mount it on the wall. Next, attach the cards or notes to the appropriate area of the Kano model. How many features fall in the category of "delighters"? What percentage of features are "delighters"?

2. In Chapter 10 we describe a single number that is a good measurement of customer satisfaction, called a "net promoter score." Compute the net promoter score for your customers. Create a simple survey that asks your customers how likely they are to recommend your product or service to a colleague on a scale of 0–10. What percentage of responses are 9s and 10s? This is your promoter percentage. What percentage of responses are 0–6s? This is your detractor percentage. Subtract the detractor percentage from the promoter percentage. Is the net score positive or negative?

3. Make a list of all the programs or projects that are currently active. For each effort, write down the name of the current leader or leaders. How many leaders does each effort have? Have you written down someone who provides both a market and technical leadership for each effort? Is there any correlation between how well a project is doing and its market/technical leadership (or lack thereof)?

4. Complete Teams: Do you have operations people on your team? Is someone from tech support regularly consulted when designing new features? Are testers involved in development right from the start? When do technical writers get involved? Are customers or customer proxies treated as full team members?

25. See "Who's Accountable for IT?" by Dan Lohmeyer, Sofya Pogreb, and Scott Robinson, *McKinsey Quarterly*, December 7, 2004.

5. Projects / Products: Are development efforts funded incrementally or all at once? What are the success criteria of a development effort? Are teams kept intact or do people regularly move to new teams? Do teams maintain the code they develop? Who is accountable for the business results of the money allocated to software development?

Chapter 4

Waste

Write Less Code

If we were to look for the root cause of waste in software development, a really good candidate would be complexity. Complexity calcifies our code and causes it to turn brittle and break. In *Conquering Complexity in Your Business*, Michael George suggests that complexity is like cholesterol: It clogs up the arteries of an organization; it is "the silent killer of profits and growth.[1] The prescription for complexity in software development is simple: *Write Less Code!*

Let's take a look at how a multibillion Euro company has gone about doing just that.

Zara

When Madonna gave a series of concerts in Spain recently, the outfit she wore at her first concert could be spotted on teens in the audience at her last concert. The Madonna look-alike outfits were purchased at Zara, a rapidly growing fashion clothing chain. Zara's strategy is to be a fast follower, betting that it's more profitable to develop the fashions its customers ask for than to push new designs onto the market. In the rapidly changing world of fashion, a fast follower has to be very fast—Zara can go from concept to cash in two weeks; entirely new lines might take a month.

Zara accounts for perhaps 70 percent of the sales of Inditex, a clothing company located in western Spain with over €5 billion in annual sales. Zara competes fiercely with Stockholm-based H&M, Venice-based Benetton, and San

1. Michael George and Stephen Wilson, *Conquering Complexity in Your Business: How Wal-Mart, Toyota, and Other Top Companies Are Breaking Through the Ceiling on Profits and Growth*, McGraw-Hill, 2004.

Francisco-based Gap and posts some of the healthiest profits in its industry. It has close to 900 stores as of this writing, about three-quarters of them in Europe, and is expanding rapidly.

Instead of hiring high-profile designers to create new products for each selling season, Zara hires the best graduates from design schools and has them work with store managers to create the garments that customers are looking for. Zara launches 11,000 new items a year, compared to its competitors' 2,000 to 4,000.[2] Zara stores are lightly stocked. Orders are sent to headquarters twice a week, and the clothes arrive two or three days later—on hangers, priced, and ready to sell. Zara eschews economies of scale and manufactures in small lots, usually in cooperatives in western Spain. It operates well below capacity so as to be able to maintain its ability to fill orders twice a week even during peak demand.[3]

Zara gets more revenue for its efforts than its competitors: It sells 85 percent of its stock at full price, compared to an industry average of 60 percent to 70 percent. Unsold items account for less than 10 percent of sales compared to an industry average of 17 percent to 20 percent.[4] Zara also spends less money than its competitors: Zara's parent company, Inditex, spends about 0.3 percent of sales on advertising compared to a 3 percent to 4 percent industry average. And it spends perhaps 0.5 percent of sales on IT compared to an industry average of about 2 percent.[5]

Stop. Let's go over that one again. A €5 billion company spends 0.5 percent of sales on IT? Indeed, Inditex's IT staff of about 50 people develops all of its applications. Stores use PDAs and a modem to transmit sales totals and new orders to headquarters. Instead of a CRM system, designers are expected to talk to store managers. And with such low inventories, an ERP system would be overkill. Interestingly, the CEO who orchestrated this minimalist approach is a former IT manager.[6]

2. Data from "Inditex: The Future of Fast Fashion," *The Economist*, June 18, 2005.
3. Kasra Ferdows, Michael A. Lewis, and Jose A.D. Machuca, "Rapid-Fire Fulfillment," *Harvard Business Review*, November 2004.
4. From "Rapid Fire Fulfillment," Ibid.
5. Andrew McAfee, "Do You Have Too Much IT?" *MIT-Sloan Management Review*, Spring 2004.
6. From "Do You Have Too Much IT?" Ibid.