Applying lean to application development and maintenance

Noah B. Kindler, Vasantha Krishnakanthan, and Ranjit Tinaikar

Burdened by high costs for application development and maintenance (ADM),1 many businesses have offshored up to half of their application development to low-cost locations, renegotiated rates on outsourced projects, and tightened the governance of new projects. In spite of these efforts, the costs of developing and maintaining applications now account for about half of the average IT budget and continue to rise. Labor costs make up more than 80 percent of application development, so many organizations have already reduced head counts or labor expenses where possible. Now they must begin to focus on improving the productivity of their development and maintenance staffs. In the past companies tried many methodologies, with mixed results (see sidebar, "Software-development productivity: Traditional methods"). Companies that apply the time-tested principles of lean manufacturing, which hunt for and eliminate waste from the production process (Exhibit 1), are seeing a significant impact within a matter of months.

EXHIBIT 1

Wasteland

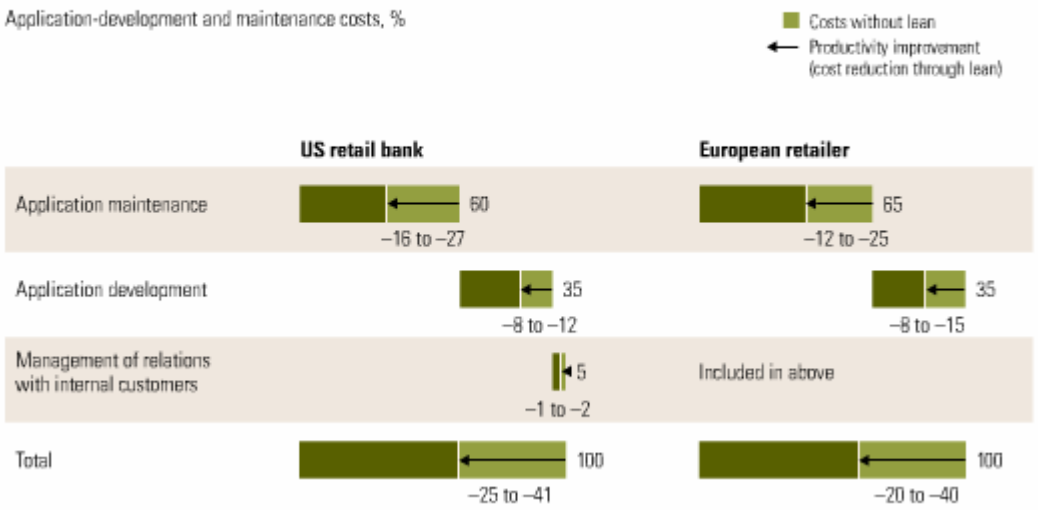
Waste in application development and maintenance

Type	Example
Overproduction/overprocessing	• Fulfillment of requests that won't be used within next 3 months • Unnecessary functionality
Rework	• Changes in business requirements during development • Application bugs
Wasted motion	• Requests not tied to business priorities • Ineffective prioritization of maintenance requests • Unplanned task switching
Wasted intellect	• Limited cross-training of developers across different applications • Poor usage of employees and offshoring resources
Wasted time	• Key resources not available • Developers idling because of incomplete information on the request
Inventory waste	• Maintenance backlogs • Many partially completed requests

Although lean principles were originally developed for manufacturing environments, they are increasingly (and successfully) being applied to service businesses, especially those with many routine processes.2 Application development and maintenance is a prime candidate for lean methods not only because it involves a great many processes with the potential to be optimized but also because large differences in productivity among organizations suggest that some are far less efficient than others. In our experience, applying the principles of lean manufacturing to ADM can increase productivity by 20 to 40 percent (Exhibit 2) while improving the quality and speed of execution.

EXHIBIT 2

Lean with IT



Each category of waste in manufacturing has a counterpart in ADM, which can be thought of as a kind of factory that develops new applications according to business requirements. Changes to an application's requirements are one common source of ADM waste, causing many of the classic varieties identified in lean: designers rework their specifications, coders wait for specifications to stabilize, testers overproduce as their testing environments have to be set up repeatedly, unmet requirements pile up in a large backlog. As in manufacturing, systematically eliminating these sources of waste improves the delivery time, quality, and efficiency of the ADM end product.

Just as each element of waste has a counterpart in ADM, so too does each of the traditional principles for reducing waste:

Flow processing reduces overcapacity or excess inventory by aligning the rhythm of output with the flow of production. A release schedule helps prioritize projects, so it prevents the waste inherent in delays to accommodate new requests.
Load balancing forces the organization to make use of development staff across several locations, as well as outside vendors.
Standardization is common in most application-development organizations but can be more widely applied to reduce the waste that results when requirements are defined in an ad hoc way.
Segmentation of projects by complexity eliminates waste by helping managers to route projects to the proper resources and by avoiding unnecessary overhead for simple tasks.
Quality ownership should extend beyond the testing group to encompass the business (which must increase its discipline in specifying project requirements), the designers (who must build "use cases" fully aligned with business needs), and the coders and testers (staff resources that must be allocated more flexibly). Unfortunately, in most application-development groups end-to-end quality ownership remains fragmented across multiple functional silos and is hardly ever tied to end-to-end performance incentives.
Each of these principles requires not only changes in processes (the technical part of the equation) but also shifts in behavior and new management tools. To improve the way project requirements are communicated, for example, the business must make experts available to IT. But IT's access to these experts often takes a backseat to more pressing priorities. Similarly, with flexible staffing ADM managers must share personnel—a practice that individual managers might resist because they have learned to rely on certain people over the years. Thus, a lean transformation requires simultaneous changes in the technical system (changes to tools, methodologies, standards, and procedures), the behavioral system (convincing staff of the value of these changes), and the management system (new roles, metrics, and incentives to encourage the shift).

A lean transformation demands a substantial investment of time, as well as a sustained focus from upper management and an ongoing revision of incentives and metrics. Implementing the lean philosophy is a continuing and long-term goal that can deliver some results quickly, but it may take years before the approach becomes a core aspect of an organization's culture.

Applying lean

A lean transformation begins with a diagnostic phase that estimates the level of waste in ADM processes. Since most ADM organizations don't track waste, the assessment is based on interviews and questionnaires asking how information (such as the requirements for new applications) and materials (such as the code under development) move through the system. In the parlance of lean, this diagnostic method is called a "materials and information flow analysis." One of its goals is to discern how much time is spent productively and how much is wasted. The wasted time is then examined todiscover the root causes and to determine the opportunity for improving productivity (Exhibit 3).

EXHIBIT 3

Where's the fat?

Stages in application development

	Time spent as % of total time in process	Share of activity in category that does not add value, %	Sources of waste
Initial user request	5	20	• Lack of guidelines, so business user struggles to describe what is wanted
Single point of contact	10	50	• Ambiguous or incomplete maintenance requests that must be clarified and finalized
Technical analysis	15	20	• Need to clarify technical aspects of maintenance requests
Prioritization	5	50	• Frequent reprioritization of projects because of lack of clear rules or canceled requests
Planning and design	10	0	
Building	40	20–25	• Requests for clarification of requirements too late in process • Failure to bundle similar requests
Testing	10	50	• Lack of bundling, resulting in repeated test setups and quality testing for each individual request • Poor quality assurance specifications
Release	5	25	• Release overhead from repeating release setup each week

A large financial institution going through a lean transformation of its application-development department discovered two primary drivers of waste within application maintenance. First, the process for defining project requirements was chaotic and inefficient, involving more back and forth than it does in organizations where processes were more efficient. IT had no standard procedure for obtaining a comprehensive description of the business's requirements for maintenance requests, so developers had to keep going back to the business to clarify requirements—an approach resulting in delays and a lot of rework. Furthermore, there was no clear and effective way to prioritize projects. As businesses requested exceptions (for example, rush jobs), developers had to shift focus from one application to another, and some projects were never completed. The ADM department measured a project's cost and staffing, but not waste, and had no specific goals to improve productivity. There appeared to be little quality ownership and scant incentive for individuals to exert extra effort.

After the diagnostic phase, the financial institution decided to launch a pilot program to learn how to implement the lean philosophy and to create momentum for a broader transformation

of the entire ADM organization. The pilot managers, basing their moves on the diagnostic findings, decided to apply three lean principles that would help reduce waste: a process redesign for improved flow, load-balanced work groups, and end-to-end performance management.

Redesign processes to improve flow

In the pilot program, the team redesigned the application maintenance process to improve the way work flowed through the system. First, they set a schedule of bimonthly releases with clearly defined steps and a fixed capacity based on available resources (that is, designers, coders, and testers). Deadlines for final requirements from the business were clearly spelled out, as were the dates for finishing the development of code and delivering applications. This predictable schedule allowed the business to plan for current and future releases and diminished the tendency to rush late requests into the process.

The team then replaced the ad hoc prioritization practices with a formal process that involved regular meetings between the business and IT. The team also established a more formal set of procedures for handling any exceptions, such as high-priority changes that might come in after deadline.

Balance the load of work groups

This aspect of the pilot solution was based on a more flexible definition of work groups. Developers and testers were cross-trained so that they could work on projects throughout the organization, not just within the group they had focused on. This move allowed managers to use these people more efficiently; for example, when a group was particularly busy, it could borrow developers or testers from another group with available resources. Managers could also tap offshore resources and the staffs of third-party vendors to meet peak demands without hiring additional personnel in the more expensive locations.

Manage performance across the entire process

New metrics to track performance at the group and individual levels focused on measuring and reducing waste. A new management “dashboard”—essentially a spreadsheet that tracked performance and highlighted trouble spots—allowed managers to identify potential problems before they happened. In one case, managers saw that a particular task was taking longer than estimated, so they redistributed that developer’s workload and minimized the disruption. The tracking of individual performance encouraged developers to take on more tasks, since their efforts were now more visible.

The pilot surpassed expectations, boosting productivity in the targeted application maintenance areas by 40 percent in less than two months. Furthermore, IT’s business counterparts were more satisfied with the process, and employee morale was up. As a result of this successful pilot, the company rolled out process redesign, load-balanced work groups, and performance-management systems to the rest of the application maintenance organization over the following year. In addition, the champions of the successful pilot helped to extend it to other parts of IT, including the development of new applications and infrastructure provisioning, further broadening the impact of the initiative.

>>> Priorities of a redesign will differ from one financial industry to another. For one example, see “Rethinking wholesale-banking operations.”

Overcoming stubborn resistance

Applying lean to an application-development organization is a substantial transformation taking two to three years. From discussions with chief information officers (CIOs) and other executives at 30 companies across a range of industries, we have learned that three challenges are the most difficult to overcome: changing behavior, broadening the focus from specifics to general principles, and setting up the right incentives.

Perhaps the most difficult part is changing the behavior and convincing the staff and managers of the value of lean approaches. The key to making the case for change is to demonstrate

positive results in a pilot program, generally undertaken in an area open to change. The financial institution in the earlier example publicized the results of the pilot in a series of company meetings that created support and momentum for a broader initiative. In another organization, management decided to emphasize the importance of the program by appointing a senior executive, who reported directly to the CIO, as the manager of lean transformations. It also assigned specialized teams to help implement lean efforts across the organization.

A common pitfall can be fixating on the specifics of a particular lean implementation rather than the more widely applicable lean principles. At the financial institution, the bimonthly release schedule is one such specific change. Two-month release schedules may not be appropriate for dynamic environments requiring shorter times to market, but the underlying principle—establishing predictability and eliminating delays caused by the ad hoc definition of requirements—still applies. Segmenting projects by complexity and flexibility will help IT determine the appropriate implementation program for achieving alignment with the business.

Finally, no transformation can sustain itself without the proper metrics and incentive systems that ensure change. In application development, “function points” measure the level of effort devoted to a project. A successful lean transformation requires new metrics to identify waste and set goals for reducing it. Leaders must adjust incentive programs, including financial awards and public recognition, to track and reward managers and staff for meeting goals on both measures.

Addressing the barriers to change is no small feat and requires management to sustain a commitment to change. Given the urgent need to improve productivity and the opportunity at hand, a lean transformation is a journey well worth the effort.

Disponível em: <http://www.mckinseyquarterly.com>. Acesso em 3/4/2008.