Agile Logic
Deliver Better Software Faster

Lean Is More

How Lean Methods Can Benefit Your Agile Enterprise

Paul Hodgetts
Founder & CEO – Agile Logic, Inc.
phodgetts@agilelogic.com

Lean Software Development is gaining increasing attention in the software development community. Just like agile methods did nearly eight years ago, lean methods offer new approaches to long-standing challenges in software development. But also just like agile methods, lean methods require us to reconsider how we think about developing software products.

Lean? Agile? Scrum? Extreme Programming? It's easy to get confused with the multitude of methods and practices offered by the new generation of software processes. Given the apparent commonalities between the agile methods, it would be easy, and not entirely incorrect to say that the differences are more in the branding of the strategies and practices than anything fundamental. While sharing some similarities to agile methods, Lean Product Development, however, does have some key differences.

The Roots of Agile and Lean

Agile methods have their roots in the experiences of a segment of the software community during the late 1980s and early 1990s. Many of these people had strong ties to the technical disciplines, and were deeply involved in other innovations of the time, such as object-orientation and the patterns movement. Agile methods emerged as they captured what worked for them on their projects—the practices and core strategies that contributed to successful software deliveries. In response to the key challenges of the time, most of the methods focused primarily on the engineering and testing disciplines, with just enough supporting coverage of project and product management to enable these disciplines to deliver.

Lean methods have a different lineage that dates back before 1950, and traces its roots to one primary source, Toyota. As Japan emerged from the devastation of World War II, the fledgling Toyota Motor Corporation was struggling to develop its ability to produce automobiles with limited available talent and resources. While the full story is beyond the scope of this whitepaper, Toyota's approach was quite different from the agile pioneers. Instead of focusing on specific practices and strategies, Toyota focused more on the culture and overall way of thinking. It is this contrast in approaches that can help us understand where Lean Product Development can complement and enhance Agile Software Development.

Beware of Naïve Lean

Before we explore Lean Product Development further, we first need to consider and avoid one of the most common mistakes when applying lean principles to software development. When the lean concepts were first developed at Toyota, and later captured in the 1990s by the lean process community, the primary focus was on the Toyota Production System, a.k.a. Lean Manufacturing. By far the greatest body of lean knowledge is in the area of Lean Manufacturing, and it's not surprising that much of the initial application of lean concepts to software development has drawn from this area.

Unfortunately, software development is not manufacturing. In manufacturing, the goal is the production of physical parts and products. Time frames are short, the work is linear, tangible and repeatable, and the scope is more constrained with limited communications channels. But this is not the primary paradigm of software development. Treating software development as Lean Manufacturing leads us down a path of optimizing the mechanics of developing software, which can yield limited benefits but fails to address the much larger software product development issues.

Fortunately, a growing body of lean knowledge exists about the other key component of Toyota's success, the Toyota Product Development System. In product development, the goal is the production and management of knowledge. The flow of information within a complex network of participants is vital. Taking place over much longer time frames, with iterative and parallel activities, the product development process is much less tangible than manufacturing. Lean Product Development provides an approach much more in alignment with software development, and its application can yield much greater benefits.

A Holistic Approach

The entire chain of activities that begins with a customer's need, and ends with the delivery of a capability satisfying that need, is often long and complex. It touches all areas of the business, from product management to quality assurance, from business management to operations and delivery, and all points in between. The ultimate goal of a successful software delivery is to optimize this entire chain so that a customer's true needs are satisfied in a way that benefits both the customer and the business.

As previously mentioned, agile methods grew from a focus on the core development activities, programming and testing, and were designed to enable the delivery of software for a wide range of simpler projects. Agile methods were not initially intended to span the full range of product development activities, and to this day have largely assumed that agile projects are wrapped in the necessary business, product and program management activities necessary for overall success. Agile methods provide a narrow, but deep solution to software development. Narrow in the sense of covering a subset of the overall product development lifecycle, but deep in the sense of providing very effective, specific practices for the activities they do address.

To be fair, agile practitioners are actively working to extend agile methods to other areas of the organization. For example, in the Scrum community the core strategy of the "scrum"—a collaborative team working closely to deliver solutions in cycles—is being extended and applied to other areas of the enterprise, including product management and business management. But these approaches have not yet achieved widespread use, and are years away from being mainstream practices.

Lean approaches, on the other hand, take a fundamentally different view of the product development process. Lean focuses on the entire chain of product development activities—the "value stream"—and works to maximize the effectiveness of the whole process. However, lean approaches recognize that all processes are context-dependent, and while industry-specific applications of lean may develop a set of practices that are more commonly applied, it is largely understood that each organization and each team will actively work to find and optimize their best practices. From this perspective, lean methods offer a broad, but shallow solution. They cover the broad scope of the product development lifecycle, but are shallow in the sense of stopping short of prescribing specific development practices.

The different scopes addressed by agile and lean methods provide an important insight into how they can work synergistically together. Where lean methods are extremely effective for analyzing and focusing on the overall software development value stream, agile methods offer successful, targeted strategies and practices for implementing desired capabilities.

The Ultimate Lean Tool

Lean methods offer a variety of tools and practices that implement the key process improvement strategies of lean. From the strategies of single-piece, pull-driven flow to the more specific activities of Kaizen and Poka Yoke, the lean toolkit is stocked with effective, time-proven techniques. We see the influences of these strategies and practices in agile methods, for example, agile retrospectives are similar to the Kaizen strategy. But agile strategies and practices focus much more on the specifics of developing software, rather than on the more general goals of overall process improvement.

Of all the lean tools, none is as vital or important as Value Stream Mapping. The activity of Value Stream Mapping, and the resulting Value Stream Maps (VSMs), provide an extraordinarily effective way to analyze, capture and communicate the flow of a process. VSMs help visualize the process, identify and target waste and bottlenecks, and act as a common language and blueprint for process improvements. Their visualization power is so significant that one of the leading texts on Value Stream Mapping is called "Learning to See."

As with the prior caveat regarding manufacturing vs. product development processes, care must also be taken to use the appropriate tools for mapping a software development process value stream. Manufacturing VSMs are designed to effectively capture the shorter, more serial workflows and limited scope of assembly lines. But to capture the complexity of product development flows, an extended variant of manufacturing VSMs is needed.

Product Development Value Stream Maps (PDVSMs) are specifically designed to capture the unique aspects of knowledge-based workflows. They capture the flow of information between separate, even distributed teams, and can map parallel and iterative activities. They are particularly effective at identifying capacity constraints, hand-offs and scheduling issues, and can help identify the root causes of lead time and cycle time problems.

While a variety of lean tools can be used in support of an agile process transition, Value Stream Mapping is most often recommended as the one tool to start with, and the one to incorporate as an important part of an overall process toolkit.

Lean Is More

If the lean approach is different than the agile approach, how can both be leveraged most effectively as part of an overall agile process strategy? At first it may seem lean and agile are in conflict with each other, or at least represent two very different approaches to process improvement. But in fact lean and agile approaches can both contribute to successful process implementations.

Agile methods were developed by assembling the best practices from successful projects. While the combinations of best practices found in agile methods exhibit the core values and principles of a lean approach, it can be difficult to understand the reasons why agile processes work just from examining their practices. The lean approach, with an almost 10-year head-start on agile methods, has a more highly developed theoretical background. Mapping agile practices to lean concepts such as value, flow, pull and waste can help explain why agile works, and offer new insights to guide process improvements.

Lean Product Development offers several strategies and practices that can supplement agile methods, particularly in areas where agile methods have been criticized as lacking. For example, the strategy of set-based design offers a balanced alternative to heavyweight up-front design approaches on one hand, and reactive, emergent approaches on the other.

While agile approaches offer recipes for process implementations, most agile processes offer little guidance in selecting and tailoring the strategies and practices most appropriate for each project's context. Lean approaches excel at stepping back, taking into consideration the larger context of an agile transition and offering tools and strategies to help guide the transition. Lean methods offer a framework for the transition itself, while agile methods offer a framework for the specific process implementation.

Agile methods offer effective practices for the development and delivery of software systems. Yet many organizations face challenges integrating agile methods into their overall product development lifecycle. Although agile methods offer limited help at the enterprise level, the proven approach of lean product development addresses just these issues. Combining the principles and practices of lean with those of agile give us a new, richer toolkit for expanding agility to the enterprise.

References

For More on Lean Development:

The Toyota Product Development System, James Morgan, Jeffrey Liker

Learning to See: Value Stream Mapping to Create Value and Eliminate Muda, Jim Womack, Dan Jones

Lean Software Development: An Agile Toolkit, Mary Poppendieck, Tom Poppendieck

Implementing Lean Software Development, Mary Poppendieck, Tom Poppendieck

Lean Thinking, James Womack, Dan Jones

For More on Agile Processes:

Agile Software Development with Scrum, Ken Schwaber and Mike Beedle

Agile Project Management with Scrum, Ken Schwaber

Extreme Programming Explained, First Edition and Second Edition, Kent Beck

Crystal Clear: A Human-Powered Methodology for Small Teams, Alistair Cockburn

DSDM: Business Focused Development, DSDM Consortium, Jennifer Stapleton

A Practical Guide to Feature-Driven Development, Stephen Palmer, John Felsing

Agile and Iterative Software Development, Craig Larman

Agile Software Development, Alistair Cockburn