

PRODUCT DEVELOPMENT

The Agile brand has more or less won the methodology wars. However, much of what has been implemented is *faux* Agile—people following some of the common practices while failing to address wider organizational culture and processes. For example, in larger companies it's still common to see months spent on budgeting, analysis, and requirements-gathering before work starts; to see work batched into big projects with infrequent releases; and for customer feedback to be treated as an afterthought. In contrast, both Lean product development and the Lean startup movement emphasize testing your product's design and business model by performing user research frequently, from the very beginning of the product lifecycle.

Eric Ries' book *The Lean Startup* (Ries 2011) created a surge of interest in lightweight approaches to exploring new business models and product ideas in conditions of uncertainty. Ries' work is a synthesis of ideas from the Lean movement, design thinking, and the work of entrepreneur Steve Blank (Blank 2013), which emphasizes the importance of taking an experimental approach to product development. This approach, based on our research, includes building and validating prototypes from the beginning,

working in small batches, and evolving or “pivoting” products and the business models behind them early and often.

We wanted to test whether these practices have a direct impact on organizational performance, measured in terms of productivity, market share, and profitability.

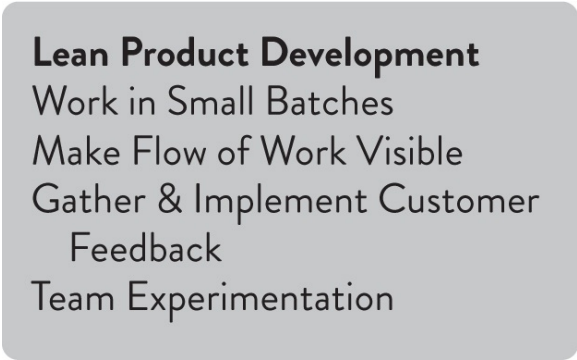
LEAN PRODUCT DEVELOPMENT PRACTICES

We examined four capabilities which make up our model of a Lean approach to product development (see also Figure 8.1).

1. The extent to which teams slice up products and features into small batches that can be completed in less than a week and released frequently, including the use of MVPs (minimum viable products).
2. Whether teams have a good understanding of the flow of work from the business all the way through to customers, and whether they have visibility into this flow, including the status of products and features.
3. Whether organizations actively and regularly seek customer feedback and incorporate this feedback into the design of their products.
4. Whether development teams have the authority to create and change specifications as part of the development process without requiring approval.

Analysis showed that these factors were statistically significant in predicting higher software delivery performance and

organizational performance, as well as improving organizational culture and decreasing burnout. By conducting our research over multiple years, we also found that software delivery performance predicts Lean product management practices. This reciprocal relationship, suggested by the literature, forms what is known as a virtuous cycle. Improving your software delivery effectiveness will improve your ability to work in small batches and incorporate customer feedback along the way.



Lean Product Development
Work in Small Batches
Make Flow of Work Visible
Gather & Implement Customer
Feedback
Team Experimentation

Figure 8.1: Components of Lean Product Management

Working in Small Batches

The key to working in small batches is to have work decomposed into features that allow for rapid development, instead of complex features developed on branches and released infrequently. This idea can be applied at both the feature and the product level. An MVP is a prototype of a product with just enough features to enable validated learning about the product and its business model. Working in small batches enables short lead times and faster feedback loops.

In software organizations, the capability to work and deliver in small batches is especially important because it allows you to

gather user feedback quickly using techniques such as A/B testing. It's worth noting that an experimental approach to product development is highly correlated with the technical practices that contribute to continuous delivery.

Gathering customer feedback includes multiple practices: regularly collecting customer satisfaction metrics, actively seeking customer insights on the quality of products and features, and using this feedback to inform the design of products and features. The extent to which teams actually have the authority to respond to this feedback also turns out to be important.

TEAM EXPERIMENTATION

Many development teams working in organizations that claim to be Agile are nonetheless obliged to follow requirements created by different teams. This restriction can create some real problems and can result in products that don't actually delight and engage customers and won't deliver the expected business results.

One of the points of Agile development is to seek input from customers throughout the development process, including early stages. This allows the development team to gather important information, which then informs the next stages of development. But if a development team isn't allowed, without authorization from some outside body, to change requirements or specifications in response to what they discover, their ability to innovate is sharply inhibited.

Our analysis showed that the ability of teams to try out new ideas and create and update specifications during the development process, without requiring the approval of people outside the team, is an important factor in predicting organizational performance as measured in terms of profitability, productivity, and market share.

We're not proposing that you set your developers free to work on whatever ideas they like. To be effective, experimentation should be combined with the other capabilities we measure here: working in small batches, making the flow of work through the delivery process visible to everyone, and incorporating customer feedback into the design of products. This ensures that your teams are making well-reasoned, informed choices about the design, development, and delivery of work, and changing it based on feedback. This also ensures that the informed decisions they make are communicated throughout the organization. That increases the probability that the ideas and features they build will deliver delight to customers and add value to the organization.

EFFECTIVE PRODUCT MANAGEMENT DRIVES PERFORMANCE

We conducted our analysis of Lean product management capabilities over two years, from 2016-2017. In our first model, we saw that Lean product management practices positively impact software delivery performance, stimulate a generative culture, and decrease burnout.

In the following year, we flipped the model and confirmed that software delivery performance drives Lean product management practices. Improving your software delivery capability enables working in small batches and performing user research along the way, leading to better products. If we combine the models across years, it becomes a reciprocal model or, colloquially, a virtuous cycle. We also found that Lean product management practices predict organizational performance, measured in terms of productivity, profitability, and market share. The virtuous cycle of increased delivery performance and Lean product management practices drives better outcomes for your organization (see Figure 8.2).

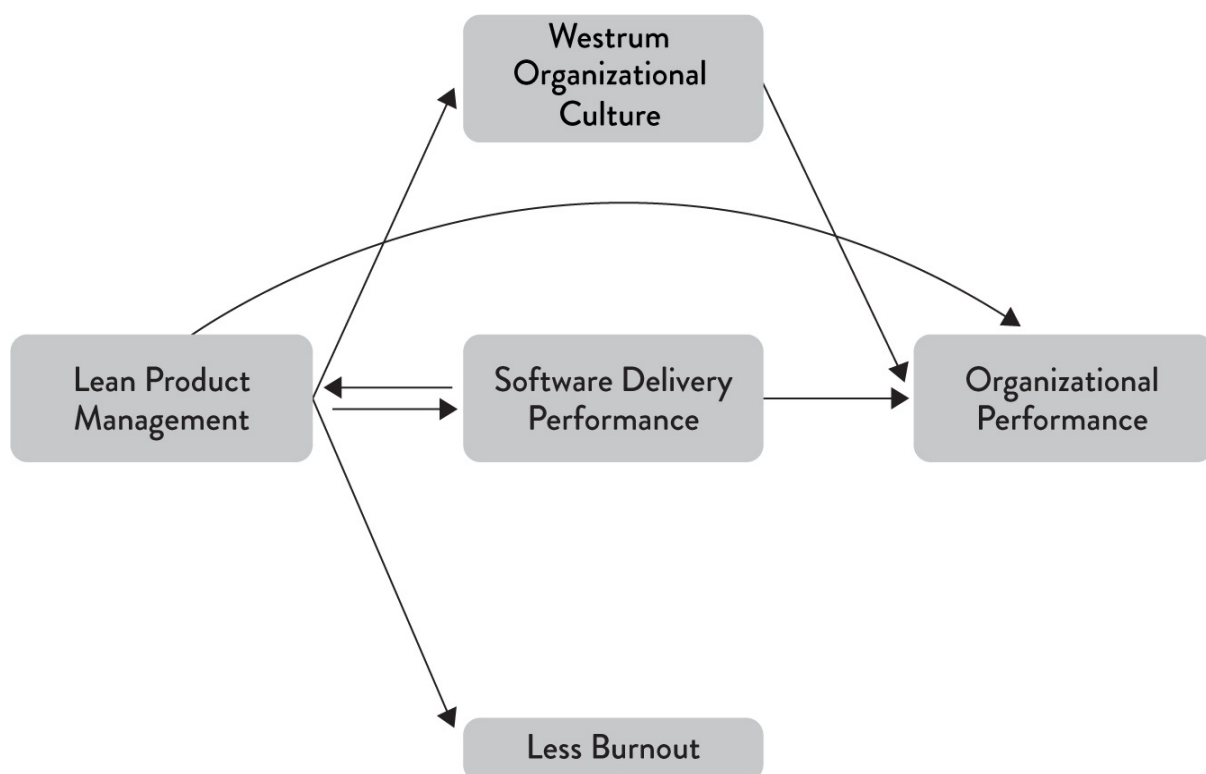


Figure 8.2: Impacts of Lean Product Management

In software organizations, the ability to work and deliver in

small batches is especially important because it enables teams to integrate user research into product development and delivery. Furthermore, the ability to take an experimental approach to product development is highly correlated with the technical practices that contribute to continuous delivery.