

Making Your "Product Development Factory" Lean



The term *lean* was first explained to the world by Jim Womack, Dan Jones, and Daniel Roos in the book "The Machine that Changed the World". Two decades ago this 5-year, landmark MIT study of the automobile industry found that Japanese manufacturing companies were typically much more productive and efficient than their Western counterparts.

Several years earlier, Taiichi Ohno had published the "Toyota Production System" which explained the main foundations of "lean" manufacturing that guided top companies in Japan. Ohno originated 7 categories of waste whose elimination has been the basis of lean improvement since. Simply put, waste is the use of more resource than what's needed to produce the product as defined by the customer (i.e. the customer won't pay for it). Ohno's 7 wastes:

1. Overproduction (producing beyond orders)
2. Motion (movement of equipment or people that add no value to the product)
3. Transportation (handling more than once, delays in moving, unnecessary moving/handling)
4. Waiting (time when no value is added to the product)
5. Processing (work carried out on the product that adds no value)
6. Inventory (unnecessary raw material, work in process, finished stock)
7. Defects (bad units or scrap)

Some of these have a good analogy in product development, some do not. The key is to have an understanding of key differences between product development and manufacturing. Here are a few of the important distinctions:

- Manufacturing improvement is typically about eliminating variability - without some variability in product development there is no innovation
- It is mostly information, not physical objects, that flows through a "product development factory"
- Important information is often unknowable at the beginning of the process
- The value-add content created in product development is different than in manufacturing so performance needs to be assessed differently

What wastes exist in your product development system?

Here are a few questions to kick off your thinking about where waste may exist in your "product development factory".

Are you overproducing work? Does your process force you to produce too much information such as frequent, detailed status reports or documentation that doesn't add value? Might you over-design because you aren't sure what's needed when early choices are made? Does some work expand to fill time? Do you keep polishing beyond what a customer will pay for?

Are you wasting time with work that sits in queues? Do you have organizational bottlenecks? Do you partially staff work to get started only to have it sit and wait for other resources to free up? Does your work flow unimpeded through your product development system?

Do you waste time and energy having to learn the same things over and over again? Do you get all the leverage you can from experience gained throughout your organization? Are you learning from the best practices in the industry?

Do you waste time and money on late changes? Does your risk management capability give you the visibility to address uncertainties as early as possible? Are you designing with sufficient design margin? Are you getting all the parallelism you can by verifying the quality of your work as early as possible?

Have you architected your product designs to make it easy for work to flow? Have you allowed for some decisions to be made late when more information is available? Do you consistently evaluate make/buy design choices and minimize work on non value-add activities?

Do your product developers have to expend energy fighting inadequate tools or overly rigid processes and systems to get things done? Alternatively, do they have to invent processes too frequently where none exist? Can you distinguish between "good variability" required to innovate and variability that has nothing to do with value-add and should be eliminated?

Are your people getting paid to do what they do best every day or are you limited by the talent you have? Are your managers spending their time doing the technical work or fighting 'administrivia'? Or do they take the time and have the talent to lead people in an engaging way that maximizes performance?

Most product development organizations will answer many of these questions in a way that suggests a potential opportunity to get much more lean - and in doing so improve both top and bottom line business performance. Just as we've seen lean help revolutionize manufacturing, there may be a tremendous business opportunity in eliminating waste and creating higher value-add in product development.

Summary

There are sound approaches you can take to look for opportunity inside the "product development factory". Implicit in the questions above are a combination of people related challenges, challenges with the work itself, and potential system and process impediments. It is important to systematically look at what gates your performance leveraging a methodology like our [WhoWhatHow™](#) framework. No matter what method you use, some useful rocks to turn over include examining where work queues up, how the work flows, how serial or parallel key activities are, how much work adds little value, what variability is innovation related versus waste, how much overhead is imposed by management demands and process compliance, how much re-learning there is, how much leverage occurs between different groups, and how risks and uncertainties are dealt with.

By default, many firms settle for 'waste-full' product development because they don't know how to maximize their opportunity in an environment where creativity and innovation need to co-exist with execution discipline. They intuitively realize that approaching product development just like manufacturing won't work. On the other hand, by applying proven methodologies that account for inherent differences in the "product development factory" you can open the door to improved competitive differentiation and develop much better business leverage from your R&D investments. There is both opportunity and benefit to getting leaner - but how you go about it makes all the difference.