October 11, 2010
Volume 9, Issue 41

Lean Thoughts

# GOOD ENOUGH NEVER IS

## (OR IS IT?)

***As Lean Practitioners, we are in the constant pursuit of the perfect process. This pursuit typically revolves around existing product designs. Of course my good friend Mike Shipulski constantly challenges me that he can do more to reduce cost through design and typically he focuses on the consolidation of pieces and reduction of parts.***

***Recently a good friend shared a link to a blog posting about a different perspective of product design through the eyes of a software developer and being a start-up company. It has a different perspective about product design, but yet lessons that we all tangle with regularly with our products and processes. The blog post is extremely lengthy … so I have done my best to edit out the best lessons to learn.***

One of the sayings I hear from talented managers in product development is, "good enough never is." It's inspirational, always calling the team to try harder and do better. It works to undermine excuses for poor or shoddy work. Especially in teams that are managing by objectives (or OKR's), the pressure to deliver is intense. Under such pressure, the temptation to cut corners, to quit prematurely, or to hand off shoddy work to another department is overwhelming.
On the other hand, there are many stories of companies achieving a breakthrough by shipping something that was only "good enough." One such rumor, which I've heard from several sources, tells of the launch of Google Maps. The Leaders were impressed, even though the team considered it still an early prototype. Larry and Sergey, so the legend goes, simply said: "it is already good enough. Ship it." The team complied, despite their reservations and fear. And the rest is history: Google Maps was a huge success. This success was aided by the fact that it did just one thing extremely well – its lack of extra features emphasized its differentiation. Shipping sooner accentuated this difference, and it took competitors a long time to catch up.

This is precisely the dilemma that the doctrine of minimum viable product is designed to solve. And it's really hard.

Most of us intuitively have a "split the difference" attitude when faced with recurring difficult choices. That is not a long-term solution. The reason: it actively encourages factional strife. Everyone naturally falls along a spectrum, from "ship anything soonest" to "always build it right, no matter what it takes." When members of a team realize that the final answer will be some kind of average, they face an overwhelming incentive to express desires in the strongest possible terms. After all, someone else's view will be averaged in, too. Any excesses are likely to be moderated by others. Of course, this logic applies to members of all factions. Over time, such teams either explode due to irreconcilable differences or dramatically slow down. The latter is actually more dangerous. Divided teams usually can't agree on facts *or* interpretations.

But action/paralysis are not the only options. As in many false dichotomies, we can find a third way that gives both factions a positive message to rally around.

Which takes us right back to the original definition of minimum viable product: the minimum viable product is that version of a new product which allows a team to collect the maximum amount of validated learning with the least effort.

In other words, the minimum viable product is a test of a specific set of hypotheses, with a goal of proving or disproving them as quickly as possible. An important of these hypotheses is always: what will the customer care about? How will they define quality?

**Where Lean Thoughts can become Reality**

*"Unless you try to do something beyond what you have already mastered, you will never grow."*

*Ronald. E. Osborn*

KUNST Solutions Corp.

***Richard Kunst***
President and CEO
*Tel: 519 651 2341 E-mail: rkunst@kunstartofsolutions.com Web: www.kunstartofsolutions.com*

Lean Thoughts

# PIVOT OR PERSEVERE ?

Minimum viable product is an attempt to get startups to simplify, but it is not itself simple. How do you know which features are essential and which should go? There is no formula, it requires judgment. Any scientific method requires the choice of a hypothesis to test. This leads to two questions:
By what standard is this hypothesis to be chosen? Minimum viable product proposes a clear standard: the hypothesis that seems likely to lead to the maximum amount of validated learning. How do you train your judgment to get better over time? Again, the answer is derived from the hard-won wisdom of the scientific method: making specific, concrete predictions and then testing them via experiments that are supposed to match those predictions helps scientists train their intuition towards the truth.

(Fans of the history of science will recognize this as Thomas Kuhn 's theory of scientific paradigms. Minimum viable products are not a single hypothesis. They should therefore be properly understood as product paradigms. As in science, the paradigms that survive will be those that allow practitioners to discover the most productive experiments to try, during the period Kuhn calls "normal science." A paradigm crisis is analogous to a pivot.)

What do you do the day after you just did it? It really doesn't matter if you took a long time to build it right or just threw the first iteration over the wall. Unless you achieve instantaneous overnight success, you will be faced by difficult decisions. Pivot or persevere? Add features or remove them? Charge money or give it away for free?

So what about the question of whether good enough really is? What's needed, I believe, is an alternative discipline that teams can get excited about. When we're talking about being disciplined, following our methodology with rigor, continuous improvement, there is no such thing as good enough. Our pursuit of learning is ongoing and our commitment is absolute. But when it comes to the specific of a product release, business plan, or marketing launch, all that matters is: do we have a strong hypothesis that will enable us to learn? If so, execute, iterate, and learn. We don't need the best possible hypothesis. We don't need the best possible plan. We need to get through the build-measure-learn feedback loop with maximum speed.

Over time, I believe we will build a new professional discipline that will seek excellence at this kind of product-centric learning. And then that new breed of managers will, I'm sure, confidently go around saying: good enough never is.

There are many stories of companies achieving a breakthrough by shipping something that was only "good enough." One such rumor, which I've heard from several sources, tells of the launch of Google Maps. The team was demoing their AJAX-powered map solution, the first of its kind, to senior management at Google. They were impressed, even though the team considered it still an early prototype. Larry and Sergey, so the legend goes, simply said: "it is already good enough. Ship it." The team complied, despite their reservations and fear. And the rest is history: Google Maps was a huge success. This success was aided by the fact that it did just one thing extremely well – its lack of extra features emphasized its differentiation. Shipping sooner accentuated this difference, and it took competitors a long time to catch up.

KUNST Solutions Corp.

*Richard Kunst*
President and CEO
*Tel: 519 651 2341 E-mail: rkunst@kunstartofsolutions.com Web: www.kunstartofsolutions.com*