

# Frustrated By Development Bottlenecks?

## Have You Considered Asking for LESS?

---

As a product leader, your success is dependant not only on your ability to drive the right products into the pipeline but also on the rest of the organization to be able to build and support them. Sometimes (or perhaps *often* in your case) no matter how hard you push, nothing seems to happen due to bottlenecks in your organization. This can be a high source of frustration for the product management team (and other executives) and the cause of much conflict in your organization (this I know personally). So what can you do about it?

The first seemingly logical place to start is to try to address the bottlenecks directly. It usually means throwing more resources or money at the problem activity to try to improve its capacity. Sometimes it works. Sometimes (nearly always) you don't really have the money to throw at the problem. Sometimes, you can't make it significantly better just due to the complexity of the problem. Sometimes you can fix a specific activity, but the bottleneck moves downstream to a new activity and no overall gain is achieved. And sometimes, because the activity is outside of your functional area of control, there's actually nothing you can do about it without an internal war breaking out.

So, the next seemingly logical step is to try to bypass the bottleneck. This is where the company (or maybe Product Management unilaterally) attempts to move the bottleneck activities outside the organization to a more cost-effective means for improving capacity, such as outsourcing. This has proven to be workable at the Testing or Support activities, as these are towards the end of the process and not necessarily core to the company. However, as you move up the value chain into Development, things get more complicated.

Employing this strategy is most successful if the new product is totally standalone and is radically different than the products produced in-house, such as if attacking a lower end market segment or attempting unfamiliar technology. It can also work for existing or similar new products outsourced by the Development team directly so they can maintain control over architecture and internal standards. In order for this to work effectively, the Testing function for the new product must also be moved outside to avoid creating another bottleneck internally.

Another variation on this theme is to license the finished functionality from a supplier. This should actually always be a consideration for any non-critical functionality and to focus your internal resources on your core competencies. However, the closer the purchased capabilities get to your core value proposition and differentiation, the less likely the strategy will work as your value-add will be significantly reduced. And with outsourcing or licensing you may find yourself in months of contract negotiations, and/or locked into a partnership you don't really like, and/or the purchaser of exaggerated promises from the external party.

So is there a better way? One rarely-used method to fix bottlenecks is to REDUCE the amount of input into a process so as to match its capabilities. This can be accomplished by reducing the number of features being requested to be built.

Now I can hear the screams and protests already coming from the product management team (and even the CEOs). “We can’t even deliver the huge backlog of needed features now, and you’re saying we have to reduce that further? Blasphemy! You call yourself a Product Manager!”

The reason you should consider this is because it is the most **LIKELY TO SUCCEED** due to:

- a) It does not require changing the development process
- b) It reduces implementation risk by *not* introducing new unknowns
- c) It does not require new investment or resources

It also puts control of the success equation back into the hands of Product Management, not leaving it to the mercy of a downstream process. The normal planning scenario pushes a desired feature set into the Development process and arbitrary pieces fall prey to the bottlenecks such that you don’t have a good idea as to what will actually emerge from the other side. Towards the end of the project you are always playing the Feature versus Schedule trade-off, with Features often losing.

One way to try to beat this game is to put more into the request than we think we’ll get out, such as by elevating priorities of features in the requirements doc even though they truly cannot all be Priority 1’s. This gives us the illusion that we’re improving our odds that what emerges from the other side will be enough to put into the marketplace. We also want our product to be feature rich to create a (seemingly) more powerful message to the marketplace, so we try to heap as much as possible into the message. In both cases, we’re fooling ourselves.

*The real problem is we as Product Managers are playing it too safe.* We have a zillion feature requests from important customers and we want to look responsive and capable. Our reaction is to push features that are merely “annoyances” felt by customers hoping that if we push enough of them they will sum up to something really valuable. Unfortunately, stuffing a velvet bag full of cut-glass jewelry is nowhere near the value of a single diamond ring.

On the other extreme, we may be driving a capability that does have real potential in the marketplace but we don’t define the **MINIMUM** possible feature set that creates the core value proposition (and messaging). We just can’t seem to accept the concept of **GOOD ENOUGH**. We wrap the offer in lots of fluff that we’re convinced is needed now and make it too big for Development to bite off. It thus never gets tackled or the fundamental capabilities never make it through the pipeline and we put a lackluster offering into the market. We’re at fault for not being able or willing to truly prioritize what’s important, yet we tend to blame the Development organization for what we view as their problem in delivering it.

It is this shotgun approach that can create large feature backlogs as we attempt to push too much through our limited resource organizations, primarily the Development team. One response from Development is to “Go Agile” to control the incoming pipeline in a more methodical manner. This only makes the feature set more bite-sized, not necessarily better. But it is also likely that our own Product Management team is spending way too much time and effort on activities associated with inconsequential or too-big feature sets. Our organizations are stuck in delivering low value/low impact features for subsequent low (or no) gain in the marketplace.

To embrace a new **LESS BUT WITH IMPACT** approach requires changes in how we function as Product Managers.

First, we need to be market experts for our product and discern what is truly compelling and what is merely annoying that our customers can live with. We need to be out in the market, learning our customer's business and seeing how they use our products. We need to find out real opportunities and create innovative solutions to real problems. We can't do this by sitting in a conference room and having intellectual discussions internally with others who are just as market unaware as us.

Second, we need to learn to say "NO" to the majority of feature requests coming in the door. If they are not increasing or adding to the core value proposition or differentiation of the product, they just don't matter to our business in the long run. We're wasting valuable time and effort in our organization even discussing them.

Finally, we need to be bold and focus on defining the minimum possible feature sets required and delivering them in the best possible solution for the customer. Instead of shooting for the moon and hoping our organization can deliver it, let's focus them on ascending the next summit with a route that's achievable and everyone believes in.

If you're frustrated in your ability to drive your product vision through your organization, realize that you actually do have the keys to the car. It's just smaller and more focused than you were probably thinking. Maximize your existing resources by asking for LESS but containing more IMPACT.

For more information, see [www.product-arts.com](http://www.product-arts.com) or send email to [info@product-arts.com](mailto:info@product-arts.com).