



Strategic Digital Product Management

Jan Bosch



STRATEGIC DIGITAL PRODUCT MANAGEMENT

Engineering is concerned with building systems. We look at the requirements, design an architecture, do detailed design, implement, test and release. This may be a bespoke system for a specific customer, a product or a platform used to build other products with.

Whereas engineering is concerned with how to build systems, there's another activity concerned with what to build in the first place. And, perhaps more importantly, why we should build the system in the first place. This activity is typically referred to as product management.

Defining the requirements for a system is the output of the product management activity, but a whole lot of work happens before that. This includes research to understand the market, interacting with customers, determining the way to release the system, what to charge for it, and so on. Because of these many facets, activities and responsibilities, product management is often viewed as fuzzy and difficult to understand.

In this series, we'll dive into the world of product management and see if we can bring some structure to the topic. The intent is to identify the key activities, tools, mechanisms and approaches that are available to us, when to use these and to understand how product management has changed with the emergence of digitalization. Although I've worked on many techniques that could be considered part of product management, I've never worked as a product manager. So, I hope that you'll let me know where I went wrong as well as what I missed.

In my view, one way of deconstructing product management is to discern five main areas: principles, scopes, activities, factors and roles. These will be discussed here as well as in the coming posts.

First, there are a number of principles underlying product management. Although we can identify several, we'll focus on three. The primary principle is to maximize the return on investment on the R&D efforts that we put in. Second, we need to embrace uncertainty. Third, we need to recognize and manage tradeoffs, between different scopes, between stakeholders, between time scales, and so on.

The second dimension is the scopes at which a product manager operates. On the one end of the spectrum, there's the individual feature, whereas at the other end, we seek to engage with the entire

business ecosystem. In between, we may find components, subsystems, products, portfolios, business areas and the company as a whole. In the upcoming posts, we'll focus on three scopes: features, product and ecosystem.

The third dimension is the activities of a product manager or the activities conducted as part of product management by anyone in the company. Although these can be structured in many different ways, the way we'll explore this in this series is to divide product management activities into exploration, strategy and execution. Traditional companies separate corporate strategy from product management. For digital companies, however, the technology drivers of software, data and AI are so fundamental for the business that strategy becomes an interactive, bi-directional process between technology and R&D leadership and the business leadership.

The fourth dimension is the factors or viewpoints that product management is concerned with. There are many that can be considered, but we focus on three: stakeholders, technology and exploration. First, product management involves a wide variety of stakeholders, both inside and outside the company. Second, although many view technology as part of the R&D realm, in practice product management tends to be involved in high-level strategic choices. Third, as product management is so concerned with maximizing ROI, it's tempting to focus on improving existing functionality that's already proven to deliver value to customers. However, we need to sufficiently invest in the exploration of new areas of functionality.

The final dimension concerns roles and responsibilities. In many traditional organizations, product managers have significant responsibility but surprisingly little authority. The roles and responsibilities dimension seeks to answer the question of who takes care of the various product management activities, for what scopes and from what viewpoints. And, of course, what alignment mechanisms are used to accomplish this. In the upcoming posts, we explore three main responsibilities: focusing on the vital few rather than the worthwhile many, organizing product management and balancing qualitative and quantitative data and insights.

Whereas engineering is focused on how to build a system, product management is concerned with what to build and why. In a digitalizing world, the role of product management is changing in that new technologies such as software, data and AI allow for much faster feedback cycles and, consequently, the use of experimentation to reduce uncertainty and significantly increase the ROI on R&D investments. To accomplish this, we need to clearly define the principles, scopes, activities, factors and roles and responsibilities of product management. As Marissa Mayer said, "Product management really is the fusion between technology, what engineers do – and the business side."

TABLE OF CONTENTS

Strategic digital product management.....	2
Maximize ROI.....	5
Embrace uncertainty	7
Trade-offs.....	10
Features	12
Product	14
Ecosystem	16
Exploration.....	19
Strategy	22
Execution	24
Stakeholders	26
Technology.....	29
Imaginate.....	31
Vital few.....	34
Organizing.....	36
Qualitative and quantitative data.....	38
Conclusion	41



MAXIMIZE ROI

One of the most obvious principles of product management, and consequently the first one we discuss here, is to maximize the return on investment on the R&D efforts that we put in. As we're a for-profit business, the goal has to be that we seek to maximize profit. This requires us to prioritize those R&D activities that are most conducive to accomplishing that goal. In most companies, the R&D budget is allocated as a percentage of revenue and the intent is to get a return on investment. Prioritizing R&D efforts that are conducive to that goal is central to product management.

This is so obvious that it's almost painful to write. So, why do I do so anyway? My experience with many companies is that product management in many ways becomes a victim of a host of different obligations, need-to-do's and must-haves to the point that there's hardly any space for new functionality that has at least some potential to move the needle from a revenue perspective.

There are three main reasons why companies fall into the trap of prioritizing R&D efforts that don't support this first principle of product management: customization, technical debt and lack of automation, especially around testing. We dive a little deeper into each of these.

First, customization is a very natural pattern in many companies. Discussions with a new potential customer are proceeding well and we can close the deal if we're willing to implement some adjustments or customer-specific features. Company management gives in and asks R&D to do so. Typically, this is resolved by branching the code base and providing a customer-specific version of the offering.

The initial investment in creating this customer-specific version is often quite limited. The challenge is that with an increasing number of customers all asking for their own customization, the number of branches quickly goes up. In addition, customers tend to request additional customizations over time and as we agreed in the beginning, they expect us to continue to agree. So, what started as a small branch in a small part of the code base grows over time.

The consequence rapidly becomes that any cross-cutting change requires changes in all branches. As there are customizations, this isn't simply a matter of copying but calls for careful checking of the implications of the change in each specific branch. Any defects that are found require the same treatment. In the end, it's not that it's impossible to run the business in this way, but rather that it's extremely effort-consuming, causing less and less time to be available for value-adding new functionality.

The second area of concern is technical debt. Especially on the business side of companies, there's little understanding of the need to invest in managing technical debt. This may easily result in a situation where refactoring and paying off technical debt is always pushed off to a later point in time and all resources are allocated to building new functionality and fixing customer issues.

As we now all know, when technical debt becomes high, the cost of adding new functionality and fixing defects increases as the complexity of the software increases. When taking this to the extreme, it becomes impossible to add any new functionality as the return on investment is simply too low.

Third, despite all the talk and publications on continuous integration and test, I still meet surprisingly many companies where test automation is severely lacking. Often, some elements are in place, but these aren't integrated to provide an end-to-end validation cycle. The challenge in this case is twofold. First, significant resources are allocated to manual testing of functionality and these resources aren't available to build value-adding functionality. The second problem is that companies relying on manual testing tend to release infrequently due to the high cost of testing. The long release cycles cause them to build large chunks of functionality without getting feedback from customers.

To deliver on the first principle of product management, ie maximizing the ROI of R&D effort, two mechanisms can provide significant benefit: fast feedback loops and careful definition of differentiating versus commodity functionality.

Product management is to a significant extent concerned with selecting functionality that doesn't yet exist in the offering. That means that we're predicting the impact on users, but we don't know for sure that the expected impact is actually accomplished. To ensure that we only build functionality that indeed adds significant value, one effective mechanism is to build thin slices of functionality, deploy each slice and measure the impact of that slice to determine whether our hypothesis was accurate. However, this only works when we have short feedback cycles, allowing us to rapidly collect feedback and make decisions. This is one of the key reasons behind the evolution of DevOps, but many companies fail to make use of the mechanism. A few years ago, I discussed this in more detail.

The second mechanism is concerned with explicitly defining what functionality in the offering is considered to be differentiating, ie drives revenue from customers, and what functionality is commodity, ie necessary for the offering to function but not driving sales and revenue. Once we're clear on this, we can more easily reject requests for changes to commodity functionality and prioritize extensions and changes to differentiating functionality. As I discussed several years ago, our research shows that 80-90 percent of all R&D resources are spent on commodity functionality. This is of course far from optimal if our goal is to maximize the ROI on R&D.

The first principle of product management is to maximize the return on investment in R&D. Although this is painfully obvious in theory, practice is showing that many companies struggle with this. The reasons include inappropriate customization, accumulating technical debt and lack of automation. Two mechanisms that are helpful in delivering on the first principle include fast feedback cycles and clearly defining differentiating versus commodity functionality. As Kris Gale said, the value is in what gets used, not in what gets built.



EMBRACE UNCERTAINTY

Humans hate uncertainty. Even saying “I don’t know” feels very uncomfortable for many. There are very strong evolutionary drivers for this as our prehistoric selves experienced existential risk whenever they entered situations with high uncertainty. As a consequence, we organize our lives to maximize certainty, ranging from the houses we live in, the jobs we select and the vacations we take. Everyone who says they like surprises is lying through their teeth. People only like positive surprises. We only like it when things turn out better than expected.

One of the irrefutable aspects of life is that the future is fundamentally unpredictable. We simply don’t know what’s going to happen tomorrow. Even if the COVID-19 pandemic is starting to become a bit of a memory, I think that many remember being completely taken by surprise as to how one stupid virus could disrupt society and our lives to such an extent.

Whenever something unexpected happens, our first reaction is to try to explain how it could have happened. My favorite example is news reporting on the stock market. Even though the stocks on average go up all the time and there are very few 10-year periods where they didn’t end up in positive territory, the day-to-day movements in the market are largely random. However, that’s such an unsatisfactory idea that we demand other answers. Hence, we watch people on television explaining to us what happened and why, even though they’re talking complete rubbish. It simply is post-rationalization that removes cognitive dissonance.

Until the Enlightenment in the 17th and 18th centuries, the primary answer was to use religion. In virtually all religions, unexpected things, both negative and positive, are attributed to the will of some god. As we’re eager to take action to reduce uncertainty, making offerings to the gods is a way to gain their favor. I find it hilarious to see people laugh at the antics of our ancestors while engaging with full commitment in corporate planning activities that often are based on such a shaky set of assumptions that we might as well take a chimpanzee to throw darts at the wall. In many ways, we’ve

replaced the old religions with new ones without calling these as such, simply because we can't stand dealing with uncertainty.

In product management, uncertainty is even more prevalent. All decisions we make are based on an assumption or prediction of the outcome or consequence of a decision. Whenever we decide to add a feature, build a new product or enter a new market, we're doing so based on a prediction of the effect of doing so. However, our predictions have a very high likelihood of being off to some extent.

Especially in companies where many are involved in the decision-making process around R&D investments, we can't simply place our bets by ourselves. We also need to convince others that the initiatives we propose will have the expected impact and are the highest-impact things we can do. The most effective strategy is to project certainty, which results in us creating an illusion of certainty that in many companies tends to become a self-fulfilling prophecy. When successful, everyone has drunk the Kool-aid and we all are convinced that the expected outcomes are 100 percent going to happen. Until reality catches up and kicks us in the shins.

Obviously, this approach, though widely prevalent, is very inefficient as significant amounts of R&D resources are wasted on initiatives that have no or only negative impact. Instead, we need to embrace uncertainty and use constructive techniques to systematically and iteratively reduce it. The primary mechanism is to simply use the scientific method, ie hypotheses and experiments, combined with fast feedback cycles.

Those who have followed my writing for a while know that I'm not a fan of requirements. The primary concern is that the one specifying the requirement implicitly claims that building the functionality realizing the requirement will predictably lead to positive outcomes. As research by us and others shows that somewhere between half and two-thirds of all features in a typical system are never or hardly ever used, this is obviously not the case. Instead, we should adopt the notion of hypotheses. A hypothesis is a testable statement of the "if this then that" type. In R&D, this often takes the form of "if we build this, that will be the effect."

The advantage of using hypotheses is that we can organize R&D as a set of bets on outcomes. This allows us to balance big bets that are high risk and high return with small bets that are low risk but obviously have less return. In addition, simply using the terminology of "hypothesis" instead of "requirement" captures the uncertainty and keeps it front of mind for everyone involved.

Viewing R&D initiatives as experiments to achieve a certain outcome has the same benefits as using the term "hypothesis." An experiment is intended to test a hypothesis in a reliable, preferably statistically validated, fashion. Hence, organizing units of R&D work as experiments ensures that we keep a very strong connection to the intended outcome. It facilitates the conversation between product managers and engineers and allows more minds to engage around the problem area.

The third mechanism is fast feedback cycles. The general principle should be that the more uncertain the outcome of a specific course of action is, the shorter we should aim for the feedback cycle to be. For instance, many software companies traditionally used annual release cycles, which meant that the "bets" the company was placing were measured in dozens or hundreds of person-years. With the adoption of DevOps, we're shortening the feedback cycle to some person-weeks of effort between customer proof points. Fast feedback allows us to move away from bad ideas much earlier than yearly release cycles. This is where A/B testing shines as it allows us to rapidly have quantitative feedback on our hypotheses.

There are of course situations where a large and long-term investment is unavoidable or at least difficult to avoid. In the automotive industry, for example, new generations of platforms tend to be developed every five to seven years and the required investment easily runs into the hundreds of millions if not more than a billion. However, even in these situations, we can triage the design

decisions we need to take into those that by necessity need to be taken early and under high degrees of uncertainty and those that can be delayed and hopefully tested before being made permanent. That brings us to the notion of reversible and irreversible decisions that we'll return to later in this series.

Humans abhor uncertainty and many of our behaviors are concerned with reducing it. This can easily lead to an illusion of certainty, which is dangerous as we start to ignore the risks associated with the actions we commit to take. Instead, especially in product management, we need to embrace uncertainty, use the notions of hypotheses and experiments instead of requirements and seek to shorten feedback cycles as much as possible. To quote Voltaire, "Doubt is uncomfortable, certainty is ridiculous."



TRADE-OFFS

One of my favorite sayings is that you can do anything in life, but not everything. This is sometimes hard to accept as it requires choices as to what not to do. And if there's one thing most people prioritize, it's optionality. Making choices that close off paths that were available to us up to now isn't a popular activity for most.

Companies are, in many ways, similar to humans in that many leaders, when confronted with a choice between A and B, will say that they want both. These leaders don't want to make the hard choices that are required for a good outcome but instead focus on maintaining optionality. Although understandable from a human nature perspective, this brings many companies in trouble.

In my experience, there are at least three main problems caused by not making decisions on priorities: low-quality solutions, random prioritization by individuals and the nothing-to-no-one syndrome. First, when asking a team or R&D department to take on more work than they have the capacity for, they'll look for ways to deliver on these requests, however unreasonable these are. The typical result will be that only the most basic aspects of the functionality are built. Quality assurance will often also take the back seat, leading to low-quality realizations of all requested functionality due to late and limited testing.

Second, when an R&D organization is loaded with too much work and a lack of clear priorities, the prioritization falls on individual engineers. So, as each individual has too much work anyway, people can easily start to cherry-pick what they think are the most interesting tasks, what they deem most important for the company, and so on. As each individual sets their own priorities, the overall result will be a mishmash as the end-to-end integration of new functionality and features will be impossible.

Third, the lack of willingness to make choices often originates from wanting to serve all customers and segments equally well with all functionality that's asked for. The hope to be everything for everyone easily results in being nothing for nobody. The outcome is simply subpar for everyone. No stakeholder, customer or segment will be happy with or even willing to use the resulting solution.

For product managers, this is very hard as they're beleaguered by everyone. With lots of responsibility and little authority, there's constant pressure to give everyone what they asked for. However, doing so will result in the aforementioned challenges. In my experience, three tactics can help in such contexts: comparative selection, group-based governance and DevOps.

One of the techniques that proves to be surprisingly effective is to offer two options and ask the other party to select one. The idea is that even if they'd prefer to have both options, the way the question is phrased forces a selection of one. Product managers can use this as a mechanism to compel choices where leadership is loath to do so. There are many more advanced versions of this game, but the essence is to use pairwise comparison to force prioritization.

Especially in organizations with multiple business units and a central R&D department, product management often is under a lot of pressure from each business unit to prioritize 'their' features. As individuals from the business units often are unaware of the requests from others, it's easy to create a perception that central R&D is slow and useless. One strategy that I've used is to establish a governance board with representation from all business units where all the requests are discussed and prioritized. If done well, it ensures that R&D resources are optimized for the best outcome for the company overall. In addition, it allows for effective alignment between business units where a feature can be used by multiple BUs.

Finally, when product releases are infrequent, eg yearly, everyone is fighting hard to get their features in the specification that forms the basis for the next release. This can become quite contentious as everyone realizes that if they don't get prioritized, they have to wait a full year for the next opportunity. As a result, the release content will be overloaded to begin with. And as we're generally poor at predicting required effort for such long periods, content will likely need to be cut anyway. However, when the organization adopts DevOps and starts to release every couple of weeks, we see a much healthier behavior. On the one hand, business units are much more willing to delay their functionality by a few weeks as it's not a lot of time. On the other hand, the predictability of R&D is much higher when asked to estimate sprints.

Recognizing trade-offs is another key principle of product management. There are trade-offs between different scopes, between stakeholders, between time scales, and so on. Failing to prioritize and trade off between various options leads to problems, including low-quality solutions, random prioritization by individuals and the nothing-to-no-one syndrome. To address this, there are tactics we can deploy such as comparative selection, group-based governance and DevOps. Product management will never be able to create a mathematically optimal solution, but rather we need to focus on continuously balancing a variety of forces to end up in a close to optimal position. To quote Michael Porter, "The essence of strategy is choosing what not to do."



FEATURES

The word “feature” is used in very diverse ways in the companies I work with and means different things to many. In some companies, it represents the key differentiation that we offer to customers and that helps us stand out from the competition. In other companies, a feature is a nice-to-have but rather irrelevant chunk of functionality that we should only add to the product if we have copious amounts of slack in the R&D organization – meaning never.

Features, however, represent the key, atomic unit or scope that a product manager deals with. Although there are many definitions of the term, we’ll use it as a unit of functionality that’s identified by the customer as an atomic, separate slice of functionality. The feature scope is only one of the scopes that product managers work with. We may, among others, focus on components, subsystems, products, portfolios, business areas, the company as a whole and the ecosystem in which the company operates.

The challenge with these scopes is that they aren’t orthogonal but rather interdependent. It’s easy to think that the company level should dictate the portfolio level and so on, but in practice, there are upward influences as well where features influence products and products influence the company. Also, inward influences from outside the company can affect all levels. In the upcoming posts, we’ll discuss three scopes: features, product and ecosystem.

As we’re focusing on the second dimension of our series on product management, which is concerned with the scopes in which a product manager operates, it’s good to start at the lowest level. One reason is that in the normal state, when the product has been released and we’re entering the periodic software updates stage, it’s typically the features that are the unit of discussion and prioritization.

Most companies allocate R&D budgets as a percentage of company or product revenue. Once the budget for the product or offering has been defined, we need a governance mechanism where we can prioritize the various activities we could do and focus on the highest-priority ones. The process of selecting these is where the challenges start.

In general, R&D organizations need to deal with three types of development efforts: building new functionality (features), managing and fixing defects and refactoring the system to reduce technical

debt. Each of these is prioritized based on predictions of what the impact of the effort will be. This is where the problem starts as, so eloquently stated by the infallible Yogi Berra, prediction is hard, especially about the future. Those involved in prioritization activities have two main options: going binary or going unreasonably optimistic.

There are three ways of going binary: using regulatory or compliance arguments, claiming that the functionality has already been sold and threatening that we lose the customer or many customers. Regulatory or compliance arguments are used to claim that unless certain functionality is included, the product isn't allowed to be offered to customers. This automatically puts the functionality at the top of the list, which is of course the intent in the first place.

The second tactic used especially by people in sales looking to influence the R&D agenda is to claim that certain functionality has already been sold to customers. As customer focus is key for any company, this leads to the functionality too often being promoted to be included in the next release as "we can't break our promises to our customers."

The third tactic is where some in the company, often sales but it can also be others, claim that we'll lose some or many customers unless we build certain features or functionality. Similar to the other tactics, the intent is to get the functionality prioritized for the next release.

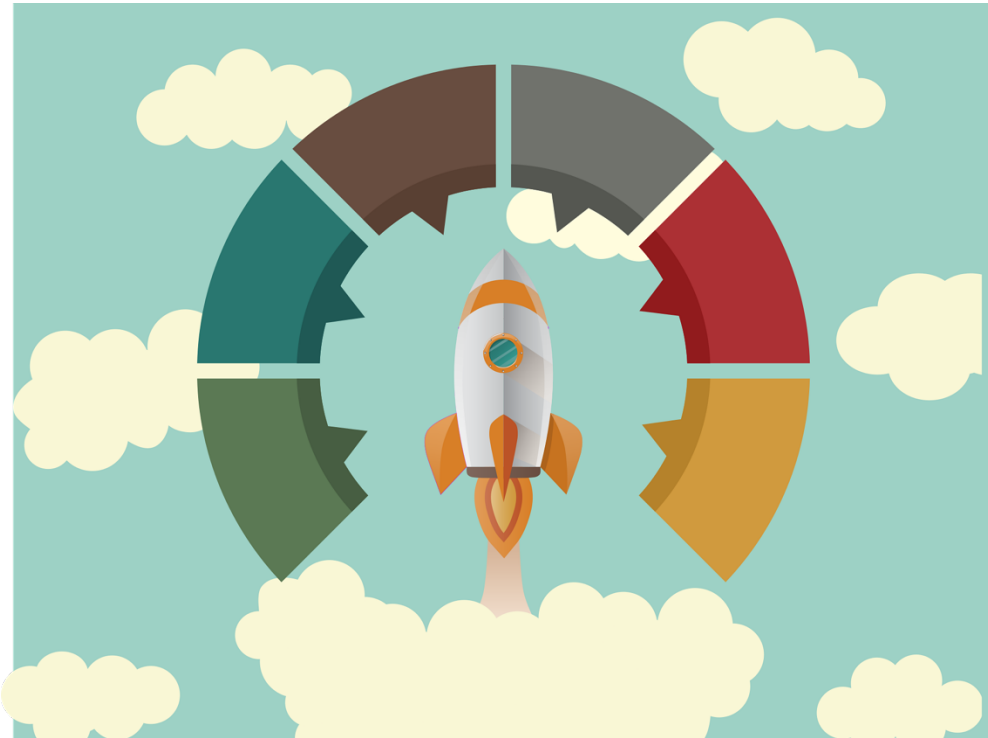
The second main approach is to go unreasonably optimistic and make outrageous claims about the impact some functionality or feature will have on the company's competitive position. The idea is to create in the heads of everyone in the company an idealistic vision of the outsized impact this functionality will have. Humans are very susceptible to the "promised land" syndrome where we create a shared illusion that we'll reach some fabulous place if we only deliver on this set of features and functionality. In this case, the best storytellers will typically be the most impactful. Especially those who can repeat the same performance release after release by making people forget the promises of the last time.

When reading the above, it must be obvious to everyone how destructive these behaviors are. By focusing on local optimization for individual goals and priorities, very often the global optimum is negatively affected. Although everyone has their own prediction on the impact of development activities, the prioritization process will become highly politicized unless we find effective ways to make rational and objective decisions. The key strategies to do so are short feedback cycles and quantification.

Short feedback cycles allow us to rapidly measure the impact of development activities. If we have yearly release cycles, we can't afford to test the impact of new functionality. However, if we use DevOps and can measure that impact in a few weeks, we can adopt approaches such as the Hypex model to conduct iterative development of new functionality while measuring that the intended impact is indeed realized.

Second, it's not sufficient to make qualitative statements about the impact of new functionality as it's too easy to redefine success once the feedback comes in. Instead, we encourage companies to quantify the impact of features and new functionality before deciding to build it. That allows us to make more balanced decisions about the required investment and the expected return, which then allows us to maximize the ROI of development.

Product management operates at multiple, interdependent scopes, ranging from features to the company's ecosystem. Here, we discussed the feature level where traditionally we can identify several less-than-constructive behaviors in prioritization that tend to cause the process to turn political. Instead, we recommend that companies focus on short feedback cycles and quantification as it decreases the amount of uncertainty that needs to be incorporated into the prioritization process. In a digital world full of data, you can't afford not to use it!



PRODUCT

To state the obvious, product management is about managing products. Often, the product manager is described as the CEO of the product, expected to decide on everything that touches it. In practice, things prove to be significantly more complicated and product managers often spend a considerable part of their time on stakeholder management.

Previously, we discussed the feature level of product management, which is of course related to the product. Even if features are important, no feature makes sense without being integrated into a product. This post focuses on the product scope. Of course, this includes cases where multiple products are derived from a common platform.

Although it seems obvious that product managers are concerned with products, there are at least three main challenges they run into: the scope of the product, identifying the real customer need and balancing new features, architecture refactoring and quality.

The first challenge frequently comes as a surprise to people not closely involved in product management. Especially in large companies, it often is surprisingly difficult to draw a clear line around the scope of “the product.” Many customers have specific extensions and configurations, which results in a wide variety of versions, even if the company may view them as one product.

In addition, within the company, it’s quite typical to have an asset that’s managed as a single, configurable product from an R&D perspective, but that’s put on the market as a family of products. The challenge then becomes whether the product manager is responsible for one of the customer-facing products or the asset as it’s viewed by R&D. This can easily be exacerbated when a family of products is derived from a common platform as each product can have product-specific as well as customer-specific extensions. In addition, the configuration opportunities in the platform and each customer-facing product may cause a situation where thousands if not millions of configurations can be created. The challenge of product management is to constrain the space of possible products to those that make sense from a business perspective.

There often also is an organizational challenge as in these contexts, the amount of work rapidly exceeds the abilities of a single product manager. This then results in a product management team where each team member has priorities that don't necessarily align with all the other members. In that case, achieving and maintaining alignment within the team often proves to be challenging as each team member has an alternative set of priorities.

The second challenge is concerned with uncovering the actual needs of the customer base. This is difficult for three reasons: noise, the gap between espoused theory and theory in use and the different needs of different customers and customer segments. First, there tends to be a surprising amount of noise when it comes to suggestions for new functionality. Everyone and their brother have ideas as to what to add to the product and many have their pet peeves that they've been peddling for a long time. Ensuring a proper assessment and prioritization of these requests is quite challenging.

Second, there's ample research out there that shows that there's a significant gap between what customers say they'll do and what they do in practice. In the B2C context, we see many companies moving toward measuring customer behavior and by and large ignoring what customers say. However, in B2B contexts, product managers need to carefully balance what customers say and what they do. This is because the product is selected and acquired based on the customers' perception concerning the product, but the day-to-day use is driven by actual behavior, which may be significantly different from what customers say. This is known as the "espoused theory versus theory in use" problem.

Third, products often serve different roles for different customers and customer segments. Although quite a bit of the functionality will overlap, there are a lot of cases where there are mutually conflicting desires and wishes. A typical example is the difference between expert users who want to be able to tweak and adjust every aspect of the product behavior and the users who are simply looking to get the job done as quickly and smoothly as possible and consequently want the simplest user interface without any bells and whistles.

Due to the above, product managers often struggle with uncovering the highest priority use cases and functionality and deeply understanding the real customer needs. In hindsight, it's usually obvious what the right prioritization was, causing product managers to catch quite a bit of flack for failing to prioritize what turned out to be important. However, identifying and prioritizing this functionality before it becomes obvious is what makes the job of a product manager so complicated at times.

The third challenge is that every digital product needs three types of R&D activities: new functionality, quality management, such as testing and defect removal, as well as architecture refactoring. The first two are focused on the short term whereas the latter is focused on the long term. Product management is concerned with ambidexterity, meaning addressing the short-term needs of the product while safeguarding its future. However, finding the right balance is very, very difficult and generally, we see that the long term is sacrificed for the short term.

Finally, the product scope isn't orthogonal but rather interdependent with the feature and ecosystem scopes. It's easy to think that the company level should dictate the portfolio level and so on, but in practice, there are upward influences as well where features influence products and products influence the company. Also, inward influences from outside the company can have an effect at all levels.

Although product management is concerned with products, it often proves surprisingly challenging to demarcate the boundaries of the product and the various product configurations that are available. This is especially the case when there's a platform involved in the equation. Second, identifying the real customer needs for prioritization of R&D activities is far from trivial for a variety of reasons. Finally, balancing the short-term activities of building new functionality and managing quality levels with the long-term activity of architecture refactoring and managing technical debt is far from trivial either. The result is that product management proves to be challenging in most contexts. To quote Thomas Schranz: "Good companies manage Engineering. Great companies manage Product."



ECOSYSTEM

Over the last decades, the interest in software ecosystems and business ecosystems has been increasingly strong. Many companies have observed the outsized success of the Apple app store for iPhone and are interested in achieving a similar situation for themselves with a large set of complementors building extensions to one product and having it act as a platform for the ecosystem. This would allow them to become the keystone player in the ecosystem, which is an enviable place to be.

In practice, we can identify multiple challenges with companies aspiring to become the keystone player in their business ecosystem. Three of the most important ones include the behavior of competitors and customers, scaling a multi-sided ecosystem and the interconnectedness of business ecosystems.

First, for a company to become the keystone player in its ecosystem, it will have to declare the strategy and start acting on it. Competitors will typically respond by taking actions of their own to avoid a situation where the company becomes dominant. This can take different forms, including offering solutions for free that the company wants to get paid for through the ecosystem partners, deploying fundamentally incompatible architectures or warning customers of the risks of vendor lock-in. Similarly, customers often dislike a situation where they're condemned to using one particular vendor without available alternatives. Consequently, there's a fundamental behavior in existing markets that seeks to avoid these kinds of situations.

Second, for a company to become the keystone player in an ecosystem, it typically needs at least a two-sided market consisting of producers of solutions complementing the platform and consumers of these solutions. The challenge is that initially, producers aren't interested in developing these solutions as the total addressable market tends to be very small and customers aren't looking for complementing solutions as these aren't available. To break this, the company needs to fuel the ecosystem by paying complementors to develop solutions and in other ways encourage the development of these solutions. In addition, it's very difficult to predict when the ecosystem reaches its ignition point where it becomes

self-supporting. As it's very expensive to fuel the ecosystem and the predictability is quite low, the leadership in the company may easily start to experience the initiative as a bottomless pit and pull the plug before the ignition point is reached.

Third, even if the company is the dominant player in its own ecosystem, the challenge is that this ecosystem typically is part of a larger ecosystem where the company is but a minor participant. The interconnectedness of ecosystems in many industrial contexts further complicates any attempts to become the keystone player. For example, several companies are currently jockeying for the position to become the dominant player in autonomous driving solutions that can be integrated into, in principle, any vehicle. Due to the nature of autonomous driving, there will likely be a single winner as the company that receives the most data back from vehicles in the field will typically have the best solution. However, the automotive OEMs are actively resisting this situation and to avoid the emergence of a single dominant player, they'll likely support alternative vendors, despite these vendors offering inferior solutions.

In practice, most companies operate in a business ecosystem where they're one of several players. The primary responsibility of product managers is to optimize their company's position in a network of competitors, suppliers and customers. There are at least three priorities for them to achieve this optimal outcome: focus internal R&D resources on differentiation, seek to establish collaborative innovation with customers and, if necessary, reposition the company and the product in the ecosystem.

First, when dividing the functionality in the product into differentiating and commodity functionality, our research shows that companies typically put 80-90 percent of their R&D resources into commodity. One of the reasons is that they spend too little time and energy to outsource the development and maintenance of legacy functionality. One priority for product management is to seek to move commodity outside the company. Of course, simply paying a supplier to maintain some code frees up our own R&D staff, but it doesn't lead to economic savings. For that, the supplier has to be free to offer the same functionality to others as well as this achieves economies of scale. This is often viewed as problematic in the company, but as the functionality is commoditized anyway, it really isn't a problem.

Of course, there are other strategies to decrease investment in commodity functionality, such as replacing internally developed software with open-source or commercial software components or open-sourcing our own software in the hope that others will engage. The goal is, however, to proactively seek to reduce the investment in commodity to free up the resources for differentiating functionality.

The second priority is to engage in collaborative innovation efforts with customers. Although some or several of these efforts may not lead to successful outcomes, the few that do tend to position the company exceptionally well for delivering novel differentiating functionality to customers and share the upside of the innovation with the customer. In addition, it tends to cement the relationship and allows for a strategic rather than a cost-centric and transactional relationship.

The third priority is to continuously evaluate the strategic position of the product and the company and explore potential repositioning. This may involve forward integration into the value network, causing our customers to become our competitors and our customers' customers to become our customers. It can also mean bundling additional products in our offering to provide an integrated solution, causing current partners to become competitors. Alternatively, it may require withdrawing from certain areas and yielding the ground to current competitors, suppliers and partners with the intent of forging new partnerships and focusing the resources of the company on the areas where it's the strongest and has the best opportunities for differentiation.

The ecosystem scope is a difficult scope to manage because of its strategic importance and the, often, unrealistic expectations of company leaders concerning the ambition to become the keystone player in the ecosystem. Typically, the behavior of competitors and customers, the cost of scaling a multi-sided ecosystem and challenges related to the interconnectedness of business ecosystems make this an uphill

battle for most companies. For a product manager, there are at least three priorities: focusing internal R&D resources on differentiating functionality, establishing collaborative innovation with customers and continuously exploring strategic repositioning of the product and the company in the ecosystem. To end with a quote from Pearl Zhu: “Innovation happens at the intersection of people, process, technology, customers and business ecosystem.”



EXPLORATION

Product management consists of multiple dimensions and so far, we've discussed the underlying principles as well as the scopes. The third dimension concerns the activities of a product manager or the activities conducted as part of product management by anyone in the company. These activities can be structured in many different ways; here, we'll divide them into exploration, strategy and execution, and we're going to start with exploration.

Exploration is concerned with investigating unknown areas of functionality that may, potentially, be relevant for the offering in focus or the company as a whole. In this activity, we aim to identify new, currently unknown, functionality, learn more about it, evaluate its relevance and decide whether it's a candidate for inclusion in the strategy activities.

In many ways, exploration is, by its very nature, a highly inefficient activity. Most of the new areas that are explored will result in a decision to not pursue them further, making all the work leading up to that decision waste. Or, at least, perceived as being waste.

Although we've focused on the functionality and features of the product, it's important to remember that there are several aspects to consider. In addition to the performance of the product itself, we can look at the product system around it, including complementing systems and services.

Next to the offering, we can also focus on what Doblin in its "10 types of innovation" model refers to as "configuration." This includes the business model, the ecosystem in which the offering is deployed, the process, ways of working and tools, and, finally, the organizational setup. Although some might consider them to be outside of the purview of product management, I take the position that the product manager is the 'mini-CEO' of the offering and should either be in charge of these aspects or have a strong say in them.

That means that product management is concerned with exploring alternative business models, experimenting with repositioning the offering in the ecosystem, trying out alternative ways of working and tools, and finally testing alternative organizational structures for maximizing efficiency and effectiveness. As we discussed earlier, the goal of product management is to maximize the ROI of R&D investments, which may mean maximizing revenue through alternative business models and better positioning in the ecosystem and decreasing cost by more effective ways of working, tooling and organizational structures.

The third area in the 10 types of innovation model is concerned with experience. This includes the customer experience, brand, channel and service surrounding the offering. As some allude to, the offering is what you pay for, but customer value is what the customer feels once leaving the store with the product. In a more general context, customer experience, brand, channel and service are all concerned with creating customer value connected to the product and product system.

The point I'm trying to get across is that there's a lot more beyond the product itself that needs to be explored. In fact, research by Doblin suggests that investment in product performance, including new features and functionality, has the lowest return on investment. Instead, the other areas, especially related to the business model, the customer experience and services, tend to have a much higher ROI. Hence, anyone working with product management needs to seriously explore these areas as well.

One of the key challenges is to conduct non-destructive experiments. Many easily feel that conducting an experiment with an alternative business model, go-to-market strategy or position in the ecosystem needs to be permanent as the experiment may upset customers and partners. Or, also very common, an experiment is highly appreciated by one key customer who refuses to accept the experiment being shut down and tries to cajole the company into keeping the functionality alive just for them.

As an example, many companies want to build a direct connection with end customers to simplify the continuous delivery of new value through DevOps and get data back from these customers more easily. This will typically upset the partners who today are in between the company and the end customers. Many companies respond with either a categorical rejection of these kinds of experiments or they jump in with a hail-mary approach, burning the ships behind them.

The solution is, of course, to find niches where these experiments can be conducted with limited downside and bring partners into the experiment as they also need to experiment with new positions in the ecosystem. Such a niche can be a particular market segment that's not that important for the company but may still provide significant learnings. Or we can run an experiment in a particular country or geography, preferably one that isn't critical for the business.

A second misconception is that many companies seek to always experiment with their most valuable and most critical customers. This is the right approach when we're looking to provide sustaining innovations, typically in terms of new functionality, to our offerings. However, in the case of more radical, Horizon 3 innovations, we want to take the other extreme: we seek to engage the customers on the edge of our business. The weird, quirky and not very profitable ones. The reason is twofold. First, an experiment that's unsuccessful and damages the customer relation is much easier to accept if the customer wasn't that important to begin with. Second, your core customers are already happy with the offering and there's no reason to upset the apple cart. Instead, we're looking to make the customers on the edge core as this allows us to significantly grow revenue from them if we're successful. It allows us to expand our customer base.

As a related challenge, it's important to remember that although sustaining innovations have a rather predictable bell curve of return, returns on radical innovation follow a power function, meaning that the vast majority of experiments fail, but the few that are successful have such outsized results that these justify the total investment in radical innovations several times over. Many companies that are used to

sustaining innovation fail to realize this and shut down radical innovation experiments after a few tries due to the lack of success.

The activities in product management can be categorized into exploration, strategy and execution activities. Exploration isn't just concerned with identifying new functionality to be included in the offering but should cover all aspects, including business model, ecosystem positioning, customer experience and services. Although this may easily be experienced as very inefficient, it's a critical activity if we seek to avoid sinking into the morass of commoditization. As Frank Borman so beautifully said: "Exploration really is the essence of the human spirit."



STRATEGY

Few words in corporate lingo are more overused than “strategy.” Frequently, problems in companies are attributed to the lack of a clear strategy. In practice, however, it often is the lack of alignment around the strategy. This tends to lead to a situation where there’s an official strategy in the documents and presentations and another, implicit, one where people are doing what they think is best. Obviously not a recipe for success.

From a product management perspective, there’s a significant need for developing a clear strategy connecting the different levels at which a product manager operates – feature, product and ecosystem. Developing this strategy should focus on formulating a clear why and the consequences for the what. This might be more involved when it concerns the company and portfolio levels and less effort-consuming when it concerns features, components and subsystems, but in practice, it’s important to have a clear “why” and “what” there as well to focus the efforts of the R&D teams.

In the companies I work with, I’ve seen at least three challenges: making choices, lack of alignment and bailing. A product manager often has lots of responsibility but very little authority to make decisions. Instead, the role is concerned with taking input from a wide variety of stakeholders, combining this and putting together a strategy that optimally balances the various forces. The problem I’ve seen play out many times is that it proves to be surprisingly difficult to say no to certain ideas. Instead, the product manager is incentivized to give everyone a little bit of what they’re asking for.

This leads to at least two problems. First, the plan tends to get overloaded and the R&D team implementing the prioritized functionality loses motivation as it’s clear that it simply is impossible to deliver. The second problem is that the result often fails to meet the needs of anyone. As I wrote in an earlier post, trying to be everything to everyone results in being nothing to no one.

The second challenge is concerned with the lack of alignment. Even if there is a strategy that has been formulated and formally accepted by key decision-makers, it doesn’t necessarily mean that it will also be executed. Frequently, some parts of the organization start executing the strategy whereas others pay it lip service but by and large keep doing what they’ve always done. This lack of alignment generates lots of activity and frustration but very few results as the different parts of the strategy don’t get together.

This challenge isn't just limited to R&D, but it's even more common across functions in a company. A typical example is when the company decides to offer customers digital solutions that are monetized using a subscription model or another continuous mechanism. R&D has built the solution, but the sales function doesn't manage to sell it. Typically, this isn't because customers don't want it but because the bonus is so much higher for selling traditional products in a transactional model that the sales staff refuse to spend time on something that doesn't generate income for them.

The third challenge is bailing on the strategy. Even when a company puts an ambitious strategy in place and manages to engage everyone, there still is a significant risk that people lose hope and abandon the strategy when it doesn't immediately lead to success. Often, this can be identified when key opinion leaders start to talk about where the revenue comes from, that the old way of doing things is still working and there's no need to change, and that they never believed in this new strategy anyway. That leads to a gradual abandonment of the strategy and we end up in a lack of alignment situation, even if we didn't start there.

The best way to address this is to make choices. My favorite definition of strategy is by Michael Porter: strategy is deciding what not to do. Product management has to simply cut away from all the requests and demands until they're left with a consistent, holistic and realistic strategy that can be executed realistically and leads to a focused, clear outcome.

As it's quite challenging to manage this for many product managers, tactics can help reduce the difficulty. In my experience, there are three that are particularly useful. The first is what I often refer to as the Barbara Bush approach: Just Say No. In many cases, there's a constant nudging by stakeholders that ceases when a clear no has been communicated. This is often uncomfortable as we're all human beings who like to get along, but it can cut out a lot of hassle.

The second strategy is morphing. We take multiple requests and morph them into one manageable chunk of functionality. The important step here is to clearly communicate how the original request is now part of this morphed functionality and therefore people got most of what they wanted.

The third strategy has become much easier with the adoption of DevOps and that's to sequence. It's basically saying "yes, but not now" and then promising the functionality to be built in an upcoming sprint or release. There's the risk of kicking the can down the road as future sprints may also require this functionality to be punted, but at least it placates some of the key stakeholders who don't want to take no for an answer.

Although defining and executing a strategy is easy from a theoretical perspective, it's surprisingly difficult to accomplish in practice. Companies often don't want to make choices, the strategy lacks alignment or teams and individuals bail on the strategy when there isn't immediate success. Still, product management needs to define a strategy that's realistic, focused and consistent. Some strategies for managing stakeholders include saying no, morphing and sequencing. To end with a quote by Max McKeown: "Strategy is about shaping the future."



EXECUTION

Thomas Edison, the alleged inventor of the light bulb (he wasn't), famously said that genius is 1 percent inspiration and 99 percent perspiration. The same is true for product management. We can explore and strategize all we want, but if we don't act on the findings and the strategy, all these efforts are in vain.

As the saying goes, the road to hell is paved with good intentions. In many of my interactions with various companies, it rapidly becomes clear that everyone knows what needs to be done and claims to want the right things to happen. However, for a variety of reasons, it stays with ideas and the execution never gets beyond ambitions and some half-baked attempts.

In my experience, the main reason that execution is so hard is threefold: personal inertia, organizational constraints and the gap between saying and doing. The first constraint that I see quite a bit is the inertia that all of us have in our lives. We all entertain a set of habits, both personally and professionally, that, according to research, dictate up to 95 percent of our actions. Even if we're all convinced that we want to do things differently, breaking out of the trigger-action-reward pattern can prove to be extremely difficult.

The second main reason why executing strategies or changes in general is so difficult is that modern organizations are optimized to an extent that every individual, team and function is tightly integrated with other individuals, teams and functions. The organizational structure dictates certain processes and ways of working, deviating from which is extremely difficult as it breaks everything in the chain.

This is often exacerbated by all the tools that were introduced or developed to automate part of the processes. Automation is amazing and very important, but the fact is that many tools dictate a certain way of working, the changing of which often requires changing the tools and their incarnations of the ways of working.

The third reason is concerned with the gap between what in management literature is referred to as espoused theory versus theory-in-use – the gap between what people say they do and what they really do in practice. This is found in all parts of life but even more so in corporate contexts. There have been several situations where it became obvious to me, after a period of trying, that even though the team claimed that they wanted to change, deep down they really didn't want to and consequently nothing happened. As a consultant, this is when you have to fire your customer and tell them that there's nothing you can do to help them.

For product management, the execution activities are typically concerned with interacting with the teams involved in building the prioritized functionality and the enabling infrastructure, the release of new software versions, the collection of feedback from the field as input for the next iteration of exploration as well as the aggregation of data to establish business impact.

First, product managers work with R&D teams to realize the prioritized functionality and implement the declared strategy. The engineers in these teams are human beings and as such, they need to know why they're building what's asked of them, how they know whether they're successful and in what way the functionality should be realized. As product managers typically are closer to the customer than engineers, in a way they become the proxy for the customer. Of course, this isn't just the case for teams building functionality that's directly used by external customers but also for teams working on infrastructures that are required to get things out to external customers. In the latter case, the product manager is concerned with an internal set of customers whose demands to meet.

Second, while DevOps is going a long way to take the drama out of releasing new software to customers and products in the field, there's often still a bit of tension and careful monitoring when new releases go live. Although the more operational parts are managed by operations staff, the performance of new functionality and the response of customers to new functionality released is a key focus area for product management. In the end, we build new functionality with the intent of delivering value and not just more complexity and crud.

The final activity is concerned with collecting qualitative and quantitative feedback from the field. It is best practice, in my view, to predict, preferably quantitatively, what the impact of a new feature or some novel functionality will be. In addition, when the level of uncertainty as to the impact is very high, we shouldn't develop the entire feature or functionality in one go but rather build thin slices and proceed iteratively, evaluating with every release whether the KPIs are moving in the right direction or not. If it turns out, after one or two releases of partial functionality, that the impact is negligible or negative, we should stop development and remove the related code from the system.

Although exploration and strategy development are critically important to get right, these have no impact whatsoever unless we execute. Many view execution as the easy part, but in my experience, this is where the rubber meets the road and all the challenges surface. For a variety of reasons, many of us, both personally and professionally, get stuck in the gap between strategy and execution. Product management is concerned with working with teams to realize the strategy, carefully tracking the impact of new software releases and using qualitative and quantitative feedback to continuously adjust and optimize the release content for each DevOps iteration. As Lawrence Bossidy so beautifully said: "Execution is the ability to mesh strategy with reality, align people with goals and achieve the promised results."



STAKEHOLDERS

After having discussed principles, activities and scopes, we're now moving on to the different viewpoints product management needs to consider. The coming posts will explore these. Specifically, we're going to focus on stakeholders, technology and exploration.

Stakeholders, as the name implies, are parties that have a stake in whatever is going on. In our context, they're typically concerned with planning the content of the next release as well as changes to make to everything around the offering. This includes the business model, the go-to-market strategy, the brand and the customer experience.

Product management involves a wide variety of stakeholders, both inside and outside the company. Inside the company, this ranges from the R&D teams to the sales organization and from customer support to the C-suite – everyone has questions, opinions and dependencies on the products in the portfolio. Outside the company, partners, suppliers, customers and competitors may all need to be managed as stakeholders.

Stakeholders all represent their perspective as well as their interests. The very nature of this process is that these interests aren't aligned and often conflicting. They can be conflicting because different people want different solutions for the same problem, which is the hardest type of conflict. However, the primary conflict typically is one of resource contention: we simply can't do everything stakeholders ask for and we need to prioritize. Managing and trading off all the needs, wishes and priorities is a critical task of product management.

In my experience, there are at least three main pitfalls product managers fall into: mixing needs and wants, failing to identify critical stakeholders and missing strategic opportunities.

When my kids were young, they often asked for candy. Once we've got the taste of something sweet and the sensory explosion that comes from consuming candy, it's hard to not ask for more. As adults, we all know that sweets are bad for us and we limit our intake. Instead, we eat broccoli and Brussels sprouts. Many customers act similarly: they focus on what feels good, their wants, and fail to identify

that this might collide with their real needs. One task of a product manager is to distinguish between the wants of customers and their needs.

As an example, one of the startups I was involved in prided itself on being exceptionally customer-focused. This had translated itself into a culture where every whim of any customer immediately led to an R&D activity being fired off. The result was a heavily overloaded R&D organization that only built small ‘nice-to-haves’ for customers, but as a whole, the offering was rapidly commoditizing as we failed to build the truly differentiating functionality and features that would meet the ‘must-haves’ for customers.

The real risk is that when failing to distinguish between wants and needs, we easily lose the competitiveness and differentiation of our products and, as a consequence, customers vote with their feet and leave. As the saying goes, the road to hell is paved with good intentions. Many organizations have good intentions, but the result of acting on them is failure.

The second problem I see product managers struggle with is the failure to identify the critical stakeholders. Not all stakeholders are created equal and we need to treat their input with different priorities based on their relevance. Many companies fall into the ‘loudest customer’ trap – going for the one making the most noise. The individual who threatens to call the CEO, or even actually calls the CEO to complain, often gets prioritized.

As a product manager, we need to identify who our critical stakeholders are. In practice, they’re the ones providing the revenue to the company. This may be customers who sign the contracts that generate the revenue or stakeholders who indirectly drive revenue, e.g. influencers in B2C companies.

Especially highly data-driven companies tend to focus on user KPIs. The implicit belief is that by serving users well, the customers will automatically follow. This is undoubtedly true in B2C companies where the user and the customer typically are the same individual. However, in B2B companies, the customer is very different from the user. As the customers pay the invoices, they’re the critical stakeholders. Not the users, who are still important but not critical.

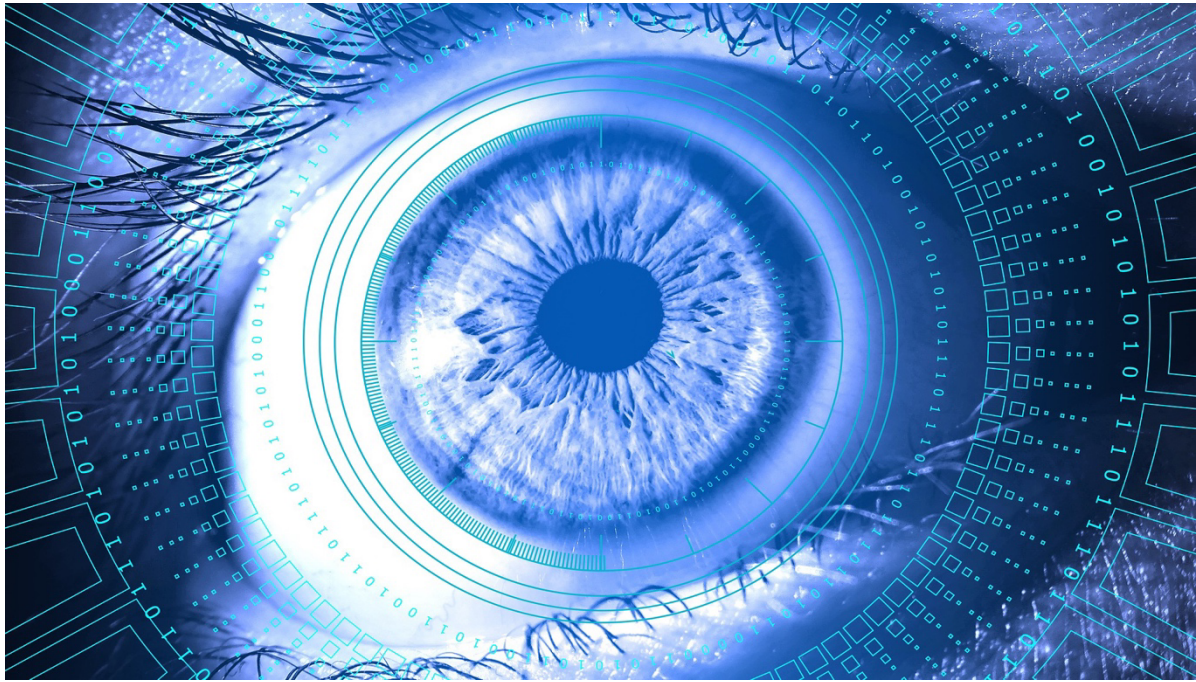
As an illustrative example, one of the big ERP companies is infamous for having a horrific user interface. At the same time, it has nine golf courses around the town where it’s headquartered. Early on, the leadership in the company realized that it was the CEO, CIO, CTO or any other C-suite individual who signs the contract and that they need to be romanced. Users will have to live with whatever experience the company decides to offer.

The third problem is that product managers only receive input from existing stakeholders. Potential stakeholders, such as new customers, strategic partners or markets, don’t have a voice yet and hence their input is missing. The challenge for many product managers is to identify and act on new, unrealized opportunities at the expense of not meeting the immediate requests of some existing stakeholders.

As an example, a company I worked with realized, after several years of catering to a specific function and department in its customer organizations, that the neighboring department could be served with the very same data, had a much larger budget and was open to engage on the use case that we were offering. Identifying these opportunities and acting on them is of course critically important, but it does take away from meeting the needs of some existing stakeholders.

Some companies distinguish between roadmap development and customer-driven development and allocate budgets to each bucket. Although that may work for you, it’s my view that product managers need to prioritize all development such that the return on R&D investment is maximized. This may mean putting all resources on a new type of customer as part of a high-risk, high-return prioritization.

Product management is also concerned about the various viewpoints that we need to investigate. These viewpoints are typically provided by stakeholders and are, by their very nature, conflicting, if only because of resource constraints. Product managers need to clearly separate the needs from the wants of stakeholders, identify the critical stakeholders and deprioritize less important ones, and act on strategic opportunities even if no stakeholders are advocating for these. As the late Steven Covey so beautifully said, “The key isn’t to prioritize what’s on your schedule but to schedule your priorities.”



TECHNOLOGY

Many, including those in R&D, may view technology as part of the R&D realm and outside the scope of product management. The thinking is that the product manager decides what needs to be built and R&D decides how to build it. In practice, however, the technology choices are so influential on the strategic options available to the company that technology selection can't be left to R&D alone. Product management needs to be involved and actively drive suitable selection among technology alternatives.

As part of this series, I've aspired to clearly communicate the interdependent nature of the business side and the technology side. Unfortunately, in many companies, one side tends to dominate. In some companies, the business side is dominant and R&D is treated as a servant where you simply order functionality and then have it delivered.

The challenge in these companies is that there's a significant amount of tension between the business side, where R&D is considered to be a bunch of incompetent morons who take forever to build anything and then make the product suffer from a quality perspective, and the R&D side, where they feel that the idiots on the business side don't understand anything about the complexity and intricacies of the product. This antagonistic nature between the two parts of the business tends to lead to short-term, knee-jerk reactions such as outsourcing R&D for a new product to a third party and completely losing the connection between products in the portfolio.

Where the technology side tends to dominate, R&D effort gets prioritized based on technical capabilities more than on anything else. If we can build it, we should and it's up to the business side to simply sell the amazing products we've created. If they fail to do so, the typically technology-savvy leadership simply replaces the sales team with a new generation who gets to prove their metal.

The optimal balance between being business-driven and being technology-driven actually changes over time and depends on the maturity of the product and the market. In the early stages of a new technology, more technical capability simply is better and being technology-driven is the way to go. For instance, the first cameras on mobile phones were pretty awful. Any improvement of the resolution, which was technology-driven as it requires better lenses and CCD image sensors, would lead to a superior product

that would sell simply because of better technical capabilities. In these situations, user experience and customer intimacy are nice-to-haves, but the main differentiator really is the technical capability.

Somewhere in the evolution of the product and the associated industry, there comes a moment when improved technical capability is no longer as differentiating or even ceases to be differentiating to customers. At this point, the technical capability of the product is commoditizing and we need to shift our attention to other aspects of the innovation landscape (see also our discussion on the ten types of innovation) to drive the company's continued business success.

The easiest way out for many companies is to start to drive down costs. Although there's definitely a place and time for this, I think that the primary focus should be on differentiation. If it's difficult to differentiate through the product itself, other aspects such as the business model, brand, customer experience, complementary services and so on can become the focus of innovation and R&D investment.

One of the classic mistakes I see many technology-driven companies make is to keep investing in new technological capabilities even though the value to customers has already plateaued or has even started to become negative. For instance, modern mobile phones have extremely high-resolution cameras leading to ridiculously large images. This is great for the happy few who use these images for specific purposes, but for the large majority of us, it's just a hassle to schlep around those large image files.

The place where we can resolve the tensions between the business side and the technology side of the company is, of course, in product management. This is where R&D brings new technological opportunities that can be used to serve customers better while sales, customer support and others can bring market-driven insights on how to serve customers better. By engaging in a constructive, curious and transparent discussion around the various options available, experimenting with these and then prioritizing effort for the next iteration, we can achieve a situation where business and technology are joined at the hip.

Product management is where business and technology meet. As technology choices are core to creating strategic opportunities for the company, product management needs to be involved and we can't leave technology choices to R&D alone. In practice, many companies are either business-driven or technology-driven, causing challenges as they fail to identify when to shift and prioritize when needed. The role of product management is to balance business and technology in a way that optimally serves the market and the customers and, by extension, the company. To quote Bill Gates, "Information technology and business are becoming inextricably interwoven. I don't think anyone can talk meaningfully about one without talking about the other."



IMAGINATE

We've already talked about exploration as an activity in an earlier section. We need to conduct exploration activities, of course, to find new functionality to add to the existing offerings and to identify new offerings to create. However, exploration is also a mindset and a viewpoint we need to take for the activity to even be successful in terms of creative breakthroughs. Rather than use the word "explore" again, we'll define a new term: "imagine."

The challenge many of us suffer from is that we have too much to do for the amount of time that we have available during the day. This leads to a situation where we become focused and concentrate on what's right in front of us. That's great for getting things done that we know need to be done, but it typically doesn't result in novel insights and ideas.

A podcast I recently listened to talked about the close alignment between mental activities and the physical response. When we get focused on some work task, also our vision changes and we actually stop seeing what's happening around us. Our brain starts to ignore all the inputs outside of the immediate task in front of us. This is of course the death of creativity and new insights and we need to fundamentally change our approach if we seek to adopt an exploratory and creative viewpoint.

In my experience, seven things help me adopt an exploratory mindset. First, we need to make room in our calendar. Although we can't schedule creative insights and ideas, we know for sure that we won't have them unless we create the space where they can occur. This means allocating significant chunks of time on our calendar. Two hours is the minimum slot where I've had any measure of success. Also, if at all possible, schedule these hours late in the afternoon when most of the urgent work for the day has been taken care of and, according to research, we're more creative than during other hours of the day. It does require some discipline to not simply conk out but instead stay focused on the intended goal of imagination, creativity and exploration.

Second, avoid thinking about scaling. Many people tend to kill an idea as they imagine how it would need to be operationalized and how it would need to scale to be practical. Every potentially large tree can be easily killed as a sapling if you don't protect it. New ideas need to be given the chance to germinate and grow a little before you become too critical and kill potentially good ideas. Rather than thinking about scaling, we need to focus on imagining and creating the most bespoke, manual and unscalable thing we can do that would really solve a customer problem. If we solve a real customer problem, the rest will follow over time.

Third, creativity is a team sport in many contexts. If we can bring other people into these imagination sessions who we can bounce ideas with, the results can be quite astounding. The main learning I've had is that people aren't allowed to judge and criticize each other's ideas. Instead, ask clarification questions and try to build on top of what others contribute. Ideally, no idea is left behind but instead, we seek to morph multiple ideas into a new concept that can be used for exploratory testing and validation with, potentially prospective, customers.

Fourth, it's crazy, but we have no idea where ideas come from. We've all had situations where a solution to a problem that we'd been noodling on for a while suddenly popped into our heads. We can't control when this happens or if it happens at all, but in my experience, virtually all challenges I've tried to tackle with creativity have resolved themselves simply by sticking with it and continuing to spend brain cycles on it. So, the fourth practice is to use our subconsciousness wisely. As Thomas Edison said, never go to sleep without a request to your subconsciousness. Novel, creative and innovative solutions aren't created by our rational brains but rather pop up out of our subconsciousness.

The fifth practice is to use positivity. According to psychologists, as humans, we're three times more sensitive to losing something than we appreciate gaining something. So, rather than spending a lot of conscious thought on what might go wrong, we need to intentionally think about what happens if we're right! In the venture capital community, this is a well-known practice to overcome our natural dislike of losing money on a risky investment. Instead, the practice is to imagine how large the startup could become if everything goes right. In many cases, this will result in 100x to 1000x the original investment. Even if the chance of success is only 10 percent, it still makes it a positive bet that we should take. If we simply make enough investments, we're going to win big-time and on each investment we can lose our money only once.

Sixth, relax constraints. Very often, we end up in a deadlock situation in our minds because the problem we're looking to solve is overconstrained. There are so many constraining aspects and dimensions that the solution space is effectively empty. The way around this is to start relaxing or even removing some of the constraints. A typical one is that we focus on our existing customers and can't imagine a solution that would serve their needs. So, initially, we should remove that constraint and focus on serving any customer, not our current ones.

Finally, as the seventh practice, move. We've all heard the saying that sitting is the new smoking. And many of us, including me, have changed to desks that also allow us to stand while working. In my experience, however, the real breakthroughs occur while moving, and walking and running are by far my favorite exercise forms for creative thinking. Running is best when I'm chewing through things by myself. Walking works well when I'm interacting with someone else, either on the phone or in person. In fact, a close colleague and I have a habit of going for a walk at the same time, calling each other and exploring a problem by bouncing ideas off each other. Virtually every walk ends with both of us having developed new insights and ideas and quite a few of our recent research articles originated from these walks.

As product management is so concerned with maximizing ROI, it's tempting to focus on improving existing functionality that's already proven to deliver value to customers. However, we need to invest sufficiently in the exploration of new areas of functionality. Exploration isn't just an activity but also a

viewpoint and mindset. To get to the “imagine” state, I use at least seven tactics and practices, including creating space in my calendar, avoiding scaling, interacting with others, exploiting my subconsciousness, using positivity, relaxing constraints and moving. Exploration, creativity and imagination are experienced as very inefficient from the perspective of sustaining innovation and improvements as so many experiments fail. However, the few experiments that are successful will have an outsized impact and pay for all the failed ones. To end with a quote by Albert Einstein: “Creativity is intelligence having fun!”



VITAL FEW

The final dimension in this series on digital product management concerns roles and responsibility. Surprisingly, in many traditional organizations, product managers have significant responsibility but surprisingly little authority. Instead, they need to align all the relevant stakeholders and reach decisions based on consensus.

Traditionally, this challenge is addressed by building a product management function where a hierarchy of product managers matches the system architecture. Within the function, the product managers align. Presenting one front to the rest of the company, they use their combined weight to ensure that their prioritizations are realized.

We can identify a trend in some companies, like Airbnb, where the product management function is disbanded. Instead, product management becomes a part of everyone's role. Each individual is expected to spend time aligning the why and what with their peers, team and the rest of the organization.

The roles and responsibility dimension seeks to answer the question of who takes care of the various product management activities, for what scopes and from what viewpoints. And, of course, what alignment mechanisms are used to accomplish this. We're going to explore three main responsibilities of product management: focusing on the vital few rather than the worthwhile many, organizing product management and balancing qualitative and quantitative data and insights.

Although we've discussed the topic before, it bears repeating: product managers often need to manage multiple stakeholders and there's a very strong tendency to look for ways in which each stakeholder is offered a slice of what they want so that everyone is somewhat happy. We can often come up with great justifications for why the selected content delivers value for customers and other stakeholders. So, for a product manager, it's extremely tempting to take this approach.

However, by peanut-buttering the available resources over so many different things, we're making only small, if not negligible, progress in each of the areas prioritized by the stakeholders. This leads to the "worthwhile many versus the vital few" challenge: it's not that each of the prioritized actions is bad in

and of itself – each of the items in the prioritized backlog is valuable in some way; the problem is just that it won't deliver as much value as a highly limited, focused set of prioritized actions, ie the vital few, would have had.

As every industry is in a digitalization journey, we now need to add all kinds of software, data, AI, cloud and other solutions to our offerings. In addition, we seek to adopt continuous delivery of value through DevOps, DataOps and AI/MLOps. However, that doesn't mean that the requests for the older parts of the offering will disappear. Of course, stakeholders and customers still want improvements there as well. However, the return on investment on these improvements often is very low as compared to exploiting the novel opportunities.

One of the tactics I frequently use to start a conversation about delivering value is to calculate the amount of business value a single agile team has to deliver every sprint. With 8 persons in a 3-week sprint, for example, the team spends 24 person weeks or 0.5 FTE per sprint. A fully loaded FTE in many companies is well over 100,000 euros, but let's use that for now. That means that one sprint for one team costs 50,000 euros.

The question is how much value this team has to deliver. That depends on the percentage of revenue that the company spends on R&D. For a typical company in the embedded-systems space, it's around 5 percent. That means that every euro invested in R&D has to deliver 20x in terms of value, meaning that every agile team in the company, every sprint has to deliver 1 million euros of business value.

I'm sure that everyone has stories where a team delivered a feature that was a run-away success, producing outsized value. However, I'm talking about consistently and continuously generating 1 million euros of value for every agile team and every sprint. This can typically not be achieved by peanut-buttering effort over a gazillion initiatives and focusing on traditional, often commoditized functionality. It has to come from growing the business by delivering new customer value that wasn't present before.

Focusing the entire organization on one or a very small number of large initiatives is scary as it brings a significant amount of risk. If you're wrong and customers don't care about it, you've wasted a significant amount of effort. It's much safer to have many small initiatives of which some are wins, some are draws and some are losses. It's easier to defend against criticism from the rest of the organization.

To reduce the risk of large initiatives failing, we rely on our exploration and "imagine" activities. By running small-scale, low-cost and fast experiments with customers, we can de-risk large, focused efforts and increase the success rate of these initiatives. This is yet another reason why exploration and ensuring that we adopt the right mindset during exploration is so important.

Product management is highly susceptible to the "worthwhile many" syndrome where we divide our effort over numerous small initiatives and make small progress in each. Instead, we need to identify the vital few that will really move the needle for the company when successful. These initiatives are high-risk, high-reward, but we can de-risk them by focusing on small, low-cost experiments as part of our exploration activities. Not all will be successful, but even the ones that don't pan out will give us incredibly valuable learnings. To close with a quote by Mark Pincus: "I like to bet on people, especially those who have taken risk and failed in some way, because they have more real-world experience. And they're humble."



ORGANIZING

Few topics are discussed more often in companies than the organization. How to organize people into teams? How to organize functions into departments? Where to allocate responsibilities? The problems of the existing organization. How much better things would be if we did it differently? And so on.

This is particularly the case with product management as it sits right on the boundary between the business and R&D side of the company. Interestingly, we can identify a trend where in some companies, like Airbnb, the product management function is disbanded. Instead, product management becomes a part of everyone's role and each individual is expected to spend time on aligning the why and what with their peers, the team and the rest of the organization.

Although the Airbnb approach isn't for everyone, it highlights an important principle: we should focus on the activities that need to be performed and spend less time on the specific organizational structure. Earlier in this series, I mentioned the BAPO model where we start with business strategy, define architecture based on that, then define processes and ways of working and only then talk about organization.

However, no matter how we think about this, product management activities need to be organized and we need to discuss how we go about this. Rather than offering an organizational blueprint, we'll discuss a set of principles that are helpful when thinking about the organization. These include "span of control", avoiding specialization, minimizing handovers, end-to-end responsibility and alignment through architecture and KPIs.

First, in management, there's a concept of "span of control," indicating the number of direct reports a manager can handle. The notion is that the more hands-on a manager needs to be, the smaller the "span of control" is. In product management, similar principles are used concerning the scope a product manager can handle. One of the things that generally is highly underappreciated is the role of automation in the ability to increase the scope or span of any function. By automating all the lower-value, more mundane and repetitive tasks, the idea is that each individual can manage a much larger scope than without the automated support.

This is critically important as every time we seek to scale an activity by breaking it into two roles, we're introducing a significant cost for coordination and alignment between the two individuals now occupying the two roles. Anywhere we can avoid separating an activity into multiple roles or, where feasible, reintegrate a split activity into a single role, we increase the efficiency and effectiveness significantly.

Second, avoid specialization. As humans, we have a natural tendency to seek to specialize – it can easily be claimed that much of the progress of humankind is driven by specialization. However, specialization only works if the task we specialize in has longevity. The challenge in digital companies is that things move so quickly that something that was critically important yesterday is no longer relevant today. As we often consider our specialization as something very important, it leads to a situation where we stick with it even if it's no longer relevant. Instead, we need to be generalists who can rapidly move from one area to the next without getting stuck in any specific one. We have to focus our energy on that which is most important right now and move on as soon as it stops being that.

Third, few productivity killers are more effective than handovers. The idea of having one person do the first part of a task and then handing it over to somebody else for the next part is a beautiful concept in theory but pretty awful in practice. It works if the activities are highly standardized and repetitive, but in that case, the activities should be automated anyway. For work that requires human intelligence, we typically run into the “knowledge iceberg” problem: what's documented in the handover is only 1-10 percent of the total knowledge accumulated. The remaining knowledge sits tacitly in the head of the person who did the first part of the task and isn't transitioned as part of the handover.

This brings me to the fourth principle: end-to-end responsibility. In my experience, individuals and teams should to the largest extent own a slice of functionality from the early stages of exploration to the evolution of the functionality after it's been released to customers. Of course, there are always “buts and ifs” associated with this, but where we can accomplish it, we succeed in minimizing handovers and keep all the relevant knowledge in the heads of the team members.

And then, finally, this requires us to think about architecture as a careful design that can significantly simplify the end-to-end responsibility for teams with minimal interaction between them. Most of us waste a significant portion of our lives in meetings. These meetings are typically there to synchronize and align. Although it works, it's much less efficient than synchronizing through architectural interfaces. As long as none of the teams materially changes the interface, both sides can work independently of each other.

The second part of the fifth principle is concerned with KPIs. Rather than bringing everyone together in meetings to discuss, defining KPIs that each team is responsible for as well as defining guardrail KPIs that aren't allowed to deteriorate beyond a certain level allows for teams to work independently in situations where one's work does affect the other. Defining and tracking KPIs can minimize process-driven interaction, which goes a long way to increasing performance.

Instead of providing a blueprint for organizing product management, I've discussed five principles to consider while organizing the product management activities. These include “span of control”, avoiding specialization, minimizing handovers, end-to-end responsibility and alignment through architecture and KPIs. We need to organize the work, but we shouldn't get stuck in one way of thinking about it. As Peter Senge said: “The learning organization is an organization that's continuously expanding its capacity to create its future.”



QUALITATIVE AND QUANTITATIVE DATA

In this series, we have, time and again, stressed the importance of using data. We presented a variety of arguments to justify our focus on data, but they all center around the notion that, as humans, we're highly able to create stories to explain why things are happening that, when more carefully inspected, turn out to be completely unfounded.

In addition, as we all collect experiences as we operate in our industry and work for our company, we increasingly use these experiences as a basis for our decision-making. The more we use them, the faster we can make decisions and the better we can explain why we advocate for a specific decision. However, in rapidly changing areas of expertise, for instance due to digitalization, the experiences we have are rapidly becoming invalid and we need to continuously critically investigate whether what we believe to be true indeed still is the case.

Few techniques are more effective in validating our beliefs than the use of data. We distinguish between qualitative data, like comments on our website or quotes from interviews with customers, and quantitative data, like results from surveys and A/B testing. Most often, however, people refer to quantitative data as the mechanism for making data-driven decisions.

The challenge is that working with quantitative data isn't entirely trivial. Among data scientists, it's well-known that by using the right statistical technique and carefully selecting data from a larger set, you can basically prove anything. As the saying goes, we have lies, big lies and statistics.

In addition, many moons ago, William Edwards Deming insisted that everyone should bring data. In an age where little quantitative data was available, the insistence on quantitative, statistically validated data was entirely understandable and the right focus. With the emergence of the big data era, however, we've entered a situation where the amount of available quantitative data is phenomenal and we've reached the other end of the spectrum: we have so much quantitative data that it becomes almost impossible to determine which data to use and for what purposes.

In our work with a variety of companies on data-driven practices, we've seen companies go through three phases. First, the company operates on traditional opinions, experiences and selected, qualitative customer input. In this phase, storytelling is an important part of the process and decisions are made

based on the loudest customer, the stakeholder with the best rhetoric or the beliefs of the product manager.

When more data is being collected from the field and some of the decisions made earlier prove less than optimal, the second phase is entered. Here, everything the company seeks to do needs to be based on quantitative data. More and more data is collected, often as part of a “just in case” mindset: gather as much data as feasible and we’ll figure out ways to use it later. During this phase, two problems pop up: how to store, process and use all this data coming back from the field and, as the data savviness improves, the semantics of the data become increasingly unclear.

A typical example of this pattern is when the company starts to use A/B testing. When running multiple A/B tests at a time, statistically determining the impact of each experiment becomes increasingly challenging. The interaction problem between different A/B tests is a well-studied problem from an academic and research perspective and clear solution approaches, such as customer base segmentation and multi-factor analysis, are available. However, in practice, it’s surprisingly hard to avoid the challenges and interpret actual data to draw valid conclusions.

When the organization matures and becomes aware of the limitations of data, it enters the third phase. Here, qualitative and quantitative data are combined to draw conclusions that are both mathematically solid and have a clear qualitative meaning in the context of the company. When the company reaches this level, we see several behaviors occurring, including a proactive understanding of the why behind data, a healthy skepticism of conclusions drawn from data and a continuous discussion around the KPIs the company is optimizing for, ie value modeling.

First, whenever interesting and surprising insights are found in the data, more mature product managers seek to qualitatively understand the underlying explanation behind the data. This often requires collecting qualitative data from customers and other stakeholders, which may involve traditional techniques such as ethnographic studies, customer interviews and surveys with open questions.

Second, as it’s so easy to intentionally or unintentionally draw incorrect conclusions from data, there’s a healthy skepticism in mature organizations toward data-driven findings. In medicine, the rule is that for any new scientific truth to be accepted by the community, the same outcome has to have been achieved in at least three different studies by completely unrelated researchers. In mature, data-driven companies, the same principle is often applied: exceptional results require exceptional evidence.

Third, data-driven companies use data at all levels in the organization and the starting point often is a quantitative definition of the desired business strategy outcomes. Rather than a vague, qualitative “hand-waving” business strategy, these companies define clear, quantitative outcomes as well as the relative priorities of these outcomes in case initiatives affect multiple KPIs. In our research, we’ve used the term “value modeling” to refer to these activities. Ideally, there’s a clear, quantitative link between the business strategy and goals and the activities of individual teams in R&D and elsewhere in the organization.

Interestingly, even in traditional companies, finance and sales tend to be highly data-driven. The goal is to adopt data-driven practices in all other functions as well, including R&D. Done well, this can increase the effectiveness of R&D significantly, ie at least double it. Of course, it’s not necessarily trivial to achieve this, but considering the atrocious effectiveness of R&D in most companies, I remain flabbergasted at the fact that most companies stay with storytelling, qualitative approaches to prioritizing work in R&D.

Adopting data-driven practices is of critical importance to digital product management, but it’s not without challenges. It’s very easy to, basically, prove anything you want by selecting a suitable slice of the available data and applying a suitable statistical technique to it. Instead, we need to combine quantitative and qualitative data to ensure that we hold a set of beliefs about the product, the customers

and the market that's validated by quantitative data. For this, we need to constantly seek to understand the 'why' behind data, maintain a healthy skepticism toward conclusions based on data and ensure that we know what we're optimizing for using value modeling. As William Turner so beautifully said: "You may have heard that the world is made up of atoms and molecules, but it really is made up of stories. When you sit with an individual who has been here, you can give quantitative data a qualitative overlay."



CONCLUSION

We started this series with an exploration of the importance of extending our focus from only how to build the systems we're responsible for to the why and what of these systems. When we only look at the how, we tend to focus on efficiency: cranking out as many features as possible per unit of R&D resource. More important than efficiency, however, is effectiveness.

The separation between the how and the what has caused, in most organizations, a very low return on the investment of R&D resources. Research by us as well as others shows that somewhere between 70 and 90 percent of all R&D resources are spent on commodity functionality that no customer really cares about, except for it needing to work. However, this functionality has been there for several product generations, so it's surprising that it requires so many resources. Second, research shows that somewhere between half and two-thirds of the new features and functionality added to the system are never used or used so seldomly that the R&D investment in the feature should be considered wasted.

Over the years, my colleagues and I have developed a variety of techniques and approaches to address this issue, including the Hypex model, the Three Layer Product Model (3LPM), various approaches to A/B testing, value modeling and so on. However, the uptake of these techniques has been far from universal in the software-intensive systems companies that we predominantly work with. My assessment after many years of banging my head against the wall isn't that the techniques and approaches don't work, they do, but that product management and R&D have a Kiplingesque relationship in that East is East and West is West and the two shall never meet.

When product management and R&D are separated from each other, the main mode of interaction is through requirements and specifications. This requires product management to make a series of decisions around the product while making, often unfounded, guesses about the impact of these decisions on the customers. And, of course, R&D is doing its best to build what many know is completely useless functionality in the best way possible based on what it says in the specification.

In my view, the best way to address this is by adopting three main principles: removing the dichotomy between product management and R&D, treating requirements as hypotheses and value modeling. The

dichotomy between product management and R&D is at the heart of the challenge I believe needs to be addressed. As we discussed when we wrote about organizing product management, the central tenet needs to be to move the interaction between product management and R&D from inter-team or inter-functional coordination to intra-team coordination. The product management activities and R&D activities need to occur within the same team to optimally align and coordinate.

The dichotomy between product management and R&D has caused the requirement specification to be the boundary object between the two functions. The problem is that product management tends to claim that the requirements are written in blood and are immutable in order to not look like idiots toward R&D. Meanwhile, R&D has no ambition to challenge product management about the relevance or viability of these requirements in order to maintain a reasonable working relationship across organizational boundaries.

However, requirements are simply hypotheses about what would add value to customers and we need to treat them as such. Once we treat each requirement as a hypothesis, we can start doing two important things. First, we can engage in a discussion around the impact, preferably measurable, that the unit of functionality is expected to have if built and included in the system. Second, once we've agreed upon the expected impact, we can conduct experiments to validate that belief. These experiments typically are concerned with developing a thin slice of the full feature and measuring the impact of that slice to establish more confidence that the hypothesis indeed holds and the requirement will have the expected impact.

More likely than not, the hypothesis will fail to hold and then we can decide to remove the functionality related to the requirement, hypothesis and experiment from the system and move on to the next hypothesis. The primary advantage is that instead of developing the entire feature, we've only built a thin slice, spending a fraction of the R&D effort, establishing that it doesn't hold and not wasting more effort.

Finally, we need to seek to establish quantitative metrics or KPIs that capture what's considered to be the value of the system to customers. It never ceases to amaze me how little agreement there is in companies about what the quantitative, measurable goals are concerning delivering value to customers, what constitutes customer value in quantitative terms and what is the relative priority of these various metrics in the case where we have to trade these off against each other. In various posts, the topic of value modeling has been brought up, but the end goal is that every agile team, every sprint, can quantitatively determine what value they delivered to customers. That requires a hierarchical value model where value factors aren't just related horizontally, ie with a relative priority at the same level, but also vertically, where lower-level factors contribute to higher-level ones.

Product management is a challenging, multi-dimensional set of activities that live on the boundary between business strategy and technology strategy as well as sales and R&D. In this series, we've discussed five dimensions of product management: principles, scopes, activities, factors and roles. For each of these dimensions, we focused on three aspects that we considered to be the highest priority.

Although product management has been studied for decades, digitalization, including software, data and AI, has caused a significant change in the way we conduct it. We can go for fast feedback loops through the adoption of DevOps, run experiments, such as A/B testing, use the data collected from systems in the field and employ machine learning models, trained by the same data, to provide solutions superior to what we can create algorithmically. This series has sought to study and discuss what the implications are on product management and the rest of the organization. While this may not seem so relevant for some in product management, in the age of digital, as Eric Pearson said, it's no longer the big beating the small but rather the fast beating the slow.