The Dangerous Animals Of Product Management

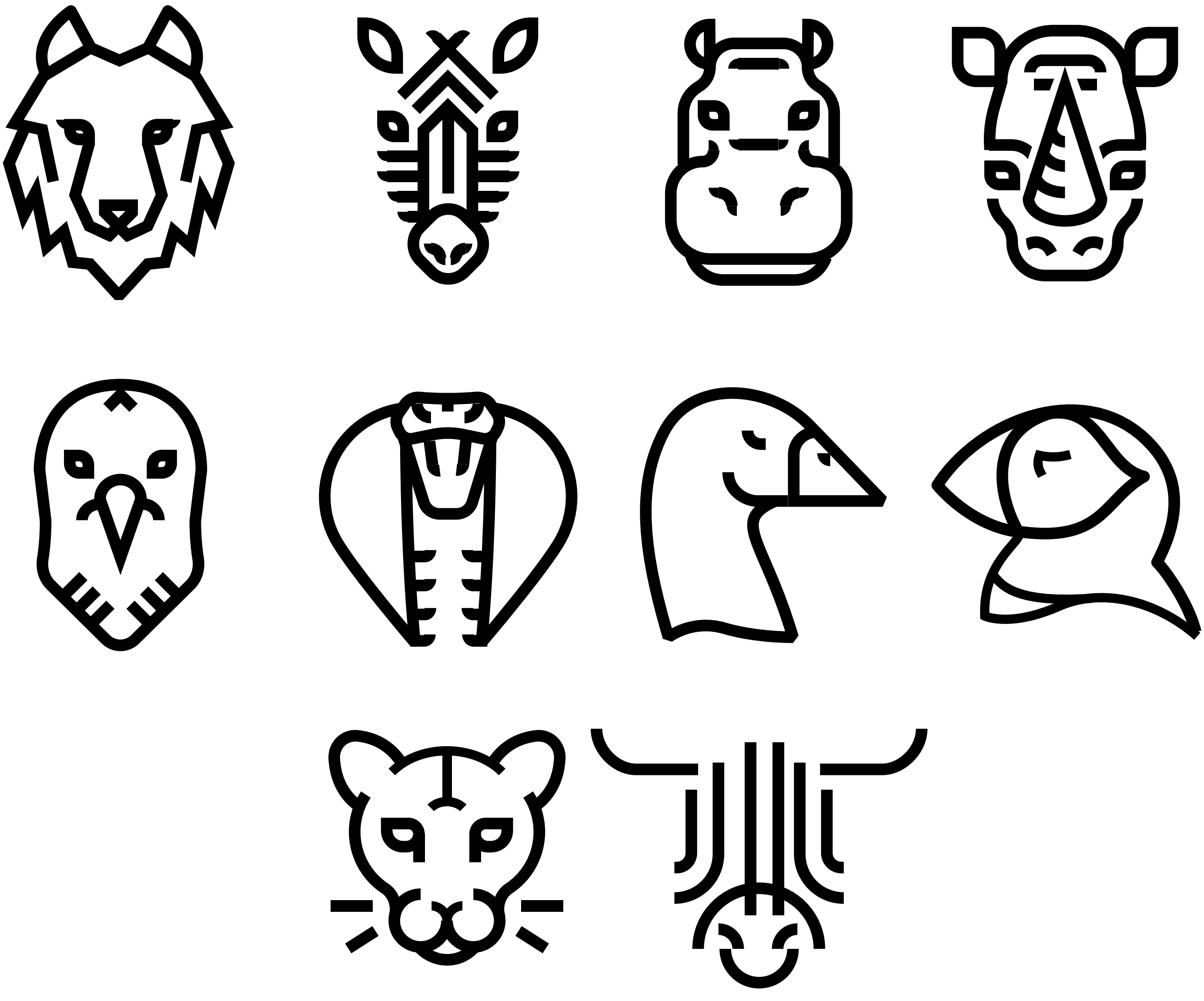
The art of managing stakeholders & staying true to your product strategy

A Productboard publication in partnership with Dean Peters

productboard

The Dangerous Animals Of Product Management

The art of managing stakeholders & staying true to your product strategy



A Productboard publication in partnership with Dean Peters

productboard

Table of Contents

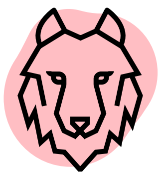
Introduction

Picture this. You are a forward-thinking product manager with a clear product vision, a value-driven prioritization framework, and a context-rich roadmap. You make it a point to listen to customer needs and incorporate feedback from many different perspectives.

Yet throughout the product management process, stakeholders come to the table with unvalidated requests. “This must be built,” they insist, “and right away.”

Introducing the Dangerous Animals of Product Management: stakeholders and situations that—if left untamed—can get in the way of your carefully planned product vision and strategy.

The Dangerous Animals of Product Management



WoLF (Works on Latest Fire)

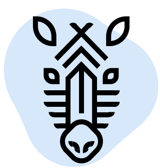
When you neglect technical debt and issues like security and product functionality to focus solely on new features, you can end up with wildfires that require your immediate and complete attention. The WoLF's cycle of reactivity demands all your resources, seriously hindering productivity and innovation.

RHiNO (Really High-value New Opportunity)

You can recognize a RHiNO by their call, "If we just had feature X, we'd be able to make this sale/land this particular customer." When you address the RHiNO's one-off requests too often, you are focusing on solutions rather than solving real problems for a broad range of customers.

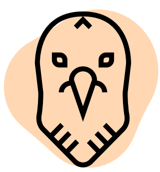
HiPPO (Highest Paid Person's Opinion)

It can be tempting to give in to HiPPOs (founders, CEOs, or other leaders), but letting them make all the decisions leads to products or features that haven't been validated. This can kill morale and introduce risk, making way for what product leader John Cutler calls a "Feature Factory".



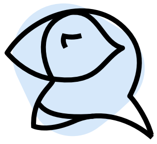
ZEbRA (Zero Evidence But Really Arrogant)

ZEbRAs think they know it all, but rely on their gut rather than any actual evidence. ZEbRAs might luck out and occasionally get things right—but not nearly as often as when you take the time to test assumptions and gather data to support your decisions.



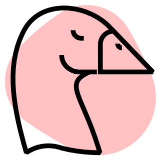
Seagull Manager

Seagull Managers tend to be a little bit removed from your day-to-day work. However, they occasionally swoop in, cause a ruckus, then swoop out again, leaving the team to clean up the mess. Their intentions are good, but they aren't always aware of the long-term effects of their proposed solutions.



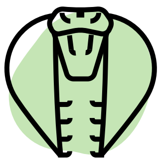
PUFFIn (Plans Unending Feature Factory Initiatives)

PUFFINS are content to jump from one shiny new feature to another without any concern for long-term vision or strategy. Let PUFFINS rule the roost and everyone will have plenty to do—but it's unlikely that the work will guide you towards a North Star or lead to a positive impact on the business.



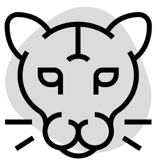
GOOSE (Guesstimating Overly Optimistic Scheduling Estimates)

The GOOSE has good intentions but a shaky understanding of how long it actually takes to get things done. It may be because they're new to product, fail to see the bigger picture, or forget to account for engineers in their estimates. Whatever the cause, it leads to over-promising and under-delivering.



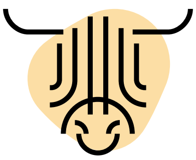
CoBRA (Cognitive Bias Related Assertions)

The CoBRA can strike at any time, and often its work is invisible. There are dozens of cognitive biases that can impact our decision-making without us even knowing. And while making snap decisions can keep us out of harm's way in real life, we should take more time and be thoughtful about our judgments in the business world.



PUMA (Promotes Unusually Meaningless Assumptions)

PUMAs use their lightning-quick reflexes to pounce on an idea or solution. But those sharp claws only scratch the surface, and they rely on assumptions instead of digging deeper into what a customer, user, or data point is really saying.



YAK (Yet Another KPI)

YAKs have quite the appetite for metrics and numbers—so much so that they often gorge themselves without a second thought! And while metrics can be a good thing, they can be gamed or turn into noise if they're not aligned with desired behaviors or actual business goals.

Every business has these Dangerous Animals roaming wild. Implementing their suggestions can interrupt months of strategic planning and prioritization, impacting product performance and your team's morale.

The problem is that product managers often lack the authority to tame Dangerous Animals and cannot fence off the product to keep them out. To avoid being devoured, product managers instead have to excel at "Dangerous Animal husbandry"—the art of collaborating with key leaders and stakeholders and getting them on the same page with your product vision and strategy.

This guide to taming the Dangerous Animals of Product Management first analyzes these different personalities and pain points. We then share how to use a combination of soft skills and hard frameworks to handle each animal and turn them into a productive part of your product management process.

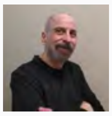
Let's go on a safari!

Written in partnership with Dean Peters, "zookeeper" of the Dangerous Animals

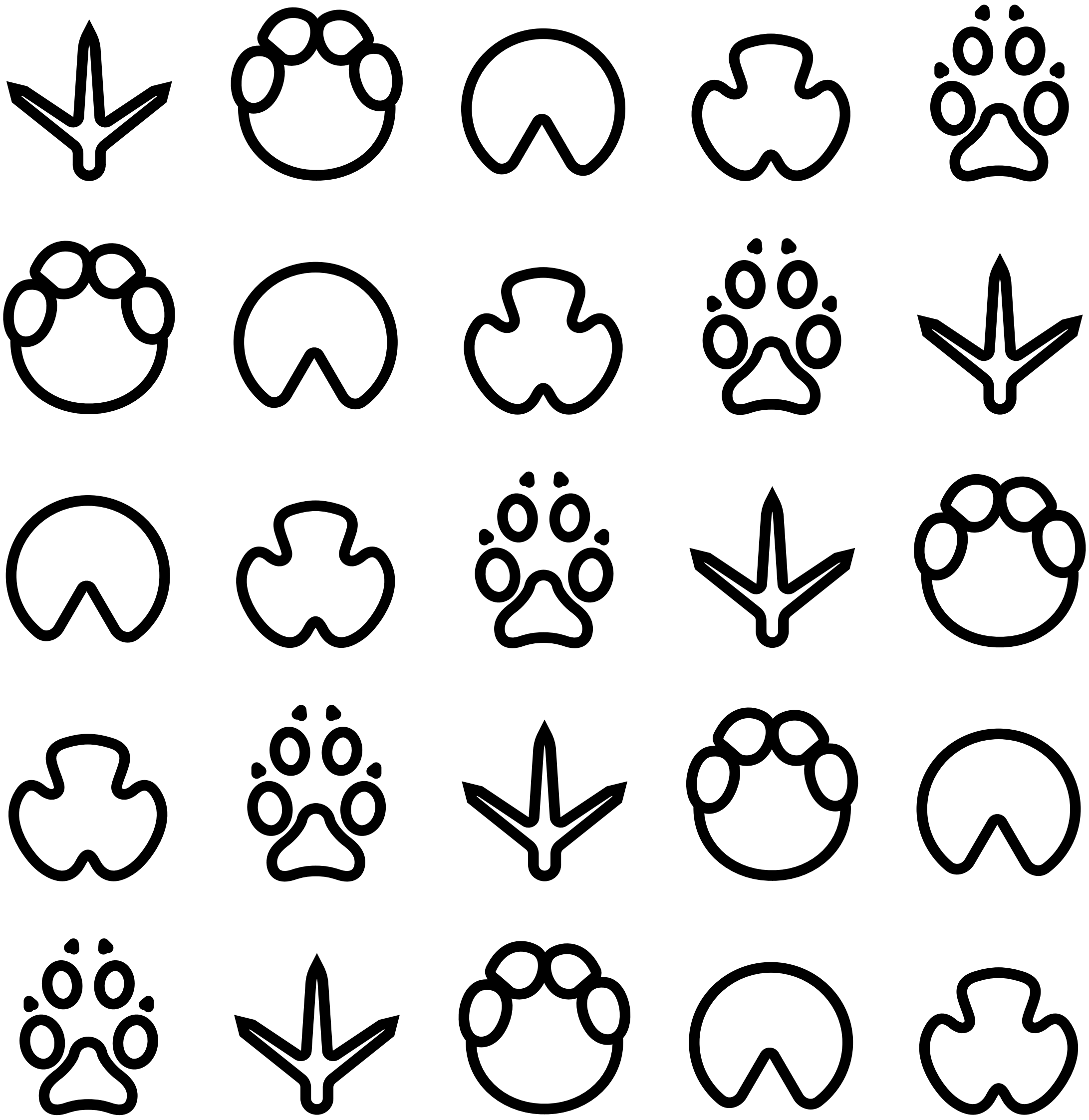
An energetic and enthusiastic innovator delivering nearly three decades of delightful user experiences, Dean Peters is a product manager and recovering software engineer working across numerous domains and technologies. He is passionate and experienced in product management, Agile, artificial intelligence, coaching, analytics, cultural transformation, mobile, natural language processing, search, software engineering, and team leadership.

Dean offers innovative and out-of-the-box approaches to effectively cut through the chaos and bring people together. His superpower is servant-leadership that helps teams create effective solutions early and continuously—all while guarding and growing the underlying culture so critical to long-term success.

Dean enjoys spending time with his family, mountain biking, and gardening at scale while trying to leverage his secret past life as an opera singer and electronic music composer to launch a product management podcast. Dean always enjoys comments, questions, and constructive criticism at his blog DeanOnDelivery.com, Twitter, or LinkedIn.

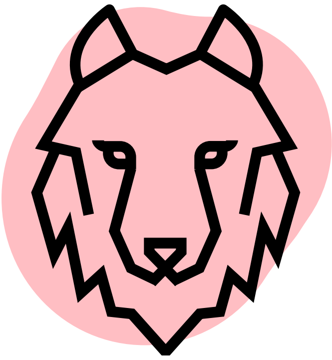


Dean Peters
Product Management &
Agile Consultant



CHAPTER 1

The Dangerous Animals of Product Management 101



WoLF: Works on Latest Fire

What is a WoLF?

The WoLF is not necessarily an individual, but a disruptive situation that requires product managers to drop everything and go into firefighting mode.

In the world of software development, there's always a negotiation between working on new features and addressing technical debt. Over-indexing on technical and security problems means you're not focusing enough attention on delivering new features and value. But the opposite—neglecting these issues to focus solely on new features—can lead to several types of fires that require your immediate and complete attention, whether it's the loss of data, security breaches, or an all-out product failure.

Why are WoLFs dangerous?

The WoLF will jump from one fire to the next, even when it comes at the expense of value delivery. In this cycle of reactivity, where all resources are dedicated to firefighting, you compromise productivity and a long-term, strategic approach to your product. Plus, you're making it impossible for your team to do anything truly new or innovative.

WoLFs in the wild

When a WoLF is left to run wild, you can end up with a wildfire that destroys your business. Take the example of Knight Capital, a financial services firm that was brought to its knees by technical debt.

In 2003, Knight Capital introduced the Power Peg algorithm. Designed to "buy high and sell low" in the stock market (the opposite of conventional "buy low and sell high" wisdom), it was meant for use in a test environment. In just one trade, the loss of a few cents would be insignificant.

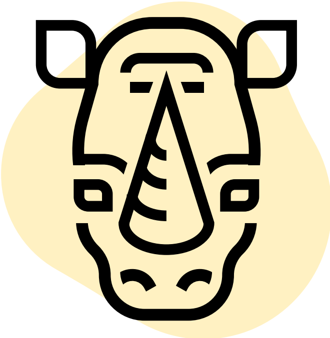
Power Peg was then left unused in Knight Capital's codebase for a very, very long time, leaving it vulnerable to the prowling WoLF.

Nine years later, Power Peg was accidentally enabled during a software update. Thousands of "buy high sell low" trades were executed each second in the real world. And it quickly added up. Within 45 minutes, Knight Capital had lost $460 million.

So, what caused this catastrophe?

1. The flag that was used to enable Power Peg in the test environment was repurposed for use in a new functionality. Thus, the program believed it was still in the test environment and executed trades as quickly as possible.
2. There was no formal code review or QA process, nothing in place to ensure the software had deployed correctly, and no thresholds to stop automated trading after a set amount of losses.
3. There was no existing catalog or naming convention for feature flags, leading to a situation where engineers had no understanding of the true nature of the feature flag.
4. There was no practice of removing old or unused feature flags.

The lesson is clear: neglecting technical debt in your product and letting a WoLF run wild can seriously cost your company.



RHiNO: Really High-value New Opportunity

What is a RHiNO?

Often, people outside of the product team—especially those in sales and marketing—can be motivated by a high-value new opportunity such as the possibility of making a sale or landing a particular customer. A RHiNO might say something like, “If we just had feature X, we’d be able to make this sale.” And while the RHiNO’s intentions are good, their approach can be myopic and disruptive.

Why are RHiNOs dangerous?

There are a few potential problems with giving in to a RHiNO’s requests. First, while they will often try to tell you that it’s simple, that’s rarely the case. Any new feature you build will take on a life of its own. It can introduce complexities to your codebase, may require modifications to your product training, or present support issues. These unintended consequences are costly and time-consuming.

Second, when you address too many unvalidated requests, you are focusing on solutions instead of identifying core problems experienced by broad range of customers. This clutters your product with features that very few customers use, impacting usability and perceived complexity. Rather than a unified set of products and features, strong go-to-market offerings, and unique differentiators, you’ll have one-offs. Product managers, meet the Frankenstein product.

When all’s said and done, there’s no guarantee that you will even make the revenue that a RHiNO is promising in the first place.

Great software is a matter of focus. You start with a vision, come up with your strategy, and finally decide on your tactics. But giving in to a RHiNO flips that approach. It jumps straight into the tactics without testing your assumptions or attempting to get validation that it's the right direction. Follow the RHiNO and you'll end up with a wandering roadmap—or worse.

RHiNOs in the wild

ScaleFactor offers a cautionary tale in what can go wrong when a RHiNO rules and you chase money instead of creating value for your customers.

In a single year, the company raised $100 million while claiming to be developing sophisticated AI software that automated bookkeeping for their customers. Instead, they secretly employed accountants who would manually complete their customers' books. Not only were they failing to deliver what they promised, but their work was often full of errors. Customers then had to redo the work or hire someone else to correct the mistakes.

In June 2020, ScaleFactor announced that it would be closing, laying off its employees and returning cash to investors.

HiPPO: Highest Paid Person's Opinion

What is a HiPPO?

A common conundrum for product managers is figuring out how to influence without authority. Product managers can't hire or fire—they answer and report to stakeholders—and VPs or CEOs ultimately control the budget and business decisions.

When these VPs, CEOs, or other high-performing individuals listen to the product manager's perspective and engage in open dialogue about the direction of the product, this is the best-case scenario. But sometimes, leaders can turn into HiPPOs.

What does this mean? "HiPPOs are leaders who are so self-assured that they need neither others' ideas nor data to affirm the correctness of their instinctual beliefs. Relying on their experience and smarts, they are quick to shoot down contradictory positions and dismissive of underlings' input," writes Chris DeRose in Forbes.

Why are HiPPOs dangerous?

Since their jobs are on the line, product managers can find it hard to contradict HiPPOs. This leads to those with more budget and power always making the rules.

Working in an environment where HiPPOs rule is disheartening for product managers. And HiPPOs don't just hurt morale, they introduce risk by pushing through products or features that haven't been validated.

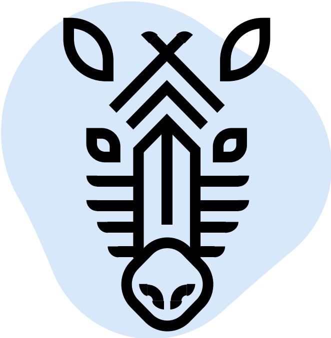
The situation might escalate into what John Cutler calls a "Feature Factory." In a Feature Factory, teams hyperfocus on output—the code and features they're shipping—without understanding or assessing the impact their work is having.

HiPPOs in the wild

The story of Ron Johnson, former CEO of JC Penney, shows us what can happen in a worst-case HiPPO scenario. While Johnson had previously been highly successful in leadership roles at Apple and Target, he made some serious errors at JC Penney. As Forbes reports, "When making changes, Johnson trusted his gut rather than the data in front of him."

Although he reportedly was shown focus group results clearly indicating consumers' strong preference for discounts, Johnson pressed ahead with mandating a fixed pricing matrix for all merchants to follow. The ensuing confusion and consumer defections were at the heart of the company's 25 percent sales drop.

During Johnson's time at JC Penney, the company burned through $1 billion in 17 months.



ZEbRA: Zero Evidence, but Really Arrogant

What is a ZEbRA?

It might be easy to look at some of these Dangerous Animals and say, “That would never be me. I don’t have the authority to be a HiPPO and I’m too clued in to be a RHiNO.” And that may be true. But we can all be guilty of being ZEbRAs from time to time—yes, even the product managers and engineers among us.

Sure, we theoretically understand the importance of having evidence to back up our opinions. But sometimes when we’re dealing in our areas of expertise, it can be tempting to skip over some of the validation process.

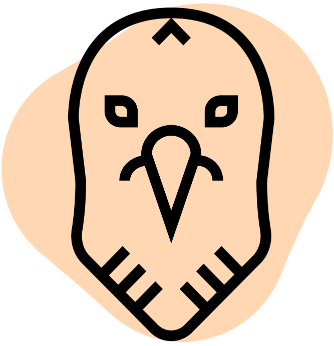
Why are ZEbRAs dangerous?

When ZEbRAs take control, you risk building products or making decisions that aren’t validated (sound familiar?). You might luck out occasionally and still get things right.
But you could also lead the product in the wrong direction. Maybe you build things that your customers don’t need. Or you miss out on key market opportunities.

ZEbRAs in the wild

Consider the case of Blockbuster CEO John Antioco, who had the opportunity to buy a small company called Netflix for $50 million. He dismissed Netflix executives’ data-driven presentation and said “The dot-com hysteria is completely overblown.” We all know how this story ends.

An Inc. article describing this exchange sums up the problem with this classic ZEbRA behavior: “...There's also an important business lesson and this is it: Never, ever be arrogant. Even if the person sitting in front of you has said something that seems absurd, even if it makes you want to laugh, pause for a moment and give it your serious consideration.”



Seagull Manager

What is a Seagull Manager?

There's no need to decode an acronym here—Seagull Managers are pretty much exactly what they sound like. They're people who tend to be a little bit removed from the day-to-day work on a project, so they occasionally swoop in, make a lot of noise, leave a big mess, then swoop out again, leaving the team to clean up after them.

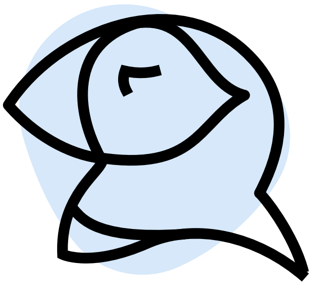
Why are Seagull Managers dangerous?

Like many other dangerous animals, Seagull Managers often have good intentions. When they swoop in, they're often trying to accelerate a process or "save the day" with their idea. The problem is that they often lack context and nuance, and they take away a team's agency and ability to find solutions on their own.

Seagull Managers in the wild

You might see a senior VP of engineering, for example, who walks in one day and tells the team, "I thought about this problem over the weekend and wrote some code for you." Their goal is to help the team, but because they have a bird's-eye view rather than an on-the-ground perspective, they might be bringing in technology that doesn't work within your infrastructure.

Dr. Travis Bradberry, author of *The Seagull Manager*, reports there are real dangers to this type of management. Employees whose managers often use Seagull-type behavior are 30% more likely to develop coronary heart disease. Employee disengagement and dissatisfaction is directly tied to company performance since it takes considerable time and money to hire replacements for employees who leave. So much for those good intentions!



PUFFIn: Plans Unending Feature Factory Initiatives

What is a PUFFIn?

Crows aren't the only birds attracted to bright, shiny objects. PUFFIns are content to jump from shiny new feature to shiny new feature. They may be guided by the desire to please others, make a sale, or, in some cases, to simply "keep the engineers busy." Regardless, the hallmark of the PUFFIn is making decisions without any concern for long-term vision or strategy.

Why are PUFFIns dangerous?

Let PUFFIns rule the roost and everyone will have plenty to do, but it's unlikely that all this work will make a positive impact on the business. Your roadmap, rather than being a strategic document that guides you towards a North Star, will just be a collection of chaotic points leading nowhere. And hustling all the time without driving any real results is one way to kill your team's motivation and engagement.

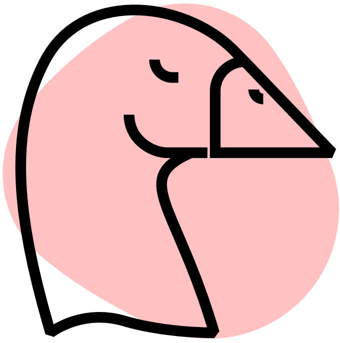
PUFFIns in the wild

Andrew Jordan knows what it's like to work in a PUFFIn environment. He was a product manager at Modern Message, which in January 2020 was acquired by RealPage, a loyalty rewards program that helps marketers and property management teams engage with their residents. The acquisition prompted the product and engineering teams to take stock of their existing processes.

When they examined the way they were working at the time, they identified a few major issues. Leadership, UX, product managers, and engineers were siloed by their roles and ideas tended to flow only in one direction. "Leadership typically came up with ideas, they then partnered with UX to flush it out. From there, a *feature* was assigned to a PM who would then partner with Engineering to plan development," writes Andrew.

While this process worked in terms of building features and shipping them (hello, Feature Factory!), it was focused more on output than outcomes. It was hard to change direction once they started working on a feature, and it put a lot of stress on engineering leads.

To get out of the Feature Factory model, they began a gradual process of change, which involved introducing cross-functional product teams, defining team charters, and eventually transitioning to a dual-track Agile model.



GOOSE: Guesstimating Overly Optimistic Scheduling Estimates

What is a GOOSE?

The GOOSE—while good-intentioned—has a shaky understanding of how long it actually takes to get things done. It may be because they're new to product management, fail to see the bigger picture, or they place too much trust in engineers who are also susceptible to being overly optimistic.

We're all guilty of being a GOOSE sometimes, believing projects will take much less time to complete than they actually do (anyone who has renovated their home can attest to this). And we often forget that it's never as simple as just building a feature and releasing it into the world—all features require care and feeding beyond the initial time of development.

Why are GOOSE dangerous?

Being overly optimistic about timelines can lead to over-promising and under-delivering. This can inconvenience, upset, or in some cases enrage our stakeholders, customers, and users. Ultimately, it erodes their trust in us. They'll be less likely to believe anything we say, which can hurt our working relationships and damage our ability to collaborate effectively.

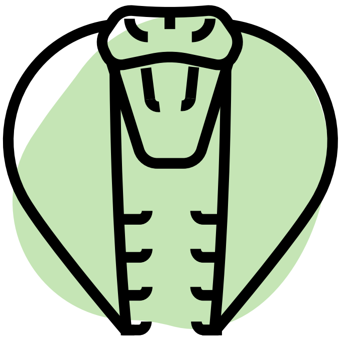
A GOOSE in the wild

MyFavorites was intended to be “the Like button for everything,” writes founder Steve Poland. You’d be able to know your friend’s favorite blog posts, beers, drinks at Starbucks, things to do in Baltimore, etc. And you wouldn’t just be limited to following your friends, either—it’d be possible to follow celebrities and brands, too.

Instead of choosing a single platform, MyFavorites was trying to build an iPhone app, an Android app, and a website where people could favorite things. “It was all too much,” writes Steve. Their dev platform of choice, Titanium Appcelerator, was supposed to allow them to write everything at once and push it out to both iPhone and Android. Turns out, it takes a lot of work to manipulate features for both of these operating systems.

Adding even more to the GOOSE scenario was the fact that the team was trying to build a web app at the same time. Steve says, “Having a web app being created at the same time was ridiculous too—especially since we still hadn’t nailed down the favoriting process or tried it with any users. I was blowing cash at a ridiculous pace. I had seven guys working on this thing at once, as we were hustling for the SXSW launch deadline.”

Reflecting on the experience now, he says, “Focus on one platform. Get it out there—let people use it—nail down the UX with user input.”



CoBRA: Cognitive Bias Related Assertions

What is a CoBRA?

The CoBRA can strike at any time, and often its work is invisible and hard to detect.

There are dozens of cognitive biases that can impact our decision-making without us even knowing. And while making snap decisions can keep us out of harm's way in the real world, we should take more time and be thoughtful about our judgments in the business world. A few common biases to look out for include:

- **Confirmation bias,** where we only notice and remember information that confirms our beliefs (and ignore or forget anything that disproves them)
- **Sunk-cost or escalation of commitment bias**, when we've already invested time and money into an idea and feel we must continue to pursue it
- **Experience bias,** where we believe that since something worked for us in the past that it will work for us in the future, too
- **The escalation of commitment,** when we don't want to give up on an idea we've already invested time and money into, even if the initial results indicate it won't be successful
- **Optimism bias,** which leads us to project a rosy picture of the future and fail to account for negative outcomes

Why are CoBRAs dangerous?

Biases lead us to make snap judgments, which we should really only be doing in very specific circumstances (like when we're dealing with an actual or metaphorical fire). In the majority of cases, snap judgments will skew our interpretation of evidence and our execution of tasks. This can lead to building the wrong thing or unintentionally leaving out—or even harming—certain groups.

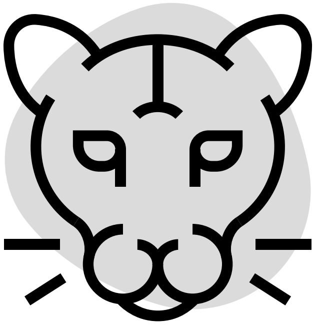
CoBRAs in the wild

Robert Rosenberg was the CEO of Dunkin' Donuts. He'd come into the job at 25, and by his own admission still had a lot of growing up to do. He almost took the business off a cliff by changing the mission from a focused coffee and donut shop business to a chain of franchises.

"I got seduced by Wall Street and wanted to keep the earnings growing at 50% a year, which is impossible," he told Christopher Lochhead in a podcast interview. He says his youth and cockiness led many franchisees to become disgruntled and he had started to lose focus on Dunkin' since he'd started other chains.

Luckily, at the time he happened to read David Halberstam's book *The Best and The Brightest* about the Kennedy administration during the Vietnam War. The gist was that the best and the brightest in the U.S. were suffering from hubris, which led to the catastrophic outcome of the war. And when Robert read that, he said, "He could be talking about me!"

This transformational moment led him to go back to senior leaders at Dunkin' Donuts, acknowledge what was happening, and create processes that would allow franchisees to share their own experiences and feedback. Instead of making all the strategy decisions himself (which was causing significant problems for the company), he created lines of communication with the franchisees, managers, and stores.



PUMA: Promotes Unusually Meaningless Assumptions

What is a PUMA?

PUMAs use their lightning-quick reflexes to pounce on an idea or solution. But those sharp claws only scratch the surface. After talking to a single customer or looking at a single data point, they're ready to make a decision. Or perhaps they see a competitor is doing something and scramble to one-up them. PUMAs rely on assumptions and arrogance instead of digging deeper into what a customer, user, or data point is really saying.

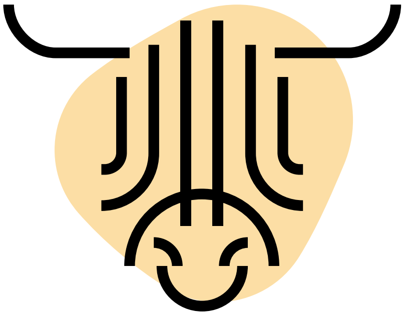
Why are PUMAs dangerous?

Relying on a single data point or signal is unlikely to lead you in the right direction. So many ideas fail to gain traction, and this happens much more when we haven't taken the time to validate them first. Letting the PUMA chart your course can ultimately lead to a lot of wasted effort and loss of team morale.

PUMAs in the wild

Votertide was a social media analytics company that tracked data from social media in real time to find out how people were reacting to political figures and issues. After some initial investment and fanfare, the company ran out of money as headcount and burn rate grew. Reflecting on what went wrong, Jimmy says he relied too much on assumptions and didn't talk to customers enough. "I wasn't getting out there in the field to understand what people wanted and what was good and what was bad."

He says the insights they got from customers were what led them to build a few key things that were by far the best-used features. And the features he personally thought were great were the ones that nobody cared about. "Don't waste your time on something that you have a gut feeling about," he now says.



YAK: Yet Another KPI

What is a YAK?

YAKs have quite the appetite for metrics and numbers. So much of an appetite, in fact, that they'll keep gorging on them until they make themselves sick. Metrics can be a good thing, but they can also be gamed or turn into noise if they're not aligned with a desired behavior or actual business goals.

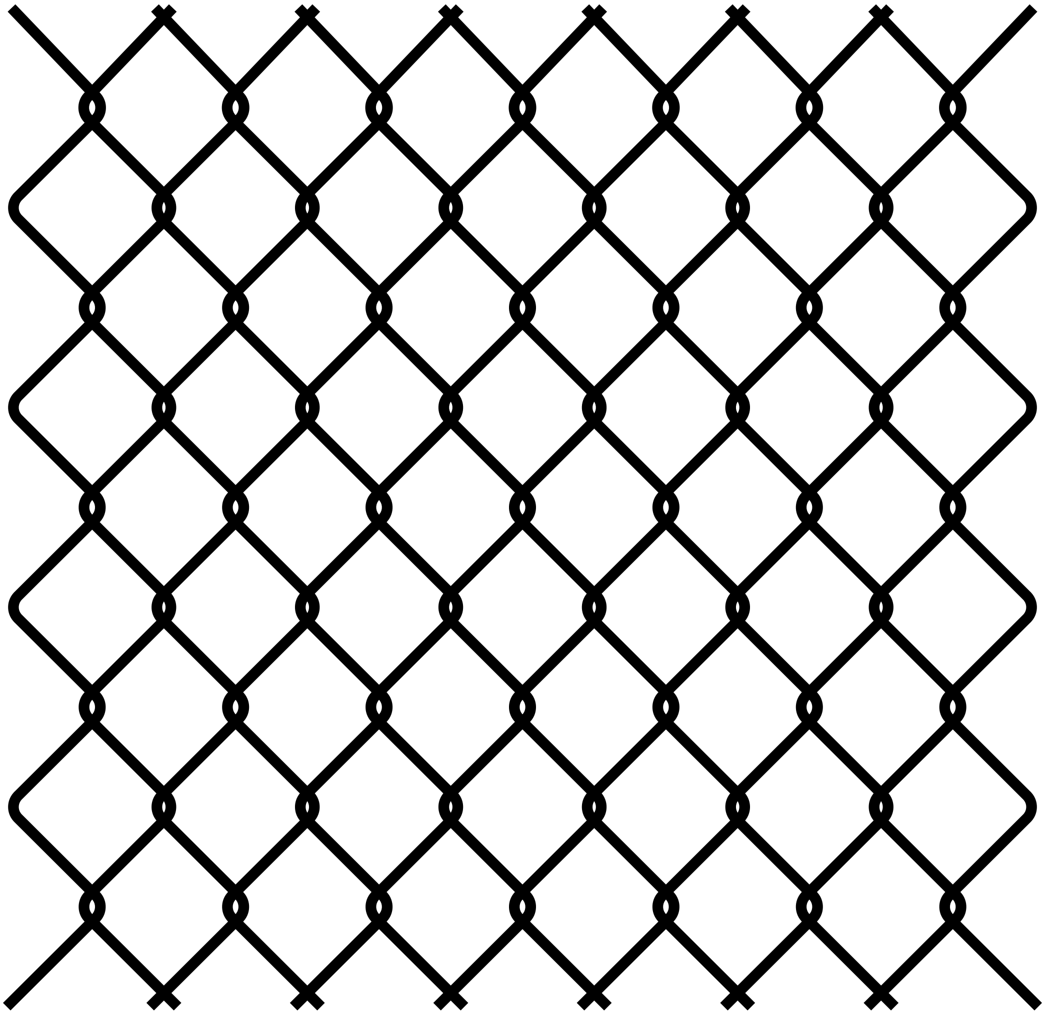
Why are YAKs dangerous?

Measuring the wrong thing, like vanity metrics, can give you a false sense of security and lead you in the wrong direction. In some cases, it can even incentivize bad behavior or lead to people gaming the system. In the early days of software development, for example, using lines of code as a metric encouraged developers to write inefficient and overly complicated code.

YAKs in the wild

Stayzilla, an India-based home aggregator, suffered a YAK on the loose and announced they would halt operations in hopes of rebooting in the future. The former CEO Yogendra Vasupal admits: "The initial seven years were all about having negative working capital, positive cash flow, and a sustained ability to fund our own growth. Those were the only metrics we tracked. In the last three to four, though, I can honestly state that somewhere I lost my path. I started treasuring GMV [Gross Merchandise Value], room-nights, and other 'vanity' metrics instead of the fundamentals of cash flow and working capital."

Reflecting on why this happened, Yogendra writes, "while there exist many benchmarks for valuation of a company, its intrinsic value starts from inside and is tied very closely to the metrics that founders value and their comfort with that selection." And while he'd realized that Stayzilla had veered too far from their initial value system, it was too late. "12 months was just not enough time to shift paths, when we were already 36 months down a different path."



CHAPTER 2

The Dangerous Animal handler's toolbox

It's important to note that the Dangerous Animals of Product Management are usually well-intentioned. Still, when you give them too much power, it's easy to lose sight of your priorities. Or end up misaligned.

To avoid being devoured, product managers must master *influence without authority*—an art that involves a mix of hard skills and soft skills. And while each animal requires a different approach, dealing with them usually involves using one or more of the following skills and tactics.

Skills for managing the Dangerous Animals — an overview

1. Exercise your empathy

The best starting point for most interactions with a Dangerous Animal is to understand their perspective. Finding out where they're coming can help you better align their motivations with your product goals.

WORKS BEST FOR: HiPPO, RHiNO, Seagull, PUMA, and GOOSE

2. Embrace transparency

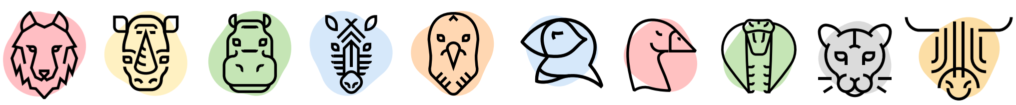
Alignment is easier when everyone understands not just the WHAT, but also the WHY of your product management process. Go the extra mile to provide visibility into your tradeoffs, decisions, customer insights, data, and more. Context like this can empower your dangerous animals to come to the table with more-informed ideas.

WORKS BEST FOR: WoLF, HiPPO, RHiNO, GOOSE, and PUFFIn

3. Empower stakeholders with technical know-how

They say a little knowledge can be dangerous, but not having enough can be even worse. Giving your animals enough technical insight—especially about the impact of technical debt—can help them understand why an idea may not work or be the right choice for your product.

WORKS BEST FOR: The entire zoo!



4. Do "tiny acts of discovery"

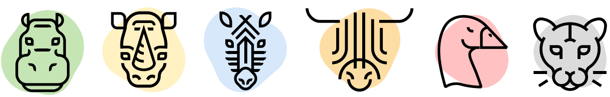
"Tiny acts of discovery" is a concept coined by Dean Peters, the Dangerous Animals "zookeeper." Whether you first release updates to a limited audience, use A/B tests to try out new ideas, or research what competitors are up to, simple acts of discovery can quickly support or challenge the assumptions made by your Dangerous Animals. You can also do tiny acts of discovery by regularly paying off your technical debt.

WORKS BEST FOR: RHiNO, ZEbRA, Seagull, GOOSE, and YAK

5. Train stakeholders to think like product managers

Develop and use simple frameworks to evaluate new ideas, then invite your Dangerous Animals to join in the exercise. These processes can feel more like objective assessments, taking the emotion and ego out of product decisions. It also helps to demonstrate what it looks like to seek underlying customer needs. Ideally, Dangerous Animals will start trying to understand customer needs themselves

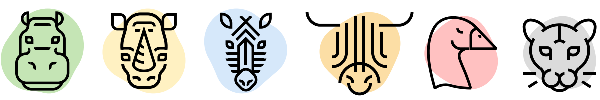
WORKS BEST FOR: HiPPO, RHiNO, ZEbRA, YAK, GOOSE, and PUMA

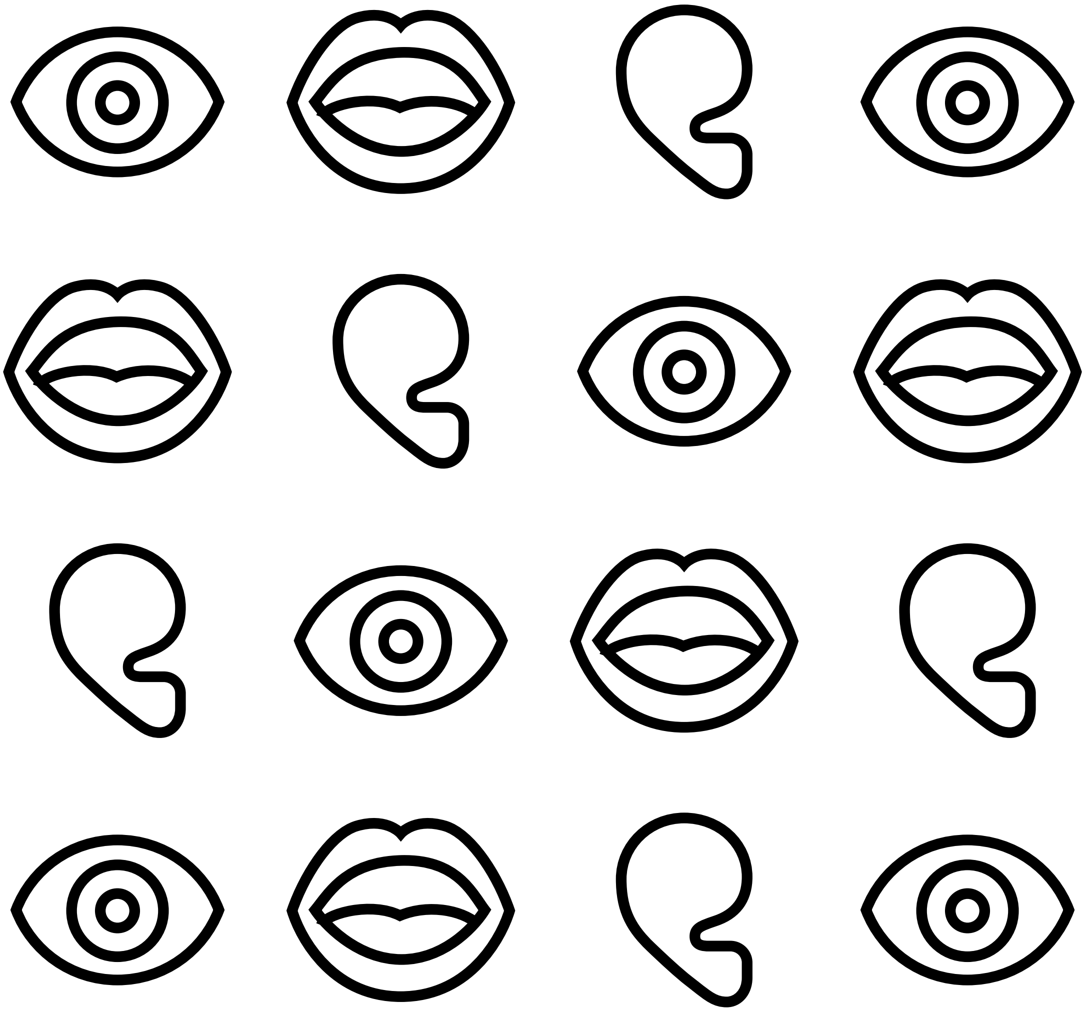


6. Connect proof with purpose

Whatever arguments and evidence you or your Dangerous Animals put forward in favor of or against an idea, always tie them back to your company's mission. If it doesn't fit with your business purpose, then it probably doesn't belong in your product.

Works best for: HiPPO, RHiNO, ZEbRA, YAK, GOOSE, and PUMA

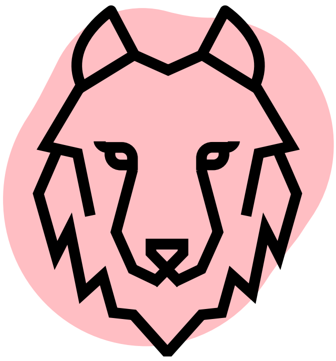




CHAPTER 3

Soft skills for managing stakeholders

Mastering influence without authority



Keeping the WoLF (Works on Latest Fire) from the door

Remember, a pack of WoLFs howling about technical issues is usually a symptom of more systemic product issues. While the immediate challenge is to put out the latest blaze, your ultimate goal is to reduce both the number of future fires and their destructive power. Here are two ways to keep the WoLF from the door.

1. Use "firebreaks" to limit future damage

Having rigorous code reviews and QA testing is essential to avoiding catastrophes like Knight Capital's "45 minutes of hell" or more mundane bugs that frustrate users. But even the most thorough testing cannot replicate a live product environment. While you'll never completely eliminate the possibility of technical problems, you *can* reduce their impact.

Firebreaks are a common fire management technique that involve creating empty spaces around vulnerable forest areas or buildings that can slow or even stop the spread of flames. When you first launch a new feature or release an update, look to contain it in a specific area so you can control the inferno should things go wrong.

Work with your DevOps and IT teams to understand which segments, regions, or other limiting factors you can use to restrict a new live deployment. This way you can be sure that it won't bring down the entire service. You could:

- Release new code to a limited segment of users
- Launch in an off-hours time zone where users are less likely to be online
- Share with a beta test group that will be more tolerant of bugs

The key is to release the update in a way that you can reverse it quickly if you notice an issue. Then you can work with DevOps and IT to analyze whether it was a problem with the new feature or if it was due to a knock-on effect to the rest of the product.

Meanwhile, you've limited the number of customers that get burned by the experience, which should keep the WoLF from your door.

2. Pay down your technical debt regularly

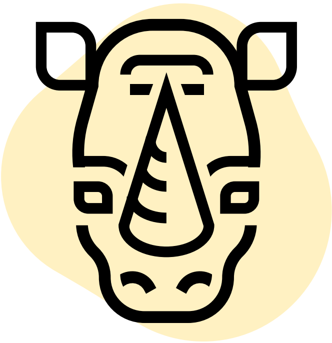
Every product manager is burdened by technical debt: the shortcuts, quick fixes, and duplications in existing code that make new updates harder to implement.

Technical debt is like borrowing money from a loan shark who charges an extremely high rate of interest that compounds dramatically with each passing week. One day, when the loan shark decides that it's payback time, your product ends up crippled by the amount of debt it has accrued.

In the case of Knight Capital, that $460 million loss was all down to the mistaken deployment of an obsolete feature that was forgetten in the codebase for nine years.

To avoid unexpected combustions of built-up technical debt, build regular fixes into your workflow. Every release of exciting new features should include at least one dull but necessary fix of a long-standing issue.

Few people will understand the point of dedicating resources and budget to paying off technical debt. You'll need to explain how WoLFs are always on the prowl, ready to pounce. Unless you reinforce your weaknesses, the waiting pack can devour your product.



How to stop a charging RHiNO (Really High-value New Opportunity)

A salesperson with a big client deal in their sights can be as unstoppable as a charging RHiNO. To avoid being trampled, work side-by-side with RHiNOs rather than trying to face them head-on. These three tactics can help turn charging RHiNOs into calmer, more collaborative creatures.

1. Serve a target market, not individual customers

When you address one-off requests—like those from a RhiNO—you shift the focus to solutions. What you need to do is identify core problems that are experienced by a broad range of customers. After all, your job as a product manager is to meet the overlapping needs of your target market, not individuals.

2. Share your technical know-how

Salespeople often instinctively respond to a potential client's new feature suggestion with an assured "we can make that happen". Usually because they have no idea of what it would actually involve. Empowering RHiNOs with enough technical knowledge can give them a sense of the different complexities involved and help them be more realistic with customers. Not making a bold claim in the first place is much better than having to backtrack on it later.

3. Look for validation

Sales and marketing like to keep up with what the competition is doing. Assign your RHiNO the task of researching whether similar products offer the features or services they've requested. If it's industry table stakes or a potential differentiator, the idea might be worth pursuing.

Find quick and easy ways to assess the assumption your RHiNO and their client has made about the value of their requested feature—what "zookeeper" Dean Peters calls "tiny acts of discovery". Think simple A/B tests, interviewing current customers, getting feedback on hand-drawn wireframes, and more.

Satisfying a hungry HiPPO (Highest Paid Person's Opinion)

Unfortunately, objecting to a HiPPO's idea is essentially telling a leader that their instincts are wrong—a no-no in most people's books. Instead, the key to satisfying HiPPOs is to get them to change their own mind.

Here are three ways to give them a taste of their suggestion in action and help them realize they're not that hungry after all.

1. Take their side

You need to be the HiPPO's partner in discovery, not an adversary. Frame the next steps in affirmatory language like "confirming your hypothesis," then lay out a clear plan of quick and easy discovery activities. If the results do not back up the HiPPO's idea, they'll know that you considered their ideas and that your intentions were positive.

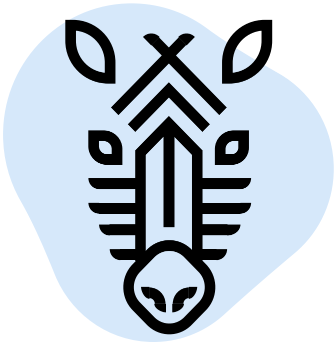
2. Amass a mixture of evidence

You can't always rely on one data source to change your HiPPO's mind. Remember that Ron Johnson, former CEO of JC Penney, backed his own instincts instead of the focus group results suggesting that his strategy for JC Penney was flawed.

Instead, arm yourself with a mix of evidence, including customer testimonies and opinions from the broader business.

3. Start defining success

If your HiPPO is still chomping at the bit despite a lack of validation, try discussing what success looks like. Setting out specific targets of what this idea should achieve can help push the plan from a simple pipedream to an actual project with numbers to hit. Only an extremely bullish HiPPO will proceed with an idea when its success is in doubt.



Make ZEbRAs (Zero Evidence, but Really Arrogant) earn their stripes

So far, one of the keys to dealing with the Dangerous Animals of Product Management has been the ability to see the world through their eyes. However, empathizing with a ZEbRA can be challenging because they don't bring any backup of their plans to the table.

As empathy is less likely to help, challenging a ZEbRA can require a slightly more confrontational stance. Try these three tactics to make them earn their stripes.

1. Produce your own evidence

When ZEbRAs submit suggestions that aren't validated, your first step is to ask them to provide evidence of their claims. If they have none, bring you own. The burden of proof then shifts to them if they want to stay part of the discussion.

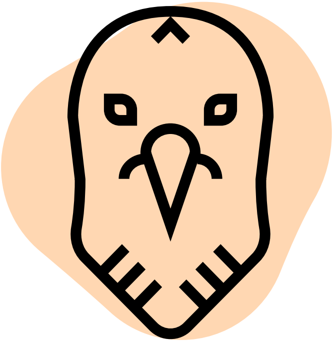
2. Gather more voices

ZEbRAs can easily assume their opinion is better than that of one other person. But as more voices chime in to contradict them, you can chip away at their sense of superiority. If the situation is important enough, elevate the conversation to senior levels. After all, stripes are very important to ZEbRAs.

3. Shift the suggestion

If possible, try to move the ZEbRA's idea towards something that's more in line with your strategy. Instead of challenging their ego, you'll need to feed it by making them think the new suggestion is still essentially their request.

It's not the most satisfying way of taking on a ZEbRA, but as long as you're doing right by your product vision, it will be worth it.



Turn Seagull Managers from pest to partner

Unlike the other Dangerous Animals, your goal is not necessarily to stop a Seagull Manager from helping out. However, you will need to optimize their value.

Here are two ways to ensure that your Seagull Manager is giving direction, not directives.

1. Validate their proposals

Seagull Managers typically have strong domain knowledge, which is why they assume that coming to you with a ready-made solution is helpful. What is truly useful to your team is the insight that led them to that specific suggestion.

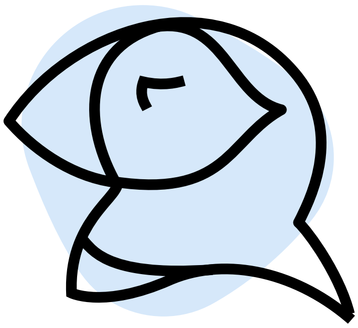
Ask Seagull Managers about the problem they envision solving with their idea. Once you validate that their area of focus is aligned with your product strategy, collaborate *together* on a context-rich solution.

2. Slow the swoop rate

Over the long-term, you can reduce how often your Seagull Manager feels it necessary to swoop in. You just need to provide the right amount of transparency.

Find ways to connect them with the day-to-day of product development so they retain critical context and nuance. Then you can regularly access those gems of experience and insight that will help your team create the right solutions.

It's a win-win for everyone.



Help PUFFIns (Plans Unending Feature Factory Initiatives) set their sights on strategy

Left to their own devices, PUFFIns won't have any trouble keeping themselves busy. But they're simply checking items off a to-do list rather than making an impact on the business. Try a few of these tactics to help them align their work to actual business outcomes.

1. Recognize when runways are at risk

Feature Factories tend to occur when runways are at risk. For example, when funding is running out, sales aren't being made, or something like COVID catches everyone by surprise.

While you can't prevent these things from happening, you can try to avoid being reactive when they do. When you find teams chasing every dollar or task as a quick-fix solution, remind them why you're focusing on higher impact activities instead.

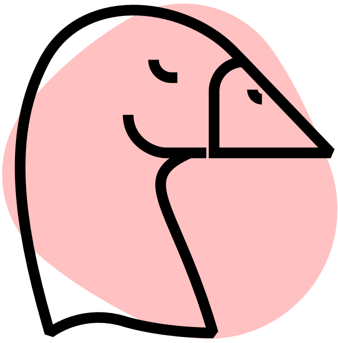
2. Paint a clear picture of your product vision and strategy (and evangelize the heck out of it)

One of the best ways to keep everyone aligned on your product vision and strategy is by having a key artifact like a strategic roadmap that focuses on outcomes rather than outputs. Make sure everyone has access so they never veer too far from your North Star.

With a tool like Productboard, you can invite stakeholders from across the organization to view or contribute to your roadmaps. You can tailor specific roadmaps to their needs, and they won't see any roadmaps that are private or that have only been shared with specific colleagues.

3. Validate, validate, validate

Make sure your work is led by a validation process. This can be the "tiny acts of discovery", seeking input from your customer advisory board, or anything else that helps keep your users and desired outcomes top of mind.



Guide your GOOSE (Guesstimating Overly Optimistic Scheduling Estimates) to more accurate estimations

Engineering math is highly optimistic. This often occurs because engineers think in terms of code being written, but just because the code is done doesn't mean the delivery of value is done. As a product person, you need enough technical understanding so you're aware of the constraints and impact of different decisions.

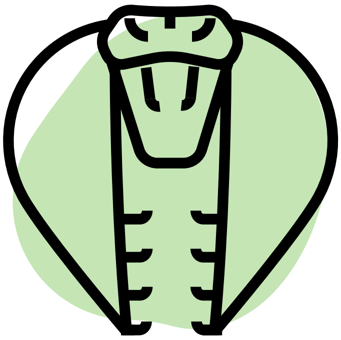
Try these tactics for generating more accurate estimations.

1. Don't estimate with tunnel vision

Keep in mind that there are different factors to consider like quality assurance, go-to-market, and onboarding for your Customer Success team—and don't forget to account for all these factors and steps in your timelines. It can help to keep a history of past estimations—both good and bad—and review these so they can inform your future estimates.

2. Keep the stories tiny

Make sure that your user stories are as tiny as possible. You're looking for one outcome for your user based on a set of given conditions, one "when" and one "then" (as in "Given this set of conditions, when the user does X, Y happens"). Any more than this and you could be aiming too high, which could slow down progress and impact your timeline.



Remove the fangs from cognitive biases so CoBRAs (Cognitive Bias Related Assertions) aren't so dangerous

Cognitive biases came about to help us avoid getting eaten in the wild. If we see a tiger about to charge or a fire that's going to burn our hand, we move quickly out of a self-preservation instinct. But in the business world, we're rarely dealing with actual fires, so we can (and should) take the time to be more thoughtful about our decisions.

1. Understand how common cognitive biases work

There are hundreds of cognitive biases and you definitely don't need to catalog all of them, but it can be helpful to familiarize yourself with some of the common ones that are likely to impact your work.

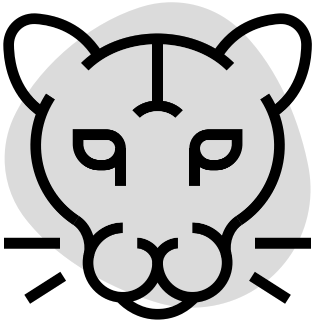
For example, confirmation bias occurs when we only highlight or remember information that confirms our pre-existing beliefs (and ignore or forget anything that disproves them). The escalation of commitment or sunk cost bias occurs when we feel like we've already invested a lot of time or resources into something and don't want to give up, even if all signs are showing that it's not working.

2. Negotiate more time for learning

The more time pressure we are under, the more susceptible we are to cognitive bias since it's a form of mental shortcut. To limit this, try to break large tasks into smaller ones so you can buy yourself some time to experiment. Look for ways to take a "now, next, later" approach so you don't have to make all your decisions at once.

3. Invite in other voices and expand the conversation

Whenever you can, don't make decisions on your own. Seek input from other sources, whether it's your broader organization, feedback from your customers, past conversations, or anything else that helps you broaden your perspective.



Prevent PUMAs (Promotes Unusually Meaningless Assumptions) from pouncing too quickly

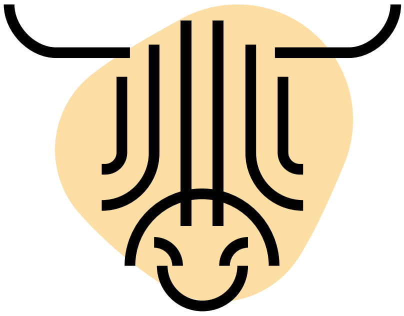
Similarly to the CoBRA, PUMAs like to make snap decisions and move fast. They'll often hear a single anecdote, data point, or tactic and immediately want to act on it without taking the time to validate their assumption. But it doesn't matter how fast you're going if you're moving in the wrong direction.

1. Stay in the problem space for longer

Customers often speak in solutions because they don't know any other language. But it's our job as product people to translate their solution space language into problem space language. Make sure to ask lots of questions so you fully understand their context and motivation. An exercise like the "five whys" can really help here.

2. Once again, try "tiny acts of discovery"!

Try to elevate assumptions into hypotheses so you can test them and measure the results. Remember, this doesn't need to be a full-blown mockup or anything that's labor intensive to create. Your goal is to run small experiments on an ongoing basis so you can collect results and add data to the conversation.



Give YAKs (Yet Another KPI) something meaty to chew on

YAKs are quick to jump on any kind of number or metric. But just because something has a number attached to it doesn't necessarily mean you should pursue it. Here are a few tips to help ensure your KPIs are actually beneficial to your users—and your business.

1. Make sure you're setting outcome-driven metrics

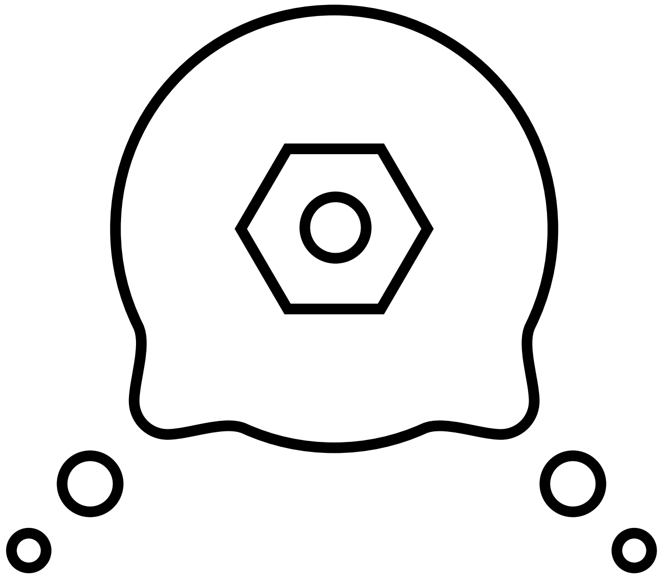
Remember that metrics can have both intended and unintended consequences. Assessing engineers on lines of code written doesn't make your software better, it just incentivizes writing more code.

Whenever you're tempted to set a new metric, make sure you can tie it to desired outcomes for your users and your product.

2. Reframe the organizational mindset on metrics

You can run into trouble when your organization uses metrics to judge or punish people. There should be psychological safety around metrics so people aren't afraid to share "bad news."

Don't forget the "I" in "KPI" stands for "indicator," which means it's important to treat your KPIs as opportunities to course correct rather than immutable final results.



CHAPTER 4

Processes & frameworks

Training Dangerous Animals to think like product people

While empathy, evidence, and experiments go a long way in controlling the occasional dangerous animal rampage, the best long-term strategy is to help them think more like product managers. When executives, salespeople, and other stakeholders have a clear understanding of how your team assesses and prioritizes new ideas, they may begin to evaluate their own thoughts before suggesting them to you.

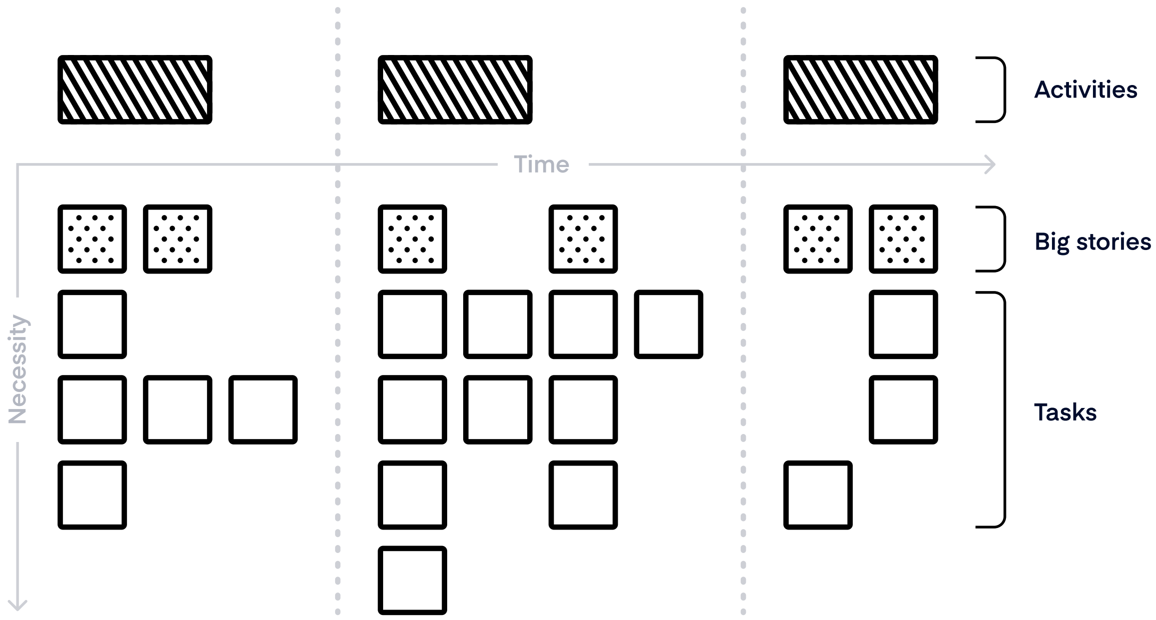
To train Dangerous Animals to think like product people, invite them to take part in short sessions where you use some simple processes to evaluate new ideas as a team.

The following four frameworks all give structure to conversations about feature requests and facilitate objective decision-making based on clear prioritization criteria.

1. Story mapping

Looking at the world from the perspective of your users and the outcomes they want and need can often clarify what your product should be. The key is for everyone at your business to develop a shared understanding of what those needs and pain points actually are.

Story mapping is the process of externalizing this thinking, usually with a blank wall and a lot of post-it notes, but now often using virtual whiteboarding tools like Miro. Having this shared view of what you're trying to achieve gets everyone on the same page and makes it easier to integrate and refine ideas from a range of stakeholders.



Adapted from Folding Burritos

2. Cynefin

Whether you're trying to solve a problem or integrate new ideas, it's not always easy to know what needs to be done. Sometimes things can be confusing and your next step is unclear. Other times, they can feel too complex to know where to begin.

Cynefin is an exercise that helps bring clarity to these situations by categorizing where you currently stand, from Clear to Confusion and Complex to Chaos. Once you establish your category, the process defines what types of action to take next, which is especially useful in more difficult scenarios.

Complex

- Enabling constraints
- Loosley coupled
- probe-sense-respond

Emergent practice

Complicated

- Governing constraints
- Tightly coupled
- sense-analyze-respond

Good Practice

Chaotic

- Lacking constraint
- De-coupled
- act-sense-respond

Novel practice

Obvious

- Tightly constrained
- No degrees of freedom
- sense-categorize-respond

Best practice

Adapted from Wikipedia

3. IDEA/E

Standardized scoring systems can bring more rigor to decision-making and reduce the possibility of bruised egos when ideas are refined or rejected. The simpler the system, the better. IDEA/E can work by just answering YES or NO to five questions:

IMPACT	How does this issue affect customers?
DISSATISFACTION	Is it a big deal rather than just a minor annoyance?
EVIDENCE	Does the data show that it impacts a lot of our customers?
ADVANTAGE	Will a solution benefit us?
EFFORT	Can we achieve this quickly?

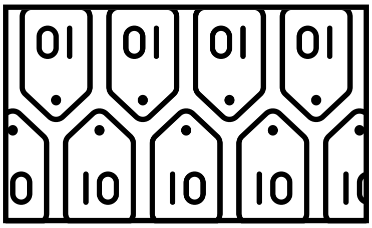
	MARKET			ORGANIZATION		
	Impact of problem	Dissatisfaction with current situation	Evidence	Advantage to us	Effort to deliver	PRIORITY
Use 1-5 or Fibonacci sequence	Where 1 = low importance	Where 1 = totally satisfied	Where 1 = few; 5 = all	Where 1 = low advantage	Where 1 = easy (or small)	I*D*E*A/E

Adapted from Steve Johnson, Under10 Consulting

4. Buy a Feature

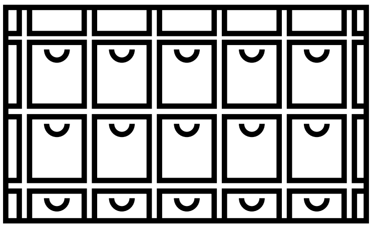
It's often useful to remind product stakeholders of the very real trade-off between chasing opportunities and managing resources. Buy a Feature is a fun and effective way to tease out the true value of competing ideas.

The game involves putting a price on each potential feature, then giving each stakeholder a set budget to purchase their favorites. Pooling their money helps them afford more expensive features but requires negotiation and compromise over how to optimize their limited resources so everyone gets what they need.

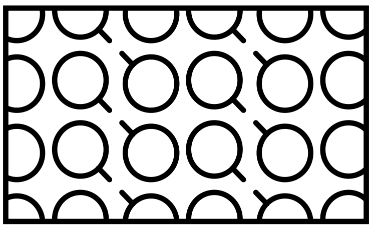


Make a list and price each item. Your list can be about anything.

Give your players some money and invite them to the game.



Players then collaboratively purchase the items they think are most important.



After the game, you can review the players' purchases to learn how they prioritized your list and why.

Adapted from InnovationGames.com

Conclusion

Unlike facing down a real-life beast, the key to handling the Dangerous Animals of Product Management is always a conversation. Only by creating a shared understanding around the opportunities and challenges of each new idea can product managers continue to do what's right for the product and its users—without leaving a trail of disgruntled stakeholders.

The best conversations generate empathy. Even if there is disagreement about the value of an idea, each party should end up with an understanding of why one wants it done or the other does not believe it's a priority.

All of these conversations, disputes, experiments, and frameworks can finally be resolved by your product's ultimate arbitrator. If a suggested new feature or fix does not fit with the overall vision for what your users want and need, it does not belong in the product.

At the end of the day, the Dangerous Animals of Product Management should never be more powerful than your customers.

About Productboard

Productboard is a customer-centric product management platform that helps organizations get the right products to market, faster. Over 5,400 companies, including Microsoft, Zoom, 1-800-Contacts, and UiPath, use Productboard to understand what users need, prioritize what to build next, and rally everyone around their roadmap. With offices in San Francisco, Prague, and Vancouver, Productboard is backed by leading investors like Dragoneer Investment Group, Tiger Global Management, Index Ventures, Kleiner Perkins, Sequoia Capital, Bessemer Venture Partners, and Credo Ventures.

Learn more at productboard.com

The Dangerous Animals Of Product Management

The art of managing stakeholders & staying true to your product strategy

A Productboard publication in partnership with Dean Peters

productboard