PERFORCE

WHITE PAPER

6 Tips for a Leaner Product Backlog

Introduction

Your backlog is, in essence, the future of your product. When it becomes oversized (or overloaded), you risk delaying time to market and compromising quality. In this paper, learn how to restore your backlog, get back on track, and unleash innovation.

The Backlog and the Modern Paradox of Innovation

Yes, we can now design, build, and deliver more, with greater efficiency, and across barriers of space and time than any of us thought possible merely a decade ago.

But has our ability to generate and collect hard drives of ideas paradoxically become one of the biggest impediments to organizational innovation?

Put another way:

Are bad, or worse — mediocre, ideas dimming the bright sparks of innovation in your product or company?

There is no better place to discover and resolve this paradox than in your product backlog, the place where ideas mature into innovations.

In this whitepaper, we'll share why your backlog is so important, what is threatened, how to prevent backlog overload, and what to do if your backlog is holding back what could be a happier, more productive team and a better, more profitable product.

3 Common Problems of Overloaded Backlogs

In an Agile development organization, the main tool for managing and implementing the roadmap is the backlog. When considering a backlog size, we encounter two opposite extremes. One, it's empty. Two, it's overloaded.

If it's empty, consider yourself either lucky, extremely productive, or both. You and your team are delivering on-time and on-demand and, in all likelihood, outpacing your competition. If, however, the opposite is true, — if it has grown uncontrollably, if it cannot be looked at without shaking your head — then it is endangering your product.

Ask any Agile team and they'll nod their heads that an unreasonably large and unkempt backlog creates fear, stress, and anxiety. Eventually, that oversized backlog will be abandoned by that team. Without a proactive vision, they will resort to reactive sprint planning, where the sheer amount of information becomes unmanageable and irrelevant. Such teams will lose sight of long-term goals and end up in task mode — because it's far easier, and less stressful, to think about what task you need to complete right now than to consider what string of connected tasks will accumulate into an important achievement.

An oversized backlog causes many problems. Here are three notable ones.

1. MAINTENANCE COSTS FROM QUEUES

How many user stories is a team member working on at any given time? When the backlog becomes oversized, the team can easily become overcommitted, and the amount of time it takes to deliver any given item prolongs. Once part of a backlog is committed, it becomes a queue and start following the rules of queuing theory, which means it requires a small amount of continuous attention to remain valid. Attention, as we all know, is a finite resource.

What makes a queue especially problematic is that very few of us actually see them. In fact, we suffer a blindness to queues (Reinertsen, “The Principles of Product Development Flow,” 2009, p. 5-6). Queues sneak into the gears of productivity and bind them up.

2. INFORMATION-VALUE REDUCTION

Grooming an oversized backlog can seem like an insurmountable task and, as with most activities that appear unconquerable, we often avoid it. When this happens for too long a time, the entire backlog becomes obsolete. Like a zombie, it represents a living-dead product.

More than that, an individual item, cast into a backlog of tens of thousands of items, feels massless and insignificant — no matter how robust or brilliant or transformational. Adding it will feel pointless. In such cases, the backlog becomes an oversized trash bin. It is where all work you want to do, but never will, ends up.

3. INHIBITED INNOVATION

Oversized backlogs compromise innovation. Since reorganizing an oversized backlog is considered a chore (which it is), brilliant ideas are either added at the front or the end of the backlog. When you add items to the front, you are, in effect, invalidating the rest of the backlog. On the contrary, by doing the opposite and adding them to the end of the backlog, you all but ensure they will never be completed due to problems stemming from the backlog size.

While stifling innovation is, in itself, a concern for the company, we can’t forget the source of this problem, which is that team members have become demotivated to contribute their best ideas. Their productivity declines as they begin searching the job boards.

Is Your Backlog Too Big? 2 Formulas to Find Out

When does a backlog become oversized? How big is too big? Do you need to undertake remedial steps immediately? Or simply begin to monitor its size so you don’t reach the point of no return? If you don’t know — if your gut alone doesn’t give you direction — here are a couple methods for measuring a backlog.

METHOD 1: BACKLOG ITEMS PER QUARTER

For the sake of comparison (and knowing that each company, product owner, and team is different), here is a typical example of a maximum backlog size. Pushing far beyond this number may mean you’re hitting your threshold and should become more proactive about backlog management.

Let’s begin with the product owner, who is responsible to maintain a backlog for the next six to eight sprints. As for the team, let’s assume they are working in two-week sprints, and in the last couple of sprints they completed an average six to eight user stories. That means your maximum backlog should be seven stories times seven sprints, which equals 49 stories. That will take you 14 weeks (seven sprints times two weeks) to complete. So, if you have substantially more than 50 stories in your backlog per quarter, do some investigating.

METHOD 2: TIME NEEDED TO REVIEW

Another way to measure is to look at your backlog as an investment of time. The formula below takes the measurement one step above the product owner role and looks at scaling teams. The result will help you understand whether or not to be more proactive about backlog refinement.

1. Write down the number of product owners in the entire team of teams.
2. Write down the size of your backlog.
3. Divide the number of items in the backlog by the number of product owners.
4. Multiply the resulting number by 5. (Each item in the backlog needs about 5 minutes of discussion for everyone to grasp it.)
5. The resulting figure is the approximate time (in minutes) it will take for the teams to go through their backlogs.

Look at that number.

Given the availability of your team members and the demands of production, is such a time commitment remotely feasible? If the answer is "no", then refining the backlog, with a focus on managing its size, should become a priority.

4 Culprits of Backlog Overload (And How to Neutralize Them)

Why do backlogs balloon into serious problems? Rarely is there a single cause. It is often a mix of reasons that, at their core, are driven by human nature. Here are four common culprits and how to overcome them.

1. HOARDING

Humans are, by nature, hoarders. Many of us have difficulty discarding anything, especially what we think may be good ideas. We believe that the more ideas we have in our backlog, the more insulated we are from future failure, and will bear less risk as a result. The opposite is true. The trick to successful backlog management is not that we know what should go in, but that we decide what should come out. Overcome your nature. Has an item been in a backlog for two years? Time to retire it. Have faith that a truly great idea will find its way back in.

Why do backlogs balloon into serious problems? Rarely is there a single cause. It is often a mix of reasons that, at their core, are driven by human nature.

2. POOR TIMING OF GRANULARITY

Many product owners or project managers, particularly more conservative ones or those at the onset of Agile adoption, believe that tracking everything at a granular level lets them predict and control long-term scope issues. This rigid thinking runs against Agile development. It assumes we know exactly what needs to be done far into the future, when in truth we know the future is uncertain. Granularity should reflect that.

There is no hard-set rule, but the acronym DEEP, coined by Agile coach Roman Pichler, provides a helpful, memorable way to consider the essential characteristics of a good backlog. The D stands for "detailed appropriately," which perfectly describes the solution to this predicament. (The rest: E — estimated, E — emergent, P — prioritized.) Meaning: leave distant backlog items as coarse as possible. Forego granularity until it is absolutely necessary.

3. UNNECESSARY DEPENDENCY RESOLUTION

In some cases, often when our work is not completely Agile, we may strive toward resolving dependency chains in the backlog far into the distant future. To identify these dependency chains, we break down larger items into their components and reduce goals into tasks. However, in doing this — by unmasking dependencies by turning as much of our backlog into tasks as we can — we create a double-negative effect on backlog quality. First, it makes the backlog balloon in size. Second, it refocuses our backlog from goals and values to tasks that need to be completed. Overcome this urge by resisting your inclination to manage the distant future. Trust the Agile process.

4. OVERCOMMITTED PRODUCT OWNERS

Today's product owner is pulled in many directions. Foremost, she decides what features the product will have by transforming raw customer ideas into polished product deliverables. She also must achieve consensus among many stakeholders — each with different ideas of success. She also must understand the market, the customer, and the business to make informed decisions about those features. Trouble is, most organizations have far too few product owners in comparison to team sizes. The backlog suffers. The team suffers. The product suffers. All of it prevented by taking the product owner role seriously within an organization. The better the ratio of product owners to their respective team size, the more manageable the backlog, the more productive the team, the more successful the product.

6 Tips for a Leaner, More Valuable Product Backlog

How do we keep the backlog lean or even right-sized? Simply: have a strategy, be selective, and don't cheat.

1. INVEST IN THE PRODUCT OWNER ROLE

There should be one person — no more and certainly no less — responsible for the backlog of each scrum team, the product owner. Ideally, that person is responsible for only one team backlog as well. Reason being, that person needs plenty of time to maintain the backlog in collaboration with the team and external stakeholders. She needs to be extremely knowledgeable about the product and possess authority to make decisions within the backlog without intervention from other parties. When the product owner is invested in the backlog and has time to curate it, everyone benefits.

2. LIMIT YOUR DIP INVENTORY

Begin by investigating the sum total of partially finished work, or Design in Process (DIP) inventory, a term popularized by Donald Reinertsen who is an expert in lean product development workflows. Consider how much inventory can stockpile in the backlog at a given time, and also how much inventory you want to plan for in the future.

How many items should you have in the backlog at a given moment? There is no single number of truth, but a starting point is to measure it as a size per product owner; since a product owner is principally responsible for a backlog, her capacity to administer the information is the true limitation. Thus, a backlog for a large product with ten POs, each responsible for a different area, may be twice the size of a backlog for a product with five POs.

General advice? Use Dunbar's number. This number, arrived upon by anthropologist Richard Dunbar in the 1990s and based on the brain's finite ability to maintain relationships, suggests that, because of the size of their neocortex, a human being can effectively maintain only 150 relationships at a time. Given the contextual similarities (a product owner must manage relationships with interrelated backlog items), Dunbar's number provides helpful guidance in this case as well. Therefore, estimate that a product owner should comfortably be able to maintain approximately 150 different items at any given time.

Finally, consider the DIP inventory for the amount of future work to ensure the pipeline isn't planned so far in advance — for years and years — that it stifles innovation. The scope and detail of a winning product roadmap lands outside the topic of this paper but depends on many factors, such as product type and market maturity.

3. DEVELOP AN INCLUSIVE BACKLOG STRATEGY

Create a simple, clear strategy that prescribes how you will manage your backlog and how you will continue to include the team in that process. The second component, including the team, is often forgotten but arguably the most vital. While the product owner is generally responsible for maintaining the backlog, she is not the only one contributing to its vision. Everyone on the team should continuously participate in the process of keeping the backlog fresh. For this to work, everyone must understand the backlog, its overall vision, and its alignment with the roadmap — at least roughly. A regular, collaborative backlog refinement session can set you in the right direction.

4. BE SELECTIVE

Yes, it can be hard to discard things — "What if there is a great idea hiding in this mediocre one? Shouldn't we keep it?" — but making the difficult decision to either keep or remove an item is a critical trait of a product owner. Restrain yourself. Do not enter every idea that occurs to you. Keep it in your head. If it still is in your mind after a week, it might be a worthy addition to your backlog. Similarly, you should have a soft "one in, one out" policy as you bump against the ceiling of your self-imposed DIP limit. Add nothing without removing something of similar size and scope.

5. RETIRE OLD IDEAS

Both the product backlog and team backlog can reach a point where the DIP limit can become nearly impossible to enforce. The simplest solution is to quickly restructure your backlog into two sections. Allocate one section of the backlog to new ideas only (while continuing to thoroughly groom and restrict the size of the rest of the backlog), while maintaining the other portion as your active backlog. Give ideas in this newly partitioned section an age limit, so that older and continuously overlooked ideas disappear over time. If or when the idea is promoted into the active backlog, it indicates the product owner has committed to implementing it.

6. FOLLOW YOUR OWN RULES

At some point, in a moment of desperation, you may be tempted to bypass your own rules by saddling each backlog item with an abundance of information. Please do not do that. It keeps the backlog lean at the cost of fattening its individual items. Both are inefficient, but the second one is perhaps more dangerous in that it conceals the problem. Instead, be honest with yourself and your team members. Follow your own rules.

LEANER BACKLOGS BUILD BETTER PRODUCTS

Backlog management is too often and easily overlooked for the important work of dreaming, designing, and developing your product. But, with an understanding of causes of core backlog problems and methods for addressing them, even the most challenged manager can step out from under the shadow of a massive backlog. From there, managing your backlog becomes a matter of maintenance and housekeeping, so your team can continue to innovate, unhindered.

Discover how simple backlog management can be. **Learn about Hansoft**, the Agile project management tool with a single scalable backlog.

About Perforce

Enterprises across the globe rely on Perforce to build and deliver digital products faster and with higher quality. Perforce offers complete developer collaboration and agile project management tools to accelerate delivery cycles – from agile planning tools to requirements, issues and test management, which then link to all source code, binary assets and artifacts for full build and release tracking and visibility. The company's version control solutions are well known for securely managing change across all digital content – source code, art files, video files, images, libraries - while supporting the developer and build tools your teams need to be productive, such as Git, Visual Studio, Jenkins, Adobe, Maya and many others. Perforce is trusted by the world's most innovative brands, including NVIDIA, Pixar, Scania, Ubisoft, and VMware. The company has offices in Minneapolis, MN, Alameda, CA, Mason, OH, the United Kingdom, Finland, Sweden, Germany, and Australia, and sales partners around the globe. For more information, please visit www.perforce.com