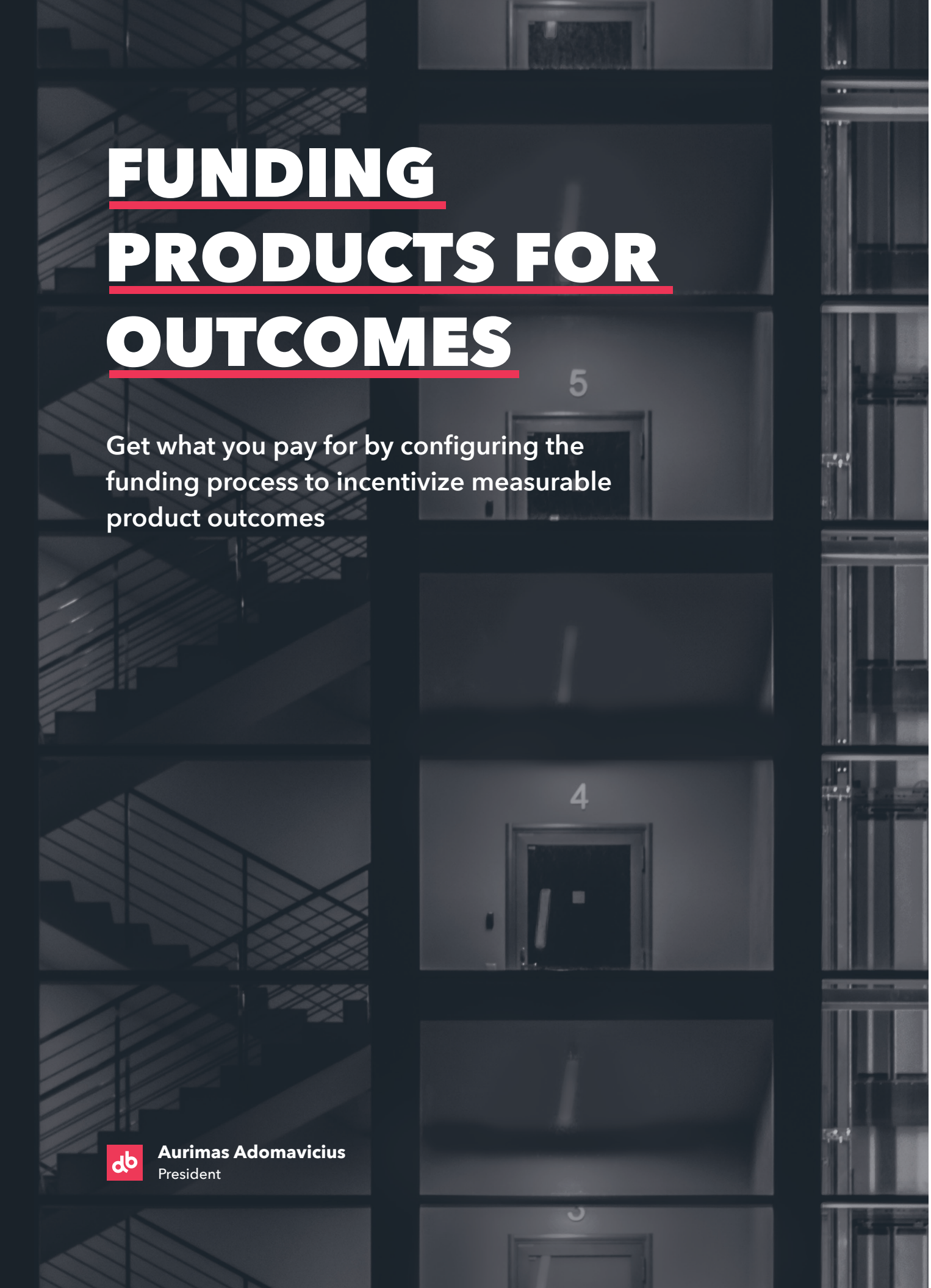


FUNDING PRODUCTS FOR OUTCOMES

Get what you pay for by configuring the funding process to incentivize measurable product outcomes

Aurimas Adomavicius
President


DEVBRIDGE®

01

INHERENT RISKS OF CLASSIC FUNDING TECHNIQUES

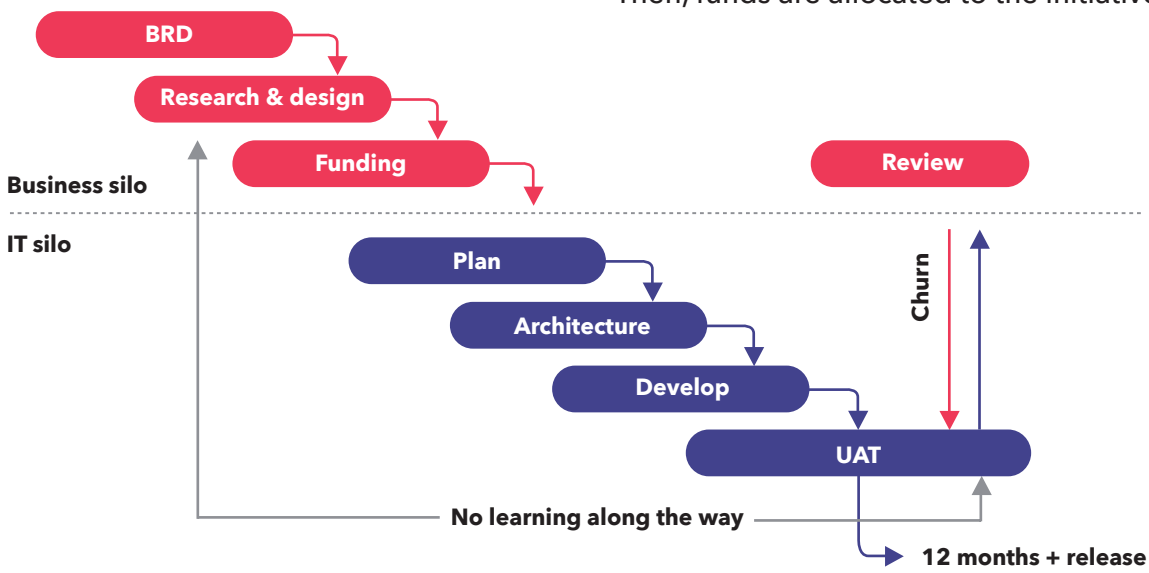
The software industry doesn't have the best reputation when it comes to delivering on value promised. It's a familiar tale. A business sets out to build a product. They turn to a supplier for support. Promises are made between the two organizations, but somehow the outcome doesn't line up with the original intent and things cost twice as much as expected.

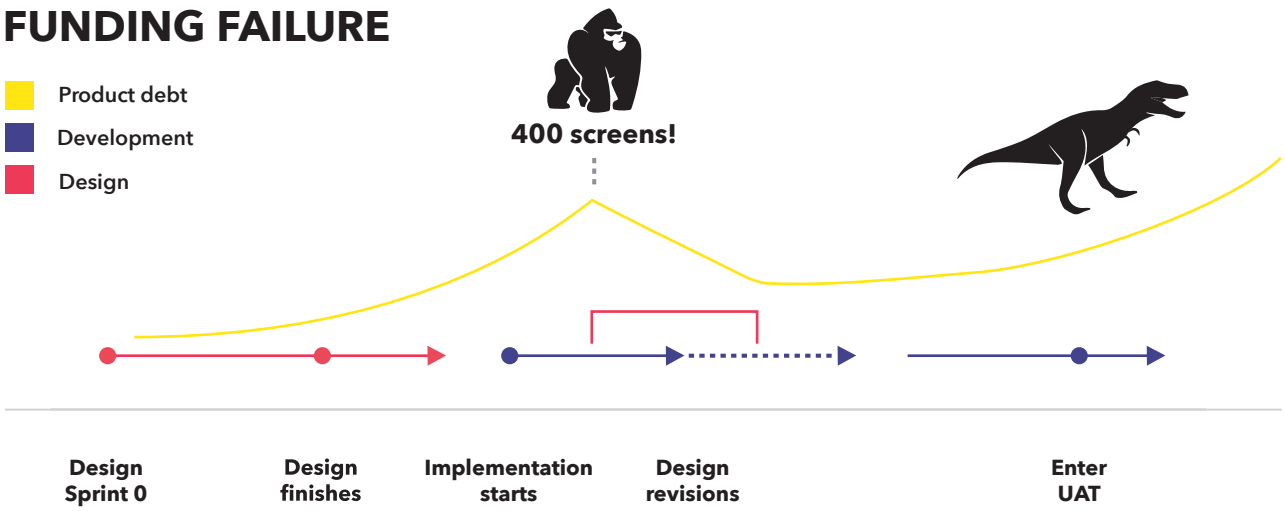
According to McKinsey, "On average, large projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted." Organizational culture, engineering, maturity, and outdated processes all have ample research pertaining to delivery failure. Taking this a level deeper, how does enterprise funding predestine a software project to fail?

Identifying typical triggers that lower value allows us to establish best practices that help the enterprise CIO, CPO, and CTO deliver more ROI per dollar spent. As it turns out, the required shifts for funding are mostly behavioral, retaining the expected level of accountability and risk management.

THE BRD FAILS

A large international bank sets off to build a custom web banking experience. A group of analysts gathers requirements for six months to identify the features necessary to achieve feature parity (compared to current, rigid, outdated off-the-shelf product), and new features that the business perceives as necessary. The requirements get packaged into a business requirements document (BRD), which is reviewed and approved by the organization's senior leadership. Then, funds are allocated to the initiative.





Everyone is slightly anxious, as this is a significant dollar amount and a single opportunity to get the project completed correctly. While trying to try to manage risk and avoid adverse outcomes, the business works to illustrate all identified features and workflows with wireframes and design. These steps help the business uncover features missed by the analysts. The design project takes six months and results in a library of four hundred screens.

Due to the complexity introduced in the design phase, the engineers—who were not initially consulted—build a large, padded estimate. No one wants to catch the bullet should this build go sideways. After reaching an agreement, the engineering team starts building the project three months later. Halfway through development, the engineering team identifies a few requirements from the design wireframes. To get clarity, they ask the business to review the work completed to date. When reviewing the new workflows, the business team quickly realizes the product is not as intuitive as originally proposed by the design agency. Development is put on hold while designers take another couple of months to revise the four hundred screens based on the new direction. In the meantime, some front-end refactoring takes place. Finally, the development team resumes their work, and after another few months, the project is finally ready to go into User Acceptance Testing (UAT).

By now, the budget is depleted from unplanned revisions and necessary refactoring. UAT surfaces additional challenges with several of the services being held during development, further delaying the launch by another few months. QA costs continue to rise. Timelines continue to slip. Once the product finally launches, the analytics indicate that customers seldom use the expensive new features.

This is a true story, a real-world example of where the Devbridge process methodology was intentionally overthrown to comply with a risk-averse culture and lack of trust. A retrospective revealed that the 2.5-year project could have been completed in under twelve months using modern, iterative delivery methods.

02

ENTERPRISE ORGANIZATIONAL STRUCTURES IMPEDE PRODUCT DESIGN

The traditional enterprise operates using a modular architecture with separate disciplines and segregated responsibilities. For example, marketing, sales, customer success, IT, support, and many other groups co-depend on each other for successful business outcomes—yet exhibit siloed behavior.

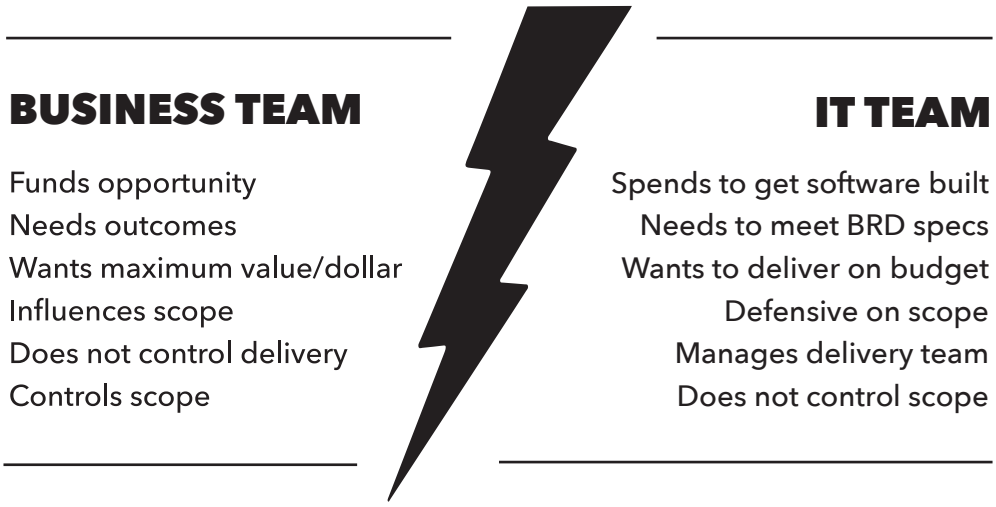
Software projects often originate inside the customer success teams (e.g., an SVP of lending identifies an opportunity). However, the projects get delegated to either internal IT or partner vendors for implementation. In other words, customer success is not equipped to build software. Moreover, while the budget originates from the customer success team, the IT team consumes the budget to produce results and ultimately delivers on the ROI presented in the business case.

This misalignment presents a conflict of interest. The customer success team wants the maximum number of features to guarantee success with customers, while the IT team wants the minimum amount of risk (and shortest path to completion) to make sure the project stays within schedule and budget.

The IT group is not necessarily interested in the customer as their ultimate responsibility is to build the product according to the specifications. Now, if the business does not adhere to the specification, IT is free to request additional funding, shift resources, or extend timelines.

Over time, the business stakeholders become jaded and lose all trust in the IT department's ability to deliver. IT continues to play defensively, shifting blame to the requirements, inadequate specifications, rigidity of underlying legacy architecture, and other limitations or demands from the business.

The issue with this traditional operating model is the scope and success of the product fall outside the influence of the team building the product. True ownership is not possible. Luckily, we can leverage learnings from the broader software industry and our tenured experience on how to best address these issues by treating software as a product.



03

TREAT SOFTWARE AS A PRODUCT

Luminaries like Netflix, Amazon, and other tech giants paved the way for treating software as a product—instead of as a project. As a result, the concept has taken center stage in the industry. The notion behind software as a product is that funding, planning, execution, and support for an application should be thought of as a perpetual cycle, instead of a one-time project. Furthermore, even aspects of software that are not necessarily customer-facing should be treated as products. In reality, a web service's lifecycle is continuous and requires maintenance, improvements, and refactoring as the service continues to support the shifting needs of applications.

RALLY THE TEAM AROUND PRODUCT SUCCESS

The challenges and lack of alignment dissipate when the whole team organizes around a single objective: product success. In response, the industry established the role of a Chief Product Officer (CPO), which encapsulates both the projected business outcome and the delivery capability aligned to these objectives.

ADOPTING A PRODUCT ORGANIZATIONAL STRUCTURE IS ADVANTAGEOUS

- The product team owns requirements and self-informs through user research, user testing, and analytics.
- The success of the customer and the success of the business are evaluated by a single team sharing context.
- The scope and gold-plating are evaluated in context with the goals of the product.
- The team composition is fixed to retain domain expertise and improve long-term velocity and throughput.
- Building transparency and trust within the team results in better estimates, more predictable output, and higher retention.

DON'T

- × Rely solely on business subject matter experts.
- × Fund a one-time project.
- × Design a product without considering ROI or clear metrics of success.
- × Let teams develop each aspect independently.

FOR EFFECTIVE OUTCOMES, SUCCESSFUL PRODUCT TEAMS:

Integrate with customer through research.

Own design and metrics.

Own product roadmap.

Own delivery.

Own ROI.

DO

- ✓ Gather customer input (primarily in the form of user research, user testing, and analytics).
- ✓ Approach each initiative as an evergreen application that requires ongoing investment to prevent it from succumbing to legacy rot.
- ✓ Make each application demonstrate the associated ROI so investment is fully justified.
- ✓ Require a dedicated product team with a more comprehensive skill set that includes engineering, product design, product management, engineering, and DevOps.

04

FLEX FUNDING MODELS BASED ON PRODUCT TYPES

"We have $10 million in funding, and we have one try to get it right. How can you guarantee we won't fail?"

During a recent innovation workshop with a Fortune 500 company, one of the executives in the room challenged the team with this question. The unease in the room was palpable. None of the company's prior "innovation initiatives" had ever gone as planned, so why would this situation be any different? You could see the proverbial cogs spinning in everyone's heads, calculating the amount of time remaining before the purging flames of failure consumed them.

Generally speaking, digital efforts fall into one of these four categories.

The category	The ask	The characteristics	The approach
Greenfield opportunities	New service offering or new product	Minimal dependencies on current technology, loosely coupled integrations, and undefined or open-ended roadmap	Minimal change management for current employees
Process transformation or digitization	Technology enabling efficiencies	High dependencies on current technologies with more well-defined requirements, often driven by internal customers to address an obvious need	Significant change management around rollout
Legacy debt resolution	Feature parity rebuilds or monolith sharding	Legacy applications with tightly coupled, monolithic workflows and high dependency on current technologies, challenging change management process, high technical debt, and embedded business logic	Multiple dedicated product teams for oversight, internal insight on an existing product during migration to a new system
Mature product	Scaled approach to building predictable output from multiple dedicated product teams	Establish product in need of rebuild and/or feature delivery at scale	Multiple dedicated product teams for oversight, theme-based funding, predictable velocity output

GREENFIELD FUNDING MODEL

Since greenfield products are entirely new, a great funding model to leverage is one used by venture capital for startups.

- Establish the hypothesis.
- Fund with small, incremental rounds.
- Validate using data on adoption or retention.
- Fail frequently, fail fast. Only when the hypothesis pans out, and the data backs it up should the organization double down on investment.

There are some prerequisites that the enterprise needs to adopt before implementing this funding model.

Establish a product-centric org structure. Enable cross-functional teams that represent stakeholders from both business and development groups.

Embed the product team within the group pursuing the opportunity. Don't isolate the product team in an innovation department.

Designate seed capital. Establish a pool of seed funding that can be distributed across several greenfield products each year, fostering a competitive environment where winners get rewarded.

Create a culture of failure tolerance. Celebrate the failures as much as the successes. The key to empowered teams is that they aren't afraid to test ideas and fail.

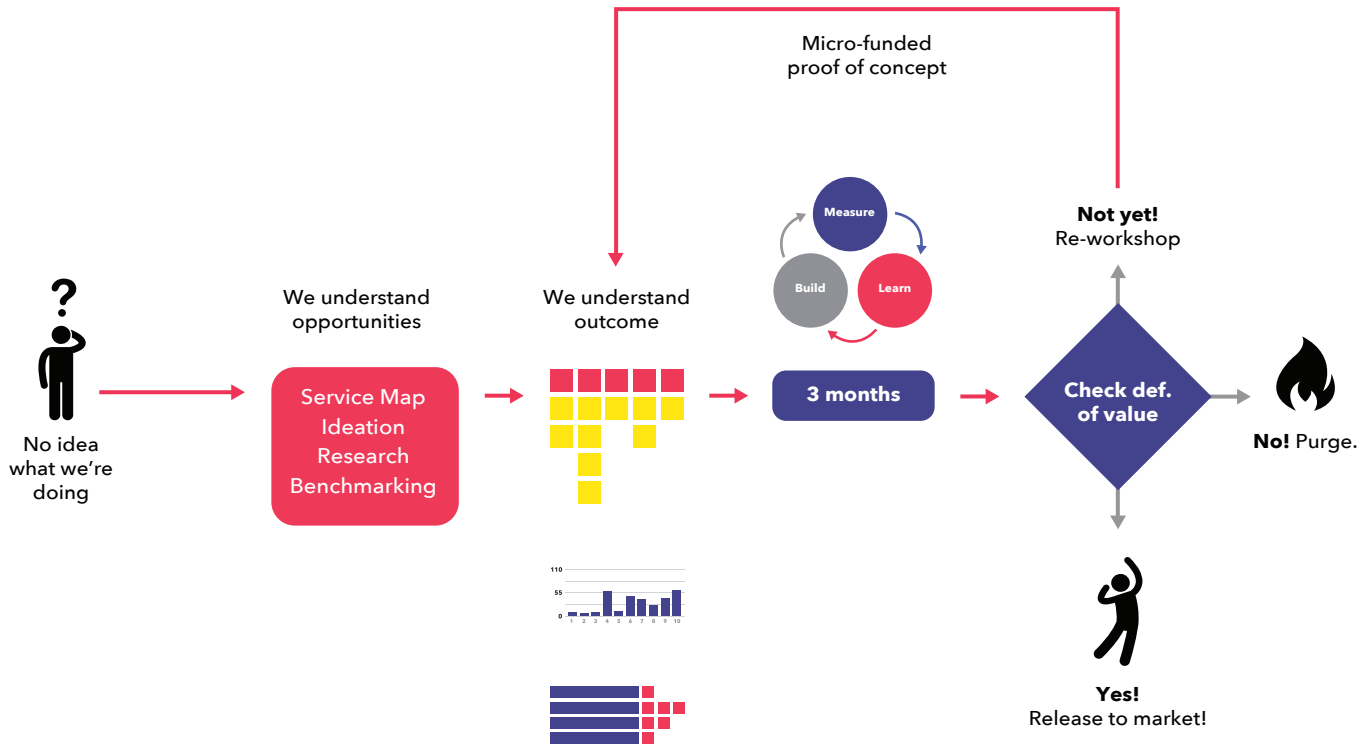
Set realistic expectations. Nine out of ten greenfield products experience failure in the early stages.

CORRELATE THE AMOUNT OF SEED FUNDING TO THE PRODUCT

Two constants apply to both internal and external product teams, as well as types of builds in terms of complexity.

1. The size of the cross-functional product team is fixed. An ideal team size is six to ten members. Less than six members implies the team is not covering all of the necessary skillsets (e.g., product design, product management, front-end, back-end, testing automation, DevOps). More than ten members reduces the team's ability to communicate and collaborate effectively.
2. The MVP should be feature-lean and viable to release in three to four months. Any longer and the seed investment grows to a point where failure becomes harder to tolerate. With known variables, calculate the monthly burn rate of the product team at roughly 100k. Set the first release and validation gate at four months, with anticipated seed funding for the product capped at $500K and four months of runway plus a minor margin.

FUNDING SUCCESSFUL OUTCOMES



CASE STUDY

A SECURE MESSAGING AND SCHEDULING APP FOR HEALTHCARE

The company
A privately held company helps hospitals and providers deliver high-quality patient care through a network of 7,200-plus partner providers and over 400 independently run facilities nationwide.

To provide emergency and hospital care, the company's schedulers book new and existing providers for shifts at the network of healthcare facilities. On an annual basis, the organization treats more than eight million patients. The clinicians need to communicate effectively to staff the shifts to provide high-quality care to patients.

The need
The organization identified an opportunity to collaborate with providers via a mobile application. They wanted shift scheduling, group chat, provider onboarding, and more to be facilitated in one HIPAA-compliant mobile app.

"We have been able to go to market and solve the business need in fewer than six weeks. With traditional waterfall development, this would have been impossible."

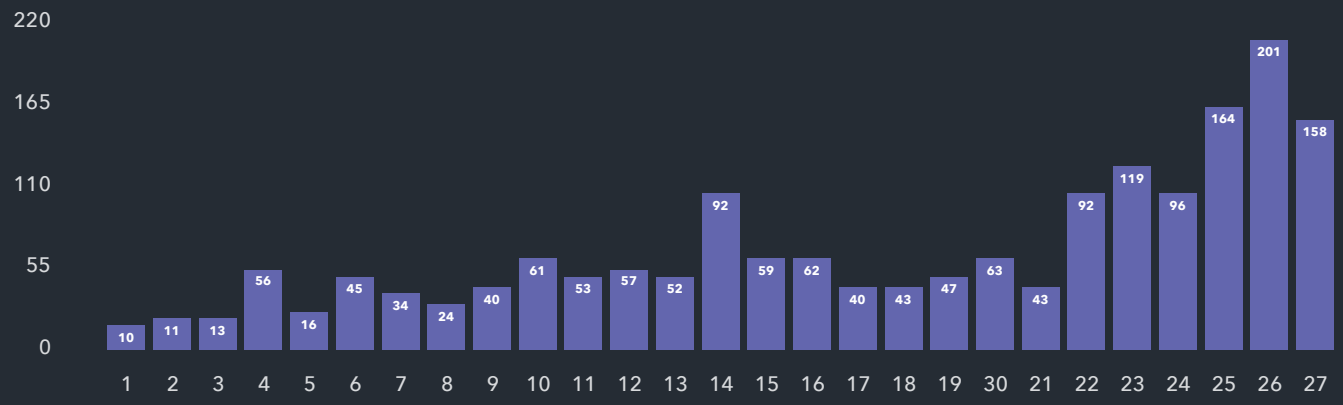
Senior VP of Information Technology

MICRO FUNDING AT WORK

ADOPTION

Weekly active users (WAU)

A user who has opened the app to send a t least 1 message in the last 7 days, on either iOS, Android or Web.



Positive trend of increasing WAU

THE APPROACH

The company enlisted Devbridge to de-risk and validate the new product before making a significant investment. After defining a full roadmap, MVP features were limited to those that would provide immediate value to the business.

In just three months, an MVP was delivered with encrypted communications for providers and schedulers to correspond and fill the open shifts faster. Focusing on a subset of features allowed the organization to get the MVP in the hands of users quickly. This approach allowed the organization to measure and evaluate user adoption before making additional investments. The initial adoption trends validated the product and helped secure funding for the extended roadmap.

THE ANALYTICS

MixPanel analytics provided the business insight into not only active sessions, daily active users, but also business KPIs such as shifts filled, the number of messages exchanged, and more. The company justified the additional investment to continue building the application feature set by using customer data. In seven months, Devbridge shipped over twelve releases to production, each funded incrementally. The strategy of micro-funding, testing, and validating was applied to several other greenfield products, paving the way to a digital transformation of the company.

PROCESS TRANSFORMATION FUNDING MODEL

Process transformation is less ambiguous than a greenfield build as these opportunities are tech-enabled. For example, the business notes an obvious upside based on the automation of a manual workflow or technicians for a generator manufacturer move to capturing payments directly in the field on an iPad app, drastically shortening the time between the origination of the work order and the collection of funds.

These types of opportunities don't need market validation, so the funding model should be slightly more risk-tolerant than a greenfield model. This tolerance allows for funding chunks to be larger, with fewer gates in the product roadmap.

For process transformation work, it is essential to validate assumptions with frequent market releases—as well as get feedback from internal customers using the application.

Identify necessary funding and structure delivery to maximize value for spend by using this workflow.

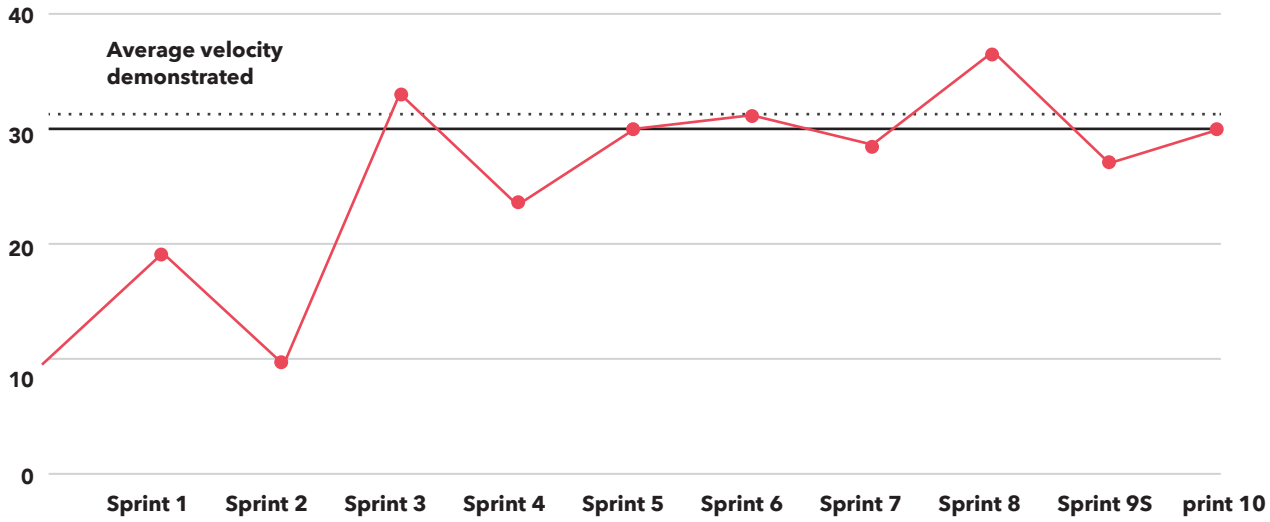
- **Execute a service blueprint to identify opportunities for automation.**
- **Create a workflow map for a particular touchpoint. Identify dependencies and the necessary change management to roll out a new product.**
- **Run a lean requirements workshop to build out the product roadmap.**
- **Estimate the roadmap and establish release gates (e.g., beta, friends-and-family, V1, V2, V3...)**
- **No phase should be longer than four to six months. Divide and conquer if product complexity is higher.**
- **Perform user testing after each release to refine the roadmap and re-estimate the backlog.**

LEGACY PRODUCT DEBT FUNDING MODEL

Legacy system replacement is expensive and extremely challenging to estimate accurately. When the software is monolithic, business logic gets embedded in the codebase, environments are hard to replicate, testing is often manual. However, there are still ways to manage funding and outcomes better than through lengthy waterfall requirements and delivery.

The most successful strategy currently available is a microservices-first approach that obfuscates the underlying legacy applications. Once the services are in place, a gradual rebuild takes place-slicing away verticals of functionality from the legacy application, building new functionality to parallelize processing across both legacy and new apps, to finally decommission the legacy workflows in favor of the new components. Microservices help hide a lot of the changes taking place behind the scenes so that customer-facing, front-end applications operate seamlessly. To effectively implement this approach, the executive team must accept that legacy re-platforming takes years to complete.

First, establish an overarching investment theme for the effort and execute a proof of concept technical spike to establish a baseline for velocity, complexity, and constraints. Once the proof of concept implementation is complete, the product management team can use the demonstrated team velocity to forecast the overall effort. Multiple workflows are parallelized across several product teams. The demonstrated burn rate and the velocity indicate the overall funding needs. Lastly, each product team should own a vertical (not lateral) cut of the functionality. In other words, a product team works across the full stack—from services to front-end of the new application. This vertical ownership reduces dependencies and helps teams stay self-sufficient.



05

TARGET STATE: PRODUCT FUNDING

Each greenfield opportunity, digitize, transform, or capture new market share, should either be killed off due to a lack of ROI or transformed into a mature product. A half-state may exist for middle-tier software (such as services), but even those need an active roadmap to maintain, support, and evolve as the business evolves.

When an opportunity demonstrates the desired outcomes (e.g., required adoption, revenue trends), then shift into a theme-based funding model. At this point, the business should be confident in the ROI of the product and evaluate how much throughput is necessary to meet customer demands, resolve growing technical debt, and stay ahead of the competitors.

These considerations dictate the number of teams to fund and which themes each team owns. Teams should have vertical ownership of components with oversight from product architecture.

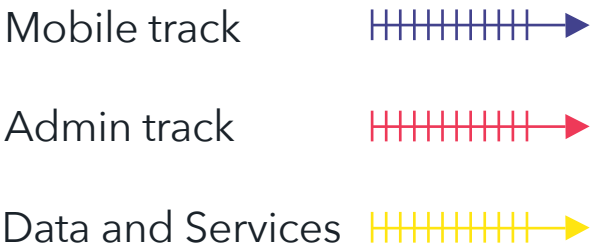
Once teams and associated themes are identified, the product managers should leverage demonstrated velocity and burn rate to predict both the annual cost of the team and the maximum amount of scope that a single team can deliver to market. These metrics then inform the sponsors, who decide if additional teams are necessary to reach the objectives.

NEW OPPORTUNITY

Micro-funding

MATURE PRODUCT

Investment themes



A FORTUNE 1000 INDUSTRIAL SUPPLY DISTRIBUTOR IDENTIFIES FIVE TEAMS WITH FIVE THEMES FOR INVESTING

TEAM	THEME
Customer workflow teams	Implementing a roadmap of features requested by customers as well as those necessary to stay competitive
eCommerce team	Implementing a roadmap for the eCommerce experience and checkout
Search team	Implementing a roadmap for product search, associated keywords, and bundle products
Merchandising team	Implementing a roadmap to enable merchandisers to add additional products to the catalog
Performance team	Resolving technical debt, continually optimizing the performance of all the components above

06

USE PRODUCT METRICS TO JUSTIFY INVESTMENT

The correct set of metrics, as well as aligning the product team on the right outcomes, is critical. It's best to use product metrics to justify spend, terminate projects, or accelerate delivery to meet market opportunities. Furthermore, outcomes are influenced by backlog priorities. Product metrics are even more crucial for the team to align with a clear understanding of what to build first.

Many customer-centric design philosophies often omit the needs of the business; after all, priorities need to be balanced for the customer to succeed and the business to make money. Aligning customer-centric and business priorities is an essential step in establishing product metrics. Failure to align on these priorities results in products that appear to perform well through metrics, but often suffer negative consequences long term, such as, customer attrition.

MISALIGNED OBJECTIVES

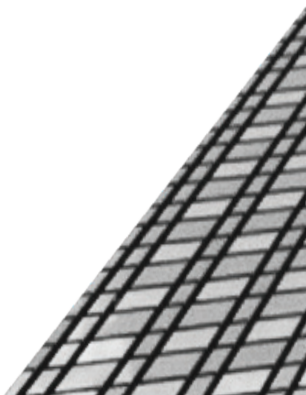
A great example is Groupon.com, a service that helps consumers save money through virtual coupons. Efficiencies are reached due to volume, and in theory, the platform allows the business to attract and retain a large number of new customers quickly. In theory, the influx of new customers makes up for the lost profits from offering products/services at a discounted rate.

Due to the popularity and viral nature of the platform, however, Groupon attracts (and partially created) a new type of bargain hunter. This new type of consumer does not demonstrate brand loyalty, and so participating businesses struggle to demonstrate ROI. While Groupon.com has evolved and adjusted its business practices, this is an excellent example of how keeping an eye on both customer success as well as the business objectives become critically important for long term growth.

TARGETED OUTCOMES AND METRICS

Product teams should establish outcomes that are specific, measurable, and achievable. Capturing 50% of total market share may be specific, but highly unlikely and thus not a great target outcome.

Targeting three to five outcomes is reasonable. Having a smaller set of outcomes allows the team to recall the target and recalibrate if necessary, quickly. Once the product goes through various releases, outcomes can adjust and evolve.



EXAMPLE OUTCOMES FOR TIER 1 BANK

A lending group at the bank identifies outcomes for the rebuild of their legacy loan processing application. They include:

1. **Speed.** The average time the customer application takes to cash disbursement reduces from three weeks to five business days. This could be achieved through automated decisioning, improved collaboration, and simplified workflows.
2. **Automation.** 90% of the applications are approved or rejected using an automated decisioning engine based on historical evidence and data provided by underwriters.
3. **Self-service.** The product allows actuaries to design, build, and test new risk models within the platform. Since the legacy platform handles risk models as code in the application, this is a simple one to measure and achieve.

THE POWER OF QUANTITATIVE AND QUALITATIVE DATA

Product metrics combine quantitative and qualitative data to inform the product team throughout delivery. Quantitative data typically originates from a product analytics tool such as MixPanel, Adobe Analytics, or Google Analytics. Having quantifiable data helps the product team establish a baseline, monitor for trends, and inform the future roadmap. Leveraging the earlier case study as a reference, the healthcare provider uses quantitative data on a per-sprint basis to evaluate the adoption of the product.

- **Activity:** percentage of operators that have logged into the platform
- **Activity:** percentage of total users logged in
- **Activity:** total downloads, downloads per week, delta
- **Adoption:** number of messages exchanged through the platform, delta per week
- **Adoption:** unique logins per week
- **ROI:** open shifts created in the platform per week
- **ROI:** shifts filled through the platform

These examples are just a subset of all variables monitored through time.

Qualitative metrics, on the other hand, provide the means for a dialogue with end-users and customers. Capture these metrics automatically through surveys, star ratings, reviews, feedback boxes, user interviews, or job shadowing. Acknowledging that all feedback is biased, reference multiple data sources, and critically evaluate the information to inform the product roadmap. Data gathered through more extensive usability studies trumps a vocal minority.

Bringing this all full circle, outcomes, and ROI from a product can only be recognized when the software is live and in front of users. This idea is the cornerstone of all agile methodologies, funding techniques, and organizational structures. The longer the product is stuck in development, the higher the risk, the less aligned it is to real market needs.

07

PINPOINT PRIORITIES FOR INVESTMENT

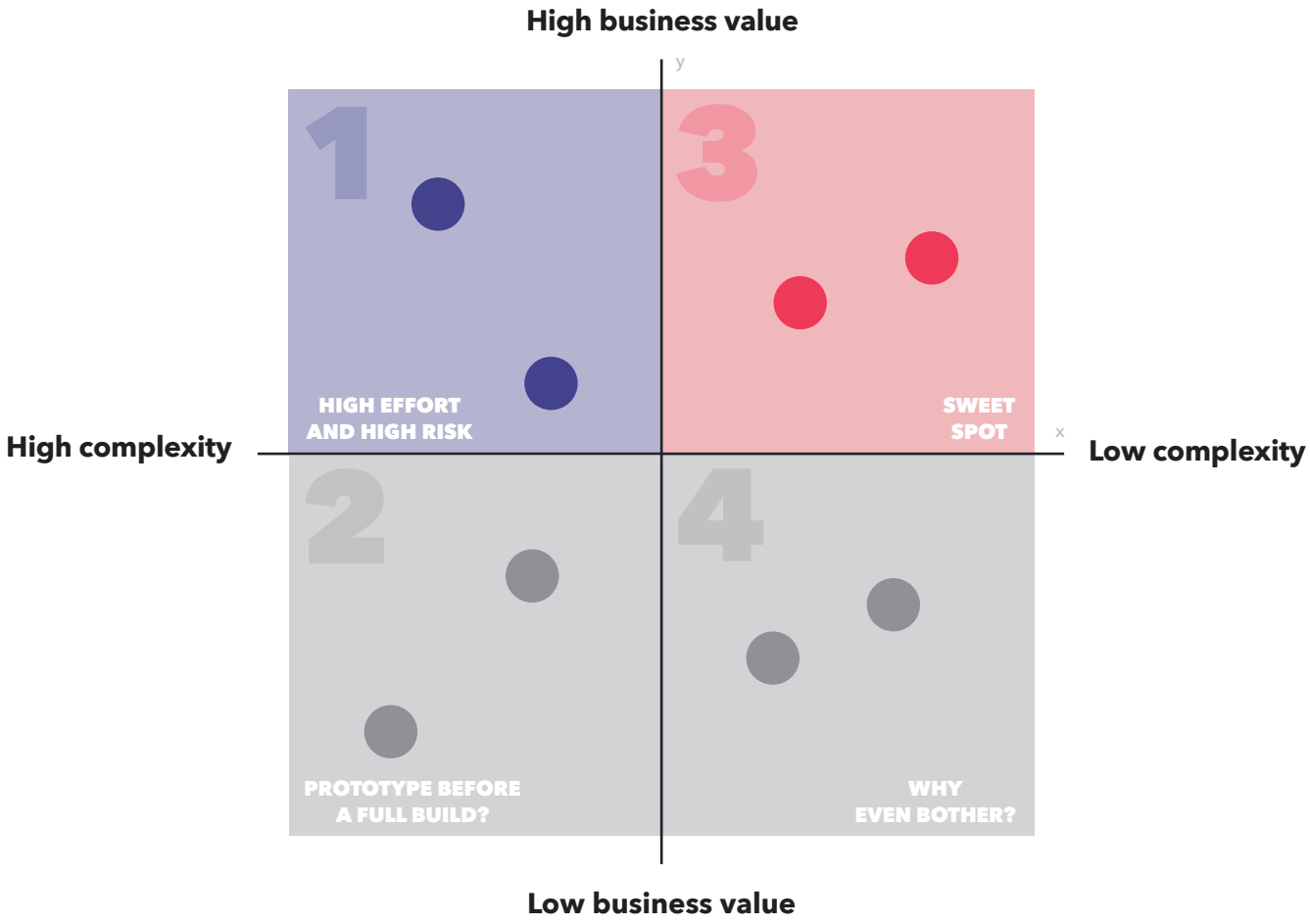
What any product organization quickly discovers is that scope creep is very real. Be it from the MVP, first versions, or version fifty…the backlog continues to expand, fueled by customers, internal users, product managers, competitors, and emerging technology capabilities. Unbridled product investment, on the other hand, can crush the usability and elegance of any product. The result often is a feature-saturated, expert-only product that requires months of training to use correctly and an exorbitant infrastructure to run. To avoid scope issues from escalating, start by establishing individual backlogs for each product investment themes around:

- **Feature roadmap** which can be further broken down by industries, workflows, and component.
- **Product debt** leveraging shortcuts taken in the past to release a feature.
- **Technical debt** pulling from shortcuts taken in the past with architecture and performance.
- **Administrative roadmap** for internal requirements to service the product.

VALIDATE AND PRIORITIZE

The initial step for the product team is to map all features on two axes: noting the business value on y, technical complexity on x. Next, take the prioritized stories and make sure they qualify to go into a specific release plan. In addition to the expected business value, a story should also be feasible and demonstrate technical readiness. Finally, use the data to prioritize product features.

User research, heuristics, usability tests, job shadowing, and user interviews should accompany qualitative and quantitative data to inform the product priorities as the team ships features on a sprint by sprint basis. A combination of these data points and techniques guarantee that maximum value gets generated from dollars spent.



1 HIGH BUSINESS VALUE, HIGH COMPLEXITY

While the risk is higher, the items in this quadrant are often competitive differentiators. Consider prioritizing these stories by the outcomes set for the product.

3 HIGH BUSINESS VALUE, LOW COMPLEXITY

This implementation effort has significant return, customer sentiment for a minimal investment, and low-level of risk.

2 LOW BUSINESS VALUE, HIGH COMPLEXITY

These stories carry a significant amount of risk, yet offer very little payback. Consider workarounds or alternatives to shift the story out of this quadrant.

4 LOW BUSINESS VALUE, LOW COMPLEXITY

These efforts may not even be worth pursuing. Discuss if there are good reasons for promoting or demoting these stories from the backlog. Use stories as training for team members being on-boarded.

08

CONCLUSION

Funding models play a vital role in a product's success. Two very similar teams can reach drastically different outcomes if funding is not conducive to customer-centric design methodologies, ongoing discovery, and dual-track scrum delivery. To summarize:

- The objective of product-centric funding is to maximize return for dollars invested.
- Waterfall requirements paired with a business case and locked scope does not mitigate risk. Agile development methodology is best.
- Siloed organizational structures don't work. Cross-functional teams do.
- Different products require different funding methodologies.
    - Shorter cycles for greenfield opportunities with validation gates
    - Longer cycles for digital transformation efforts
    - Ongoing discovery cycles for legacy re-platforming
- Key metrics informing risk management demonstrate the output from the team. Velocity helps estimate the cost needed to deliver.
- Outcomes are the North Star for product strategy with qualitative and quantitative metrics as navigational queues.
- Roadmap prioritization must be influenced by product metrics, research, release strategy, and technical feasibility.

Combining all of these strategies reduces risk, maximizes ROI, and establishes a healthy culture of innovation.

The result: Great product teams exceeding expectations and delivering more value than planned.

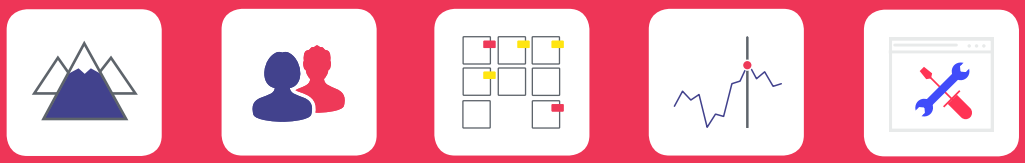
GET STARTED WITH A LEAN REQUIREMENTS WORKSHOP

Put custom software development on the fast track. First, get key stakeholders in one room. Then, let our product development team orchestrate the ultimate workshop. Without fail, we get the stakeholders aligned in a couple days—which is more than most companies can do in months.

One rule: Don't bring documentation. The interactive series of workshop activities produces the six key outcomes needed to jump-start product development, DevOps, or any digital transformation to net measurable results for your organization.

AUTHOR: Aurimas Adomavicius, President and co-founder of Devbridge. Founded in 2008, Devbridge revitalizes the largest of enterprises with custom software. When not in the trenches working with clients, Aurimas is an active speaker and writer on the best practices of product design and engineering.

LEAN REQUIREMENTS WORKSHOP ACTIVITIES



Problem & Vision | Users, Roles & Goals | Story Map | Risks and Impediments | Technical Feasibility

SIX KEY OUTCOMES

1. **Defined business goal**
   A clearly defined business goal with success metrics

2. **MVP requirements**
   The minimum amount of requirements necessary to kickoff the design process

3. **Hidden requirements**
   Bringing people from different functions together to uncover what impacts goals, scope, and priorities

4. **Shared understanding**
   A shared understanding of the business process, end users, and their pain points

5. **Scope & priorities**
   An agreement on scope and priorities to meet the business goals

6. **Product release strategy**
   A phased approach to releasing your product to market

FASTER THAN YOU'RE USED TO.

getstarted@devbridge.com
312.242.1642

DEVBRIDGE®

Our experienced teams deliver software **4x faster** than the industry average.

We are a full stack research, design, and software delivery company. We facilitate a creative, iterative process, powered by some of the most passionate individuals in the industry. We build extraordinary custom applications that solve complex problems and deliver measurable results.

Our scalable cross-functional product teams and tailored services transform businesses in agriculture, aviation, financial services, fraud & forensics, healthcare, hospitality, logistics, and manufacturing.

Mobile applications for B2B and B2C

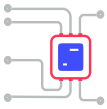
We create elegant, intuitive mobile applications to enhance the customer experience.

Product research and design

We build custom applications that solve real business problems to drive measurable value.

Scaling product delivery

We go beyond team aug and dedicate full cross-functional teams to scale feature delivery.



Data strategy and microservices

From dashboards to master data strategy, we can help you leverage your data to grow your business.

Legacy software modernization

Replatform aging systems and legacy architecture to meets the needs of your business now.

Workflow optimization

Streamline your business processes to reduce time, costs, and improve productivity.

450

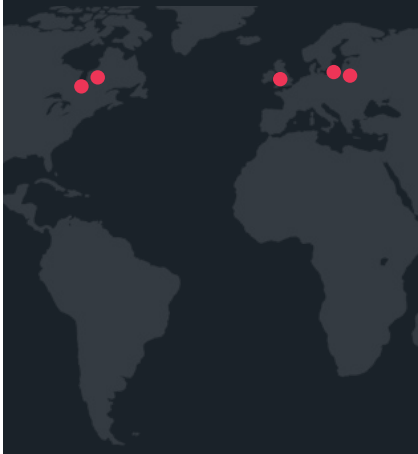
Full-time employees

5

Offices

11

Years in business



Chicago
London
Toronto
Kaunas
Vilnius