WHITE PAPER

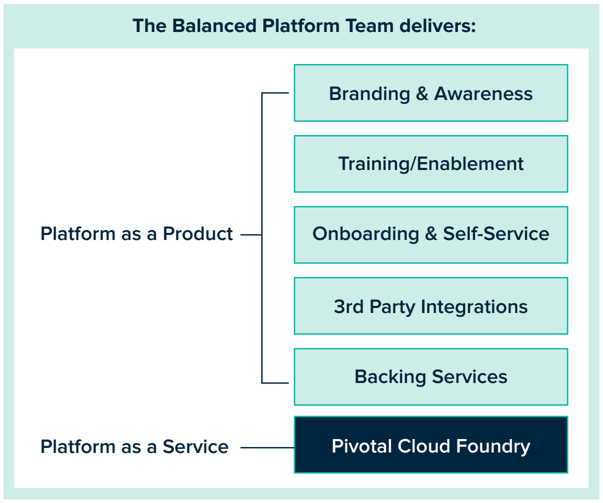
Pivotal Platform Solutions: Why You Should Treat Platform as a Product

By Joe Fitzgerald and Colin Humphreys

Table of Contents

Why You Should Treat Platform as a Product

Your platform is not an off-the-shelf piece of software[1]; it is an evolving set of reusable services, integrated with your existing systems, that creates valuable outcomes for your business. The platform's capabilities should change in response to the needs of its users — your app developers — among whom it is a recognizable internal brand. In other words, your platform should be treated as a product.



Building a Platform Product

Pivotal's platform-as-product approach combines Product Management (PM), User Centered Design (UCD), Agile, eXtreme Programming (XP) and Site Reliability Engineering (SRE) practices. A dedicated, balanced platform team uses these practices to both build and run the platform product.

All of this allows an app development team to have an idea in the morning which is running in production by the afternoon. Treating your platform as a product maximizes the value of the platform while minimizing delivery time, risk, and waste. Done well, it will change the way your organization builds and runs software.

[1] If you are considering a Managed Service from one of Pivotal's partners, you still need to treat Platform as a Product.

Pivotal’s Platform-as-Product Approach

Pivotal’s approach combines practices from three different areas: Product management, agile software development, and SRE. Product management practices determine what the platform team will build. Agile software development with XP provides a clear methodology for how they should build. SRE defines how the team runs the platform product.

Product Management

Your platform team must continuously respond to the changing needs of its customers, your app developers. Lean startup practices are used to maximize customer value while minimizing waste. UCD practices ensure that the team builds the platform capabilities that app developers actually want, by validating assumptions about their behavior in real-world tests.

Agile Development with XP

Your platform team focuses on the incremental, iterative, and flexible delivery of platform software. The team uses XP practices to build high quality, well-tested platform code that evolves in line with your app developers’ requirements. XP anticipates changes in customer requirements, avoids building features until they are needed, uses pair-programming and automated testing practices, and requires continuous communication between all stakeholders.

Site Reliability Engineering

SRE treats operations as an engineering problem by using software to manage and maintain systems. Agile software development practices can, therefore, be applied to add new features to your platform and automate away toil and waste. Toil in SRE is defined as manual, repetitive and automatable work tied to running a production service.

SRE attempts to balance the traditionally conflicting goals of velocity and reliability. Your platform team and your application teams will determine an appropriate level of reliability for the platform and each application while maximizing feature velocity. SRE regards one hundred percent reliability as the wrong target for the platform or for applications running on a platform. Instead, the platform team and app development team together define an error budget, which quantifies how reliable (as experienced by the end user) the product needs to be over a particular time period. The app development team is responsible for planning how much change they can safely introduce and on what schedule to stay within their error budget.

Prerequisites For Building a Platform Product

Pivotal's experience has shown that putting the following team and technology prerequisites in place greatly increases your chances of successfully launching a platform product.

The Push to Build a Platform

Your organization should have a strong motivation to build a platform product. In some cases, your IT organization knows that it wastes large amounts of effort building platform-like capabilities and wants to reduce costs. In other cases, a business unit in which multiple app development teams have already been forced to solve their own platform needs wants to increase velocity.

A Platform Champion

A platform champion has the motivation and the political capital to protect the platform team as they embark on the platform journey. The champion is often at, or close to, CxO level and has a track record of internal entrepreneurship or organizational transformation. This person understands and can articulate the value of a platform product and evangelizes its use and growth within your company.

A Dedicated Platform Team

Your platform team builds and runs your platform product. The team must have the authority to change the production process as well as the platform itself. A platform team consists of a minimum of three full-time employees in the following roles:

- The platform product manager builds the thesis for the platform product, tests that product thesis with the target customer — internal app developers — and iterates on the features provided to achieve the desired customer and business outcomes.
- At least two platform engineers work through a feature backlog to deliver new platform features on a weekly basis. You will add additional platform engineers as the platform scales.

Reliable Local Infrastructure

Don't introduce changes in your Infrastructure-as-a-Service (IaaS) layer at the same time as embarking on the platform journey. You need a stable IaaS capability during the platform creation period.

Establish a production platform capability which is located close, or ideally in the same physical location, as the existing applications and data the platform needs to function. Otherwise, latency issues may be blamed on the platform itself. If you must have distance between applications and their data, establish a baseline with monitoring that you use to disambiguate changes to the platform or application.

Shared Goals

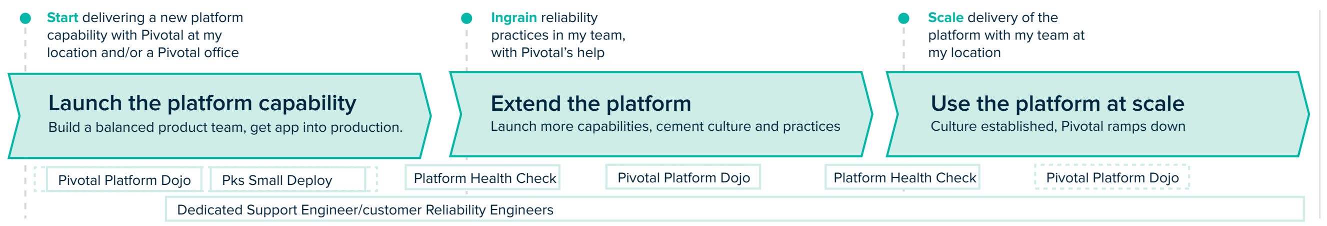
Many stakeholders are involved in building a platform product including security, change management, release management, virtualization, networking, production operations, business application owners, app development teams, and their end users. Defining shared goals ensure that stakeholders work together rather than against each other. Shared goals also change the conversation from "Here's why this won't work" to "How can we make this work?".

Pivotal uses an inception exercise to align teams on the importance of the platform and how it fits into the bigger picture for your organization. Define quarterly objectives and key results (OKRs) for the platform, e.g. deploying the first app to production during the launch phase. Variable compensation schemes, rewards, and recognition can also be linked to these OKRs.

How To Build Your Platform Product

Your platform team will start small and gradually extend the platform's capabilities. We recommend three successive phases during which your platform increases in maturity, taking you from creating a platform team to running thousands of apps in production on the new platform.

Platform Journey


Start delivering a new platform capability with Pivotal at my location and/or a Pivotal office
Ingrain reliability practices in my team, with Pivotal's help
Scale delivery of the platform with my team at my location
Launch the platform capability
Build a balanced product team, get app into production.
Extend the platform
Launch more capabilities, cement culture and practices
Use the platform at scale
Culture established, Pivotal ramps down
Pivotal Platform Dojo
Pks Small Deploy
Platform Health Check
Pivotal Platform Dojo
Platform Health Check
Pivotal Platform Dojo
Dedicated Support Engineer/customer Reliability Engineers

Launch The Platform Capability

The main objective in this phase is to launch a minimum viable product (MVP) version of the platform and deploy a single production app on that MVP.

Choose Your Platform Team

The platform team consists of a product manager and at least two platform engineers. These first members of the team will seed future teams so choose them wisely.

Product Manager

IT organizations often do not have agile product management experience so you may need to recruit an external-facing product manager from a business unit. We have found that successful product managers fulfill several of the following personas:

- The Alchemist: Takes disparate requirements and distills them down to a succinct product vision.
- The Dreamer: Doesn't allow themselves to be constrained by legacy thinking and processes.
- The Influencer: Has strong relationships with business partners, application teams, and IT.
- The Minimalist: Understands the value of shipping early and often.
- The Lean Champion: Relentlessly pursues the elimination of waste.

Platform Engineers

The platform team requires a combination of infrastructure and software engineering skills. Because SRE treats operations as a software problem, your platform engineers must code. In many cases, training will be needed to help platform engineers recruited from software teams to deepen their infrastructure knowledge and platform operators to build their software skills.

- The launch platform team should ideally include at least two of the following personas. As the team matures it should eventually contain all three personas.
- The *Infrastructure Architect* who codes: Very experienced in IaaS primitives (compute, storage, network) this person usually has experience in production operations and if asked to perform an activity multiple times will instinctively automate it.
- The *Natural Automator*: You'll often find this person doing continuous integration and deployment (CI/CD) work, automating your current release management processes or doing system automation using tools like Chef, Puppet, Salt, and Ansible.
- The *Curious Software Engineer*: You'll find this person in an application product team which has previously solved its own platform needs by automating infrastructure.

Pick a First Production App

Choose a single production application which all stakeholders agree will go into production on the new platform. The ideal launch app is not mission critical but has existing production users and is maintained by an app development team which can assist in moving the app to the platform product.

Do not fall into the trap of choosing a mission-critical app in order to prove to detractors that "important" apps can immediately go to production on the new platform. Mission critical apps can follow quickly once the platform team gains expertise by running the first real workloads in production.

Define a Platform MVP with a Brand

The platform product manager defines an MVP platform product with a self-service landing page and a product brand. An MVP is the simplest and most easily built version of the platform that can help validate or invalidate a product management hypothesis about user behavior or features.

Pivotal customers who have successfully delivered platform products in large organizations define a recognizable internal brand for the platform. A brand creates a sense of ownership and pride in the platform team and helps to build a customer base by raising awareness among app developers. "Pivotal Web Services" is an example of a branded platform.

Define a New Path To Production

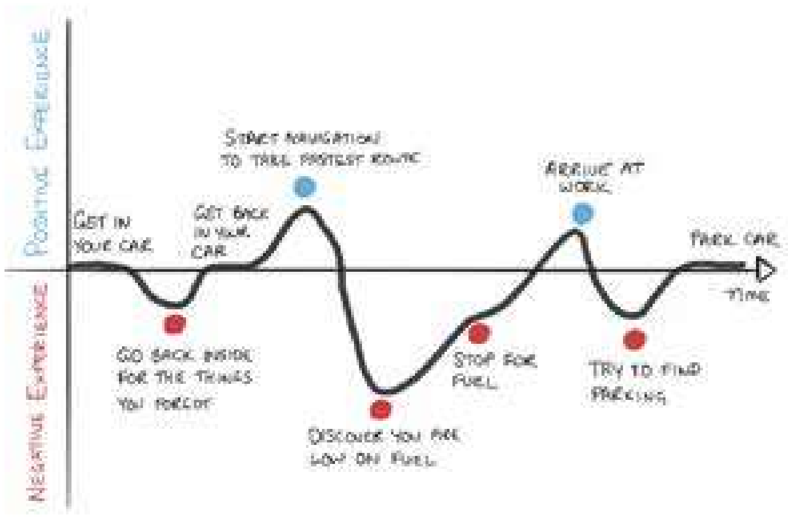
An effective platform allows application teams to release software faster, which stresses systems and processes that assume centralized control and throttle change in an attempt to achieve reliability. Do not shoehorn your platform into your existing path to production or you will simply make it easier for apps and features to pile up waiting to get to production on the platform.

Deploying the first production app on the platform will prompt important conversations among stakeholders about the processes in your current path to production and production operations, e.g. security, release management, testing, compliance, and change management. Customer journey maps and value stream maps will be created to guide these conversations.

The result should be a new, and deliberately naive, path to production that emphasizes velocity and is only suitable for a small proportion of your app portfolio. As you extend the platform, the path to production will be enhanced to meet the production needs of a larger proportion of your app portfolio.

Create Customer Journey Maps

A customer journey map is a compact visualization of an end-to-end customer experience. The platform product manager will generate journey maps of your app developers' current experience of going to production to understand where there are opportunities for improvement.



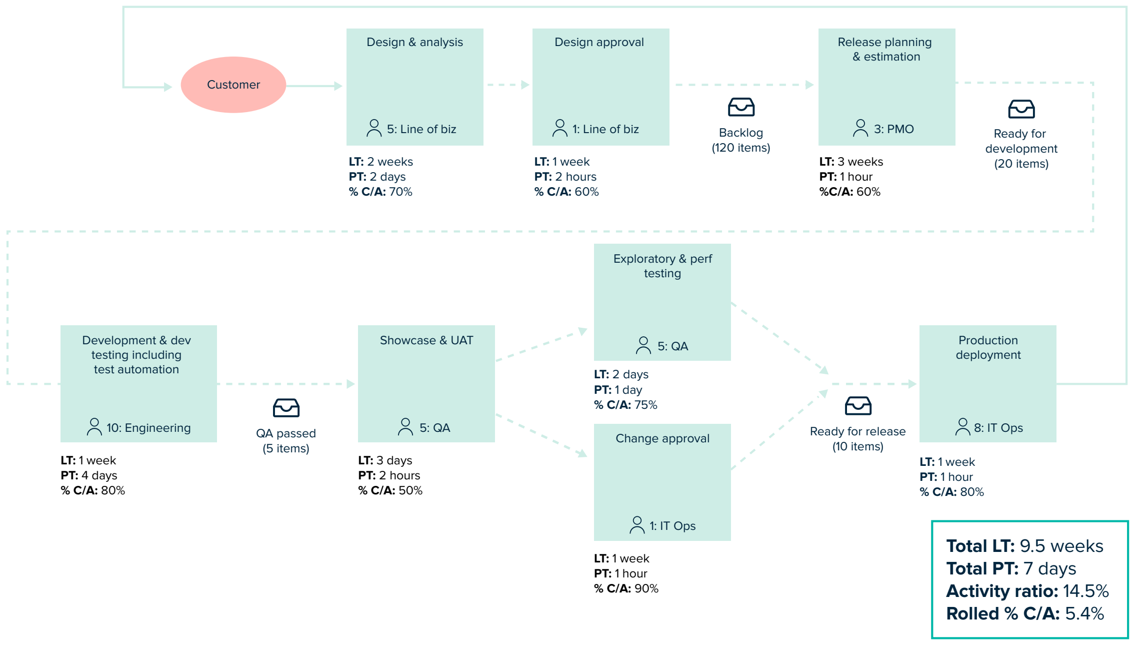
A "drive to work" journey map

The product manager will also talk to teams currently running apps in production and generate journey maps of their experience of responding to incidents and performing root cause analysis. These maps will help identify opportunities for improvement in the way teams run apps in production while still increasing the velocity of change.

Compare the path to production to the requirements of the first production app. Since the app is not mission critical, it may not need to adhere to requirements for compliance or security demanded by other apps. Does it require all steps in the process? What can you change or leave out? This exercise helps define the essential steps in the new, naive path to production.

Define Value Stream Maps

A value stream map is a visualization of the sequence of activities an organization undertakes to deliver on a customer request. In the case of the path to production, that request is getting a feature or app into production. The time between a particular trigger and the value delivery prompted by that trigger is the lead time. Shortening the lead time, by removing steps and reducing waste in the activities used to deliver value, is an objective of value stream mapping. While customer journey maps are qualitative, value stream maps are quantitative.



Software Design and Delivery Value Stream Example

The platform team will continuously evaluate the path to production and perform value stream mapping to find opportunities to remove waste and increase the rate of automation.

Deploy to Production

Launch the platform MVP and deploy the first app to production using the new, naive path to production.

Extend The Platform Product

In this phase, you will extend the features of the platform so that a larger proportion of your app portfolio can go to production while meeting security, compliance, auditing, financial reporting and other enterprise requirements.

Iterate on Features and Reliability

The platform product manager will gather information from application and IT teams to determine what platform features are required to run additional apps from your portfolio in production on the new platform. Since the platform should create value for your business, the exact features will vary but here are a few typical extensions:

- Additional service brokers to permit self-service access to application dependencies.
- Additional buildpacks to allow new applications to run on the platform or to reduce the time required to modify an app to run on the platform.
- Modifications to the onboarding portal to add more self-service features.
- Additional backing services such as MySQL, RabbitMQ or Spring Cloud Services.
- Solutions for cross-cutting concerns like security, logging, metrics, and load balancing.

- Integrations with other enterprise systems.
- Multiple variants of the platform that have different service level objectives and characteristics.
- Additional deployments of the platform in new regions and/or on new infrastructure.

Iterate on the Path to Production

The platform team will iterate on the path to production to meet non-functional and compliance objectives for security, auditing, financial reporting or other enterprise requirements and make it possible to deploy a larger proportion of your app portfolio on the platform.

The platform team will use the five whys or the infinite hows technique to determine the process objectives in your original path to production in order to achieve the same outcomes with less waste and more automation. The exercise should result in journey maps with improved sentiment (the emotional state of your app developers at each stage of the journey) and updated value stream maps with less waste.

Grow the Platform Team

Two platform engineers are only sufficient while you launch the platform and have a small number of platform tenants. If your platform engineers spend more than 50% of their time on toil or you want to provide enhanced support for the platform such as 24/7 on-call support, then it’s time to add more engineers.

Use The Platform At Scale

During this phase, you will scale the platform and the platform team to serve thousands of applications while meeting all mandatory security, compliance, auditing, financial reporting or other enterprise requirements.

Increase Self-Service

Self-service helps the platform team to scale sublinearly with the workloads running on the platform. At scale, app development teams should be able to self-service their way to production, while individual developers should be able to use the platform to learn and experiment pre-production.

Self-service changes the communication pattern between the platform team and the app developers they serve. Instead of waiting for developers to send a request to IT, the platform team will track self-service activities and proactively reach out to new and existing tenants of the platform when they see opportunities to educate, plan for the tenant’s needs or identify missing features in the platform’s capability.

Scale The Platform Team

The platform team will scale sublinearly with platform workloads and generally max out at nine or ten people. A team of this size is capable of building and running a platform that can service 3000+ applications.

Pivotal always recommends creating a sustainable work environment where each engineer works a maximum of 40 hours a week and spends most of their day pairing. Each engineer should only be on call for a maximum of one week per month, ideally with a primary and secondary engineer on call during each rotation. Providing 24/7 support coverage in that environment requires 6 - 8 engineers per site.

Pivotal has found that spreading those 6 - 8 platform engineers across multiple time zones in an effort to reduce “out of hours” work is the wrong approach. Maximizing time zone overlap between engineers is more important to ensure that context about platform changes is shared. Since platform incidents are almost always the result of planned changes, you can effectively schedule when incidents will occur.

Add a Developer Enablement Team

Adopting a new set of platform capabilities and transforming applications to run on the platform can be challenging for app development teams. A dedicated developer enablement team helps app developers improve their craft and establishes bi-directional communication between the platform team and lines of business. A developer enablement team can help transform your existing app portfolio and launch new products faster.

How to Know if You are Succeeding

Pivotal customers often ask how they can tell if they are building an effective platform product. The following signals show that you are on the right path:

- Features are delivered to end users faster.
- Developers ask for more platform capabilities.
- Deploying to production requires no special reviews, checks, balances or ceremony. Simply clicking “accept” triggers automated deployment, audit trails, and approvals.
- Application outages are quickly traced to a root cause without a fire drill.
- Events which previously generated downtime are auto-healed without paging anybody.
- The platform team reaches a level of platform mastery where they provide new insights to their peers or to Pivotal.

Pivotal does not have all the answers. We learn as much from our customers as we teach them. The successful platform journeys of Pivotal customers such as Comcast and Home Depot have informed much of the above advice. If you have questions or insights about building a platform product you would like to share please contact us at platform-practices@pivotal.io.