White Paper
August 2025

AI Platform Marketing Playbook

Increasing AI Platform Consumption for AI Native Transformation

Table of contents

Introduction

Based on recent independent research commissioned by the Tanzu Division of Broadcom, 98% of enterprise organizations have decided to make an AI-ready platform a major part of their AI strategic plan.[1] With AI being such a hot topic, you might think that your organization's AI platform will "sell" itself. But nothing could be further from the truth. To achieve your desired outcomes, you need the buy-in of your development and data teams.

No doubt AI is providing value in the enterprise. In fact, according to our recently published research,56% of enterprise organizations that are deploying AI models in applications are achieving revenue gains, and 54% are seeing increased staff productivity.[2] We have definitely leaped over the trough of disillusionment for AI more quickly than any technology trend before.But organizations that are still trying to get a handle on shadow AI need to develop an AI-friendly positioning and marketing strategy. The perception that teams will be cut off from the next big technology trend is a recipe for mutiny. It's best to think through how to artfully navigate this technology inflection point while protecting the business. Your goal should be to convince application developers and data scientists that you are not intending to take away their favorite AI tools, but rather speed up AI adoption with scalable patterns.

As a rule of thumb, we recommend adopting the following strategies before rolling out your internal AI education and evangelism strategy:

- **Drink the AI Kool-Aid –** Whoever is setting and signing off on the policy should use AI for two to four weeks before crafting AI policies. Understanding what AI can and can't do firsthand should inform the policies you adopt. There is a lot of fear, uncertainty, and doubt as well as a miscalculation of benefits that should be vetted before assigning corporate policies.
- **Adopt a friendlier approach to shadow AI –** Your employees will use AI no matter what you try. Think about the history of trying to police instant messaging ("chat"), SaaS, mobile, etc. Would you rather understand AI, and smartly manage it, so that your organization sees you as a trusted partner?
- **Make the corporate AI policy easy to follow –** Have a clear policy that is frictionless so that it becomes easy to follow. Provide easy access to AI, make the policy logical, and invite questions about it. If you keep an open-door policy, you are more likely to be seen as a trusted partner and less likely to have to wrangle shadow AI. Adopting a platform where all your AI is transparently available can really help in this regard.

[1] From Cloud Native to AI Native: The CIO's Journey to the New Digital Transformation, July 2025
[2] From Cloud Native to AI Native: The CIO's Journey to the New Digital Transformation, July 2025

The Tanzu team's long history of observing successful technology adoption trends reveals a consistent issue before even arriving at AI readiness: Technically-sound platforms suffer from low developer adoption. Despite meeting stated developer needs, internal developer platforms (IDPs) often go unused, a problem recurring with Kubernetes-based platforms. This raises a crucial question: How do we drive adoption and achieve ROI on our platform investments?

This pattern highlights three overlooked areas: **product management, community building, and platform marketing.** While product management and community building are gaining acceptance, marketing is often met with skepticism. This is a mistake, as marketing can be approached systematically like any engineering discipline. Let's explore how to systematically drive awareness, understanding, and adoption of IDPs.

Defining the customer

You can't have marketing without a customer. Good marketing strategies spend a lot of time defining the customer, or audience, that marketing activities are trying to reach. When it comes to internal developer platforms, your primary customer is right there in the title: developers.

However, instead of just settling on "developers," it's good to narrow your audience down even more. First, these are likely application developers. They're not developers writing embedded systems, for example. They're probably not the developers creating the platforms in question, nor are they programming the custom development tools your application developers use. And they are also not data experts who will be evaluating AI models for inclusion in those applications. That is for the data team to assess and address.

Another area to home in on is which types of applications and which parts of the business your developers work on. Are we talking about developers in the trading desk division of a financial institution? Or developers who work on the ecommerce front end of a car manufacturer? Are these Java developers who are adding AI functionality to existing applications? According to our research, 65% of enterprise organizations will be adding AI to existing apps,[3] which means the platform will need to support these same developers that you have been working with to date, just with a new supporting service included: AI models.

Many platform teams can cater to a wide range of developers, such as all of the ones used as examples above. Early on when developing your platform strategy, picking one or two types of developers can be handy to force you to focus on those developers and their needs. You learn what this platform stuff is all about by working with a handful of initial teams. Then you take those skills and expand to other teams.

---

[3] From Cloud Native to AI Native: The CIO's Journey to the New Digital Transformation, July 2025

We're focusing on platform *marketing* here, but a lot of this "who is the customer?" thinking is also done by platform *product management*. That function and role is incredibly valuable and, we think, what makes platform engineering different from more traditional IT service management and delivery. As you do more and more platform marketing, you'll find that it overlaps with product management a lot, and this is good!

Once you have your customer identified, you can then move on to the core parts of platform marketing.

Core marketing: Messaging, positioning, and value props

Three core elements are key to any marketing strategy: messaging, positioning, and value propositions. This is a good place to start for IDP marketing.

Platform messaging – What is it?

Messaging is how you communicate the value of your platform, i.e., the key points you want developers to understand and remember. Think of it as your elevator pitch, distilled into clear, memorable statements. Your messaging should connect platform capabilities to developer pain points and needs. Developers don't care about the platform itself, they care about how it helps them build their own software. When defining your platform, your messaging should start with how it benefits developers, not just a rundown of its features.

For example, lead with the benefit developers will gain, then specify which part of the platform delivers that benefit:

- **Deploy to production in under an hour** using the platform's automated CI/CD pipeline.
- **Zero environment setup time** thanks to the platform's infrastructure-as-code (IaC) templates and frameworks like Spring Boot.
- **AI models at your fingertips** with self-service access to curated and evaluated AI models that are right for your application. Just plug and play.
- **Less waiting and fewer security review meetings** due to built-in security compliance and integrated vulnerability scanning.

Each message has two parts: the benefit and how the platform achieves it. When brevity is required, just state the benefit. Define the platform by what it does for developers, not just its technical specifications.

Platform positioning – What is it good for?

Positioning defines where your platform fits in your organization's technical landscape. It answers the crucial question: "When and why should developers choose this platform over other options?"

Oftentimes, platforms are positioned as the everything solution that solves all problems and, thus, should be used for all applications. This might be technically true, but it's helpful to narrow things down to a set of smaller, specific positions at first.

Here are some examples of how to position your platform:

- Your platform is good for cloud native applications, not just *any* type of applications.
- Your platform is a great place to develop and run AI-enabled applications.
- Your platform is a good destination for modernized applications. Many modernized applications target cloud native architectures, moving apps to containers and microservices architectures.
- Your platform is the best place to run Java applications, especially ones that use the Spring Framework.
- Your platform is good for classic, three-tiered web applications: something with a UI, a middleware and business logic layer, and a database.
- Your platform is good for highly regulated apps that need to run in air-gapped environments.

You don't need to pick just one positioning for your platform. After all, platforms are usually general and intended to be used for many different types of applications. However, coming up with multiple positions like the above allows you to speak to specific teams, making it easier for them to sort through all the options and figure out whether your platform is the right fit for them.

Platform value propositions – What's in it for dev teams?

Value propositions, often shorthanded as "value props," are the concrete, measurable benefits your platform delivers. They answer the developer's question "What's in it for me?" with specific, provable outcomes.

Good platform value props for developers focus on specific benefits to developers, not abstract capabilities or business outcomes. For example:

- **Accelerated deployment & efficiency –** Reduces deployment time from two days to 30 minutes and eliminates 80% of security review meetings.
- **Reduced operational toil –** Automates infrastructure configuration, network routing, and load balancing, freeing developers from YAML programming to focus on application development.
- **Rapid delivery of AI value –** Empowers developers with single-command service bindings and redeployments for quick iteration of AI applications.
- **Continuous AI application advancement –** Facilitates easy evaluation and optimization of AI model performance, latency, and safety for ongoing iteration and innovation.
- **Superior developer experience –** Developers can automatically provision configured environments for their applications eliminating the need for support tickets.
- **Seamless onboarding –** Enables new developers to contribute to production-ready code in hours, not weeks.
- **Instant AI model / data access –** Offers self-service, pre-approved access to databases and AI models without ticketing.
- **Built-in observability –** Leverages automatic logging, tracing, and monitoring for faster application debugging and optimization, removing the need to build these systems into applications.

As you can see, developers like ease and speed while being able to try out the latest and greatest trends. Even more so, they hate having to file tickets and wait around for the ticket to be addressed.

What do AI/ Data teams want from Platforms?

Now that we have addressed the application developer audience, we can turn our focus to a new set of platform stakeholders for AI applications: the AI and data team. These teams are critical stakeholders for AI platforms since they will be a partner in delivering AI models to the application developers, so it's important not to forget their influence in platform adoption. When choosing a platform, ask whether it supports continuous AI model optimization so that you can adapt to ongoing input from your AI and data teams. Continuous optimization ensures that AI models remain performant and accurate and that they deliver maximum return on investment (ROI) throughout their lifespan. Some of the value props that could help you convince the AI and data teams about your platform could include the following:

- **Rapid pilot deployment –** Enables rapid deployment of lightweight AI models for pilots so you can validate AI model use cases before deployments.
- **Accelerating initial validation –** A/B model testing allows AI and data teams to compare different model versions and determine which performs best.
- **Iterative refinement and testing –** Robust model monitoring tools provide real-time insights into model behavior and enable teams to swap models as needed.
- **Ensuring accuracy and performance –** Audit and evaluation tools help teams identify and address potential drift, biases, or performance degradation.
- **Continuous optimization –** Continuous monitoring enables proactive intervention and seamless swapping of model versions in production.
- **Maximizing ROI and sustaining performance –** Organizations can adapt AI solutions to changing data landscapes and business requirements.

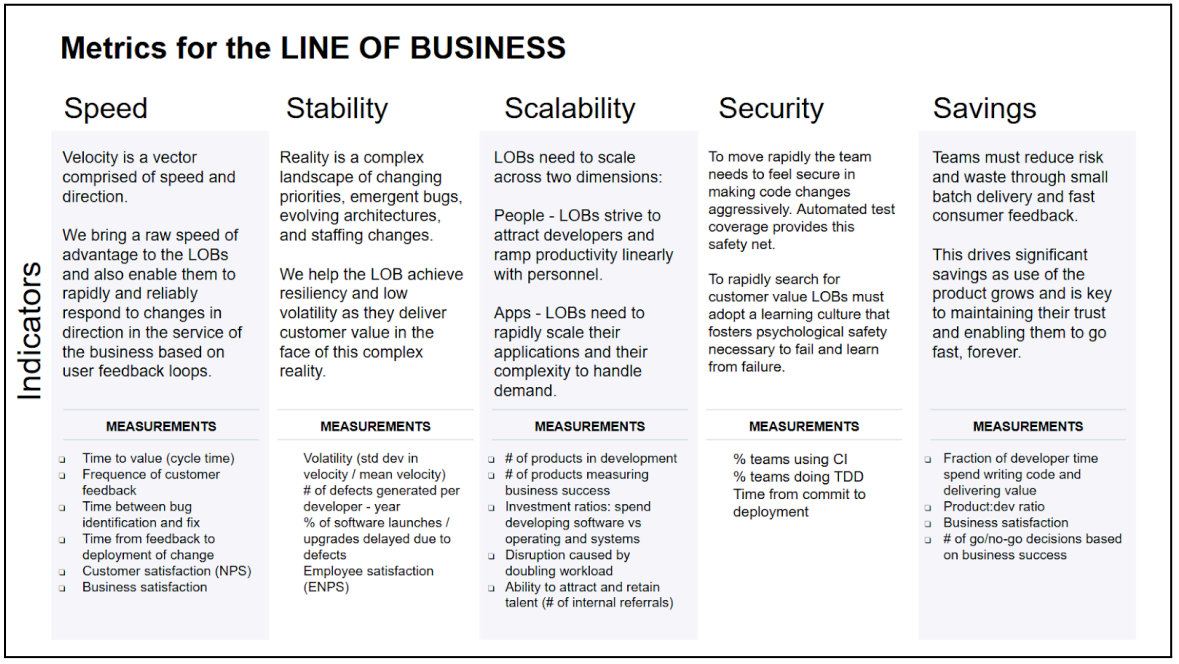
AI and data teams really want their models to work well in the real world. By quickly building prototypes and getting models into production, these teams can learn a lot about how the models are performing and make improvements efficiently. AI platforms help with that. Whether the models are built with your own data, by using GenAI/foundation models, or a mix of both, it's important to help AI and data teams see the actual value of their models. This is key to making the platform a central hub for all these models.

Executives like business outcomes

The above examples are value props that developers and AI and data teams care about, but there's another audience that's important to market to as well: executives. Executives want to know why they're supporting and paying for the platform. For example, this will come up in annual budget planning, when a new executive takes over, and especially when it's time for any license renewals for platform components you use.

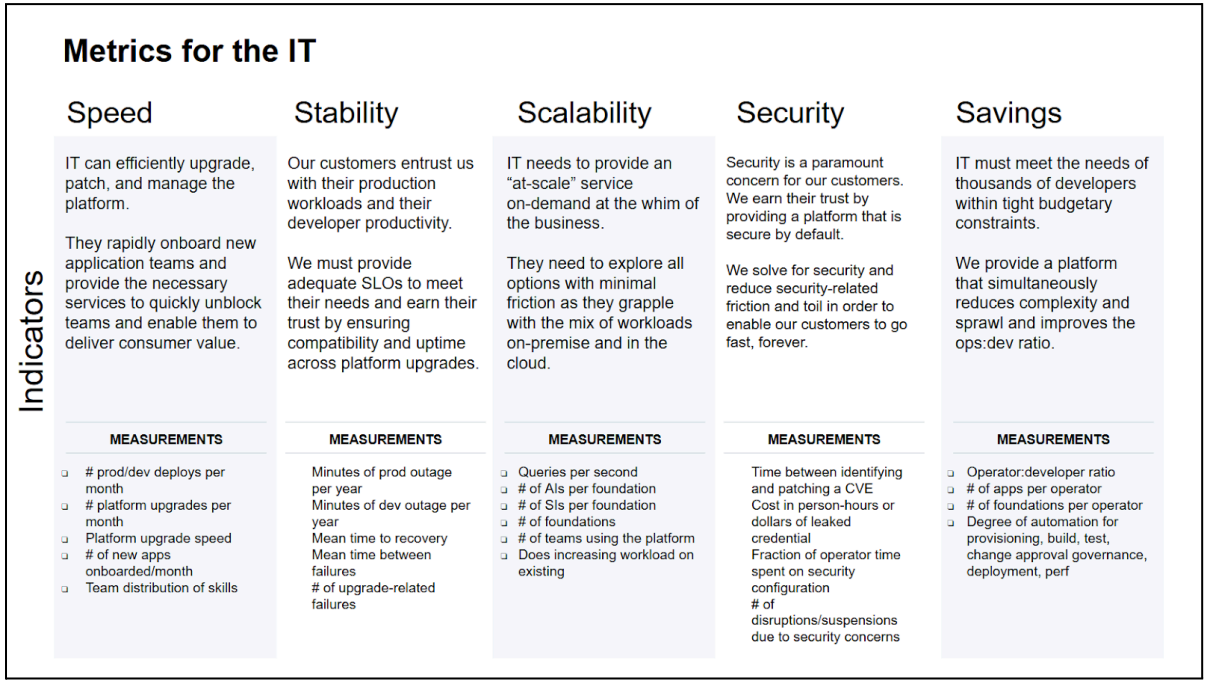
Most executives recognize the value of applications running on the platform based on business outcome: the business is either running well or it's not. However, drawing the connection between those outcomes and the platform itself isn't always obvious. It's similar to plumbing: you don't appreciate it when it works, but you sure notice when it doesn't.

Over the years we've used the "5 S" set of metrics to show the value of platforms to executives. Discussing these 5 S's is a whole topic on its own, but here's a quick overview of the clusters of metrics.



In these business-centric metrics, you can see how things the platform does are mapped to things the business needs to thrive. These metrics demonstrate the value of the platform to executives and help show them why the platform and the platform engineering team is contributing and worth supporting.

For inward-looking metrics, you might alter these to focus on IT goals.



What to measure when you add AI

According to our recent AI survey, CIOs have wishlists of metrics for AI applications. Sixty percent of them indicate that revenue growth is their favorite measurement metric.[4] However, that is closely followed by 52% who indicated staff productivity is their second-most important metric.[5] To assess your platform's support for AI, consider adopting the DORA metrics for platform engineering.

Deployment frequency should be expanded to include how often AI models are deployed, which is much more frequent than databases or other supporting services. In addition to the supporting infrastructure for applications, a higher frequency suggests teams are quickly iterating on models, experimenting with new features, and adapting to evolving business needs. Also, innovations in AI are frequent, so your ability to adapt to new AI patterns and redeploy will be valuable.

Lead time for changes tracks the duration from the start of development to the production deployment of a change. Adding AI to an application—whether a new app or an existing app—is not a "one and done" activity. New models are becoming available all the time, and new methods for optimizing those models are also constantly being introduced (for example, mini-models or model distillation). Lead time to change AI models, or changes to an AI application's code because of an AI model swap, should be tracked because shorter lead time to change enables faster experimentation and deployment of AI applications.

The change failure rate measures the percentage of production deployments that result in issues. With AI models in applications, the problems you encounter will expand to encompass problems with model accuracy, performance degradation, or unexpected behavior. Organizations should be attempting to minimize this rate to ensure not only the reliability and stability of AI applications in production, but also to keep track of the accuracy of the application.

Mean time to restore quantifies the time required to recover from a production failure. For high-performing teams, that amount of time can be hours. For less experienced teams, it can be days or weeks. When adding AI models to applications, MTTR should include model errors, data pipeline failures, and infrastructure outages—basically anything that might stand in the way of getting the AI application up and running again and performing correctly. A faster recovery time is ideal because it means minimal disruption of an application, but in the case of AI apps it is also an indicator of the strength of your cross-team collaboration since you now have a third team as a DevOps stakeholder for AI applications.

---

4 From Cloud Native to AI Native: The CIO's Journey to the New Digital Transformation, July 2025

5 From Cloud Native to AI Native: The CIO's Journey to the New Digital Transformation, July 2025

"5 S" measurement framework for AI applications

We can recommend some potential additional metrics for AI application delivery by applying the 5 S framework:

Speed	Stability	Scalability	Security	Savings
• Time to swap an AI model • Time to modify data pipelines for AI apps • Time to distillation for AI models	• % of correct model responses* • Average or % AI model latency	• % of apps using platform's AI access • # of new apps that use AI models deployed in a year • # of existing apps that use AI models deployed in a year	• % of AI models with monitoring • % of AI models with content safety filters applied	• $ AI model saving costs • # hours saved of staff time

***NOTE:** AI model stability can be difficult to track so you may want to consider utilizing another AI to evaluate responses.

These metrics are just suggestions to think through. Since every AI application is different, pick the best metrics for your platform engineering team and share regular updates with your executive team to champion your platform.

Brand: *Who* is your platform?

"Do you have a T-shirt yet?" is something our colleague DaShaun Carter asks platform teams. That seemingly frivolous question actually cuts to the heart of platform marketing: brand identity. While engineers often dismiss branding as marketing fluff, the reality is that technical teams form deep attachments to their tools—just like sneakerheads obsessing over the latest Air Jordans.

Consider how developers self-identify: Ask them what type of developer they are, and they'll immediately name their primary language—"I'm a Java developer" or "I'm a Python developer." Operations folks do the same—they're "VMware admins" or "Linux admins." And now with data teams in the mix, they are very eager to say they are the “AI team.” This tribal identification with technology tools drives both adoption and persistence through the inevitable technical challenges.

This brand affinity matters even more for internal platforms since you can't piggyback on existing open source or vendor communities. When developers weave your platform into their professional identity, they'll not only stick with it but evangelize it to colleagues. That word-of-mouth marketing is pure gold.

Little wonder, then, that many of the successful platform teams we’ve talked with over the years put a lot of effort into branding their platforms. Organizations like the US Air Force, JP Morgan Chase, and Mercedes-Benz each create brands for their internal development platforms.

Symbols: Names, logos, and color schemes

The basic building blocks of branding—logos, slogans, color schemes, and names—deserve serious attention from day one. While your platform's visual identity should align with your organization's brand, you have room to be more playful and personable. Avoid dry, bureaucratic names like "Enterprise Developer Services" or "Internal Cloud Platform Architecture Cluster." Instead, give your platform a name you'd use affectionately. If you find people referring to your platform by its initials (EDS or ICPAC, with the above), you probably need a better brand name.

Look at successful examples: the US Air Force's "Kessel Run" captures their platform's maverick spirit, while JP Morgan Chase's "Gaia" suggests global enterprise scale. Your platform's name should signal its essence to developers.

Here are some ways to start brainstorming brand names:

1. What superpower does your platform give developers? “Turbo” for speed, “Nimbus” for seamless cloud integration, “Vault” for security and compliance.
2. How would a developer casually mention your platform in conversation? “I’m pushing the app to Shipyard” or “Let’s test it in Vault.” Names that feel awkward in these contexts will struggle to gain organic adoption.

3. What metaphors or symbols from your industry, company culture, or technology space could represent your platform's purpose? Consider using terminology that aligns with your industry. The US Air Force's Kessel Run is a great example because it evokes flight and coolness.

Once you settle on a brand, you should put the logo, name, and colors on all of your platform's UIs, documentation, and maybe even command line tools. You're also going to need to create physical manifestations of your brand: stickers, T-shirts, and banners. This last part is a serious recommendation, done by most organizations I've talked with. It's why our friend DaShaun always asks about the platform's T-shirt first thing.

The platform's brand signals what your platform does for developers and what your platform is. Speaking of, let's look at defining your platform's ethos as part of its brand.

For a little inspiration, here is what the Tanzu Platform team designed for some brand association messages.

Faster and smarter AI app delivery begins with platform adoption. So market it.

To drive internal adoption of an AI platform—or any platform—organizations must strategically approach marketing, product management, and community building, rather than assuming the platform will sell itself. Key to this is defining the customer (application developers and AI / data teams) and tailoring messaging, positioning, and value propositions to their specific needs.

For developers, focus on benefits like accelerated deployment, reduced operational toil, and superior developer experience, using examples such as "Deploy to production in under an hour." For data and AI teams, highlight rapid pilot deployment, iterative refinement, and continuous optimization, demonstrating how the platform helps models work effectively in the real world. Executives, on the other hand, require a focus on business outcomes, using metrics like the 5 S's (speed, stability, scalability, security, and savings) and adapting AI-specific concerns like "Time to swap an AI model" and "Model latency" are meaningful for being able to communicate that the executive is getting value from their purchase.

Finally, fostering a strong brand identity with memorable names, logos, and physical manifestations like team swag can create deep attachment and evangelism among technical teams, ensuring the platform becomes a central hub for all applications, including AI applications.

If you would like to get started using Tanzu Platform for AI workloads, be sure to download the AI Starter Kit today!

By Michael Coté and Camille Crowell-Lee

August 2025

Precursors to this whitepaper are available in a three-part series of articles by Michael Coté:
See part one, part two, and part three.

Item No: vmw-bc-wp-tech-a4-word-2025 Dec-24