DIGITALFABRIC® AISWITCH™

prosight™
AI Engineering Capabilities Study

Series: Platform Engineering

How to Win on Platforms?

Focus Area: AI
Document Type: Knowledge Article

www.digitalfabric.group

Introduction:

In the digital age, technology and business platforms have become the cornerstone for achieving innovation and scalability. From tech platform giants like Amazon, Microsoft and Google to digital platformized businesses like Uber and Lemonade, the platform model has proven to be a powerful driver of growth and efficiency. However, succeeding in the platform economy requires more than just building a platform; it demands a strategic approach to platform engineering.

Here we delve into the strategies and principles essential for winning in the platform economy. We'll explore how to design, build and scale platforms that not only meet technical requirements but also pivot businesses into intelligent platforms combining “services as platforms.

Understanding the Platform Economy

The platform economy is characterized by businesses that create value by facilitating exchanges between two or more interdependent groups, usually consumers and producers. Platforms like Uber, Airbnb and Alibaba have disrupted traditional industries by leveraging network effects, graph and group theory constructs, gamification and data-driven dynamic operating models.

In this context, platform engineering plays a pivotal role in enabling organizations to build robust, scalable and user-centric business and tech platforms. It's about creating the services, intelligence and data infrastructure and tools that allow for seamless interactions and continuous delivery of rapid innovations.

Key Strategies for Winning on Platforms

1. Design Platforms in “Services as Software” constructs

Adopting a “services as software” mindset that integrate the best practices of services and product engineering, is crucial. This means understanding the needs of your user and customer persona’s —developers, partners, distributors —and continuously iterating to meet those needs. Recognizing the value of integration (the “glue”) and having a definite mission and well-defined roles, are critical success factors.

2. Focus on Incremental Value

Rather than attempting to build a fully featured platform from the outset, concentrate on delivering incremental value. Identify the most pressing use cases your platform can address and build features that solve those specific problems. This approach not only speeds development but also encourages early adoption. Incremental integration and traditional spiral models in software engineering can be refactored to apply in this context. MVP can now be called as Maximal Value Platforms, starting with a small set of critical features and services at core or kernel, and then adding more functionalities and story points in multiple dynamic shells, delivering unique user journeys.

3. Design for Self-Service and Automation

Empowering users through self-service capabilities, LCNC (low code no code) functionalities and automation in a flexible 'canvas and toolbox" mode, is a best practice proven in successful platforms. By reducing manual interventions, you enhance efficiency and user satisfaction. Microsoft, for example, emphasizes that platform engineering should improve developer experiences while maintaining security, compliance and cost control.

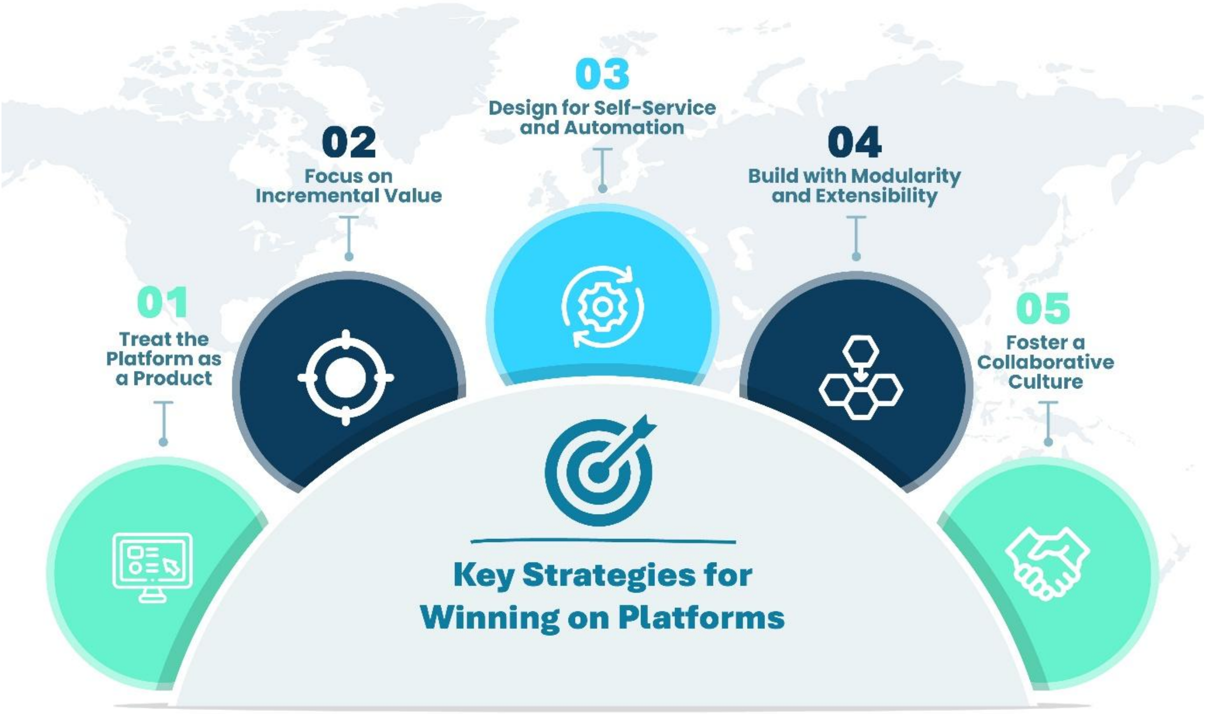
4. Build Composable Platforms with Modularity and Extensibility

A modular and extensible design for composable platforms with intelligent orchestrators, ensures that your platform can adapt to changing requirements and integrate new technologies seamlessly. Composable and dynamically orchestrated platforms with modularity and extensibility allow new tools and services to be integrated in plus and play, thus avoiding legacy lock-ins and allowing users to stay at the edge of new technologies. Modular platform designs also make them scalable and cost-elastic based on demand fluctuations.

5. Foster a Collaborative Culture

Collaboration across teams is vital for platform success. Encouraging cross-team communication and knowledge sharing leads to better design thinking and innovation. A collaborative culture ensures that platform engineering efforts align with organizational goals and user needs.

WINNING IN THE PLATFORM ECONOMY:
STRATEGIES FOR SUCCESS


01
Treat the Platform as a Product
02
Focus on Incremental Value
03
Design for Self-Service and Automation
04
Build with Modularity and Extensibility
05
Foster a Collaborative Culture
Key Strategies for Winning on Platforms

Measuring Success in Platform Engineering

Consider using the following metrics to gauge the effectiveness of your platform engineering efforts.

- Lead Time for Changes: Time taken from code commit to deployment, scalability (capacity changes), SLA changes, change-related release frequencies
- Deployment Frequency: How often new releases are deployed.
- Mean Time to Recovery (MTTR): Time taken to recover from failures, resilience
- Change Failure Rate: Percentage of changes that result in failures.

These metrics, known as DORA metrics, provide insights into the platform performance and reliability.

Conclusion

Winning in the platform economy requires a strategic approach to platform engineering. By treating your platform as a product, focusing on incremental value, designing for self-service and automation, building with modularity and fostering a collaborative culture, you set the stage for success.

Reach out to us

Tap into expert insights and transformative strategies to lead your enterprise into the age of AI, Cloud, and Intelligent Automation. Connect with us to accelerate growth, optimize operations, and unlock new value at scale.

Email ID: rishant.singh@digitalfabric.in

Learn more about DIGITALFABRIC® for more insights:
https://digitalfabric.group

Connect with DIGITALFABRIC®

prosight™
AI Engineering Capabilities Study

DIGITALFABRIC® AISWITCH™