



Make It Better, Faster, Forever: Building the Software Factory

Table of contents

I. A state-of-the-art factory for software development	3
Software that enables better outcomes	3
Achieving digital transformation	4
II. The pillars of the software factory: Fundamental principles	4
III. People: Empowering developers to deliver value, forever	6
IV. Process: Tailoring development best practices to deliver the best results . . .	7
V. Technology: The platform that powers the software factory.	9
VI. The ultimate prize: What a software factory produces.	10

What is a software factory?

Pioneered by Pivotal Labs, now called [Tanzu Labs](#), a software factory is an organized approach to software development that provides the right, customized mix of people, process and technology, to enable the production of a widget consistently and efficiently. Software factories leverage reusable patterns while providing a flexible framework that allows for adaptation, agility and evolution over time.

A widget can be anything software related: a software capability, app or product; a skilled team of developers; a common, secure environment that supports the software lifecycle, from development to production and maintenance; or any combination thereof.

I. A state-of-the-art factory for software development

Today's automotive factories are a marvel to behold: a linear constellation of human technicians and robotic assistants, working shoulder to shoulder to produce the ultimate prize—a pristine vehicle that is defect-free and in compliance with all regulations. The assembly lines that cars and trucks roll off are both time-tested (the concept was introduced by Henry Ford over 100 years ago) and modern, enhanced continuously with innovations that improve production speed and quality. As the foundation of the automotive industry and many other manufacturing domains, the assembly line is *proven, repeatable and scalable*.

State-of-the-art factories can produce more than just cars. Across industries, leading organizations including Ford, Dell, GE, Accenture, Allstate, Boeing, Toyota, the U.S. Army and the U.S. Marine Corps share a common defining characteristic: They are all software companies. Every modern firm is powered by software and, as such, these leaders have created their own *software factories*, fueling success with exceptional digital products and teams built within them. This white paper focuses on what software factories are and the assembly lines they build: a combination of people, process and technology that can produce a wide range of outputs—high-performing teams, upskilled talent pools, and software that is defect-free, secure, and in compliance with all regulations.

When done right, software factories can ignite any IT organization's ability to deliver the ultimate prize: software that is better, faster, forever.

Software that enables better outcomes

Strategically, the goal of a software factory is to produce rapidly and efficiently the output required to propel the business forward. This typically includes getting new apps deployed faster into the hands of users, modernizing legacy apps, and training or upskilling development teams, all while reducing operating costs and risks.

In the past, software was often produced with business units and IT organizations operating as two entirely separate departments; business managers specified the necessary software capabilities and IT departments built them. A more modern approach to delivering better apps, faster, starts with business and IT groups working closely together, aligned on common goals and a shared understanding of the challenges to be overcome in attaining them. In government organizations, shared goals can be described as a common mission.

Building modern software at lower cost and with less risk also calls for IT organizations to work differently. Silos must be broken down to build a DevSecOps approach incorporating culture, automation and platform design that, critically, integrates security as a shared responsibility throughout the entire software lifecycle.

What is DevSecOps?

[DevSecOps](#) is a method to integrate security as part of the software development process. Among other benefits, DevSecOps makes it easier to avoid including common vulnerabilities and exposures (CVEs) in new applications, and easier to patch newly discovered CVEs in existing code.

DevSecOps is a key pillar of the software factory. With app development, security and operations teams working collaboratively, better apps can be delivered faster, while still adhering to the highest security standards.

Achieving digital transformation

For more than a decade, organizations have been striving to use, with various degrees of success, digital technology to fundamentally change how they operate and deliver value. This *digital transformation* touches all areas of the business, but for many organizations, it is elusive because the software apps that catalyze transformation can't be developed or updated fast enough. In addition, the teams that are required to build these apps may not have the skill sets required to meet the most up-to-date, modern software standards set by technology organizations.

To accelerate app development, some IT organizations adopt new software technology such as containerization, or new development processes such as agile. Either approach, individually, can positively impact the speed and quality of software delivery. But new technology, combined with new processes and deep cultural change, can yield far more powerful improvement and a self-sustaining innovation process. This is the essence of the modern software factory.

II. The pillars of the software factory: Fundamental principles

Just as stamping, welding and painting are key stations on an automotive assembly line, a software factory's digital outputs are grounded in several core conceptual principles. Together, these principles form a modern mindset for software development, and the pillars of the software factory itself.

- **Product vs. project** – Today's software apps are anything but generic or off-the-shelf; they are evolving sets of reusable services, integrated with existing systems to drive valuable business outcomes. As such, businesses need to bridge longstanding disconnects with the IT organization to manage software development holistically. Whether internal- or external-facing, software is no longer a "one and done" IT project. Instead, software is a product, a set of digital capabilities that evolves with the target audience and its needs. From a practical standpoint, treating software as a product also minimizes delivery time, risk and waste, while allowing security issues to be detected and remediated early in the development lifecycle.
- **Building to achieve goals** – Similarly, modern software apps must deliver more than prescribed functionality. They must embody the longevity and flexibility necessary to support the business over time. Like an automotive assembly line, the software factory must emphasize producing digital output that is
 - Consistent
 - Repeatable
 - Scalable
 - Secure

A software factory’s output must also be measurable and tied to well-defined business goals. If goals are not clearly articulated at the outset, it’s difficult to understand what success looks like and how to iterate on what works.

- **Defining the product** – What does the organization want the software factory to produce? The answer can vary widely depending on goals, which can be broad, specific, or a mix of both. The exact components that make up a software factory can depend on an organization’s technical capabilities and software development goals. A traditional enterprise with a mix of mainframe, client-server, and cloud technology may have different requirements than a company born in the digital era—with different priorities, levels of product maturity, and software development skills. However, the software factory approach may benefit any type of organization implementing DevOps or DevSecOps.

For example, the U.S. Marine Corps’ software factory was designed to develop the digital literacy necessary for modern defense, including plans to deploy fully enabled Marines to provide commanders with software capability at the tactical edge. Commercial enterprises may have a goal of leapfrogging over competitors by getting to market faster with breakthrough apps that win customers. Still others may need to accelerate app modernization initiatives to achieve large-scale cost reductions, bringing legacy apps into the cloud.

With these pillars in place, organizations have already begun building their software factory. Tooling the factory floor—with people, process and technology—comes next.

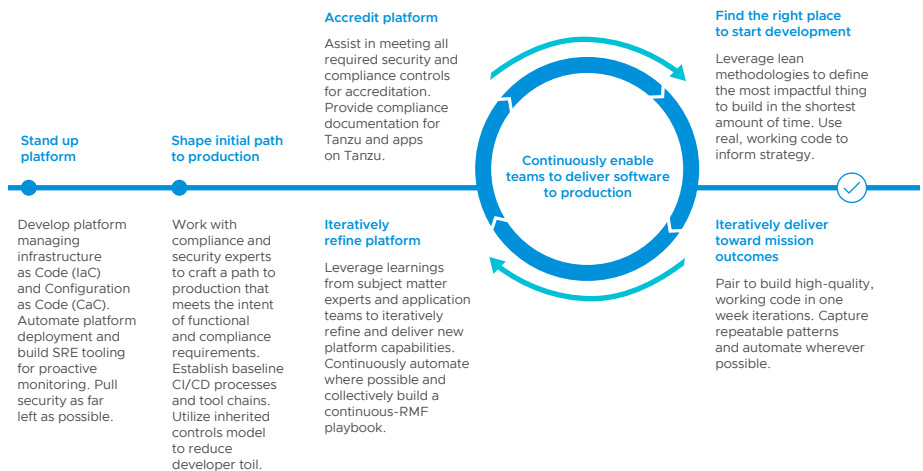


Figure 1: Software factories leverage reusable patterns while providing a flexible framework that allows for adaptation, agility and evolution over time.

Spotlight: The U.S. Army Software Factory

When The Army Futures Command discovered that the U.S. Army needed more responsive software development, they built a soldier-led software factory. After three years, this decision is bearing fruit across the armed services.

[The U.S. Army Software Factory turned to Tanzu Labs](#) to build a robust, compliant and resilient software development process to deliver applications to production more quickly. Educating the key staff up front enabled the organization to modernize, transform and achieve better software outcomes right from the start. In partnership with Tanzu Labs, the Army Software Factory trains soldiers and civilian employees in modern software development practices and is producing software that better addresses current and future mission priorities and the needs of soldiers.

III. People: Empowering developers to deliver value, forever

Improving business performance through software development starts with empowering developers and creating the right environment and culture in which they can succeed. This is the first critical step in tooling the software factory; by educating developers on how to develop and scale an organized approach to building software, software teams are equipped to maximize their productivity. IT organizations are thus immediately empowered to modernize, transform and achieve better business outcomes.

In addition to internal training, coaching and incentives to accelerate the shift from a project to a product mindset, the developer education program should ideally include

- **Alignment on goals, challenges and a common mission** – IT leaders can team with their business counterparts to jointly illustrate shared goals and how software apps play an essential catalytic role in achieving them. Developers should be able to articulate the goals and associated challenges, and understand the organization's common mission.
- **Breaking down IT silos** – The benefits of a DevSecOps approach can only be achieved when development, security and ongoing software operations processes are integrated. That typically requires IT organizations to not just dismantle these siloed functions, but to teach teams how to collaborate efficiently. New development approaches, as outlined in Section IV below, can impart many of the DevSecOps processes that will remain embedded going forward.
- **Upskilling** – New app development paradigms and their cloud migration can leave some software professionals behind. Developers upskill by learning how to solve problems in new domains outside their purview. That requires providing training resources for relevant, useful digital skills to ensure that developers can keep pace with digital transformation. A structured approach to upskilling typically includes
 1. Assessing skills and knowledge gaps
 2. Setting clear goals and objectives
 3. Creating a learning and development plan
 4. Providing learning resources and tools
 5. Promoting a learning culture
 6. Supporting ongoing learning and development

IV. Process: Tailoring development best practices to deliver the best results

Conceptually, a software factory brings together the best and most innovative approaches to software development to accelerate digital transformation. Rooted in DevOps and agile software development methods, a software factory combines tools, teams and practices to organize, standardize, store and reuse code, enabling teams to efficiently build upon accumulated knowledge. Both developers and software architects benefit from a more organized and structured approach, while the entire organization benefits from faster software delivery and higher quality code.

The foundation of a software factory is the tools, services, repositories and practices used to deliver software. This typically includes stored recipes, templates and reusable code that can be leveraged to quickly create new applications while minimizing the amount of original code needed. This foundation enables teams to plan, build and deploy software more predictably.

Best Practice Focus Area	Skills and Capabilities Developed
<p>Modern Software Engineering Practices Extreme programming (XP) and cloud native development practices empower developers to write quality code and develop apps to utilize cloud technology.</p>	<ul style="list-style-type: none"> • Foundational software engineering (Java/.Net) • Cloud native / microservices architectures <ul style="list-style-type: none"> – Spring Framework – 15-factor patterns – API-first development – SWIFT methodology – Domain-driven design – Path-to-production analysis • Agile/XP development <ul style="list-style-type: none"> – Pair programming – Test-driven development – CI/CD • Application modernization approaches • Monolith decomposition • Application portfolio assessment and modernization

Pilot projects: Small projects, big wins

Traditional software development methods break down monolithic projects into discrete phases and deliverables. However, the value of this approach is difficult to demonstrate at any point other than the eventual go-live date.

Software factories emphasize the power of the pilot project, an initial small-scale implementation that is used to prove the viability of a software app or idea. Pilot projects allow software factories to quickly produce apps that can be assessed for their ability to support business goals, allowing development initiatives to be prioritized accordingly.

<p>Agile Product Design and Management Tools, patterns and methods lay the foundation for a healthy agile product development practice.</p>	<ul style="list-style-type: none"> • Leadership development • Product office development • Stakeholder (LoB) engagement • Lean product development and management <ul style="list-style-type: none"> – Product vision and strategy – MVP definition – Lean experiments – Balanced teams • User-centered design (UCD) <ul style="list-style-type: none"> – User interviews – Ethnographic studies – Persona definition – Service blueprints – Prototype creation
<p>Platform Engineering and Operations DevOps and DevSecOps practices support a frictionless developer experience.</p>	<ul style="list-style-type: none"> • Container platform operations (Hybrid/Multi-Cloud) • DevSecOps • CI/CD and platform automation • Continuous platform compliance • Platform-as-product <ul style="list-style-type: none"> – Day 2 operations – End-to-end automation – Site reliability engineering – Developer onboarding • Developer portals/golden path

[Learn more about how Tanzu Labs can improve the way your team builds custom software.](#)

V. Technology: The platform that powers the software factory

The Tanzu Application Platform (TAP) is a modular, application-aware platform that any organization can use to build and operate its software factory, running on any compliant public cloud or on-premises Kubernetes cluster. It delivers a superior multi-cloud developer experience with a pre-paved path to production, including preconfigured components that allow apps to be built and deployed quickly and securely. TAP is a composable platform that teams can customize based on their organization's preferences and changing business needs.

Tanzu Application Platform works across any Kubernetes cluster to simplify the developer and operator experience. It provides an ideal end-to-end, streamlined DevOps experience so teams can

- **Jumpstart application development** – Developers get a head start on cloud native apps with application accelerators. These templates include skeleton code, configurations, and cloud native patterns combined with an organization's best practices and security policies.
- **Boost developer velocity** – Developers get a consistent graphical user interface (GUI) to underlining services and APIs, making them easy to discover and integrate with other apps. Developers can work in their integrated development environment (IDE) with Tanzu Application Platform developer tools. They can debug a running container on any Kubernetes cluster (running locally or on a managed Kubernetes cluster) and quickly update code in a running container, enabling them to quickly test and iterate.
- **Accelerate the path to production** – Operators can use the Supply Chain Choreographer to create preapproved paths to production, customizing this tool to bring their own opinions to meet business needs. In addition, the build service uses Cloud Native Buildpacks to automatically create containers from validated building blocks and update them with no manual intervention. These capabilities reduce friction for developers and enable a stronger security posture.

[See what others are achieving with Tanzu Application Platform.](#)

VI. The ultimate prize: What a software factory produces

Every automotive assembly line is built to produce pristine vehicles that are defect-free and in compliance with all regulations. That is the analogous goal of the modern software factory: to quickly produce reliable, highly performant, widely adopted software and achieve goals that move the business forward. In doing so, software factories can

- **Increase app quality** by standardizing app development practices across a portfolio of projects, at scale, with a focus on utilizing proven practices and reusing code.
- **Enhance developer productivity** and decrease toil through the combination of well-understood methods, reusable code, and process automation.
- **Improve application release cadence** by speeding up the process of creating and testing new application code.
- **Ensure consistency** by standardizing the process(es) necessary for software delivery across projects, while lowering training and maintenance costs.
- **Establish a useful catalog** of software and code that documents reusable code and patterns in cookbooks, helping the organization to take better advantage of knowledge previously gained.
- **Enable DevSecOps** to be achieved by integrating security into the software development process, resulting in more secure software with less effort.
- **Create a framework for change** by facilitating collaboration and catalyzing the culture shifts necessary to achieve digital transformation.

Ready to take the next step? Schedule a free hour of consultation now!

If your organization is looking to establish a new standard of software delivery, train and upskill talent, or even build out a brand-new software function, establishing a software factory may be the logical next step.

If you're ready to learn more about software factories and how to assess if building one is right for your business, why not book a free hour of consulting with our Tanzu Labs team? Our experts will guide you through the specifics of implementing a software factory and what's needed for success. Schedule your free consultation today at tanzu.vmware.com/office-hours.

