

Webhead



WHITE PAPER

Software Factory: The Ultimate Guide to Delivering High-Quality Software for Business

Webhead

Web-Hed Technologies, Inc. (dba Webhead)
1710 N. Main Ave • San Antonio, Texas 78212 • (210) 354-1661 • webheadtech.com

Copyright© Web-Hed Technologies, Inc. (dba Webhead) (1994-2022). All rights reserved.

Software Factory: The Ultimate Guide to Delivering High-Quality Software for Business

Have you ever wondered how companies like Google, Facebook, and Netflix can ship software so fast and so often? Their apps and services are part of our daily lives, and we can see first-hand just how quickly and effectively they can churn out bug fixes, introduce new features, and manage scaling.

The Googles and the Netflices of the world are able to push streamlined, clean software regularly thanks to a relatively new software development approach known as software factory. The software factory approach is helping organizations revolutionize their computing power. But what is a software factory, anyway? And how does an organization become one?

In this whitepaper, we'll introduce software factory, the history of the software factory concept, its basic building blocks, and how organizations can realize software factories. We'll also explore the potential benefits of software factories for software development projects. Read on.

What Is a Software Factory?

Software factory is an emerging concept for mass production of software and continuous innovation. A software factory describes a factory approach to software development; fast, efficient, and automated. It's simply a package containing proprietary tools, processes, and components that development teams can easily arrange and process to build software quickly.

Think of it as a factory-built for mass production of goods, but in this case, the product is software. Akin to a production line in an actual factory, a software factory will standardize its assembly process, including the build, quality, security, and deployment stages. Various development teams within the organization can reuse this process.

In other words, your internal development teams can create applications through an assembly-like process using the structured collection of related software assets. The software assembly process helps ensure that the company can consistently enforce code quality and security requirements across multiple development teams.

Economies of Scope

A software factory mechanizes software development by taking advantage of the economies of scope. Economies of scope occur when a company produces two or more goods together, resulting in lower marginal costs compared to producing these goods separately. In software development, a company might produce a range of programs and feature sets using shared resources.

Capitalizing on economies of the scope allows you to get the most you can out of your fixed costs. Thanks to the software factory approach, organizations can roll out high-quality products and features faster, easily, and frequently while maintaining high-profit margins. Otherwise, software development costs would grow exponentially.



History of the Software Factory

Software factory is such a buzzword at the moment, but it is not entirely a new concept. The idea was first introduced by R. W. Bremer of General Electric and M.D. McIlroy of AT&T in the 1960s. The latter focused on the systematic reuse of code when creating new software systems while the former emphasized the use of standardized tools and controls.

The approaches proposed by Bremer and McIlroy were a little different in their goals, but the software factories we know today are based on their original ideals. One of the first companies to adopt this term was Hitachi in 1969 with its Hitachi Software Works. At the time, Japan lacked skilled labor for the software development industry, hence adopting software factories.

How Software Factory Works / Methodology

How does a software factory work? In this section, we look at the basic building blocks of software factories that allow companies that implement this paradigm to industrialize software development.

Product Line

For a long time, manufacturers have used a common factory to assemble and configure parts for reuse across a line of similar products. The crux of the software factory concept is the creation of software from reusable parts. It involves a collection of software assets that the development team can configure and compose in different ways to create different products in the product line.

To successfully implement software factory, companies need to create a portfolio of closely related products with variations in functions and features to target the needs of prospective customers. Having a product line allows for the systematic reuse of software assets in the development process, which is the core concept of software factories.

Architectural Framework

The architectural framework provides the baseline architecture for all products of a product line. The baseline architecture incorporates the best practices and patterns for the products in a particular product line. Having an architectural framework allows internal development teams to create high-quality software with uniform architecture.

As you can see, architecture is a key component of the software factory concept. That doesn't mean that you are going back to the waterfall model. No. It ensures that you have a system blueprint when you implement features with a set of software assets/components.

Businesses that adopted the factory software paradigm saw significant improvements in productivity, process control, and software quality. Nevertheless, software factories didn't quite catch on until the 2000s with the publication and release of 'Software Factories, Assembling Applications with Patterns, Models, Frameworks, and Tools.

With the publication of their book, Jack Greenfield and Keith Short sparked a new interest in the software factory concept. The writers described an approach that streamlines and automates software development, allowing companies to become more efficient and churn out high-quality software faster. Software factories are now mainstream in the world of software development.



Automated Guidance

Software factory implementations can help provide automated guidance in context when developing an application using reusable processes and templates. Because you have already implemented similar products before, you can provide guidance to developers in the form of best practices. The guidance is automated in the sense that it is derived from previous implementations.

Guiding the development team regarding what to do, when, and how to do it has numerous advantages. It will reduce the learning curve for new/inexperienced developers and lead to more efficient development and increase the quality of software by preventing mistakes.

Model-Driven Development (MDD)

Model-driven development (MDD) is a key element of software development and one of the building blocks of software factories. This methodology allows developers to build complex software through simplified abstractions of pre-built components. Software factories use business domain concepts such as workflow or activity to achieve higher abstraction levels.

The factory software concept provides models at different abstraction levels, allowing developers, testers, and other stakeholders to fill in the required information at the levels they are familiar with. MDD eliminates the complexities of application development through abstraction, allowing stakeholders with no programming experience to participate in the development cycle.

Benefits of Software Factories

What are the benefits of implementing a software factory?

This section explores how custom software development and the software factory can improve your business and deliver high-quality software.

Continuous Innovation

When it comes to application development, the software factory is fast, efficient, and cost-effective. By allowing the better organization of the teams involved in the design pipeline, the software factory approach facilitates the continuous innovation required to maintain product leadership in the market.

Mitigate Risks

The software factory approach helps development teams mitigate the risks associated with the development process. You'll always be able to create a product that meets your current business needs through incremental releases instead of waiting for months for a new product. Improvements will be tested by users immediately, allowing you to intervene promptly in case of a problem.

Accelerate Application Time to Market

The software factory accelerates application development, delivery, as well as any corrective changes or additional features to the software significantly. The quick delivery is a result of the agile frameworks and the synergy between the various stakeholders involved in the development process.

Align Software to Business Needs

The software factory approach facilitates close collaboration between the development team, internal IT staff, and end-users. Consequently, applications can be aligned with the needs of the business. Input from all stakeholders also increases the quality of the software significantly.

Clear Roles and Responsibilities

In a software factory model, roles and functions are clearly defined. There are no gaps and overlaps when it comes to roles and responsibilities. Developers, architects, and product managers can all work together to create software without any confusion on who's supposed to do what.

Conclusion

The software factory approach will standardize, automate, and scale your business processes. Picture a software plant with multiple assembly lines. The factory lines are equipped with a set of approved tools, including templates and code, and workflows. These processes allow internal development teams to create software more quickly, securely, consistently, and with reduced risk.

The software factory approach simply offers a better way for software companies to operate, which is why the concept has become so popular in recent years. Implementing a software factory within your organization can be an intimidating task. There's no need to worry, though. We can help you get started.

Trust Webhead's Software Factory to bring the best-practice software development processes and methodologies to your web, mobile, and product engineering project. Our approach will ensure a successful launch and prepare your organization to scale. Contact Webhead to speak to an expert today about your project.

