Agile development: Building composable software for digital efficiency

contentstack.com/blog/composable/agile-development-building-composable-software-digital-efficiency

Until a few years ago, developers built software and applications using the traditional monolithic approach. And they were satisfied with it.

But, since composable software became a buzzword, the software development world hasn’t looked back.

But what led to the shift from a monolithic to a composable approach?

Dynamic business and customer requirements and businesses’ inability to adapt to such rapid changes gave rise to the new revolution called "Composable Software."

With the composable applications and infrastructure market likely to reach **$14.35 billion by 2028**, most next-gen applications will be composable and integrated with video and speed capabilities while offering enterprise-level scalability.

This article explores the fundamental principles and steps in creating a composable modular software infrastructure.

But let's start by exploring what composable software is.

What is composable software?

Composable software is built using reusable building blocks. This involves assembling loosely coupled and independent components and services to create complex applications.

Remember the time you played Lego and used the Lego block to assemble, disassemble, and create innovative things? Composable software works on the same principle. You can build composable applications by making smaller components instead of starting the development process at step one.

Watch Video At: https://youtu.be/jrVSh-uR_xs

This results in more flexibility and adaptability for business operations. A practical example of composable software is a process where you can reuse functions from existing libraries to build your application.

The role of application programming interface (APIs)

Composable software is primarily built around APIs. As a result, APIs ensure communication between software components, helping you to assemble these components and create complex applications.

Another reason APIs are critical in composable software is that they provide abstraction. They expose only the necessary interfaces and mask the complex implementation details.

This helps you focus on integrating new functionalities rather than worrying about the technical know-how of every component.

How do APIs improve application development efficiency?

With APIs, you can compose and decompose applications based on changing business requirements. As it incorporates pre-existing functions, you no longer have to develop an app from scratch, helping to save you time and money.

In addition, you can implement changes and upgrades to a single component without disrupting the entire application. This ensures the application development process is leaner and more agile, helping you meet your growing business demands.

Advantages of composable architecture

Composable architecture helps you build more scalable, flexible, and adaptable digital applications. Here are some benefits of composable architecture:

Improves efficiency and development speed

Composable architecture deconstructs a system into independent components, improving efficiency and development speed. Writing and testing unit cases becomes easier as you can test individual components. Composable architecture makes identifying and rectifying errors easy, which speeds up development.

The speed at which you can launch a feature in a composable environment is much higher. Because you only test/build small and independent components rather than fully-grown apps.

Contentstack accelerated K2 Sports website implementation by 75%

K2 Sports had difficulty managing its website, which was built on a homegrown CMS and Demandware. Its team experienced development challenges due to the highly technical nature of its CMS. So, K2 used Contentstack's headless CMS.

Contentstack's intuitive interface made the business less dependent on developers and enabled automation. **Read the case study** on how implementing Contentstack DXP resulted in **75%** faster website development, **90%** faster content delivery, and a **50%** increase in productivity.

What do our customers say?

"Contentstack is a versatile headless CMS. Onboarding was straightforward and quick. It was easy for the developers to set up and integrate applications with it. It was easy for the project managers and marketing people to use." - Nicole Fugere, Director of Web Services, K2.

Promotes innovation

The modular nature of composable architecture fosters a culture of innovation. As a result, you can test new features or functionalities without causing any disruptions to the pre-existing systems.

This ability to experiment and iterate paves the way for continuous innovation in the business environment, ensuring that businesses stay ahead of their competitors while opening new frontiers in software development.

Offers a more secure option

Compared to monolithic architecture, composable architecture provides higher security. As every component operates independently, this reduces the chances of cyberattacks. If one part of the component gets compromised, you contain the impact on that component, ensuring it doesn't disrupt the existing system.

Ensures flexibility

Composable architecture is flexible and modular, allowing you to modify, upgrade, or replace components without disrupting the entire system. This flexibility is crucial in the digital landscape, as it helps you to respond quickly to market requirements.

Today, developers seek such flexibility, as it improves the application while enhancing its stability.

Contentstack accelerated REWE content creator's workflow

REWE core website REWE.de was built using an expensive, customized, Java-based monolith CMS. This led to maintenance problems, unpleasant developer experiences, and complex dependencies.

So, REWE chose Contentstack's headless CMS as it offered a more flexible approach, making it possible to roll out implementation in phases. **Read the case study** on how this phased implementation enhanced the productivity of their content manager and streamlined the workflows towards an agile content development process.

What do our customers say?

"You don't dismantle a decade-old monolith overnight. Our phased approach meant we could course correct as we progressed, maximizing the return on the investment in the pilot phase. In addition, deeper technical sessions for the most advanced content managers established trust." - Eduardo Zamin, Engineering Manager, Rewe Digital.

Provides scalability

Scalability is one of the benefits inherent in composable architecture that helps businesses increase system size. The architecture utilizes modular components for scaling specific functionalities. When one part of the system scales, it doesn't impact the other components, creating a resilient and flexible system.

Composable architecture is cost-effective as it allows you to have scalable and responsive IT systems. This ensures optimum use of resources.

Building blocks of a composable software

The building blocks of composable software are packaged business capabilities (PBCs). These independent modular components of functionality help build a composable system.

PBCs include inventory control management, customer relationship management, payment processing, and order management.

The power of PBCs lies in their modularity, as these can be used independently or with other PBCs.

How modular components lead to a composable business

As modular components fulfill a specific function, you can combine them to create an adaptable and comprehensive business solution or software. Adding, deleting, and modifying components empowers you to create customized applications and swiftly react to dynamic business requirements - leading to a composable business.

Modular components ensure continuous improvement and innovation in a composable business. You can replace or change these components independently, helping you integrate the latest technologies and practices in the development process.

This means your business stays at the forefront of technological advancements and creates a continuous learning and adaptation culture.

Strategies for building composable software

Here are some strategies to consider when developing applications with composable design:

Build a composable infrastructure

Composable infrastructure represents physical resources like storage, networking, and computing components. You can dynamically pool and allocate these resources based on the specific requirements of applications.

In a composable infrastructure, you treat your devices as a service. When building one, ensure it allows a seamless integration with different PBCs.

Use microservices and APIs

Microservices break large applications into independent and scalable services. Each service is developed, deployed, and scaled independently, promoting agility and resilience. APIs help these microservices to communicate with each other and other systems, ensuring interoperability.

In tandem, they develop agile and modular architectures that meet dynamic business requirements.

Implement agile methods

Use an agile development approach to build composite software. Agile development relies

on iterative development, continuous improvement, and responsiveness to change, principles that align with composable designs' flexible and modular nature.

You can develop, test, and refine individual components in parallel, reduce development time, and increase an app’s quality using agile methods. Agile software uses continuous integration and delivery (CI/CD) techniques. This helps you to make changes or add new features to the existing components.

FAQs

1. How does composable software support digital transformation?

Composable software supports digital transformation by helping businesses to adapt to changing market requirements and demand. Its modular design ensures easy integration with existing IT infrastructure, facilitating seamless digital transitions.

2. What are the cost benefits of switching to composable applications?

The shift to composable applications brings about substantial cost savings. These modular components reduce the time and cost of development. As composable software offers flexibility, it can easily accommodate changing business requirements, reducing the need for costly overhauls.

3. How do composable technologies integrate with existing IT infrastructure?

To integrate composable technologies with existing IT infrastructure, use APIs. This ensures a seamless workflow between different software components, irrespective of the technology or platform they are built on.

Are you ready to embrace composable software?

Adopting composable software is the smartest thing you can do to remain ahead on the digital innovation curve, as it will revolutionize your software development process.

Embracing this innovative approach gives you the flexibility and scalability to manage changing business requirements.

Ready to take your next step toward digital efficiency? Join us for a free demo to see how composable systems work. Find out how you can use these systems to increase the efficiency of your development activities and gain a competitive advantage. Sign up now to unlock the power of composable software!