

# TCO e-Book



# THE NEW APPROACH

TO TCO FOR DIGITAL COMMERCE

Why modern, composable solutions are more cost-effective than legacy, monolithic platforms

Published in partnership with:



# TCO e-Book

## THE NEW APPROACH to TCO for digital commerce

Co-authored by Sven van Hegelsom and Tom Jones,  
commercetools Customer Value Services

This eBook has been created to address the limitations of traditional TCO calculations in capturing the true costs and value of technology investments in a world of constant change. We believe a better methodology for calculating TCO includes evaluation of additional metrics like Total Cost of Change (TCC) and Total Spend Productivity (TSP).

## Content

<b>Introduction &amp; key takeaways</b>	3
<b>What is TCO?</b>	5
<b>Standard measures of TCO for composable commerce</b>	6
TCO part 1: Total (Scope)	6
TCO part 2: Cost	7
TCO part 3: (of) Ownership	10
<b>Technical debt: The hidden costs of traditional platforms</b>	11
Business impact of technical debt	12
<b>Modern digital commerce: Additional TCO evaluation metrics</b>	13
TCC: Total cost of change	13
TSP: Total spend productivity	14
Bringing TCC and TSP together	17
<b>The benefits of a modern approach to digital commerce</b>	18
Growth enablement	18
Cost optimization	19
Risk reduction	20
<b>Next steps</b>	22
<b>About</b>	23
<b>Appendix: Cost comparison of traditional &amp; modern commerce solutions</b>	24

# Introduction & key takeaways

With a traditional monolithic tech stack, pricing is relatively easy to define in the beginning because you know what you are getting out of the box. However, this typically requires multi-million dollar contracts, long decision processes and inability to change and flex to your business requirements as the market changes over time.

The MACH Alliance global 2023 Enterprise MACHified survey of 500 large business tech leaders, shows that upgrading is a huge burden which organizations are struggling to reduce. Over a quarter of respondents run more than 20 projects each year. One in five are spending over half of their IT budget on upgrades, and the same number say upgrades are taking up over half of IT teams' time. With 41 percent of organizations' IT ecosystems still being legacy on average, there is a clear need for tools which can enable them to improve the ability to upgrade infrastructures at speed.

MACH allows you to build a best-of-need tech stack leveraging best-of-breed vendors ensuring no more vendor lock-in or inability to capitalize on new ideas. 85 percent of organizations that have increased the percentage of their infrastructure which is MACH in the past 12 months cite increased ability to respond to changes in the market faster, to build and implement new functionality quicker, and reduced costs. They're also most likely to say their infrastructure is keeping up with customer demands and that they're ahead of the competition.

"This is one of many reasons why the MACH Alliance exists, to ensure cooperation among vendors and systems integrators, and to show the value over time of a MACH approach. The pace of transformation is relentless but the cost of not innovating is much higher than investing in a modern and future-proof technology stack. While transitioning to MACH is no small undertaking, continuing the status quo is an ongoing, big responsibility which is holding companies back." Casper Rasmussen, President MACH Alliance.

The success and failure of most businesses depends on their digital commerce strategy and ability to effectively drive it. In today's rapidly changing business landscape, companies are facing increasing pressure to keep up with the evolving demands of customers, employees and other stakeholders, especially in times of economic uncertainty.

Legacy monolithic platforms have historically played a vital role in running digital commerce operations. But they now have become a hindrance by stifling agility, innovation and incurring forever-increasing overheads. Modern, composable digital commerce solutions, powered by MACH-architectures have therefore emerged to replace and resolve the numerous headaches caused by technologies derived during the 1990s.

The need for resilience and adaptability is fueled by the increasing importance of digital commerce to reach strategic objectives for achieving sustainable, profitable growth with minimal risk. Modern, composable commerce solutions are key for companies to achieve their strategic objectives as they enable key paradigms around agility, resilience, customer experience and employee retention.

However, making the decision to migrate to a modern architecture can seem daunting. But with a clear understanding of its proven benefits and comparative costs, the decision becomes very compelling and justified.

One of the key decisions organizations face is choosing between monolithic and composable solutions and the related migration path. TCO (total cost of ownership) is often used as the key metric to compare the cost of these two solutions. However, the ongoing costs associated with adapting to a changing environment are not adequately covered by the traditional way of calculating the TCO.

**The purpose of this e-Book is to help business leaders to objectively analyze and compare all the costs of both architectures while introducing a better methodology for calculating a more accurate TCO.**

This e-Book explores the importance of two additional metrics: TCC (total cost of change) and TSP (total spend productivity), and how they can provide a more comprehensive understanding of the costs related to a composable solution versus a monolithic solution in the context of today's reality.

**Total cost of ownership (TCO):** TCO is a widely used metric that measures the direct and indirect costs of acquiring and using a technology solution over its lifetime. While TCO provides a good starting point for understanding the cost of a solution, **many organizations miss the impact of the cost of change and productivity in their TCO calculations.**

**Total cost of change (TCC):** TCC measures the cost of making changes to a technology solution over its lifetime. In a monolithic solution, changes to one business application component often require changes to other components, resulting in high TCC. In contrast, composable solutions are designed to be modular, with each component being able to operate and be modified independently, resulting in significantly lower TCC.

**Total spend productivity (TSP):** TSP measures the efficiency of IT spend by evaluating how much of the budget is being spent on activities that directly contribute to the organization's strategic objectives around the top and bottom line. By reducing unproductive IT spend, organizations can allocate their budgets more effectively to meet the needs of the business.

**While TCO is an important metric, it can miss key cost-drivers that have the greatest impact on business outcomes. TCO, inclusive of TCC and TSP, provides a more comprehensive view of the cost of digital commerce solutions and operations, helping budget holders make more informed and impactful decisions about their commerce technology stack.**

# What is TCO?

TCO is a crucial measure for organizations looking to make informed decisions about their investments. By accounting for all the costs associated with a product or service, TCO aims to provide a comprehensive view of the financial impact of a decision. In today's cost-scrutinized business environment, including TCO in your decision criteria is more important than ever.

Understanding the TCO (and value) of IT for business-valued services can help decision-makers make informed decisions about cost allocation (showback or chargeback) and sourcing. This information is often used by business leaders, IT PLM, sourcing and procurement teams to understand how technology supports achievement of business goals.

When estimating TCO for technology investments, it is critical to determine the scope (the total) first, before going into the cost section.

Subsequently, when evaluating the cost it is essential to include all the costs involved in selecting, building, deploying and maintaining the solution for a specific period of time (e.g. three to five years). These costs can be broadly categorized as follows:

Upfront Cost	Implementation Cost	Run Cost (OpEx)
<ul style="list-style-type: none"><li>• Selection cost (e.g. RFP)</li><li>• Planning cost</li></ul>	<ul style="list-style-type: none"><li>• Development &amp; customization cost</li><li>• Integration &amp; migration cost</li><li>• Training cost</li></ul>	<ul style="list-style-type: none"><li>• Infrastructure, data, networking &amp; hosting costs</li><li>• License &amp; subscription cost</li><li>• Scalability &amp; performance cost</li><li>• Documentation cost</li><li>• Security cost</li><li>• Maintenance &amp; support cost</li><li>• Upgrade cost</li><li>• Retirement &amp; replacement cost</li></ul>

## Limitations of focus on cost

A limitation of a TCO analysis is that it does not consider returns. When time-to-value for two approaches are similar this might be less of an issue, but can otherwise lead to a serious misinterpretation of results and vastly different business outcomes.

TCO analysis is important as a stand alone exercise but is even more valuable when it forms part of an ROI (return on investment) calculation and overall business case. However, this e-Book is primarily focused on TCO.

# Standard measures of TCO for composable commerce

The shift from transactional eCommerce to customer-experience-driven digital commerce has led to a significant change in the way businesses operate. With digital commerce, businesses are able to leverage new technologies and platforms to create more personalized and engaging experiences for their customers.

One of the key developments that is enabling this shift to customer-centric commerce is the move from monolithic solutions to composable solutions, based on Microservices, API- first, Cloud-native and Headless principles, known as a MACH architecture. Businesses can leverage MACH-based technologies to create more flexible and scalable solutions. This allows businesses to quickly respond to changing customer and market needs, to innovate more easily at a lower cost of operations.

As companies move towards composable commerce, with new features and innovative solutions being added or removed on the fly, evaluating the true costs of implementing and maintaining these systems is sometimes considered to be complex.

In this section, we will delve into the details of each of the three main factors of TCO for composable and compare these with monolithic solutions, including what scope to consider, which costs to include and the implications of shifting ownership models.

## **PART 1**

### Total (Scope)

Traditionally, the scope of eCommerce was relatively easy to determine and often limited to the functional scope of the selected platform itself. With monolithic systems, customer journeys and engagement have proven very difficult to orchestrate and optimize. Marketing, commerce and customer services are often being compelled to use different, non-integrated systems resulting in siloed business functions and disjointed and frustrating customer experiences.

Today, leading companies organize themselves in a customer-centric model around customer journeys and are continuously innovating and adapting their offer to meet real-time customer needs. Companies can do this because modern technologies are no longer a single, one-size-fits-all platform — but composed of a set of microservices that are readily interoperable by APIs.

When comparing scope between a legacy and modern approach, it is therefore key to determine what functional domains are needed (e.g. search, content, checkout). It is also paramount to distinguish platforms that are MACH-based versus those that are not, since both architectures have substantially different cost structures.

## A lean & efficient total scope

In a composable approach you'll only pay for the functionalities needed. Most companies do not fully utilize all of the functionality in their monolithic eCommerce platforms for many reasons. Some of the functionality is simply not needed, too complex or there is already a preference for another tool. Very often, they still find it necessary to extend their systems with best-of-breed point solutions to meet their specific needs and gain more flexibility. Yet companies are still paying for the cost for the entire monolithic platform. They are unnecessarily paying double for overlapping functionality resulting in a higher than necessary TCO.

## PART 2

## TCO part 2: Cost

When estimating TCO for commerce solutions, companies typically (and appropriately) evaluate upfront, implementation and run(ning) costs. The following explains each of these primary factors:

*Gartner predicts that by 2024, the IT costs of managing software operations will be halved as a result of the adoption of composable application architectures.*

### Upfront cost

Upfront cost refers to the costs incurred during the selection and planning of the solution. In a composable approach, these costs are lower than in a monolithic approach:

#### 1. Selection costs: lower

The first move toward a composable architecture is often done via a relatively expensive RFP (request for proposal) process and POC (proof of concept). Many companies have taken the opportunity to modernize their selection process (e.g. by running hackathons) and shorten time (and effort/cost) spent.

The RFP process for monolithic systems is complex and takes months because it has to cover every aspect of a digital commerce system, whereas an RFP for a modular composable system can focus only on explicit and most critical functionalities. A POC used to be complex and time-consuming with a one-size-fits-all platform because of all the rigid interdependencies of each function. On the other hand, a POC with a modular, API architecture is simple. Modern solutions have free trial periods that are easily accessible via the vendor's website.

#### 2. Planning costs: lower

While a composable approach does require a more sophisticated architecture and MVP (minimum viable product) design phase, the solution can be incrementally built and deployed using the "strangler" pattern. This approach allows for the replacement of a monolithic system with a composable architecture function-by-function over time, while still running and maintaining the functionality of the existing system, resulting in lower risk and less end-to-end planning and coordination upfront.

## Implementation cost

The implementation cost of composable commerce compared to monolithic systems can vary, as it depends on several factors such as the size and complexity of the project, the resources, sourcing and expertise of the development team, and the specific technology stack and tools used.

### 1. Development & customization cost: lower, but depends on approach

In general, these costs for the complete scope of a modern composable commerce system will be lower, since customization of components can be done relatively easily by extending flexible APIs. Furthermore, you can launch with incremental improvements or a MVP to more quickly realize your first returns (which further offset the initial costs). In a monolithic solution, any effort spent on customization will be lost when you move to a new platform. In a truly MACH-based composable approach however, custom-built components can be leveraged even after migrating away from your commerce vendor.

### 2. Integration & migration costs: lower, but depends on approach

While it might seem that integration and migration costs would be higher in a composable setting since it requires “internal” integration of these components into a cohesive solution, this is typically not the reality. APIs play a significant role in the MACH-based composable stack. APIs are the primary means of communication between different components, which allows for decoupling and an easier, lower cost of integration. Validate with your platform vendor to what level the solution is decoupled to truly understand the related integration cost. Vendor marketplaces also contribute to the lower integration and migration cost in a composable commerce solution. Solutions from different vendors from [the MACH alliance](#) can be easily combined and integrated.

### 3. Training costs: lower

The training costs for business users and developers working on a composable commerce solution are generally lower as they only need to understand the APIs and the specific components they are working on. This allows them to focus their efforts and resources on a smaller subset of the system, making it easier for them to get up to speed and start delivering value more quickly. Training costs for developers working on a monolithic commerce solution may be higher as they will need to have a broader understanding of the entire system, including all its components and how they interact with each other. Business user training costs (e.g. on order management or merchandising) are similar for both approaches.

## Run cost

Run(ning) costs refer to the costs incurred after the initial deployment of the solution. In a composable approach, run costs are lower for the following reasons:

### 1. Infrastructure, data, networking & hosting cost: lower

In general, cloud-based services will involve a lower cost for both monolithic and composable approaches compared to an on-premise setup. However, since resources in a MACH-based approach can be scaled on a service-level (meaning you only pay for the capacity you need), it's likely that the related costs are lower. Any cost related to ongoing backend systems and digital channels integration is lower in a MACH architecture because of the API enabled connectivity.



**2. License & subscription cost: lower**

Costs are lower due to the ability to use open-source components and pay only for specific services or features. Also, many MACH-based ISVs (independent software vendors) charge on a cost-per-order transaction model or a fixed license fee rather than a percentage of GMV (gross merchandise value) transacted model — which can further reduce cost.

**3. Scalability & performance cost: lower**

Smaller, independent services can be more easily and individually scaled and maintained to ensure the right level of performance (e.g. during peak moments).

**4. Documentation cost: lower**

Leading vendors in the composable set come with extensive (API) documentation. In a composable solution there are less (technical) complex dependencies that need to be managed and documented compared to a monolithic setup.

**5. Security cost: similar**

While security requires a different approach for a service-based architecture, there is no evidence that this either decreases or increases the overall cost structure.

**6. Maintenance & support cost: lower**

The ability to use specialized teams for individual components reduces the need for a dedicated platform team. L1 resources however might be slightly more expensive as they need to have knowledge on a number of different applications for triage.

**7. Upgrade cost: lower**

In a modern composable approach there are no disruptive upgrades or required support package installations, reducing the effort spent by enterprises. Modern software is versionless and updates are continuously deployed without downtime.

**8. Retirement & replacement costs: lower**

No requirement to replace and retire the entire platform when it's end-of-life. Those days are gone. With a composable approach you'll only retire and replace the individual components when required.

*"Cloud providers like Google Cloud enable you to run e-commerce cost optimally by scaling and paying only for the resources you consume, and further lower the cost with flexible pricing plans and usage-based automatic savings. In addition, free cost management tools help you control spending within your budgets, while AI-powered intelligence allows you to forecast and manage costs."*

- Logan Vadivelu, Global Lead, Retail Industry Solutions, Google Cloud

*"Ask yourself if you are able to scale e-commerce functions independently to meet traffic spikes for peak season sales without being in the business of managing infrastructure. You also need to enable consumers to shop anywhere, anytime and on any device. We help customers by partnering with commercetools who run their e-commerce backend on Google Cloud to reduce infrastructure costs and reinvest savings in continuous innovation of features and capabilities."*

**PART 3**

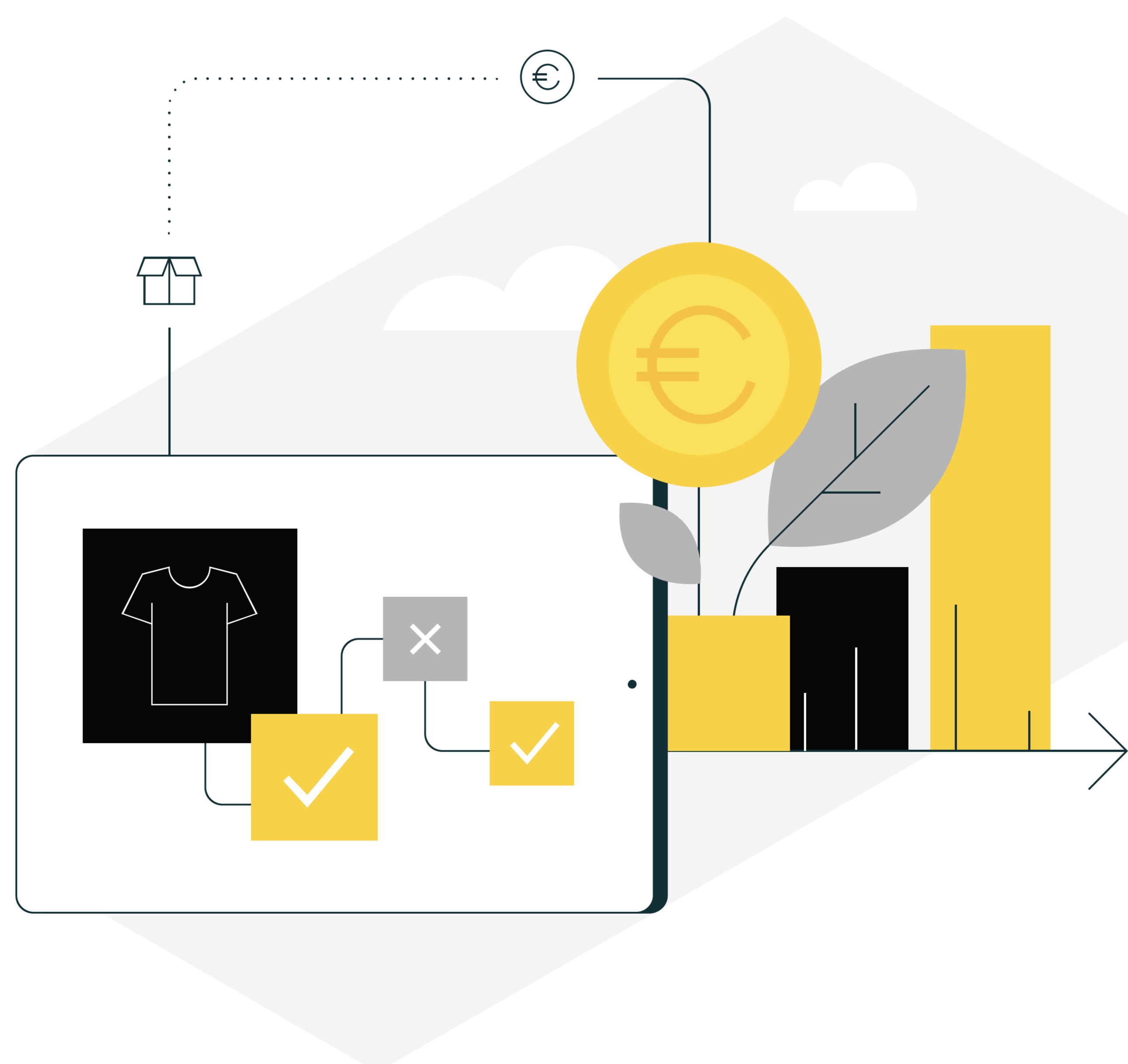
## TCO part 3: Ownership

Composable commerce solutions often require a shift in technology ownership models (both on software and hardware), which can further complicate TCO analysis. As software becomes more modular and delivered in the cloud, the lines between ownership and usage become blurred. This can make it difficult to accurately assign costs and ownership responsibilities, which in turn can make it challenging to effectively analyze, manage and optimize the TCO of the solution. It is also a strategic decision on how to allocate costs for shared resources (such as security monitoring platforms or analytics capabilities), and who should take ownership of these costs.

A composable approach allows for more granular and precise cost tracking (powered by the cloud), enabling organizations to better understand and optimize the costs of specific business services, as well as the underlying technology services that support them.

### How to compare apples with apples

When aiming to compare apples with apples, for every investment decision and TCO analysis, the total scope, all related costs and any expected incremental value should be included in the equation. However, a composable solution has cost structure and operational efficiency advantages that are very distinct from a monolithic solution — and the same holds for multi-tenant cloud solutions that are not based on MACH principles. An accurate comparison requires the inclusion of metrics that capture the cost of change and the productivity of your spending.



# Technical debt: The hidden costs of traditional platforms

## **An outdated model in a new era**

There is a common misconception that composable commerce comes with a hefty price tag. When you look at the bigger, long-term picture, it becomes clear that traditional systems are inflexible, becoming outdated relatively rapidly and you have to spend much more just to keep the lights on and eventually have to replace the technology.

This is especially true when we compare monolithic platforms to composable solutions. Let's bring in a little context from the dot.com bubble era of the 1990s. This is when the first generation of commerce platforms like SAP, Oracle, Shopify, Magento and Salesforce were born. These massive software companies provided organizations with the commerce technology to leverage desktop-based commerce via all-in-one, one-size-fits-all platforms, known as monoliths.

Fast forward 30 years and digital commerce has changed dramatically. Consumers are shopping on mobile devices and over 15 other touchpoints, from social apps to in-car dashboards. Rigid, slow-to-update monolithic platforms can't keep up with modern-day demands, having a direct impact on the success of the business while incurring forever increasing and unpredictable costs.

## **Technical debt in commerce**

As modern technology becomes a core driver of value for businesses in a digital world, the hidden peril of technical debt related to monolith solutions is emerging as a significant challenge.

Technical debt is defined as the off-balance-sheet accumulation of all the technology work a company needs to do, which can undermine efforts to compete and innovate. In a recent [McKinsey survey](#), CIOs reported that 10 to 20 percent of the technology budget dedicated to developing and launching new products is diverted to resolving issues related to technical debt. In the same survey, CIOs estimated that technical debt amounts to 20 to 40 percent of the value of their entire technology estate before depreciation, translating to hundreds of millions of dollars of debt for larger organizations.

Technical debt related to digital commerce operations can lead to costly complications, such as prohibitively expensive integration of new digital services and capabilities, unexpected challenges that cause projects to run over budget and miss deadlines, and an unmotivated IT workforce that is focused on managing complexity rather than driving innovation. Perhaps most importantly, technical debt leads to an inability to quickly adapt to opportunities or changes in the market. Over time, more resources are required to simply maintain existing systems, without driving additional value to the business.

In simple terms, technical debt is the incremental cost and loss of agility to your company as a result of legacy technologies becoming unfit for purpose. This debt further increases when corners are cut to save time or money when implementing new systems or maintaining existing ones. It occurs when systems aren't integrated correctly or code is overly complex. This is due to a variety of reasons, such as inefficiencies, time to market considerations or running outdated versions of software, among many other issues.

Similar to financial debt, technical debt is an unavoidable cost of doing business. Although it doesn't show up directly on the balance sheet (nor is it formally controlled), it must be managed effectively to ensure an organization's long-term viability. By actively managing technical debt in digital commerce, companies can free up resources and invest more time on work that supports business goals.

## Business impact of technical debt

While technical debt may not be immediately visible, it can lead to unproductive spending and negatively impact the efficiency of the development team. As said before, when technical debt accumulates, it becomes more difficult and expensive to make changes to a product or service, reducing the agility of the organization and limiting its ability to respond to changing customer needs and market conditions.

When organizations are selecting technologies, they should account for how effective those technologies are at remediating and avoiding technical debt, as this vital selection criterion is often not thought to be sufficiently through.

### **Inflexibility limits ability to respond quickly: impact on revenue streams**

Monolithic applications are created as a single, indivisible unit. This means even minor code changes require refactoring the entire software stack, leading to downtime and lost business. Also, it ties developers to bug fixing and maintenance tasks instead of doing their actual jobs: Developing and deploying game-changing features.

### **Limited scalability: impact on customer experience & cost**

Monolithic systems are often not designed to be auto-scaled, which can make it difficult for organizations to handle increased traffic, demand or move into new business models. This can lead to decreased performance and customer experience, as well as increased costs associated when mitigating this impact.

### **Maintenance & upgrades: *limits ability to innovate***

Traditional systems require significant ongoing maintenance and (forced) upgrades, which are costly, time-consuming and risky. This diverts (productivity of) resources away from strategic initiatives and limits the ability to innovate.

"As customer demand and market dynamics evolve so rapidly today, your ability to shift and change not just quickly, but also cost-effectively matters more than ever. A MACH-based approach can help companies of all sizes innovate and scale in less time and using fewer resources by decoupling backend and frontend."

- Logan Vadivelu, Global Lead, Retail Industry Solutions, Google Cloud



# Modern digital commerce: Additional TCO evaluation metric

As discussed before, TCO is often used as a primary metric to understand the direct and indirect costs and overheads of digital commerce systems. However, TCO analyses can often miss the true cost of purchasing and operating technology. To fully understand the cost of a composable versus a monolithic solution, it is important to consider two additional metrics: TCC and TSP.

## TCC: Total cost of change

The functionality and services that your business requires today will be different from the ones your business requires tomorrow. Therefore, change costs are a critical component and refer to the costs incurred when making ongoing changes to the solution — for example, for new feature development, customizations or adding integrations.

In a modern composable architecture, companies have the freedom to select and integrate best-of-breed or best-for-need components in a flexible manner. This freedom allows companies to invest in products that are the best fit meeting the immediate needs of the business in the most efficient way. Spending is made much more effective by leveraging only what you need and getting a more immediate return by targeting spending on what is best for your business right now. Additionally, a MACH-based architecture significantly increases the flexibility and agility of the underlying technology.

TCC measures the cost of making changes to a technology solution over its lifetime. This includes costs such as development, testing, deployment and support for new features and updates. In a monolithic solution, changes to one component often require changes to other components along with extensive QA (quality assurance) resulting in high TCC. Let's examine how TCC is reduced in composable solutions:

### **1. Development & customization costs**

In a MACH-based composable stack, each component is designed to perform a specific task, making it easier to isolate and modify individual components without affecting the entire system. Combined with the reusability of (micro)components or leveraging of low/no code systems, this makes it simpler and quicker to realize new developments, significantly reducing the cost of change.

### **2. Integration & migration costs**

As described in the TCO section, the nature of an architecture with decoupled microservices and API-first results in significantly lower integration and migration costs. Think about the amount of integrations with new platforms or services you'll need over the coming years — for every integration this benefit will be there.

### 3. Overhead costs

Nowadays, it is good practice to organize teams around the customer journey. In a monolithic solution every change needs to be coordinated across those teams. That requires meetings, alignment sessions and off sites.

In a composable stack these teams can truly operate independently, with less overhead as a result.

When evaluating TCC, estimate how often some of the following (management, optimization or expansion) scenarios will occur in your organization and leverage that to evaluate a monolithic versus a composable approach:

- Add new campaign pages
- Change frontend layouts, structures and components
- Add new sales channels (frontend, voice, AR, social, marketplace)
- Expand into a new market
- Add or change a loyalty, payment or shipping provider
- Add or change PIM, Search or CMS
- Add or change custom microservice
- Add single-click checkout
- Test a new business model (e.g. microsite, members-only access, etc.)
- Make adjustments to the customer experience.

Also, consider the impact of these changes when the proposed change is not supported natively in your existing platform.

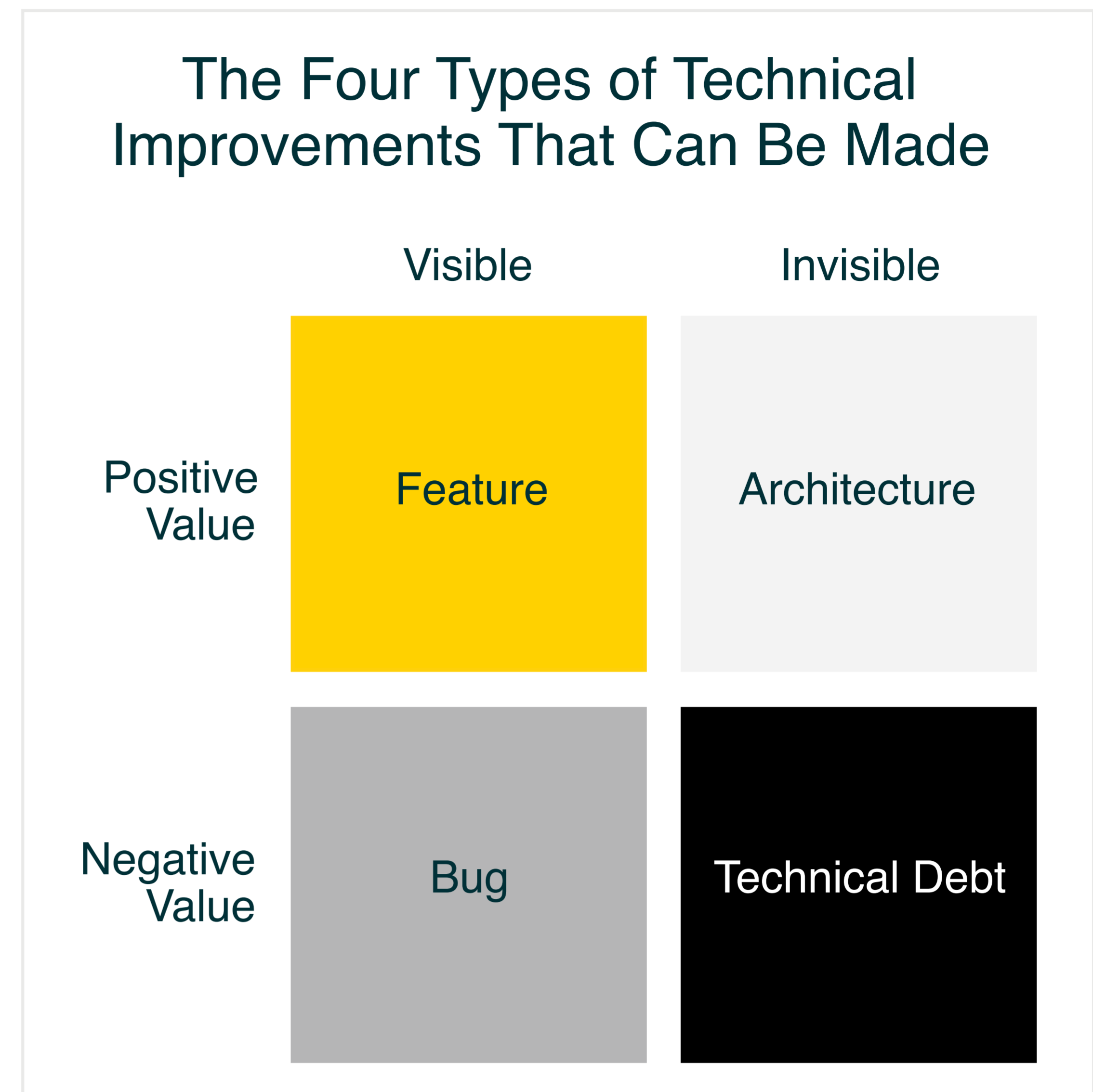
## TSP: Total Spend Productivity

Executives care about productivity and effective utilization of resources, since these drive profitability. This is not only valid for labor productivity, but also capital productivity and hence valid for IT spend.

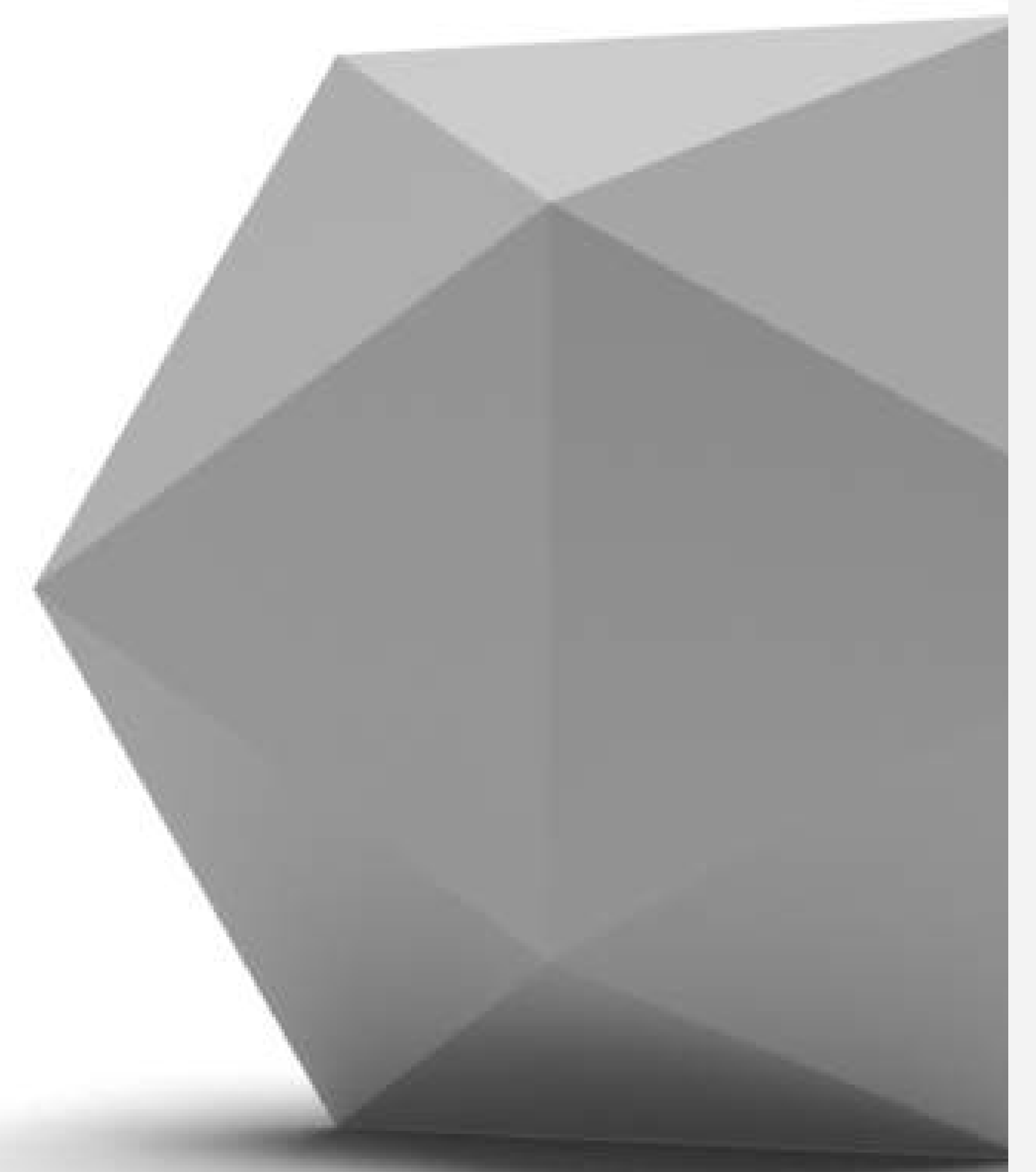
When evaluating the effectiveness of IT spend, productive spend is defined as any expense that contributes to the organization's strategic objectives to improve the top or bottom line. Unproductive spending is the opposite and can include expenses such as maintaining outdated systems, duplicate or redundant investments, and investments in technology that are not being fully utilized. Additionally, IT spend on non-core business functions, such as administrative tasks, can also be considered unproductive (especially when there is potential for automation).

# THE NEW APPROACH to TCO for digital commerce

Productivity of IT spend can vary depending on the organization's goals and priorities. For example, investments in compliance and security may not directly contribute to revenue growth, but they are important for protecting an organization's assets and maintaining regulatory compliance. Thus, it's important to evaluate IT spend in the context of the organization's overall strategy and objectives. A way to look at spend productivity is to understand how your current resources (particularly your developers) are spending their time. A developer's time can be divided into four types of effort: new features, architecture improvements, bug fixes and managing technical debt. Understanding the different types of effort and the value they create or detract from the product or service is important for making informed decisions about resource allocation and prioritization.



Source: Aequilibrium



These types of effort can be divided into two categories: positive value and negative value, and further split into visible and invisible. Here's a more detailed explanation of each type:

### **New features (visible, positive value)**

New features refer to the development of new functionality for a product or service. This type of effort can have a positive impact on the customer experience and create new value for the organization. It is also visible to the customer as it results in new or improved functionality.

### **Architecture improvements (invisible, positive value)**

Architecture improvements refer to changes made to the underlying structure and design of a product or service to improve its performance, scalability and maintainability. While these changes may not be immediately visible to the customer, they can result in a better overall product and increased efficiency for the development team.

### **Bugs (visible, negative value)**

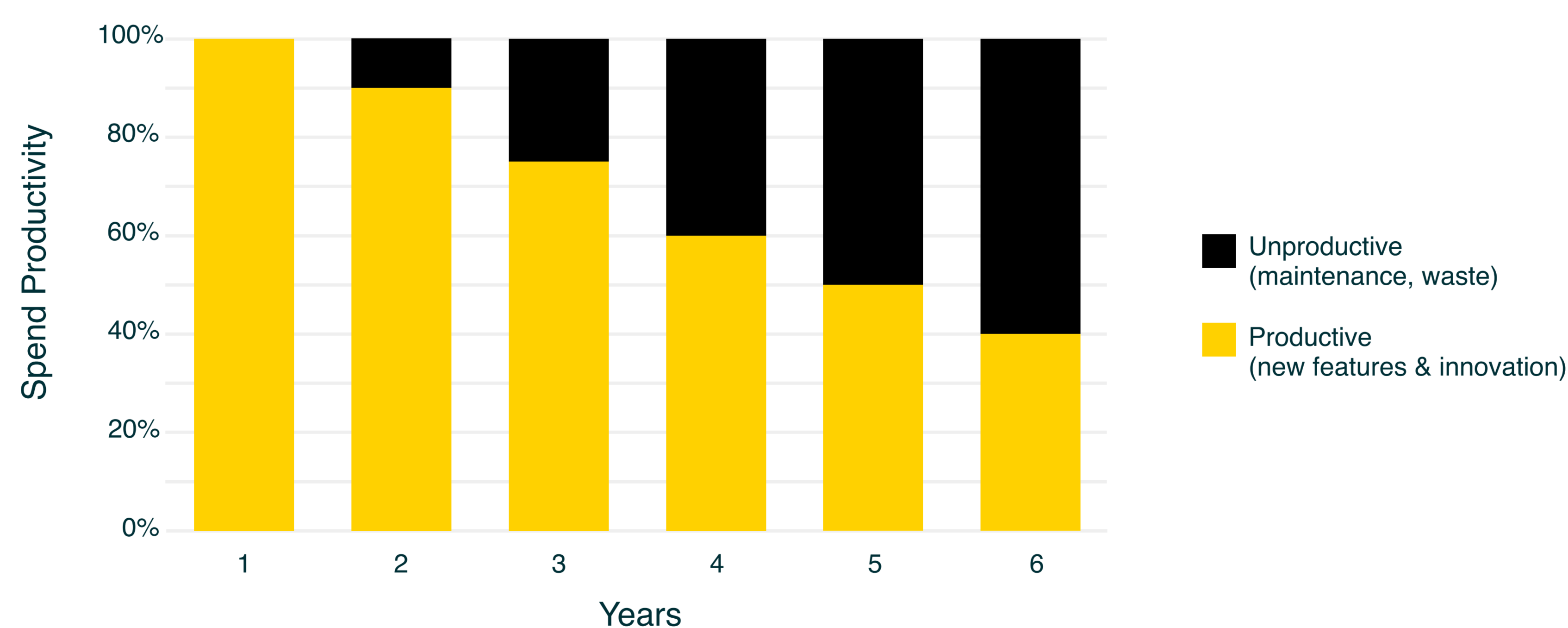
Bugs refer to issues with a product or service that negatively impact the customer experience. This type of effort is focused on fixing issues that have been identified and can be seen by the customer as a reduction in the quality of the product.

### **Technical debt (invisible, negative value)**

Technical debt refers to the cost associated with maintaining and updating the technology used by an organization. This cost can arise from the use of legacy systems, poor documentation and the accumulation of technical debt over time. Technical debt is often invisible to the customer, but it can make it more difficult and expensive to make changes to a product or service, reducing the efficiency of the development team and increasing the cost of change.

In a legacy monolithic solution, the percentage of overall spending that goes to negative- value creating, parasitic activities around technical debt, such as maintenance and upgrades, is staggeringly high and increasing over time. It is often 50% or more of the overall budget and tends to increase over time as more technical debt is accumulated. Those dollars are “wasted” on unnecessary work that are largely unnecessary in composable commerce solutions.

Monolith solutions decrease productivity over time





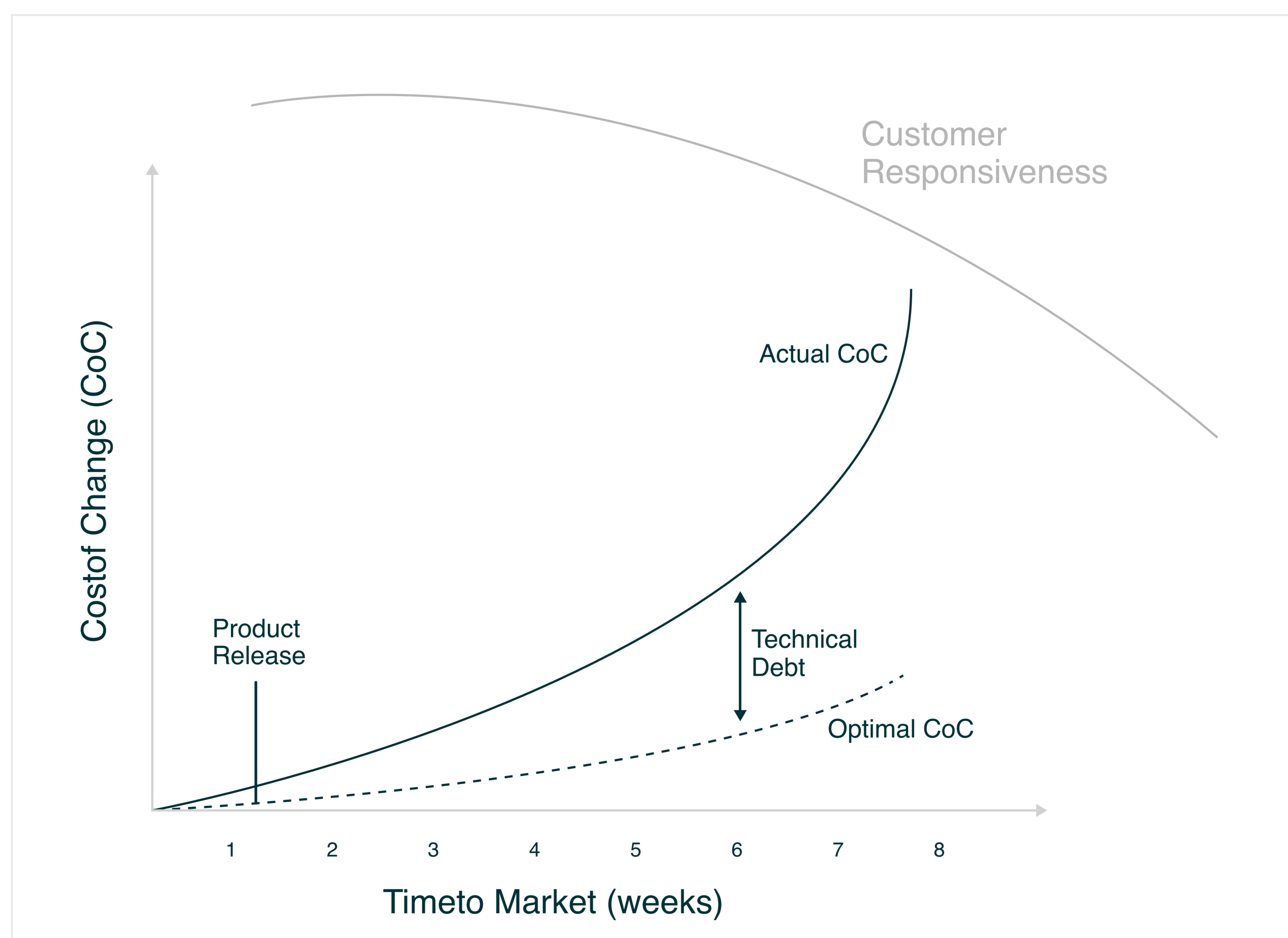
If you could reallocate your existing budget so that much more of your spend goes toward meeting the needs of the business, refining the customer experience and adapting quickly to market opportunities, how would that impact your ability to achieve your critical business objectives?

## Bringing TCC and TSP together

The previously described metrics of TCC and TSP can be brought together in the context of customer responsiveness and technical debt.

Customer responsiveness refers to the ability of an organization to respond to changing customer needs and requirements in a timely and effective manner. Organizations that are highly responsive to customer needs are able to quickly adapt to changes in the market and maintain a competitive advantage.

Technical debt derived from monolithic solutions can make it more difficult and expensive to make changes to a product or service, which increases the cost of change, decreases productivity and negatively impacts customer responsiveness. This was described by Jim Highsmith in his book Agile Project Management — see below picture.



In summary, customer responsiveness, the cost of change and spending productivity are interrelated yet can have a significant impact on the overall success of an organization. By understanding the relationship between these factors, organizations can make informed decisions about their technology investments and balance the cost of change with the need for customer responsiveness, while managing their technical debt effectively.

# The benefits of a modern approach to digital commerce

Now that we've covered TCO and additional metrics such as TCC and TSP, we must recognize that cost is not the only driver for an evaluation. Let's examine the total benefits of a modern approach to digital commerce, which collectively could also be considered your opportunity cost for staying with or adopting a monolithic solution.

## Growth enablement

In this section, we will explore the key growth-enabling benefits of this approach, including faster time to value, increased business agility and improved customer experience.

### **Faster time to value with increased business agility**

One of the most significant benefits of a modern approach to digital commerce is the ability to accelerate ROI through faster release cycles and improved agility. With a composable approach, organizations can leverage flexible technology that does not hold them back. By breaking down monolithic applications into modular, reusable building blocks, organizations can rapidly iterate and adapt to changing business needs, improving their ability to respond to new opportunities and challenges faster. This increased agility can help organizations stay ahead of the competition and capitalize on new growth opportunities by taking faster advantage of new business models, such as click-and-collect, that can drive growth and competitiveness.

### **Improved customer experience, omnichannel**

With a modern approach to digital commerce, organizations can leverage new technology and architecture to deliver fast, responsive and personalized digital experiences that drive engagement and conversion. The headless nature of composable solutions enables seamless (data) integration across touchpoints and channels. Other concrete examples would be to optimize site performance, renew the checkout process, or hyper-personalize and localize pricing and promotions based on customer analytics. This can help organizations to retain customers and grow their business over time.

### **Enable business paradigms: Dynamic pricing, promotions & inventory**

Modern commerce solutions that are based on MACH technologies enable (near) real-time data synchronization and insights for decoupled data elements such as price, promotions and inventory data. A scalable commerce engine combined with best-of-breed solutions allows organizations to dynamically optimize their pricing, promotions and inventories based on internal and external data sources. As a result, companies provide an even better hyper-personalized customer experience and may expect an increase in revenue, profitability and working capital.

## Cost optimization

Analysts like [Gartner](#) agree that IT costs of managing SaaS (software as a service) operations are expected to be halved as a result of the adoption of composable application architectures by 2024.

In a modern approach, businesses can leverage the following cost optimization levers:

### **As-a-service model**

The cloud enabled a shift from traditional CapEx models to OpEx models, also known as "as-a-service" ownership models. This shift allows businesses to pay for the use of a service or solution on a subscription basis, rather than having to make a large upfront investment in the form of purchasing or licensing the technology. This can have a significant impact on a company's cash flow and budgeting, as well as providing more flexibility in terms of scaling up or down based on business needs.

### **Cost-per-order model**

In the traditional GMV (gross merchandise value) transacted model, a business is charged a percentage of the total revenue generated from its digital commerce activities. The GMV model does not accurately reflect the value provided by the technology provider, which is to provide a selection of the enabling capabilities to drive value.

A GMV model will significantly increase costs for businesses as they grow. As the volume of sales increases, this leads to increased costs for the business — but not per se increased profitability. Software vendors might profit from other initiatives that have been growing GMV — such as additional marketing spend — and any investments done to increase AOV (average order value) will always generate higher license costs. Composable solutions typically charge a fixed fee per order, which decreases as order volumes increase.

### **Increased developer productivity**

Companies are always looking for new ways to increase their productivity. Traditionally, there was a lot of (manual) activity involved with developing software. For example, regression testing was required for the entire solution. Modern, composable commerce technology gives the ability to independently develop, test and deploy different services and can significantly increase productivity in several ways:

1. **Automation:** The modular and cloud-native nature of composable commerce components allows for automation of development activities (leveraging DevOps best practices), reducing the time and effort required for manual activities such as testing and deployment.
2. **Modern integration patterns:** Modern integration patterns, such as APIs and microservices, enable composable commerce components to easily integrate with internal and external solutions, again resulting in increased developer productivity.
3. **Decreased dependencies:** By breaking down monolithic systems into smaller, modular components, composable commerce can decrease the dependencies between components, allowing developers to work on isolated pieces of functionality without having to worry about affecting other parts of the system.

4. **Increased reusability:** Composable commerce components can be reused across different solutions, projects and initiatives across business models (B2B and B2C), reducing the amount of time and effort required for new developments, but also related license costs in case you require separate solutions.
5. **Utilization of business resources:** Traditionally a lot of effort gets spent on aligning between business and IT, especially around campaigns. Modern frontend-as-a-service solutions often have no-code capabilities to give power to the business users to “drag- and-drop” their own landing pages.

### **Decreased labor cost**

When it comes to labor costs, modern composable commerce solutions that are based on industry standards are likely to provide a cost advantage. Developers can often code using a few common SDKs (e.g. for Javascript, PHP, DotNET), which reduces the need for specialized resources. Obviously, the degree of cost advantage is dependent on sourcing strategy and (local) availability of resources.

### **Reduction of costs associated with legacy platforms**

When moving into a new solution, companies can reduce the costs associated with maintaining legacy systems that are difficult to upgrade or replace (see details in previous section on technical debt).

Composable commerce based on MACH principles allows businesses to avoid creating new technical debt by providing future-proof ways to build and maintain their technology infrastructure: versionless and without forced upgrades. This would avoid related backwards compatibility or upgrade cost.

Obviously, when shifting to a modern solution the current solution can be decommissioned over time and existing license costs and hosting costs will no longer occur.

## Risk Reduction

Composable commerce reduces risk in several ways, particularly in terms of business continuity and attrition.

### **Business continuity**

In traditional commerce systems, a failure in one part of the system can bring down the entire system, causing costly downtime and lost sales.

With composable commerce, different parts of the system are decoupled, meaning that failure in one component does not necessarily impact the entire system. Furthermore, composable commerce can reduce the risk of security vulnerabilities by partially shifting these responsibilities towards the as-a-service provider, but for the in-house maintained components making it easier to patch and upgrade these in isolation. This helps to prevent security breaches and maintain the privacy and security of customer data. Finally, a composable approach lowers your dependency on a single software vendor, offering the ability to swap vendors as needed and control their own roadmap.

### **Employee attrition**

Composable commerce solutions can also have a positive impact on employee attrition. As employees work with new technology and are able to respond quickly to business demands, they are likely to experience increased job satisfaction and a sense of being challenged and engaged in their work. This will lead to lower employee turnover, decrease recruitment and training costs, and prevent the inability to execute on strategic priorities due to lack of resources.

It might require a cultural shift in your organization, but once employees are able to work with new and modern technology, they will be better equipped to handle new and emerging business challenges, which in return will enable profitable growth.



# Next steps

## **Start the value-driven shift towards composable commerce**

The shift to composable commerce is a journey that can be commenced with the following approach:

### **1. Discover current pain points, technical debt & spend levels — and ambitions**

The first step is to understand the current state of the organization and the challenges it is facing in the rapidly evolving eCommerce landscape. This will involve an assessment of the current pain points and performance gaps, level of technical debt and related impact, TCO and spend levels, ambitions of the organization, and an analysis of the opportunities and risks associated with the move to composable commerce.

### **2. Envision required capabilities, architecture principles and target state**

Once the current state has been assessed, the next step is to envision the target state and the capabilities required to achieve it, as well as the architecture principles underpinning the transformation. With composable commerce, you can incrementally transition towards your target state.

### **3. Select your core technology — without extensive RFP**

Run a proof-of-value to demonstrate technical feasibility and financial viability for a critical business service. Understand whether a vendor offers a composable solution based on MACH principles or more traditional architecture patterns. Evaluate the potential for avoiding technical debt. Modern composable commerce technologies allow you to get started with a free trial (see below). Leverage industry analyst reports such as [Gartner's Magic Quadrant](#) for a pre-selection and start building!

### **4. Build your business case**

Use this e-Book to help you assess the ongoing TCO to be incorporated in your business case. Create and tell a compelling story on the value of composable commerce for your organization. Spell out the business needs and align the case with your strategic goals. Build the right team to test and shape your ideas. Analyze risk and opportunities and present the case to your stakeholders. At commercetools, our Customer Value Services team can assist you with this effort.

### **5. Start small and demonstrate value to the business with incremental innovation**

This will involve identifying a new, decoupled capability or MVP that will enable the organization to demonstrate the value of composable commerce to the business and get buy-in from key stakeholders. This will be a critical step in fueling the growth of the composable commerce initiative and securing the support and investment needed to scale it over time.

# About

## **About The MACH Alliance**

The MACH Alliance is a [501(c)(6)] non-profit organization, governed by an independent board and does not endorse specific vendors, members or otherwise. The Alliance has become the voice of the industry helping enterprise organizations navigate the complex modern technology landscape. It aims to guide and show the business advantage of open tech ecosystems that are Microservices based, API-first, Cloud-native SaaS and Headless. All MACH Alliance members meet strict certification principles that are published on the website. The MACH Alliance welcomes technology companies and individual industry experts who share the same vision for the future. Learn more about the Alliance and MACH certification at [machalliance.org](https://machalliance.org), and follow us on Twitter and LinkedIn.

## **About commercetools**

The inventor of headless commerce, commercetools is an innovative technology disruptor that has established itself as an industry-leading eCommerce software provider. Today, some of the world's most iconic brands and growth-focused businesses trust commercetools' powerful, flexible, scalable solutions to support their ever-evolving digital commerce needs. As the visionaries leading the modern MACH (Microservices-based, API-first, Cloud-native and Headless) architecture movement, commercetools provides customers with the agility to innovate and iterate on the fly, merge on and off-line channels, drive higher revenue, and future proof their eCommerce business.

Based in Munich, Germany, with offices in Europe, Asia and the United States, commercetools is singularly focused on leading a future of limitless commerce possibilities.

More information at [commercetools.com](https://commercetools.com).

# Appendix: Cost comparison of traditional & modern commerce solutions

cost element	Cost breakdown	Monolithic (existing) on-premise or cloud-hosted)	Monolithic (new) (as-a-Service, cloud-hosted)	Composable Monolithic Core (as-a Service, cloud-hosted)	Composable MACH Core (as-a Service, cloud-native)	Lowest cost
TCO Upfront cost	TCO Upfront costSelection	N/A	€€€€	€€€	€	Composable powered by MACH
	Planning	N/A	€€€	€€€	€€	Composable powered by MACH
TCO Upfront cost	Development & customization	N/A	€€ to €€€€	€€ to €€€€	€€ to €€€€	Similar, depends on Maturity & Approach
	Integration & migration	N/A	€€ to €€€€	€€ to €€€€	€€ to €€€€	
	Overhead	N/A	€€	€€	€	Composable powered by MACH
	Training	N/A	€€	€€	€	Composable powered by MACH
TCO Run cost	Infrastructure, data, networking & hosting (application)	€€€€	included	Included	Included	As a Service
	Infrastructure, data, networking & hosting (integrations)	€€€€	€€€	€€	€	Composable powered by MACH
	License & subscription	€€€€	€€€€	€€€€	€€	Composable powered by MACH
	Scalability & performance	€€€€	€€	€€	€	Composable powered by MACH
	Documentation	€€€€	€€	€€	€	Composable powered by MACH
	Security	€ to €€€	€€	€€ to €€€€	€€	As a Service
	Maintenance & support	€€€€	€€	€€	€	Composable powered by MACH
	Upgrades	€€€€ or N/A	€€€	€€	Included	Composable powered by MACH
	Retirement & replacement	€€€€	€€€	€€€	€	Composable powered by MACH
TCC Change cost	Development & customization	€€€€	€€€	€€€	€€	Composable powered by MACH
	Integration & migration	€€€€	€€€	€€	€	Composable powered by MACH
	Overhead	€€€€	€€€	€€	€	Composable powered by MACH
Hidden costs	Scalability	Limited	Limited	Partially Limited	Unlimited	Composable powered by MACH
	Ability to respond quickly	Limited	Limited	Partially Limited	Unlimited	Composable powered by MACH
	Technical debt cost	€€€€	€€€	€€	N/A	Composable powered by MACH