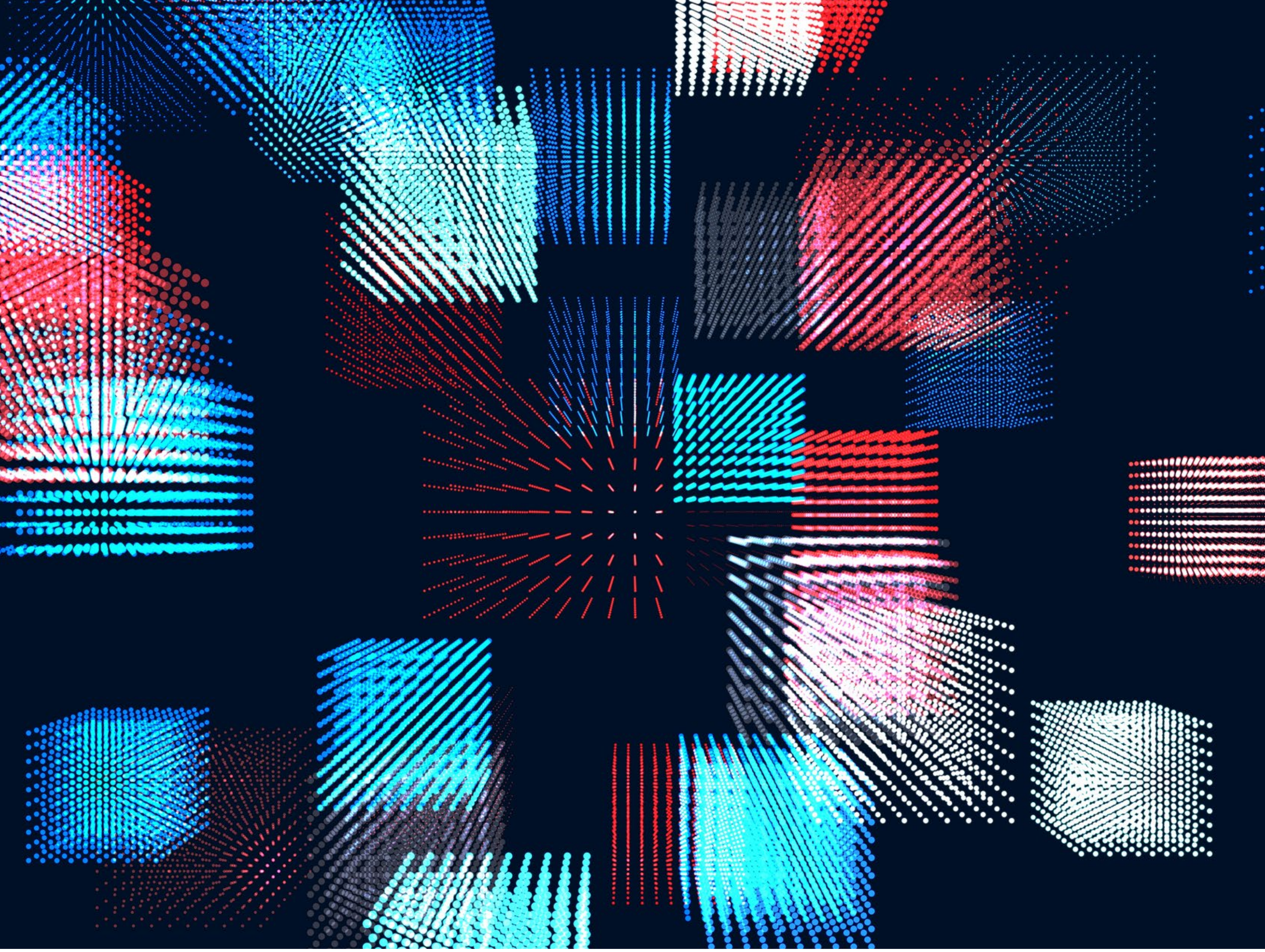


White Paper

Unlocking the secrets to managing technical debt

A guide to understanding and strategizing for your organization

We **power**
the **possible**™

Introduction

By the end of 2023, International Data Corporation (IDC) predicts that the global economy will finally reach “digital supremacy” with more than half of all gross domestic product (GDP) worldwide driven by products and services from digitally transformed enterprises. It is predicted that in 2023 more than 500 million applications and services will be developed and deployed using cloud — this is the same number of applications developed in the last 40 years combined. Most of these applications will be for industry-specific digital transformations.[1]

In the age of technology saturation and software prevalence, digital transformation has created unavoidable ripple effects across enterprises. Often, the impact is positive, ushering in improved efficiency, productivity, and transparency in business operations. But there are unintended consequences of digital transformation when the pace accelerates.

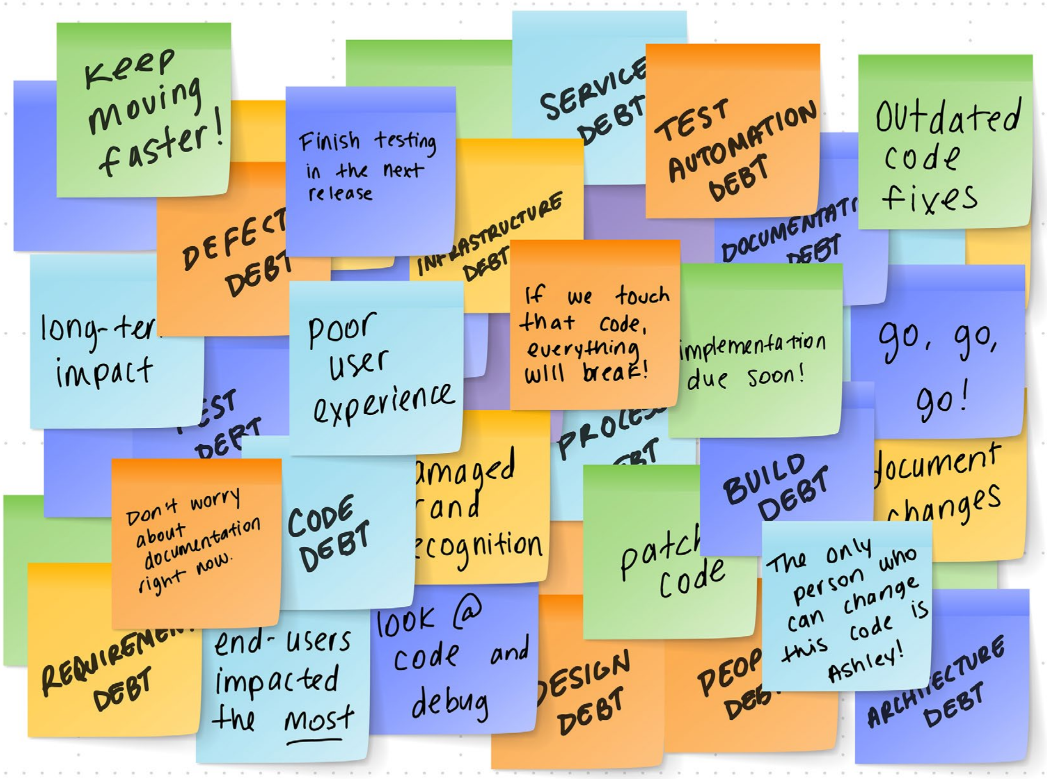
Agile development has pushed businesses to move faster, but there are other factors at play. A global 2021 study revealed 78% of organizations have accrued more technical debt in the past year than in previous years. Yet more than half (58%) lacked an official strategy to tackle this debt.[2] The COVID-19 pandemic created an increased urgency for digital transformation, as businesses struggled to react to changing demands and adapt to remote working conditions. And, with the implementation of these new tools and solutions came a rapid accrual of technical debt.

78%

of organizations have accrued more technical debt in the past year than in previous years

58%

of organizations lacked an official strategy to tackle this debt



A surface covered in multiple sticky notes with no organization or pattern. Notes include items or tasks to remember while working through a digital process. Represents how technical debt can be accumulated when speed is prioritized over strategic project delivery.

What is technical debt?

Technical debt is what happens after making swift development or software implementation decisions that prioritize speed over effectiveness. The phrase was originally coined by a software developer, Ward Cunningham, in 1992, but the term has evolved significantly since.[3]

Technical debt can manifest in a variety of ways across businesses of all sizes. It can look like adding new solutions to a tech stack that duplicate the functionality of existing tools, or when software code must be written and rapidly deployed as a quick fix or patch, in lieu of rolling out full-scale updates.

“Conceptually, technical debt is similar to financial debt,” says the CTO and EVP of Engineering at enterprise application provider, Skuid. “For the borrower, it is more important to purchase an item and have it in hand now than it is to save up the funds to purchase in all-cash.”[4]

Regulation changes, especially for cybersecurity, often call for new technology additions to be made which can be planned or unexpected. For instance, there are regulations being proposed that would require companies to maintain a detailed and up-to-date Software Bill of Materials (SBOM) so that they can quickly and accurately know all the different pieces of software embedded in their complex computer systems. This may require significant changes to the ways that software is developed and acquired.[5] And without a comprehensive IT asset lifecycle management process, these challenges can be exacerbated further.

Impact of technical debt

Accruing technical debt can have both long-and short-term impacts on all levels of a business. Much like financial debt, technical debt requires paying off, but this is not always as straightforward as making a monthly “installment.” In some cases, organizations are not even aware of the magnitude of the debt to pay off, or how to go about doing so. This section reviews the most common impacts of technical debt and offers insight into how each can affect a business.

Increased vulnerability

One of the most dangerous elements of technical debt is that, in many cases, organizations don’t even know how much they have — let alone where it is. Poor visibility into potential bugs, outdated code, and legacy practices leave organizations much more vulnerable to potential exploitation of these issues.

Technical debt can cause businesses to ignore the need for cybersecurity solutions. Or, even worse, look for solutions based on price versus actual need. For example, they may go with a cheaper solution that does not provide adequate security. Less expensive solutions that lack certain features might be purchased with the expectation that existing teams will be able to cover the need, often with intrusion detection and prevention type solutions.

Instead of investing in a solution where the provider will be the “eyes-on-glass” monitoring of their network (such as a managed detection and response (MDR)), the organization may invest in a self-monitoring application. Believing they have saved money, they may increase cost due to additional requirements put on IT teams.

Decreased productivity

When organizations are focused on paying off technical debt — patching, fixing broken code, documentation — they can’t move as quickly, which can hamstring an organization. Reactionary work, like the work technical debt creates, typically results in much longer completion cycles compared to forward progress. Furthermore, if the debt is legacy debt, teams will spend even more time trying to understand how to go about making corrections. A longitudinal study found developers report that they waste 23% of their working time due to technical debt.[6] With resources already stretched thin, focusing more time on fixing problems draws out completion cycles even further and makes an organization much less agile and adaptable.

23%

of developers working time wasted due to technical debt

Stunted innovation

When teams are focused primarily on identifying and fixing existing issues, it leaves little room to focus on improvements, updates, and future project phases. Constantly looking backwards instead of innovating leaves organizations less adaptable and behind in the market, which has proven to be detrimental in today’s hyper-digital environment. IDC predicts that, through 2023, half of enterprises’ hybrid workforce and business automation efforts will be delayed or will fail outright due to underinvestment in building IT infrastructure with the right tools and skills.[7]

Decreased customer satisfaction and damaged brand reputation

Unfortunately, in many cases, users feel the impact of technical debt before anyone can correct the issue. Rushing to complete a project leaves room for mistakes, hurries testing teams and deprioritizes quality — all of which have the potential to negatively impact the user experience.

Poor user experience and long repair cycles make for dissatisfied customers, and this can have a long-term impact on a brand's reputation for producing and delivering quality products and services.

Messy or broken code

Prioritizing speed over quality often results in code that is broken, difficult to understand, or poorly documented, if documented at all. When teams rush to build new applications without considering the long-term impact of the build, there's an increased risk of code imperfections. Messy, poorly documented code is much more difficult to fix and can take a significant amount of time to remedy. Future teams may struggle to read the code in the first place, making updates and fixes cumbersome and timely — and requiring more cleanup than if the code was properly written and documented up front.

Strained teams

Technical debt is often the result of an organization trying to move quickly, directly impacting the technical teams involved with the projects. For example, speeding up processes to meet a tight deadline puts strain on testing teams that must move at a faster pace than their usual turnaround. As such, they must rush or even skip certain tests to meet their deadlines, which places strain on the team to finish in time and can potentially expose the end product to additional imperfections that further accumulate debt. In one study, more than 1,000 developers across the globe broke down their time spent on maintenance issues caused by technical debt. Fifty-two percent reported spending 17 hours a week debugging and refactoring code, and 76% of those surveyed reported the negative impact paying down technical debt has on their personal morale.[8]

52%

of 1,000 developers reported spending 17 hours a week debugging and refactoring code

76%

of 1,000 developers reported a negative impact on morale due to technical debt

Financial strain

Decreased productivity and slower completion cycles have a direct financial impact on a business, seen in both hard and soft costs. Hard costs like work hours and equipment may be easier to measure, but there are additional soft costs that occur that are equally impactful — think outages, patching, and continued fixes that were not originally accounted for and will take more time and resources. Beyond that, employee time spent addressing problems and inefficiencies goes beyond technical teams and can impact other areas of the business.

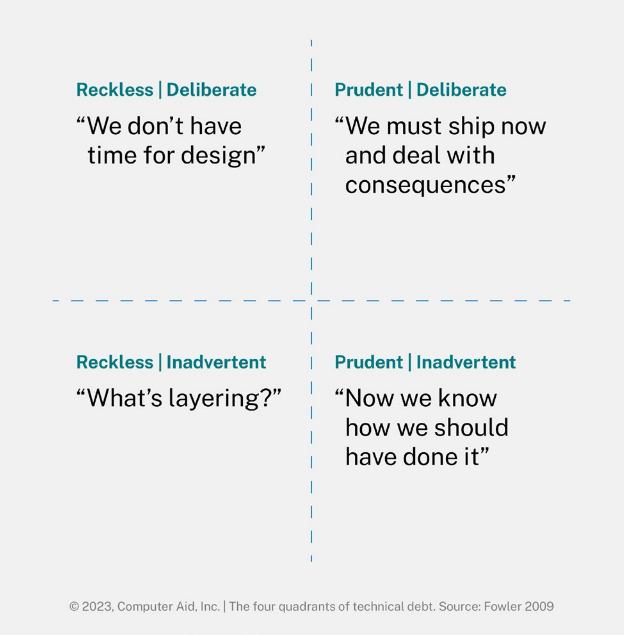
Additionally, organizations may need to seek supplemental resources to support teams trying to pay down technical debt. Hiring additional team members is difficult as-is, but it's even more difficult to find talent when the role entails reactive work instead of innovative, forward-thinking projects. Longer completion cycles, longer recruitment cycles, and increased headcount all directly impact a business's financial bottom line — all common side effects of technical debt.

Technical debt increases the total cost of ownership (TCO) of an asset. In these cases, TCO estimates the expenses associated with the procurement, deployment, development, maintenance, and usage of the asset throughout its entire lifecycle. As explored above, technical debt hinders productivity, delays turnaround times, and strains resources, all of which directly impact the TCO and, in turn, a business's bottom line.

> **"...technical debt hinders productivity, delays turnaround times, and strains resources, all of which directly impact the total cost of ownership and, in turn, a business's bottom line."**

Types of technical debt

Different classifications of technical debt have been explored and expanded since the term originally surfaced. The most widely used breakdown of technical debt came by way of Martin Fowler's Technical Debt Quadrant, considered the gold standard of measurement since 2009.



Reckless | Deliberate

"We don't have time for design"

Prudent | Deliberate

"We must ship now and deal with consequences"

Reckless | Inadvertent

"What's layering?"

Prudent | Inadvertent

"Now we know how we should have done it"

© 2023, Computer Aid, Inc. | The four quadrants of technical debt. Source: Fowler 2009

Martin Fowler's Technical Debt Quadrant square. Helpful for defining the types technical debt within an organization. Square is divided into four equal parts: (Upper left) reckless, deliberate; (upper right) prudent, deliberate; (lower right) prudent, inadvertent; (lower left) reckless, inadvertent.

The quadrant explains technical debt in four sections: deliberate and reckless, deliberate and prudent, inadvertent and reckless, and inadvertent and prudent.[9] Two of the most common types of technical debt are intentional (deliberate) and unintentional (inadvertent).

Intentional technical debt is any debt accrued that the organization knew about during development. This kind of debt is easier to reconcile, as there is a much clearer picture of where potential issues are and how to resolve them.

In contrast, unintentional technical debt occurs unbeknownst to an organization. This can happen from moving too fast in development, ignoring potential issues, or a combination of the two.

Unintentional technical debt can cause additional strain on technical teams, who must not only fix these issues, but also identify where the issues are coming from in the first place. Where intentional technical debt is often strategic and proactive, unintentional debt requires long-term reactivity and a dedicated mitigation effort.

The Software Engineering Institute published a comprehensive list of the different types of technical debt, considering the ways the debt accumulates as it relates to business and technical processes. The list is as follows:[10]

- Architecture debt
- Build debt
- Code debt
- Defect debt
- Design debt
- Documentation debt

- Infrastructure debt
- People debt
- Process debt
- Requirement debt
- Service debt
- Test automation debt
- Test debt

Considering all forms of technical debt is critical for an organization's overall strategy of paying it down. Without a thorough understanding of the debt incurred intentionally and unintentionally, as well as how it came to be, organizations will be left playing catch up and struggling to innovate.

Leading drivers of technical debt

The last few years have seen multiple factors change the way intentional and unintentional technical debt is accumulated. Taking a deeper look at the impact of rapid digitization, it's apparent the pace is not slowing down. While new technologies have a lot to offer, there can be a steep upfront cost before businesses see the return on investment. In particular, cloud infrastructure, artificial intelligence (AI), and Web3 technology often require a large investment to implement, and this can contribute to the accrual of technical debt.

Cloud

Despite the overwhelming benefits and critical need for cloud services from both employees and consumers, migrations to the cloud, as well as ongoing initiatives to ensure secure access, are just a few of the many drivers of technical debt that organizations are facing. The average technical debt for cloud management is about 23%. This means that for every $100 spent in an IT cloud budget, $23 of it is wasted on unnecessary or inefficient work. Additionally, the same study shows that companies have an average of 31% more cloud resources than they actually use.[11]

Those that are well equipped with a robust IT infrastructure can more easily pay off this debt. However, organizations such as nonprofits, government agencies, start-ups, and even enterprise organizations with smaller dedicated IT teams may struggle paying down these investments. Research indicates that through 2023, dealing with the technical debt accumulated during the pandemic will haunt 70% of CIOs. This will inevitably correlate to increased financial strain, a lag in IT agility, and forced migration to the cloud. The IDC predicts, "by 2023, an emerging cloud ecosystem will be the underlying platform for all IT and business automation initiatives and 75% of G2000 companies will commit to a hybrid work environment by intentional design, not due to circumstances."[12]

> **"…by 2023, an emerging cloud ecosystem will be the underlying platform for all IT and business automation initiatives and 75% of…companies will commit to a hybrid work environment by intentional design…"**

Transferring legacy systems to the cloud is not a simple fix. Many legacy applications must be refactored and rearchitected, both of which are indicators of a hasty digital transformation. Refactoring involves reorganizing the code while maintaining its intended functionality. The process involved addresses technical debt by making internal modifications to the code to accommodate the current version of an application, as well as any new features. This creates another application layer necessary to perform in the cloud environment, which can carry architectural limitations.

Other issues can occur when migrating to the cloud due to data flow, dependencies, security risks, and overall performance impact. In these cases, organizations choose to keep legacy applications on-premises or hybrid and choose supplemental purchases for native cloud applications in the areas they can migrate. It's important for companies to have a strategy in place for addressing and managing cloud technical debt, as well as keeping track of the debt levels.

Artificial intelligence

Organizations are also spending more on AI, which can cause further technical debt to accumulate. In a 2022 global study, 550 senior IT and data science professionals were interviewed, and more than half plan to make further investments in AI over the next 1–2 years to strengthen their security functions, basic data movement, and governance. Forty-nine percent have already deployed new technology specifically for AI and machine learning, as well as recruitment to close the skills gap. And financial investments in these areas are showing no signs of slowing down.[13]

When organizations do not properly plan for these expensive investments by hiring the right skillsets, maintaining their systems integrity, and owning quality data, they often accumulate higher technical debt than anticipated.

50%

of 550 senior IT and data science professionals plan to make further investments in AI

49%

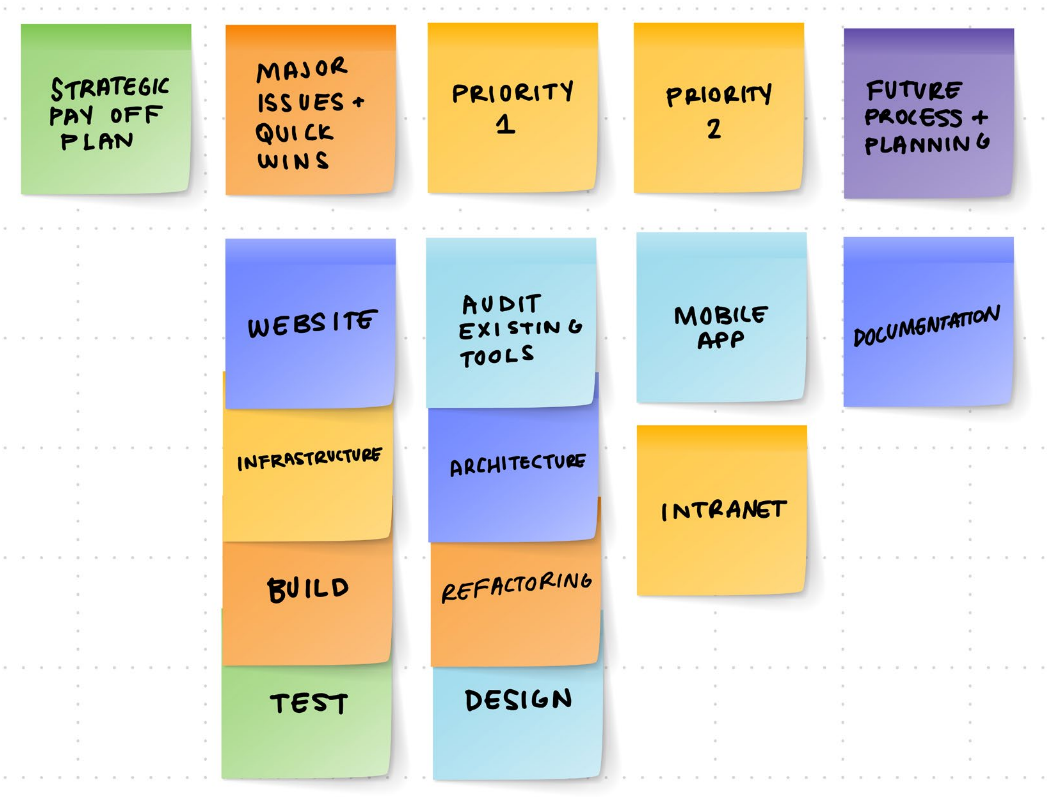
of 550 senior IT and data science professionals have already deployed new technology specifically for AI

Web3

Web3 technology is also on the rise, and its impacts are still being observed. The Web3 revolution enables new forms of decentralized economic systems and digital marketplaces, creating new business opportunities and changing the way we think about ownership, trust, and value.

Many organizations have already implemented various blockchain integrations, with future investments expected to increase. Because this emerging technology is in its early stages, these ecosystems are changing faster than organizations and developers have chances to react and maintain their decentralized applications and systems. Though decentralized systems are currently not regulated, they are rapidly becoming an elevated risk for unanticipated technical debt. The reason? Regulation is looming on the horizon that will radically change the ecosystem.

With a lack of standardization, it can be difficult for companies to choose the right technology to ensure scalability. As it is an area new to many, this constant tinkering will inevitably accumulate technical debt. Having a roadmap or action plan to address the potential debt, and how it's accrued, can be the difference between companies losing money and opportunities to innovate versus strategically navigating the debt into the future.



A surface covered in sticky notes that is organized into 5 columns and 5 rows. Each column is a step in the digital process. Written as column 1 header – Strategic pay off plan. Written as column 2 header – Major issues + quick wins. Written as column 3 header – Priority 1. Written as column 4 header – Priority 2. Written as column 5 header – Future process + planning.

Strategic payment of technical debt

Paying down technical debt is a large undertaking; it requires time, patience, and a firm understanding of all the teams that need to be involved to achieve success. Building a strategy for paying down the debt that factors future architecture and infrastructure plans into the equation can make the process more manageable.

While it requires constant work and monitoring, there are multiple things an organization can do to create a plan for tackling their debt head-on.

Plan building blocks:

- **Audit existing tools and systems,** including the cost of software subscriptions, solution functionality, and APIs. Evaluate the organization's IT asset lifecycle management process.
- **Identify quick wins and major issues first,** then balance a long-term vision with an iterative approach.
- **Create a roadmap and timeline** with repayment terms and deadlines.
  - Define long and short-term goals.
  - Be data-driven and understand the defined goals in the context of a specific business.
    - If internal teams lack the bandwidth for this task, hire a trusted partner.

- **Engage with technical teams** and communicate regularly. It is crucial to fully understand the extent of how tools are being used, how they interact with other solutions, and where development work or process efficiency is needed.
    - Ask tech teams what is and is not working.
    - Balance the needs of development with the needs of the business and be cognizant/realistic about service level agreements (SLAs) and turnaround times.
    - Consider both the hard and soft costs of the projects. Keep in mind that technical teams are not the only resources that will be needed and there may be unplanned additional costs from outages, increased issues, or longer turnaround times.[14]
- **Bake technical debt into the software development lifecycle (SDLC)** and account for it in future processes and projects.

Strategic accrual of technical debt

While technical debt is inevitable, there are ways to strategically accrue it. Being proactive about technical debt and the IT asset management lifecycle, rather than simply reacting to it, can be the difference between innovation and a constant state of “catch up.” There are occasions where being reactive is necessary, and unplanned technical debt can arise, but keeping this to a minimum is critical for business continuity.

The past few years have seen organizations take on unprecedented amounts of technical debt, much of which has been unexpected, unplanned, or unintentional and will require long-term action. It is important to be more intentional in considering future-state technical debt and include it in strategic conversations.

Here are a few things to consider in taking on new technical debt:

- **Take stock of existing technical debt.** With a thorough understanding of the technical debt already being addressed or requiring future work, it will be much easier to plan for additional costs.
- **Find balance** between paying off existing debt and incurring new.
- **Understand the workload** and availability of technical teams. Take the time to understand team structure, completion times, and existing workloads. If they are already stretched thin paying off existing debts and completing day-to-day tasks, this will impact the amount of debt an organization can sustainably take on.
- **Hire strategically,** include resources with legacy knowledge and new approaches, and find a balance in the work they are given. Identify gaps in technical teams. For example, if certain teams are busy on a new project, consider hiring teams with more availability to focus on planned technical debt payoffs.
- **Consider the following** with regard to the proposed debt:
    - What does the organization gain by incurring this debt?
    - What challenges will the debt cause later, and of what magnitude?
    - Why can’t the issue be addressed immediately?
    - What would resolving the debt look like? Will it impact more than the immediate/obvious features?
    - How long is this debt sustainable?
    - Is there a better way to address it in the future with advanced capabilities?

- **Make actionable plans** to pay off debts; prioritize maintenance and work it into daily operations. With planned maintenance included in day-to-day work, additional technical debt can more easily integrate into team workflows.
- **Plan for any known technical debt that will occur** as a project result, but also understand that things may arise that aren't planned for — and they'll need to be paid off too.

At a high level, it's important to include technical debt in every conversation with both technical and operational teams. This includes conversations about strategy, planning, hiring, and more — building technical debt into these conversations allows for more strategic accrual and payment. Additionally, consider technical debt outside of software development and price it into all IT work. Doing so allows for greater flexibility and visibility into potential issues.

"...it's important to include technical debt in every conversation...This includes conversations about strategy, planning, hiring, and more..."

As with paying off existing debt, whether intentional or unintentional, planning for technical debt will vary greatly based on the needs of the organization and the makeup of the teams. However, at all levels, it is important to consider the implications of new debt and make actionable plans to pay it off. With a thorough understanding of technical and operational positioning, organizations can be better prepared to take on technical debt in a sustainable manner.

Conclusions on tackling technical debt

For many organizations, technical debt is the reality of doing business. Beyond that, it can also be a practical investment in a business's future. Regular audits, such as CAI's IT HealthCheck, were designed specifically to shine a light on places where businesses have accrued intentional and unintentional debt. Other CAI services, like application rationalization, can show where redundancies exist in a tech stack. Effective project management, coupled with continuous integration and continuous deployment, can ensure more efficient development rollouts.

Ultimately, technical debt is not always a negative thing, especially if it's part of a bigger vision to achieve development or operational improvements and greater efficiency. This kind of debt can be taken on with the intention of improving processes down the road, and it can facilitate easier implementation of solutions for teams working with tight deadlines.

It's still best to minimize the potential pitfalls of technical debt by understanding the cause and effect of taking it on. The right strategy and approach can make the difference between doing business in the red — with few benefits to outweigh the strain that debt causes — or working toward a larger efficiency goal while taking carefully measured risks.

CAI has helped hundreds of businesses, organizations, and agencies in both commercial and public sector spaces navigate the digital transformation and modernization journey. From knowing where to start with evaluation and gaining a 360-degree view of the cost, risk, and quality of business applications and the roles they serve within an organization, to a holistic approach to analysis and management of resources, infrastructure, and the talent necessary to manage the impacts of technical debt, CAI is an expert at powering the possible on the path to optimization.

Take the next steps for honing your technical debt strategy by visiting CAI.io.

References

1. “IDC FutureScape Outlines the Impact ‘Digital Supremacy’ Will Have on Enterprise Transformation and the IT Industry.” Business Wire. October 29, 2019. https://www.businesswire.com/news/home/20191029005144/en/IDC-FutureScape-Outlines-the-Impact-Digital-Supremacy-Will-Have-on-Enterprise-Transformation-and-the-IT-Industry

2. “78% of organizations have accrued more technical debt during the pandemic, says new Software AG Research.” Software AG News Center. January 19, 2022. https://newscenter.softwareag.com/en/news-stories/press-releases/2022/0119_technical-debt-reality-check.html

3. “What is technical debt? How to pay it off (with examples).” Asana. July 10, 2022. https://asana.com/resources/technical-debt

4. Casey, Kevin. “How to explain technical debt in plain English.” The Enterprisers Project. June 3, 2020. https://enterprisersproject.com/article/2020/6/technical-debt-explained-plain-english

5. Madnick, Stuart. “New Cybersecurity Regulations Are Coming. Here’s How to Prepare.” Harvard Business Review, Technology and Analytics. August 29, 2022. Accessed February 23, 2023. https://hbr.org/2022/08/new-cybersecurity-regulations-are-coming-heres-how-to-prepare

6. Besker, Terese, Bosch, Jan, Martini, Antonio. “Technical Debt Cripples Software Developer Productivity - A longitudinal study on developers’ daily software development work.” ResearchGate. May 2018. https://www.researchgate.net/publication/325790190_Technical_Debt_Cripples_Software_Developer_Productivity_-_A_longitudinal_study_on_developers’_daily_software_development_work

7. Villars, Rick, et al. “IDC FutureScape: Worldwide IT Industry 2021 Predictions.” IDC. October 2020. https://www.idc.com/research/viewtoc.jsp?containerId=US46942020

8. “The Developer Coefficient.” Stripe. September 2018. https://stripe.com/files/reports/the-developer-coefficient.pdf

9. Fowler, Martin. “Technical Debt Quadrant.” martinfowler.com, October 14, 2009. https://martinfowler.com/bliki/TechnicalDebtQuadrant.html

10. Rios, Nicolli, et al. “Towards an Ontology of Terms on Technical Debt.” ResearchGate. Software Engineering Institute, December 2014. https://www.researchgate.net/publication/286010286_Towards_an_Ontology_of_Terms_on_Technical_Debt/

11. “Flexera 2022 State of the Cloud Report.” Flexera. 2022. https://path.flexera.com/cm/report-state-of-the-#page=1

12. Villars, Rick, et al. 2020. “IDC FutureScape: Worldwide IT Industry 2021 Predictions.” IDC: The Premier Global Market Intelligence Company. IDC Corporate. October 2020. https://www.idc.com/getdoc.jsp?containerId=US46942020

13. Bourne, Vanson. “Achieving AI: A study of AI opportunities and obstacles.” Fivetran. 2022. https://get.fivetran.com/rs/353-UTB-444/images/achieving-ai-a-study-of-ai-opportunities-and-obstacles.pdf

14. Vega, Myrna. “The real cost of tech debt.” J.P. Morgan. January 20, 2022. https://www.jpmorgan.com/commercial-banking/insights/the-cost-of-technical-debt

About CAI

CAI is a global services firm with over 9,000 associates worldwide and a yearly revenue of $1.3 billion+. We have over 40 years of excellence in uniting talent and technology to power the possible for our clients, colleagues, and communities. As a privately held company, we have the freedom and focus to do what’s right — whatever it takes.

Our tailor-made solutions create lasting results across the public and commercial sectors, and we are trailblazers in bringing neurodiversity to the enterprise.

Learn how CAI powers the possible at www.cai.io