The Impact of Technical Debt on Cybersecurity

BARR ADVISORY | HIVE SYSTEMS

Table of Contents

Refinancing Technical Debt

Have you ever done something manually in a spreadsheet promising yourself to automate, but ended up doing it manually for another year? In hindsight, if you had just adopted a different tool, you'd have saved dozens of hours in time and wages. Technical debt accumulation is a predictable outcome of human behavior. Humans aren't great at spotting subtle changes in their environment. One catchy metaphor for this phenomenon is the boiling frog story: A toad dropped into boiling water will hop away, but a toad brought slowly to a boil will not due to the slow and subtle increase in temperature. While the science of the metaphor has been debunked, the message is relatable and the fable lends well to memory. Organizations may not initially notice the small encroachments into their time and budget that result from technical debt. It can take time and an external party looking across our track record to spot the insidious shifting baseline.

> **Technical debt is the result of taking shortcuts to meet short-term objectives at the expense of long-term flexibility and security.**

The term “technical debt” has come to encompass a number of issues in the industry: bugs, legacy code, missing documentation, “silver bullet” tooling, poor system visibility, old hardware assets, weak governance, and more. With such a wide net, understanding how technical debt impacts your business can be a challenge.

In theory, technical debt is not much different from financial debt. Think about taking a financial loan—you have the money you need today to buy a house or car, but you end up paying a higher amount down the line. Similarly, incurring technical debt while building a product may improve your speed-to-market time, but will cost your organization more in terms of time and money down the road.

When people push problems into the future, they usually do so in order to get something faster. Like with financial debt, that can sometimes be a perfectly acceptable solution, as long as they pay back the debt within a reasonable period of time. However, if they fail to

pay back financial debt, even a fairly stable interest rate can quickly compound and exceed the value of any advantage they "bought" by taking out the debt in the first place. They may be willing to pay 1% in interest to have money early, but 50% may exceed our valuation of time or other resources they'd ultimately save. People often struggle with debt-related decisions even when they involve explicit quantities like in the example above—just imagine how difficult those decisions become in the relatively qualitative world of technology resource management. Cutting corners when designing or implementing technology doesn't manifest a clean number of what interest rate organizations will be charged if they carry a balance, or when exactly the balance is even due.

For a simple comparison, imagine being so pressed for time that you continuously use the same dishes and utensils for food without cleaning them. Eventually you find yourself with food poisoning, and fall behind significantly more than if you took the time to clean the dishes between uses. In the technology world, one way organizations "carry a balance" of technical debt is by rapidly developing, but not documenting. Even when employees review their own system architectures and code a couple of months after they've initially developed it, they may not recognize their own work and will need to spend a significant amount of time reacquainting themselves with the content.

Because the definition of technical debt varies between industry practitioners and academic researchers, and varies even further within each of these domains, we set out to find common themes, where it intersects with cybersecurity, and what methods of managing it hold the most promise. Because technology supports nearly every division of today's business, the accumulation of technical debt impacts many areas and in different ways. The impacted areas range from interpersonal relationships and company culture to hardware and software in data centers. Most of the research on technical debt covers the software development life cycle and in the management of systems and infrastructure. In this report, we focus on how technical debt impacts cybersecurity and compliance, and how it scales from startups with one laptop to large international organizations.

From software development to everyday IT

The term technical debt appeared as early as 1992 when developers explained the inevitable need to refactor their code. Since then, hundreds of studies and dozens of tools have been published specifically addressing code-related technical debt. Researchers identified and explored the interdisciplinary nature of technical debt in areas of

software, engineering, economics, and finance. Each of these areas quietly intersects with cybersecurity, but little research is available that evaluates the impact of technical debt on cybersecurity and compliance objectives.

Where technical debt intersects with cybersecurity

Cybersecurity and compliance professionals categorize problems into one or more of what is called the "C.I.A. triad." The C, I, and A stand for the confidentiality, integrity, and availability of information assets. Confidentiality is what people usually think of when they hear cybersecurity: keeping information visible only to specific parties. Integrity is about keeping information, such as datasets, accurate and whole, and availability refers to keeping systems and information accessible. Different information assets call for various degrees of confidentiality, integrity, and availability. The confidentiality of Wikipedia articles is non-existent by design, but it is critical in the case of medical or financial data. When at least one of the C.I.A. triad components are critical (confidentiality, integrity, and availability of information assets), cybersecurity and compliance professionals help to ensure that data remains secure.

Developers focus on how their own work impacts the availability of information assets over time, but cybersecurity professionals focus on how hackers, malware, or malicious employees impact the confidentiality, integrity, and availability of information. Resource managers then have to decide how much to fund cybersecurity efforts to minimize that impact. They do so under the constraints of limited resources and social pressures to keep productivity and innovation restrictions caused by cybersecurity controls to a minimum. This cycle inevitably results in reduced or deferred cybersecurity measures to more rapidly deliver information technology products and services, whether they are in support of employees within a company or sold to customers. Each time a company makes such a compromise, it accrues some amount of technical debt that it must then pay off in the future—and it could become the cause of a data breach.

A 2020 McKinsey survey found "CIOs estimated that tech debt amounts to 20 to 40 percent of the value of their entire technology estate before depreciation." The survey only covered the integrity and availability part of technical debt. Would including confidentiality-related technical debt meet or exceed 50%? The data breaches that hit news headlines tend to attribute blame to insufficient cybersecurity controls, but how much of that is due to unplanned or planned technical debt?

Planned and Unplanned Technical Debt

With planned technical debt, you know what you're getting into. You know that cutting corners now will make more work for your team in the future, but you've explicitly decided that it is worth it. More work in this context unfortunately doesn't mean more money, it just means more work.

With unplanned technical debt, you usually find yourself saying things like "it must have slipped through the cracks." While innovation thrives in the absence of bureaucracy, compliance, and checklists, so does error. We want engineers to design new, innovative airplanes but we don't want the pilots of our flight to try new, innovative piloting styles with us on board. Safi Bahcall sums this up: “we need to balance radical innovation with operational excellence.” In information technology, operational excellence is supported by tools like policies, ticketing systems, checklists, code version control, change management, and audits. These keep people, processes, and technology standardized where consistency is critical, preventing avoidable setbacks, which in turn reduces technical debt and gives us more time and resources to innovate.

Some industries define technical debt as only the unplanned technical debt that spontaneously emerges from the inevitable obsolescence of technology. What was designed at one point in time simply doesn’t fit the objectives and circumstances of the business any longer. Businesses are then forced to scrap, or ideally salvage, what hardware, software, and processes they find themselves with—rebuilding toward new goals within new constraints.

Between the extremes of planned and unplanned technical debt is a state in which you may be aware that some type of debt is accumulating, but circumstances or denial prevent you from intervening. The result is either more work pushed closer to your deadline, or a concentration of tedious work that may impact team morale when it is time to pay up and put in the work. Enough technical debt also makes you more fragile to operational continuity risks. As you near a deadline, your team members may call out sick, go on vacation, or leave the company. By holding too much technical debt, organizations cause the probability and/or impact of failure to increase as tasks are pushed closer to deadlines instead of closer to the present.

It is important to inspect system and software design plans for cybersecurity risks before committing resources to development. It is easy to fall into the trap of excluding security considerations until the end of a project with the intent of "bolting on" the security controls after the fact. Pushing integration of secure coding or design principles to the end of the development phase—delaying spending resources on security now with the intent of spending those cybersecurity resources at a later date—is another way to accrue technical debt. Technical debt of this kind, whether planned or unplanned, has an extra quality that is comparable to compounding interest. The cost of cybersecurity efforts tends to increase the further into development you are. Fundamental design principles, whether in software development, network architecture, or workflow design, have an impact on cybersecurity itself but also constrain the number of risk mitigation approaches available. For example, you may choose to migrate a server from your office's data-center to a cloud-based equivalent because the cloud provides superior availability and is less costly to maintain. Upon review, your security team points out that the cloud-based server is exposed to the internet, unlike your office server that was only exposed to the computers in the office building. You ask the cloud service provider to only allow log-in attempts from the computers in your office building or on your VPN. The cloud service provider says that this is not something that they are willing to do. As a result you now have to research, develop, and implement compensating security controls to attain the same level of risk you had when the server was hosted in the office data center. Whatever risk mitigating approaches you come up with will not be as effective as the IP restriction you had and will cost more than you originally planned to spend because you did not anticipate this cost in the first place.

Another example of delayed-security technical debt is when you are developing a web app and delay security vulnerability scanning of the code until later in the development lifecycle. The static or dynamic scanning of your code may reveal such a large number of vulnerabilities, or vulnerabilities that do not yet have mitigations, that you're forced to adopt a new coding framework and scrap everything that has already been developed. Cybersecurity professionals or developers experienced in secure coding will research which coding framework is the right match for what is under development, but also what has the best track record for finding and mitigating vulnerabilities.

Know The Signs

So what does technical debt look like in practice, and how can you recognize it when it starts to become a problem in your organization? Because technical debt can take a number of different forms, it can be easy for it to slip under the radar. Let's explore how to spot the warning signs before they lead to elevated cybersecurity risks and walk through a few indicators:

Signs of technical debt can emerge as early as the planning stage, such as failure to include a strategy to respond to issues as they arise. As the project goes on, poor documentation and issues with coding style may also be signals of technical debt. This might sound as simple as hearing "don't worry about the documentation for now," or "we only have one person on the team who knows how to fix this code."

Rushing to release before testing is another sign of technical debt. It might sound like "we don't have time to finish testing until after release, so we'll wait until later." Even if planned, if a company doesn't carefully track these decisions and prioritize remediating the consequences, it's easy for technical debt to become much more difficult to manage down the line.

Feeling cornered into taking risks or feeling like there are no other options is a good indicator that technical debt could cause a big problem for your organization. In these situations, having a transparent company culture is essential.

Let's take a look at another common situation of unintentional technical debt. Imagine the following scenario: your department begins using a new internet-based tool accessible via a web browser. It makes your work easier and more efficient, and the vendor chat support is great. Your entire department begins using a free version of this tool, and it becomes a

big part of how you do business. Given how this tool is improving your work, you decide to buy the paid version for even more functionality. But after purchasing the paid version, your security and compliance team sends a security questionnaire to the vendor and realizes that the tool isn't compliant with the types of data you handle. The security team informs you that there's another very similar tool available that is compliant and secure—but it would take a huge number of person-hours to transfer all of your data from the original to the new tool. In this example situation, your organization faces some difficult technical debt to overcome: either take the risk of being noncompliant and not secure, inviting a potential data breach, or spend more money than originally planned. Building internal reliance on webapps and outside tools prior to verifying their cybersecurity and compliance posture is one common way that organizations build technical debt that could otherwise be avoided.

These examples show how technical debt can impact organizations of every size, causing cybersecurity risk, financial burden, and stress.

Too Much Cybersecurity

On the flip side, too much cybersecurity or implementing cybersecurity controls too early can also be a source of technical debt. Cybersecurity controls can negatively impact productivity, innovation, flexibility, and even talent management if an organization does not apply them strategically, or does not take into account the cultural and socioeconomic context of where they are being applied.

Cybersecurity controls can have unintended consequences that degrade user trust, increase cybersecurity risk, and result in more technical debt. Controls that are perceived as invasive like computer and email monitoring agents may cause employees to avoid using the business managed computers and email accounts altogether. Some industries have very few privacy expectations due to the nature of their work, but others, such as higher education and journalism, not only expect privacy but depend on it in order to have candid conversations about controversial topics. Regardless of the industry, talent management risks also arise from cybersecurity controls that employees perceive as too invasive. Given the choice, high skilled employees may choose the privacy respecting employer over the heavily surveilled alternative. Employees who choose to stay, or don't have a choice in staying may opt for less secure consumer-grade technology and put company data at risk in order to avoid the surveillance. Today's increasingly remote workforce complicates this further as states begin to regulate or explicitly prohibit employee surveillance. As a result, even cybersecurity controls implemented with the best of intentions can result in everything from sunk implementation costs to lawsuits.

Regulation and compliance are necessary to maintain stability for an organization as it scales and are a requirement to enter certain markets. An unfortunate downside to increased bureaucracy is that it is eventually and inevitably prioritized over creativity and innovation as part of the increased responsibility of holding more liability of sensitive information and employee livelihoods.

Prevention

It's clear that technical debt can cause massive cybersecurity problems for organizations of all sizes. Luckily, there are steps that every organization can take to prevent technical debt from building. Keep in mind that these steps can evolve as an organization scales—for a small to medium-sized business, just a few basic cybersecurity measures are better than nothing.

Effective Planning

A key first step for preventing technical debt is to plan your projects effectively. Tools that manage tasks and timelines like Asana, Jira, or Trello can help with project planning, but it's important to remember that these tools can only be effective when used properly. Keep everyone up to date with shared calendars, and ensure everyone has access to important documents. When planning your project, storyboard as much as possible. While it's impossible to know every issue that might come up in the future, map out your project far in advance to refine your understanding of the risks that you are taking. Your project plan should include details on long-term operational cost, necessary support resources to reduce cybersecurity risk, and how you will keep up with annual maintenance.

Automation

Implementing an automated testing solution with continuous integration is another critical part of preventing technical debt. This ensures that each time a change occurs, engineers rigorously test your product or app for bugs or other issues, saving time and preventing problems from growing larger.

Best Practices

As obvious as this sounds, it's important for your developers to use code writing best practices. This means establishing and standardizing code with proper documentation. While this could potentially result in an increase in development time, it will pay off in the long run compared to a product written poorly.

If your organization is building an app or product in the cloud, it's critical to architect a cloud environment that can be rebuilt as needed. It's much harder to rebuild infrastructure after you've created your app.

Security Plan

Create a sound cybersecurity plan from the beginning. When you involve cybersecurity professionals in the early stages, they can help plan and build a project roadmap for proper threat modeling and help prioritize what needs to be done in the short-term, and what can wait. Not only does this prevent your business from becoming overwhelmed with the plethora of cybersecurity frameworks available, doing so can actually speed up development because it prevents late-stage restructuring for cybersecurity measures. This may involve performing a basic quantitative risk assessment to understand potential future risks. Considering cybersecurity from day one can also reduce friction in the sales cycle—when a potential client sends a security questionnaire, you can provide the necessary assurances much faster than if you still have to figure out what cybersecurity looks like for your business at this stage.

Accountability

One key mistake that many small to medium-sized companies make is failing to assign accountability for cybersecurity internally. At this size, it's just as important to have internal ownership of cybersecurity as a whole as it is to use tools and partners to help simplify cybersecurity. You can delegate everything except accountability, and this internal ownership can prevent excess technical debt from building up. If your organization doesn't have the resources for a dedicated cybersecurity professional, pick someone to be the cybersecurity advocate. The cybersecurity advocate's role for the design meetings will be to routinely ask how the confidentiality, integrity, and availability of the data and/or information systems are being protected. The cybersecurity advocate can also be included on a rotating basis—when more people take on cybersecurity, there's an opportunity for diverse perspectives and more people understanding what it's like to be on "both sides of the table" of taking cybersecurity seriously.

Culture

Finally, building a transparent culture that values and prioritizes cybersecurity is important for every company, but especially for those looking to avoid technical debt. When your stakeholders ask you for something "ASAP," imagine replying with "No problem, but can I have your approval to take out a loan to make it happen?" If they say no, they may not realize that what they are doing is taking out a loan of sorts by deprioritizing other work in favor of fast delivery. The loan isn't in the form of cash, but in offset human resources. At the very least, provide stakeholders with a best-case and worst-case outcome of offsetting resources to expedite delivery.

A culture that values cybersecurity is one that avoids the temptation to save money in the short term by cutting corners in development; it's also one that gives everyone a seat at the table when it comes to communicating about cybersecurity and risk. Give your employees a way to discuss technical debt and include them on your plan to remediate debt issues if and when they emerge. Providing everyone on your team with the space to be open and transparent about issues is a great step for every business. Acknowledge team members when problems are brought to light rather than swept under the rug or ignored—everyone must be part of the solution to bring visibility to issues.

Even for organizations that do everything right in the early stages, some technical debt may end up inevitable. And just like financial debt, not all technical debt needs to be entirely negative—so long as it's paid back in a timely and appropriate manner. Some developers and engineers argue that technical debt can be a good thing because it points to short-term wins that can be paid off eventually. From a cybersecurity perspective, before taking out a technical "loan," ask yourself: how confident are you that you'll remember to patch that vulnerability or upgrade that system next quarter at the latest? How confident are you that you'll have the resources to do so? So long as technical debt is monitored and addressed on a scheduled and regular basis, a reasonable amount of technical debt is understandable for most businesses.

Automation

Information technology, like currency, is mostly invisible. It is obvious when computer hardware or physical money are missing, compared to changes to software and digital currency. Financial monitoring and transaction reconciliation are heavily regulated and carefully managed. Even with that regulation, consumers still find themselves forgetting to unsubscribe from expensive services, discovering hidden fees, and reporting fraudulent charges. Today's financial accounts provide monitoring and alerting capabilities and at the very least statements of our transactions history. Technical debt doesn't come with the regulatory oversight, monitoring, and alerting that financial debt has. So how can organizations keep an eye on their technical debt balance?

In the case of planned technical debt, you're promising to do something in the relatively distant future instead of now, and you're using that promise as justification for cutting a corner in the short term. For example, you may decide to delay mitigating a cybersecurity vulnerability by installing a cybersecurity patch because that patch has a high probability of causing an outage during peak season for your business, so you'd prefer to wait until the peak has passed. Keeping track of these promises is important because you accepted the risk of an extended cybersecurity exposure period and there is nothing that automatically reminds you that you're still taking that risk.

Smaller organizations may automate technical debt management by setting themselves calendar reminders, using vulnerability management solutions that report on outstanding vulnerabilities after each scan, or establishing routine business processes like recurring compliance and risk assessment reviews that bring the promises back to management's attention.

There are many governance, risk, cybersecurity, and compliance automation solutions available that facilitate keeping track of promises. Traditionally, compliance monitoring solutions weren't much more than glorified calendar-based reminder systems that reminded specific people that they have some due diligence activity to complete on a monthly, annual, or quarterly basis. Today's compliance monitoring solutions automate these activities themselves wherever possible, saving companies from having to manually collect and upload the evidence. This kind of automation reduces the cost and user-error that comes with manual equivalents, and does so in parallel at scale which was not previously possible without incurring significant costs on personnel.

Small Business, Big (Technical Debt)

Startups and small businesses can “pay down” their technical debt by establishing a sustainable means of tracking it and reporting it to stakeholders (the people who decide how human and financial resources are spent on technology, even if that is yourself). Technical debt is inconspicuous until it starts causing problems, so it’s important to take special care to recognize it and keep track of whether it is increasing or decreasing. Without deliberately keeping technical debt at the forefront of our resource planning activities, it will continue to accrue unnoticed or quickly be forgotten in the face of more exciting business demands. Some organizations will encounter technical debt only occasionally, but most will carry a balance that they want to keep relatively low. In either case it is usually more practical to pay it off over time than to put the business on hold until you’ve caught up.

Paying down technical debt involves spending money. Whether it be new hardware, software, contract work, or wages, you need to spend resources to pay off neglected or obsolesced technology and the processes that support them.

If you find yourself in technical debt, begin by making your pay off plan. Create an inventory of all of your software, making note of any issues or vulnerabilities. Begin addressing them through a risk management strategy—prioritize eliminating the highest risks first, and then move your way down. Schedule regular time and resources to address and “pay off” these issues.

Once you begin to pay off technical debt, begin implementing the above recommendations on preventing debt from incurring. Whether this looks like a major culture shift in your organization or simply designating a cybersecurity advocate, these actions can help prevent technical debt from creating cybersecurity or compliance issues in the future.

Managing Technical Debt

When organizations intentionally or unintentionally put themselves into technical debt, they may not fully consider the benefits of either not taking out technical debt in the first place or better managing their technical debt situation. While technical debt may provide short-term gain, an organization that successfully avoids and manages their technical debt can expect lower overall costs, increased flexibility, increased sales, increased morale, and increased productivity.

Organizations frequently choose to put themselves into technical debt in the name of increasing short-term sales. Managing or avoiding technical debt in the name of cybersecurity, however, can actually have a positive and reliably steady impact on sales. When cybersecurity is an afterthought, you can run into major issues with partners or vendors down the line. But when cybersecurity is baked into your product early on, it can reduce friction in the sales cycle at a later date.

In the same vein, preventing and managing technical debt can lower overall business costs and increase flexibility. When there's no expensive surprises that require hours of overtime to fix, you have the flexibility to make smarter decisions on how to grow your business most effectively.

Lastly, avoiding and reducing technical debt can be an overall morale booster for your team, leading to increased productivity and better work. Accumulated technical debt can cause workplace frustration as developers work to fix problems and struggle with communication. An organization that successfully prevents and manages technical debt is one that prioritizes transparency and communication—providing an overall more positive work environment.

Conclusion

Technical debt is a balancing act. It would be naive to assume that an organization can avoid technical debt entirely—instead, preventing the accumulation of technical debt early on and strategically managing it over an organization's lifecycle to prioritize cybersecurity is the ticket for success.

Innovation is no longer just about delivering solutions—cybersecurity has become critical criteria for organizations at every stage. As we saw in the early days of research on the technical debt within software development, more research will come to light on the intersection of cybersecurity and technical debt. Organizations should stay informed on trends and best practices of this new component of cybersecurity and compliance.

About BARR Advisory and Hive Systems

BARR Advisory

At BARR Advisory, we build trust through cyber resiliency. We help protect the world's data, people, and information networks through a human-first approach to cybersecurity and compliance. Businesses looking for the accessibility of a boutique firm with the tools and expertise of a global consulting firm will find a partner in us.

Specializing in cybersecurity and compliance for companies with high-value information in cloud environments like AWS, Microsoft Azure and Google Cloud, BARR has the global network of partners, the perspective, and deep expertise every thriving SaaS provider to world-class enterprise needs to stay secure and compliant at every stage.

Brad Thies
President and Founder
bthies@barradvisory.com
888-532-2004

Hive Systems

Hive Systems provides smarter cybersecurity solutions with our trusted experts. Leveraging our collective experience, we promote a true partnership by understanding what makes your organization unique to help evaluate your cybersecurity strengths and vulnerabilities. Together, we'll develop a risk reduction strategy that best utilizes your existing investments, so you can reduce risk everywhere. Through Hive Helps, we offer pro bono consulting services to qualified non-profit organizations and communities to ensure that limited resources don't stand in the way of social progress.

So whether you know exactly what you need or have no idea, we love to talk about cybersecurity. And if you need help with something that we didn't discuss in this whitepaper, there's a good chance we can help with that too. Contact us directly and let's talk more about how Hive Systems can help make cybersecurity approachable for you and your company.

Alex Nette
CEO and Co-Founder
alex.nette@hivesystems.io
804-396-4720