Technical health isn't optional

An exclusive report from CTOs in the Asia-Pacific region

By Mike Loukides

If every company is a technology company, then every healthy company must have a healthy relationship to technology. However, we haven't seen any discussions of "technical health," which suggests that industry at large doesn't know what differentiates a company that's been through a successful digital transformation from one that's struggling.

To help us better understand technical health, we asked several CTOs in the Asia-Pacific (APAC) region what their companies are doing to prevent security incidents, how they use open source software, how they use technology strategically, and how they retain employees in a challenging job market. We hope that their answers will help companies to build their own strategies for digital transformation.

Being proactive about security

We asked the CTOs how their companies prepared themselves for both old and new vulnerabilities. The key is being proactive, as Shashank Kaul, CTO of Webjet, noted. It's important to use tools to scan for vulnerabilities—particularly tools provided by cloud vendors, such as Microsoft Azure's Container Registry, which integrates with Microsoft Defender for Cloud (formerly known as Azure Security Center) to scan containers for vulnerabilities continuously. Webjet also makes use of GitHub's Dependabot alerts, which are warnings generated when code in a GitHub repository uses a dependency with known vulnerabilities or malware. This proactive approach reflects an important shift from older reactive approaches to security, in which you deploy software and hope nothing bad happens. Tools like Container Registry and Dependabot alerts are constantly inspecting your code so they can warn about potential problems before they become actual problems.

Tim Hope, CTO of Versent, pointed to the importance of identity and role management in the cloud. Jyotiswarup Raiturkar, CTO of Angel One, said something very similar, highlighting the key role played by a zero trust policy that requires continuous validation (i.e., an identity is validated every time a resource is accessed). Raiturkar also emphasized the importance of "least privilege," in which access to resources is limited on a "need to know" basis. Least privilege and zero trust go together: if you constantly verify identities, and only allow those entities access to the minimum information they need to do their job, you've made life much harder for an attacker. Unfortunately, almost all cloud services grant privileges that are overly permissive. It's been widely reported that many—perhaps most—cloud vulnerabilities stem from misconfigured identity and access management; we'd bet that the same applies to applications that run on-premises.

(continued)

It's also important to recognize that "identities" aren't limited to humans. In a modern software architecture, a lot of the work will be performed by services that access other services, and each service needs its own identity and its own set of privileges. Again, the key is being proactive: thinking in advance about what identities are needed in the system and determining the appropriate privileges that should be granted to each identity. Giving every user and service broad access just because it's easier to make the system work is a recipe for failure. If you think rigorously about exactly what access every service and user needs, and implement that carefully, you've blocked the most important path through which an attacker can breach your infrastructure.

Sharing knowledge and solutions is the true heart of engineering culture, and that's visibly demonstrated by open source.

Threat modeling and penetration testing are also key components of a good security strategy, as Raiturkar pointed out. Threat modeling can help you assess the threats that you actually face, how likely they are, and the damage a successful attack can cause. It's impossible to defend against every possible attack; a company needs to understand its assets and how they're protected, then assess where it's most vulnerable. Penetration testing is an important tool for determining how vulnerable you really are rather than how vulnerable you *think* you are. The insights you derive from hiring a professional to attack your own resources are almost always humbling—but being humbled is always preferable to being surprised. Although penetration testing is largely a manual process, don't neglect the automated tools that are appearing on the scene. Your attackers will certainly be using automated tools to break down your defenses. Remember: an attacker only needs to find one vulnerability that escaped your attention. Better that someone on your team discovers that vulnerability first.

Open source and a culture of sharing

The rise of open source software in the 1990s has undoubtedly transformed IT. While vendor lock-in is still a very real issue, the availability of open source software has done a lot to liberate IT. You're no longer tied to Digital Equipment hardware because you bought a DEC compiler and have a few million lines of code using proprietary extensions (a very real problem for technologists in the 1980s and 1990s). More importantly, open source has unleashed tremendous creativity. The internet wouldn't exist without open source. Nor would many popular programming languages, including Go, Rust, Python, JavaScript, and Ruby. And although C and C++ aren't open source, they'd be much less important without the free GCC compiler. At the same time, it's possible to see the cloud as a retreat from open source: you neither know nor care where the software that implements Azure or AWS came from, and many of the services your cloud provider offers are likely to be rebranded versions of open source platforms. This practice is controversial but probably unavoidable given the nature of open source licenses.

So we asked CTOs what role open source played in their organizations. All of the CTOs said that their organizations make use of open source software and frameworks. Chander Damodaran of Brillio noted that "the culture of sharing solutions, frameworks, and industry-leading practices" has been a crucial part of Brillio's journey. Similarly, Tim Hope said that open source is critical in building an engineering culture and developing systems at Versent. That's an important statement. Too many articles about engineering culture have focused on foosball and beer in the company fridge. Engineering

(continued)

culture must focus on getting the job done effectively, whether that's building, maintaining, or running software. These responses suggest that sharing knowledge and solutions is the true heart of engineering culture, and that's visibly demonstrated by open source. It's an effective way to get software tools and components that you wouldn't be able to develop on your own. Furthermore, those tools aren't tied to a single vendor that might be acquired or go out of business. In the best case, they're maintained by large communities that also have a stake in ensuring the software's quality. Bill Joy, one of Sun Microsystems' founders, famously stated, "No matter who you are, most of the smartest people work for someone else." Open source allows you to use the contributions of those many smart people who will never be on your staff.

Unfortunately, only two of the CTOs we asked indicated that their staff were able to contribute to open source projects. One of the CTOs said that they were working toward policies that would allow their developers to release projects with company support. It's almost impossible to imagine a technical company that doesn't use open source somewhere. The use of open source is so widespread that the health of open source is directly tied to the health of the entire technology sector. That's why a critical bug in an important project—for example, the recent Log4j vulnerability—has serious worldwide ramifications. It's important for companies to contribute back to open source projects: fixing bugs, plugging vulnerabilities, adding features, and funding the many developers who maintain projects on a volunteer basis.

One solution is for the IT group to encourage employees in other divisions to build their own tools as they need them.

Thinking strategically about software

The CTOs we questioned had similar views of the strategic function of IT, though they differed in the details. Everyone stressed the importance of delivering value to the customer; value to the customer translates directly into business value. The best approach to delivering this value depends on the application—as Shashank Kaul pointed out, that might require building custom software; outsourcing parts of a project but keeping core, unique aspects of the project internal; or even buying commercial off-the-shelf software. The "build versus buy" decision has plagued CTOs for years. There are many frameworks for making these decisions (just google "build vs buy"), but the key concept is understanding your company's core value proposition. What makes your company unique? That's where you should focus your software development effort. Almost everything else can be acquired through open source or commercial software.

According to Tim Hope, the IT group at Versent is small. Most of their work involves integrating software-as-a-service solutions. They don't build much custom software; they provide data governance and guidelines for other business units, which are responsible for building their own software. While the development of internal tools can take place as needed in different business units, it's important to realize that data governance is, by nature, centralized. A company needs a standard set of policies about how to handle data, and those policies need to be enforced across the whole organization. Those standards will only become more crucial as regulations about data usage become more prevalent. Companies that haven't adopted some form of data governance will be playing a high-stakes game of catch-up.

(continued)

Likewise, Jyotiswarup Raiturkar at Angel One focuses on long-term value. Angel One distinguishes between IT, which supports internal tools (such as email), and the "tech team," which is focused on product development. The tech team is investing heavily in building low-latency, high-throughput systems that are the lifeblood of a financial services company. Like Versent, Angel One is investing in platforms that support data discovery, data lineage, and data exploration. It should be noted that tracking data lineage is a key part of data governance. It's extremely important to know where data comes from and how it's gathered—and that's particularly true for a firm in financial services, a sector that's heavily regulated. These aren't questions that can be left to ad hoc last-minute solutions; data governance has to be consistent throughout the organization.

Although software for internal users (sometimes called "internal customers") was mentioned, it wasn't a focus for any of the IT leaders we contacted. We hear increasingly about "self-service" data, democratization, "low code," and other movements that allow business units to create their own applications. Whatever you call it, it seems that one role for a company's technology organization is to enable the other business units to serve themselves. IT groups are also responsible for internal tools that are created to make the existing staff more efficient while avoiding the trap of turning internal projects into large IT commitments that are difficult to maintain and never really satisfy the users' needs.

One solution is for the IT group to encourage employees in other divisions to build their own tools as they need them. This approach puts the IT group in the role of consultants and helpers rather than developers. It requires building a technology stack that's appropriate for nontechnical employees. For instance, the IT group may need to build a data mesh that allows different units to manage their own data while using the data from other parts of the organization as needed, all subject to good policies for data governance, access control, and security. They'll also need to learn about appropriate low-code and no-code tools that allow employees to build what they need even if they don't have software development skills. This investment will give the rest of the company better tools to work with. Users will be able to build exactly what they need, without passing requests up and down an error-prone chain of command to reach the software developers. And the IT burden of maintaining these in-house tools will (we hope) be reduced.

Keeping employees happy and challenged

It goes without saying, but we'll say it anyway: even with news of tech sector layoffs, today's job market is very good for employees trying to find new jobs, and very tough for employers trying to hire to support company growth. In many organizations, even maintaining the status quo is a challenge. What are APAC CTOs doing to keep their staff from jumping ship?

Every company had training and development programs, and most had multiple programs, adapted to different learning styles and needs. Some offered online training experiences only; Angel One provides both online and in-person training to its employees. Offering programs for employee training and development is clearly "table stakes," a necessity for technological health.

It's more important to look at what goes beyond the basics. Webjet recognizes that training can't just take place outside of business hours. Managers are charged to carve out work time (roughly 10%) for employees to participate in training—and while 10% sounds like a small number, that's a significant investment, on the order of 200 hours per year devoted to training. It's worth noting that our *2021 Data/AI Salary Survey* report showed that the largest salary increases went to employees who spent over 100 hours in training programs. While it's a crude metric, those salary increases clearly say something about the value of training to an employer.

(continued)

Shashank Kaul also observed that Webjet keeps its IT developers as close as possible to the problems being solved, and in conversation with their counterparts at customers' firms, avoiding the problem of becoming a "feature factory." This description reminds us of extreme programming, with its regular demos and contact with customers that allowed software projects to keep on target through many midcourse corrections. It's important that Kaul also sees contact with customers and peers as an aid in retaining engineers: no one likes to spend time implementing features that are never used, particularly when they result from inadequate communication about the actual problems being solved. Webjet and Versent also run regular employee hackathons, where anyone in the organization can participate in solving problems.

Jyotiswarup Raiturkar offered some additional ideas to keep employees happy and productive. Angel One has a "permanent work from anywhere" policy that makes it much easier for employees to balance work with their personal life and goals. The ability to work from home, and the time that you get back by avoiding a lengthy commute, is worth a lot: in congested cities, an 8-hour day can easily become a 10- to 12-hour commitment. It's important that this policy is permanent: employees at many companies got used to working at home during the pandemic and are now unhappy at being asked to return to offices.

Raiturkar also noted that Angel One's employees can roll out features in their first few days at the company, something we've seen at companies that practice DevOps. An important part of Facebook's "bootcamp" for new employees has been requiring them to deploy code to the site on their first day. Continuous deployment may have more to do with software engineering than human resources, but nothing makes employees feel more like they're part of a team than seeing their changes go into production.

What is technical health?

In this brief look at the experiences of CTOs in the APAC region, we see a proactive approach to security that includes the software supply chain. We see widespread use of open source, even if employees are limited in their freedom to contribute back to open source projects. We see Agile and DevOps practices that put software developers in touch with their users so that they're always headed in the right direction. And we see training, hackathons, and work-from-anywhere policies that let employees know that they, their careers, and their home lives are valued.

If all companies are software companies, technical health is not optional.

We hope all companies will consider technical health periodically, ideally when they're forming plans and setting goals for the coming year. As the business world moves further and further into a radical technical transformation, every company needs to put in place practices that contribute to a healthy technical environment. If all companies are software companies, technical health is not optional.