White Paper
Data Center
April 2019

IT@intel

Enterprise Technical Debt Strategy and Framework

Learn how Intel IT developed a unique framework to successfully manage technical debt and position the enterprise for modernization, innovation, and digital transformation

Since implementing our technical debt framework, we have eliminated over 665 applications and platforms and have seen close to a 30 percent reduction in the enterprise landscape.

Intel IT Authors

Harish Thanneer
Principal Engineer

Catalin Baicu
Sr. Technical Program Manager

Table of Contents

Executive Summary

Large organizations gradually accumulate technical debt for various reasons. Intel largely runs its business with enterprise platform solutions that comprise a mix of commercial off-the-shelf (COTS) and custom applications. Over time, the number of platforms and applications delivering enterprise capabilities has grown significantly, leading to duplication of solutions, overlap of capabilities across multiple platforms, and layers of customization. These all have contributed to technical debt, which has a negative impact on business velocity, modernization, digital transformation, and ability to be highly innovative—we're spending the majority of our budget on Run, instead of on developing new capabilities.

Intel IT's systematic approach to reducing technical debt includes a well-defined framework that encompasses the full scope and complexity of Intel's business. Applying the framework has helped contribute to the following:

- Business outcome-driven enterprise architecture (EA) across the business, data, application, and technology (BDAT) domains
- Governance and accountability
- Prioritization of investments
- Operational efficiency and shifting budget into innovation, new capabilities, and enhancements
- Awareness and ownership of technical debt

Since implementing our technical debt framework in 2017, we have eliminated over 665 application/platforms and have seen close to a 30 percent reduction in the enterprise landscape.

Phasing in technical debt-reduction activities allowed us to focus on big wins immediately while laying the groundwork for complex items that require broader alignment with dependencies. Our strategy includes establishing standards, roadmaps, and target EA blueprints to guide technical debt reduction and prevention.

We approached technical debt holistically with a comprehensive view of the Intel IT enterprise, which enabled us to successfully establish and execute the framework. This has led to greater awareness of technical debt across the organization and a culture shift that is a key element in preventing technical debt from accruing in the future.

Intel IT Contributor

Brent Conran, Chief Information Security Officer

Acronyms

BDAT	business, data, applications, and technology
EA	enterprise architecture
EOL	end of life
TIME	Tolerate, Invest, Migrate, or Eliminate

Background

"In software-intensive systems, technical debt is a collection of design or implementation constructs that are expedient in the short term but set up a technical context that can make future changes more costly or impossible."[1] Technical debt is a metaphor introduced by Ward Cunningham in 1992, stating that "a little debt speeds development so long as it is paid back promptly with a rewrite. ... The danger occurs when the debt is not repaid."[2] Essentially, technical debt is the gap between making a change perfectly and making the change quickly.

When new technology becomes available, it is tempting to ignore legacy systems and focus on the "shiny new object"—this is how technical debt begins to accumulate. As long as an organization relies on technical debt only for quick short-term gains, the debt does not cause too many issues. But then the next emergency comes along, and a few more quick fixes get made, and the debt gets larger. When this cycle repeats multiple times, it can cripple an organization's productivity with interest payments. Eventually, the expense of the new features outweighs the value that they bring to the business. At this point, an organization is not just sinking in technical debt, it is drowning.

According to Gartner, by 2023, 90 percent of all technical debt existing today will still exist and will continue to strangle business innovation.[3] It is critical that we address Intel's technical debt, because legacy systems prevent our teams from moving quickly, innovating, modernizing, and delivering new capabilities that are aligned to Intel's digital transformation. Bi-modal IT—where some teams focus on predictability while others on exploration—can actually make technical debt worse. But most technical debt is invisible; therefore, it is easy to overlook and invest in new technologies like cloud and mobile without cleaning up existing technical debt problems. We have found that as technical debt grows as more applications, platforms, and business processes are delivered, the majority of our IT budget is spent on Run instead of on innovation and new capabilities.

Sources and Types of Technical Debt

There are three high-level types of technical debt:

- **Deliberate or prudent debt** is introduced when quick changes are done to reduce time to market.
- **Accidental or outdated design debt** is a result of systems evolving over time. When new capabilities are introduced, it takes more time to implement them because the design may not scale—thus requiring significant refactoring.
- **Bit rot debt** is the result of complexity introduced over time with many incremental changes and deviations from the original design and intent. In our experience, this type of debt is difficult to fix after the fact; therefore, we attempt to prevent this type of debt from occurring in the first place.

The technical debt framework we have developed encompasses all three types of technical debt at various levels.

Impact of Technical Debt

As shown in Figure 1, the accumulation of technical debt impacts both the cost to deliver solutions and the ability to respond to customers' needs. High technical debt leads to lower productivity, reduced quality, and a need for constant code revisions.

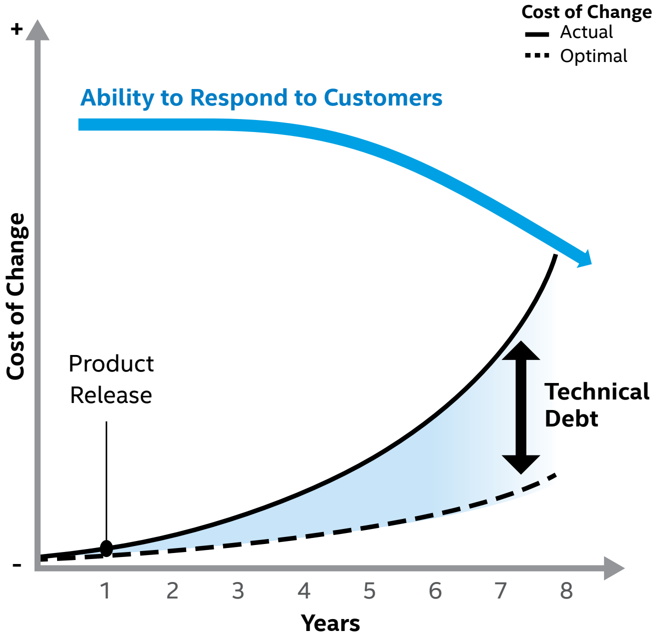


Figure 1. As technical debt increases, it becomes costlier to make changes and more difficult to quickly respond to customers' needs. (Image credit: Jim Highsmith)

[1] Paris Avgeriou, Philippe Kruchten, Ipek Ozkaya, and Carolyn Seaman, eds. "Managing Technical Debt in Software Engineering," Dagstuhl Reports, 6:4, 110-138, 2016.

[2] Ward Cunningham, March 26, 1992. "The WyCash Portfolio Management System." c2.com/doc/oopsla92.html

[3] CIO Dive, October 2017, "Expectation, meet reality: The 4 biggest takeaways from Gartner Symposium 2017." ciodive.com/news/expectation-meet-reality-the-4-biggest-takeaways-from-gartner-symposium-2/506974

Addressing Technical Debt Is Core to Digital Transformation and IT Strategy

As mentioned earlier, accruing technical debt results in higher operational costs, employee inefficiency, and slower time to market. But more importantly, makeshift solutions stacked atop legacy systems ultimately take more time and money to revise, leaving fewer resources for innovation and growth. One of the key pillars of digital transformation at Intel is technical debt reduction. Reducing technical debt and modernizing legacy systems by applying our technical debt framework will enable us to invest in new capabilities and digital transformation initiatives for future success and reduce cybersecurity risk.

Solution: Addressing Technical Debt with a Guidance Framework

Sporadically pursuing technical debt is not very effective. Instead, we use a framework to guide our technical debt efforts. This framework is holistic, in that it encompasses the full scope of technical debt to drive prioritization, aid in decision making, and fuel digital transformation. Our unique framework spans the entire business, data, application, and technology (BDAT) domains. Figure 2 provides a high-level summary of the three phases of our technical debt framework. See "Applying the Technical Debt Framework" for more details.

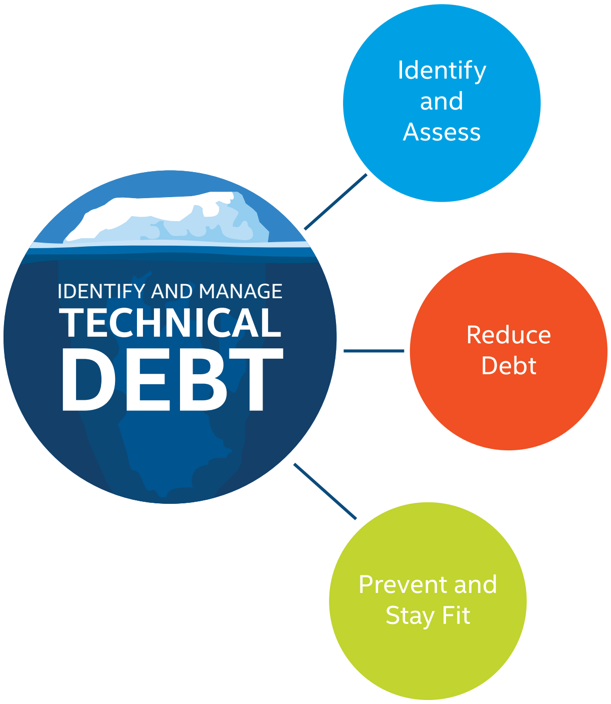


Figure 2. Consisting of three steps—identify and assess, pay and reduce, and prevent and stay fit—our unique technical debt framework spans the business, data, application, and technology domains.

How to Measure Technical Debt

We measure technical debt at two levels:

BDAT Level

We use the following vectors to measure technical debt across the BDAT stack:

- Velocity to introduce changes
- Cost of change/release
- Alignment of capability with our target enterprise architecture (EA)
- Alignment of technology with our EA standards and roadmaps
- Technical debt ratio, computed using the SQALE model, provides a consistent metric to help prioritize at the software code composition level

Using automated tools to measure and publish the technical debt metrics helps raise visibility and make technical debt reduction an enterprise-wide priority.

Architectural Level

As part of application governance, it is important to introduce defined criteria to measure and score EA debt. This helps to quantify risks and technical debt creation. The computed score will help governance bodies to approve or reject a project before it gets too far along. Adopting this model brings EA debt to the surface, making it much more visible and forcing correct decisions.

Computing the Cost of Technical Debt

We developed a method to compute the cost of a technical debt item and its impact to the enterprise to prioritize technical debt issues and focus on the ones that will most benefit Intel. The cost includes both principal and interest:

- **Principal.** The effort to address the technical debt item.
- **Interest.** The maintenance cost, difficulty to introduce changes, and the risk that the debt might get out of control.

It is important to note that the interest can continue to increase based on time and other events (that is, the cost of technical debt continues to rise if not addressed early).

Intel IT's Technical Debt Strategy

We apply Gartner's Tolerate, Invest, Migrate, or Eliminate (TIME) model characterization to enable technical debt reduction. This model enables us to characterize every application to verify that it aligns to the target EA and technology roadmap. It also helps us identify the applications that carry the most debt to determine technical debt-reduction initiatives that will bring significant immediate and long-term value. Using this approach allows us to establish the core foundation required to perform the assessment to reduce technical debt systematically at an enterprise level. Each application is assessed and tagged appropriately based on the TIME model.

For example, applications that are no longer needed or that provide duplicate capabilities are tagged for elimination. Applications that align to strategic investment and core platforms are tagged for investment. Legacy applications that are still considered business-critical are tagged for toleration.

Figure 3 shows how each application/system characterized by the TIME model is handled differently to manage technical debt.

Applying the Technical Debt Framework

As mentioned earlier, our technical debt framework consists of three phases. A phased approach prevented us from being overwhelmed and enabled us to prioritize items in an orderly fashion.

Phase 1: Identify and Assess

During this phase we identified and assessed existing debt across the enterprise using the following steps:

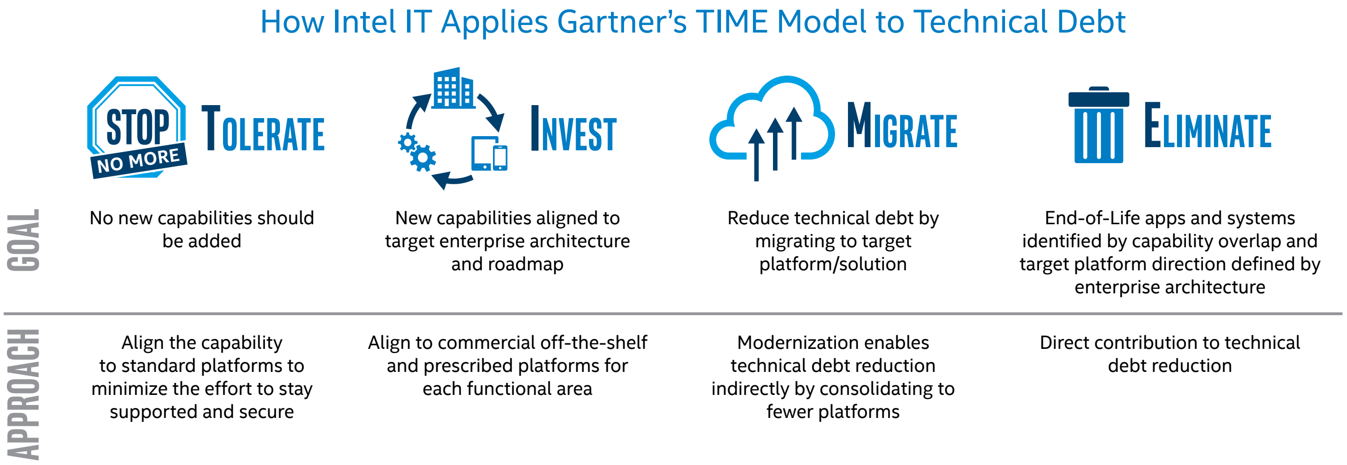
1. **Create an asset inventory.** Because most technical debt is invisible, we wanted to create a comprehensive inventory of all the applications and systems in use by Intel IT (several thousand). We recorded all the information in an enterprise asset management system. This asset inventory, along with corresponding metadata such as application owner, technologies used, and other characteristics, formed the baseline for mapping enterprise capabilities to the EA.
2. **Quickly identify which applications can be eliminated.** Once the inventory was complete, we assessed application capabilities and how they aligned to IT objectives and Intel's digital transformation journey. These capabilities were then mapped to business processes as part of the BDAT EA. This step quickly identified applications, platforms, and technologies that were no longer needed or that overlapped with others. We marked all those items for end-of-life (EOL) (that is, "eliminate" using the TIME model).
3. **Identify where investment needs to be made.** Next, using Gartner's Pace-Layering strategy, we identified the "big bets" for Intel. These are large investments in platforms and technology (such as cybersecurity) that will drive the business forward. Using this knowledge, we consolidated isolated solutions into core platforms.
4. **Identify technical debt at the code and design levels.** We used an open source platform for static code analysis. This tool continuously and automatically inspects code quality. It finds bugs, security vulnerabilities, and source code characteristics that may indicates a deeper problem.
5. **Take advantage of early wins to build momentum.** We established an EOL roadmap to eliminate the obvious applications from the above steps. This enabled us to show immediate benefit and generated enthusiasm for further technical debt reduction.

When we had completed Phase 1, we could illustrate the overall cost savings from the elimination of applications and systems to management and IT staff. All this has been made possible by implementing a good strategy and structure and endorsing the right culture across IT.

Phase 2: Pay and Reduce

Having eliminated the obvious technical debt culprits in Phase 1, we moved into Phase 2. We followed these steps:

1. **Clean up inventory data.** We revisited our asset inventory to verify that the data about applications and systems was accurate and current.
2. **Establish total cost of ownership (TCO).** Next, we established a method for calculating the TCO for each application or system. This method enabled us to assess costs consistently. We made sure to include all aspects of the system, including costs, licenses, hardware, support, and headcount.



	TOLERATE	INVEST	MIGRATE	ELIMINATE
GOAL	No new capabilities should be added	New capabilities aligned to target enterprise architecture and roadmap	Reduce technical debt by migrating to target platform/solution	End-of-Life apps and systems identified by capability overlap and target platform direction defined by enterprise architecture
APPROACH	Align the capability to standard platforms to minimize the effort to stay supported and secure	Align to commercial off-the-shelf and prescribed platforms for each functional area	Modernization enables technical debt reduction indirectly by consolidating to fewer platforms	Direct contribution to technical debt reduction

Figure 3. Gartner's Tolerate, Invest, Migrate, or Eliminate (TIME) model enables us to systematically assess applications and take the right path to technical debt reduction.

3. **Perform full characterization of all applications.** In Phase 1, we focused mainly on the "eliminate" category of the TIME model. In Phase 2, we looked more deeply at each application, platform, and technology and established a comprehensive roadmap for all categories: Tolerate, Invest, Migrate, and Eliminate. This step required a target EA for all the business capabilities aligned to the technology roadmap. Our goal was to consolidate assets, resulting in fewer platforms and solutions—hence reducing software maintenance costs.
4. **Establish a plan for each application or system.** Once all the assets were categorized based on the TIME model, we combined this information with our TCO figures to establish an EA technology roadmap. This roadmap is aligned with business strategy and business outcomes. Every application was marked for elimination, migration, or consolidation. This roadmap included dependencies. For example, we could not EOL an application that was being used by another business-critical application until we migrated that application to a more modern system.
5. **Follow the roadmap.** Once the roadmap was complete, we executed each item in the roadmap (like any program) to ensure full alignment to the target EA. We also made sure that all the assets related to eliminated systems were decommissioned.
6. **Modernize legacy apps that are critical to run the business.** It is important to provide support and secure legacy applications that are critical to business. Modernization can help identify legacy applications that are candidates for leveraging containers, microservices, public cloud, and other initiatives aligned to target EA. Modernization at Intel IT is a strategic investment that allows innovation while enabling technical debt reduction.

Phase 3: Prevent and Stay Fit

Phase 3 (our current phase) overlaps with Phase 2, but our focus shifts from EOL and migration to explicitly managing technical debt to stay fit. This is an on-going process, because no large enterprise, including Intel, will ever be completely free of technical debt. The trick is to stay on top of it and control it. During this phase (that is, for the foreseeable future), we are focusing on the following:

- Establish technology standards, EA practices, and governance, integrate them into the delivery of all new capabilities across the enterprise, and keep them current.
- Update standards and the target EA to keep pace with technology trends and the direction and strategy of the business.
- Integrate technical debt management into our DevOps model to make technical debt visible. We intend to avoid irresponsible technical debt, and capture any deliberate or prudent debt as part of the product backlog. We have dedicated a certain percentage of our IT teams (the exact percentage will vary from one enterprise to the next) to work on technical debt items with small refactoring installments in each iteration. Adopting this model keeps everyone informed and drives the right prioritization of new functionality.
- Effect a culture change by educating the organization about the importance of managing technical debt. The required new mindset embraces technical debt management as a key component of good software development practice and a key enabler of Intel's continued digital transformation and success.

Conclusion

Our technical debt journey began in 2017 by assessing more than 2,500 applications. This assessment resulted in identifying potential business benefits. To date, using this assessment, we have eliminated over 665 applications and platforms. With this reduction in the number of platforms and applications, which are better aligned to the enterprise strategy and technology roadmap, we have seen close to a 30 percent reduction in the enterprise landscape. The resulting budget can now be applied to modernization, innovation, and digital transformation initiatives (including modern cybersecurity). We have also experienced significant success in changing our IT culture, making technical debt management part of our everyday thinking. This culture change is paramount in sustaining technical debt management over the long term.

The enterprise benefits of technical debt reduction are substantial:

- **Efficiency.** Fewer required IT teams, lower support costs, and less suppliers to manage.
- **Stability.** Fewer changes to the core platform means fewer bugs to fix.
- **Agility.** Faster pace of change (faster validation).
- **Reusability.** Business units know what capabilities are available, and developers know how to introduce new functionality.

Our success hinges on our ability to optimize at every layer, then focus on what makes a difference for the business.

Related Content

If you liked this paper, you may also be interested in these related stories:

- Take the First Step Toward Digital Transformation - IT Framework for Technical Debt Reduction
- The Battle Against Technical Debt on the Journey to IT Transformation

For more information on Intel IT best practices, visit **intel.com/IT.**

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation:

- Twitter
- #IntelIT
- LinkedIn
- IT Center Community

Visit us today at intel.com/IT or contact your local Intel representative if you would like to learn more.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

THE INFORMATION PROVIDED IN THIS PAPER IS INTENDED TO BE GENERAL IN NATURE AND IS NOT SPECIFIC GUIDANCE. RECOMMENDATIONS (INCLUDING POTENTIAL COST SAVINGS) ARE BASED UPON INTEL'S EXPERIENCE AND ARE ESTIMATES ONLY. INTEL DOES NOT GUARANTEE OR WARRANT OTHERS WILL OBTAIN SIMILAR RESULTS.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS AND SERVICES. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS AND SERVICES INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.