

A Better Way to Plan the Payoff of Technical Debt



Technical debt is an ongoing issue no one should expect to square away because as technology advances, even today's top systems will eventually achieve full "legacy" status. However, if you don't keep on top of it, technical debt will eventually cause significant damage to your pocketbook and reputation. If you think that sounds like an exaggeration, get up to speed on [Southwest Airlines' meltdown](#) during the 2022 holiday season. It was a technical debt-driven debacle of the most public nature.

Instead of pretending technical debt doesn't exist or fooling yourself into thinking that you can solve it once and for all, consider addressing technical debt as a recurring cost related to maintaining your technology investments. If you're ready to get a handle on your organization's technical debt, let's talk about some ways to smooth the process through better planning and using a data operating system.

Step 1: Figure Out What You've Got

The first step in any planning process related to system replacement is to understand what is in place and how the necessary future state looks. Unfortunately, many legacy systems are difficult to navigate and hard to query. As a result, it can be quite a task for any team, technical or not, to dive into legacy systems and validate precisely what data each system contains, its structure, what the replacement system will need to create, and which subsets of that data you need to extract and migrate before retiring the system.

A data operating system, such as DataOS from The Modern Data Company, can immensely benefit this situation. A data operating system can be laid on top of any legacy system and tied into its data. It then provides a modern interface layer that easily interacts with common reporting and analysis tools. These tools can request data from the data operating system, which in turn handles extracting the data from the legacy system. This seamless connectivity lets people use the tools they are comfortable with, even when working with data from a legacy system. As a result, inventorying and profiling the data is much easier and faster than using the tools embedded within a legacy system. This is especially true given that few people will be knowledgeable about and comfortable with those legacy toolsets.



Step 2: Determine How to Combine Information

Looking into a specific system is a start. However, to unlock the maximum power of corporate data, it is necessary to mix data from different systems and allow each data source to enhance the others. Various architectures, from data warehouses to data lakes, have attempted to help solve this problem over the years. However, those solutions entail extracting data from the source systems and copying it into a more analysis-friendly environment. Even if enterprise-level data repositories of this nature are in place, it is unlikely that they contain 100% of the information contained within source legacy systems. This discrepancy is because only the most important and widely used data is typically extracted and loaded into an environment such as a data warehouse or data lake.

Before retiring a legacy system, however, it is a good idea to ensure that there aren't other data worth preserving that never made it out in the past. A data operating system makes this exploration seamless because any system already mapped can be joined and mixed with any other system. The data operating system will handle the access and movement of data from each underlying system to facilitate any query. Just putting a layer over unchanged legacy systems isn't the long-term solution. But in the short term, it allows for a thorough exploration of an organization's data to document and plan what data will be migrated and kept and what data a replacement system will need to generate.

Step 3: Design a New User Layer

Once the data from the first two phases is understood and documented, the next step is determining what functionality users will need and how to deliver them. There is no better way to help users decide their requirements than to allow them to access prototypes that can be created and updated quickly. Using prototypes to define and validate requirements mitigates risk and streamlines the development of a final solution.

Laying an application on top of a data operating system's map of corporate data enables application prototypes to be developed quickly and easily. Interfacing with the data operating system can happen through fully modern tools and protocols while the data operating system handles the messy details of dealing with the legacy systems. Over time, prototypes will evolve into production applications that can sit on top of the same operating system as the initial prototypes. Modernizing applications and access to data is a crucial component of retiring technical debt, and this phase gets an organization a good part of the way there as far as the front-end applications go.

Step 4: Finally, Retire Your Debt

While bolting modern applications on top of legacy systems is a start, it has some shortcomings. First, the legacy systems are still in place, even if users have better access to their data and more functionality. Next, performance won't be optimal because there is still a dependency on the legacy platforms to serve the data. Finally, as legacy systems age, they become more prone to failure and to having underlying code that contains now-unsupported functions and components.

The final step in retiring technical debt is to replace each legacy system over time. The beauty of doing this after a data operating system is in place is that the end-user interfaces don't need to change. As a new system comes online, the data operating system can be repointed to map to the new system instead of the legacy system. While technical work is still necessary to make that remapping occur, it will be seamless for the end users and won't require application changes. Instead, those applications will make the same data requests as always, but the data operating system will execute them differently by making the most of the modernized replacement systems. At this point, there are updated applications sitting on top of updated systems connected by a modern layer of technology that, when combined, retires your technical debt! When systems eventually need to be replaced again, repeat the process. It's a winning approach.



Develop a Plan to Pay Off Your Technical Debt Today!

Following the steps outlined, you can modernize your handling of data as you retire technical debt. While it will still require time, effort, and money to retire your technical debt, it will require much less than it would have using approaches of the past. Following the steps outlined here will also help you avoid having technical debt blow up in your face, as happened so publicly to Southwest Airlines.

A data operating system, such as DataOS from The Modern Data Company, can be used to assess and diagnose where debt exists up front, how to best mitigate that debt, and can also be a part of the final solution. Once an architecture based on a data operating system is in place, it will make future technical debt much more manageable. Just make sure to budget the costs as part of ongoing maintenance efforts so that your technical debt doesn't again grow to a dangerous level in the future.

To learn more about how a data operating system like DataOS can help your organization modernize its systems and end-user functionality, download our e-book "Modernize Your Data Architecture Without Ripping and Replacing."

BY E. YOUNANZADEH



A Better Way to Plan the Payoff of Technical Debt

© 2023 The Modern Data Company. All trademarks are properties of their respective owners.

The Modern Data Company
306 Cambridge Ave
Palo Alto, CA 94306
TheModernDataCompany.com
info@TMDC.io