Softura

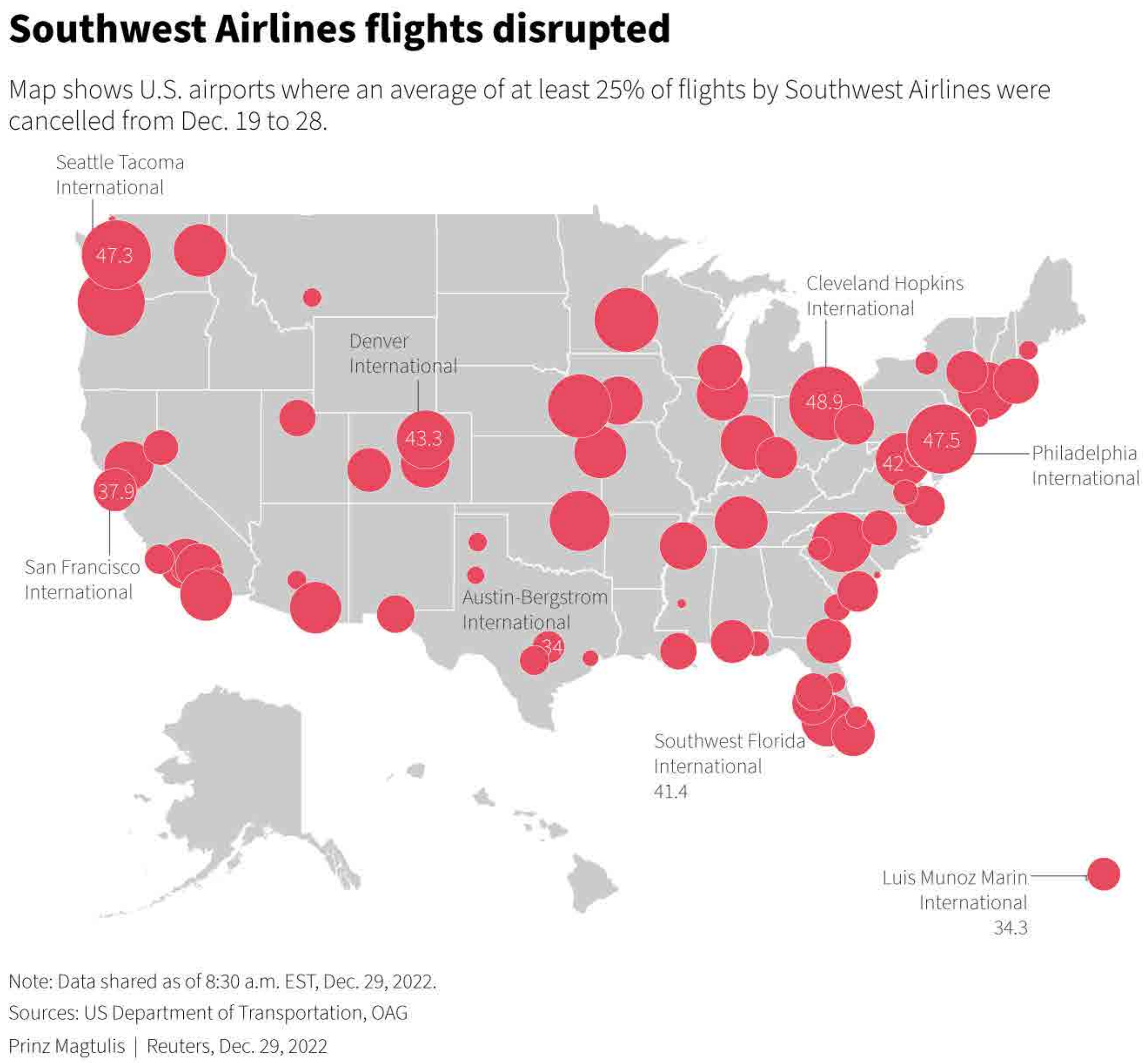
Lessons Learned from Southwest Airlines:

The Implications of Technical Debt for

Technology Leaders in 2023

Introduction

For the first time in 51 years of operations, Southwest Airlines experienced a "perfect storm" from a trifecta of factors - an intense winter storm, a risk-prone point-to-point operating model and outdated technology. 16000 flights were cancelled between December 19 to 28 which also happens to be the busiest holiday period of the year when demand peaks.



Southwest Airlines flights disrupted

Map shows U.S. airports where an average of at least 25% of flights by Southwest Airlines were cancelled from Dec. 19 to 28.

Seattle Tacoma International
47.3
Denver International
43.3
Cleveland Hopkins International
48.9
47.5
42
Philadelphia International
37.9
San Francisco International
Austin-Bergstrom International
34
Southwest Florida International 41.4
Luis Munoz Marin International 34.3

Note: Data shared as of 8:30 a.m. EST, Dec. 29, 2022.

Sources: US Department of Transportation, OAG

Prinz Magtulis | Reuters, Dec. 29, 2022

In this article, we will analyze how technical debt affected Southwest Airlines and what are the lessons that companies can learn from this incident. We will also discuss how regular technology assessments will prevent companies from getting impacted by outdated systems during a crisis or high levels of demand… the time that your customers need you the most.

How Technical Debt Disrupted Airline Operations?

Southwest Airlines began its digital transformation journey in 2017 when it committed $800 million to improve customer experience and employee productivity through software and workflow optimizations. $300 million was earmarked for "new technologies to support operations" and $500 million was allocated for developing a reservation system. The company stated that, by 2020, it aimed to boost earnings through software revamp that was to include new systems, updates, and upgrades that would allow the company to effectively track flights, passengers and cargo.

Southwest decided to develop a proprietary system known as "OpsSuite" to compete more effectively with JetBlue and Delta Airlines which were already using mobile devices to record information. OpsSuite was to enable employees to work together seamlessly on tasks such as tracking in-transit flights, informing ground crew to prepare gates for arrival and oversee maintenance in real-time instead of relying on outdated tools such as faxes to track progress.

Some of these challenges were not unique to Southwest and were seen in competing airlines as well. For instance, on January 23, 2017, United Airlines announced that it grounded all domestic flights at Newark International Airport, due to a computer glitch. This was not an isolated incident but a part of a series of systems failure that ranged from flight plan software losing functionality to losing access to reservation records. Air Canada and Porter Airlines also suffered delays and cancellations due to software glitches.

Fast forward to 2022 during a major winter event that hit airline operations during the week before Christmas, leading to a cancellation of 16,000 flights by Southwest. The company's scheduling software systems were outdated and couldn't handle the cancellations.

Michael Santoro, Vice President of the Southwest Airlines Pilots Association told CBS News that "The storm was the catalyst that started this whole event, but the major problem is that our scheduling IT infrastructure is outdated and can't handle the massive cancellations that had to happen that day when the weather event occurred. You get this snowball effect where it can't keep track of where pilots are, flight attendants are, and airplanes are."

Why do such system failures occur?

- Outdated scheduling systems (aging code) that required modernization
- Lack of integration between different software systems to ensure smooth flow of data
- Insufficient automation and reverting to manual ops at the first sign of trouble

The first reason stated above is also known as technical debt.

What is technical debt?

Technical debt refers to the accumulation of technical challenges, complexities, and shortcuts that a software system incurs over time. It is a metaphor that compares the accumulated challenges to financial debt. Just like financial debt, technical debt can be a useful tool when used responsibly. However, if it is not managed properly, technical debt can accumulate and eventually become a burden that can hinder a software system's development and maintenance.

Before we proceed to study the impact of technical debt, we will understand the main causes:

- Rushing a project or taking shortcuts during development
- Tight deadlines or lack of resources
- Lack of understanding about the long-term consequences of certain decisions
- Lack of maintenance or neglect of the codebase
- Outdated or deprecated technologies being used
- Poor design or architecture decisions that make the codebase difficult to work with or modify
- Priorities for investment to go elsewhere (other projects, shareholder returns, etc.)

Managing Technical Debt

Ward Cunningham, author of Agile Manifesto, coined the term technical debt. He said, “Shipping first-time code is like going into debt. A little debt speeds development so long as it is paid back promptly with refactoring.”

Managing technical debt is an important part of software development, as it can help ensure that a software system is maintainable and able to evolve over time. This involves making strategic decisions about when to incur technical debt, as well as actively tracking and paying off existing debt.

Here is a list of best practices on how companies manage technical debt efficiently:

- ✓ **Monitor and track technical debt :** Keep track of the amount and types of technical debt that your team is accumulating, so that you can prioritize and address it effectively.
- ✓ **Prioritize debt repayment:** Not all technical debt is created equal, so prioritize the most critical or time-sensitive issues first.
- ✓ **Invest in code review and testing:** Regular code reviews and testing can help prevent the accumulation of new debt and can also help identify existing debt that needs to be addressed.
- ✓ **Refactor code:** When repaying debt, take the opportunity to refactor the code to improve its design and architecture, which can make it easier to maintain and modify in the future.
- ✓ **Allocate time and resources for debt repayment:** Make sure that you have the necessary time and resources dedicated to addressing and repaying technical debt. This might require setting aside dedicated time for refactoring or hiring additional team members to help with the effort.
- ✓ **Incorporate debt repayment into your development process:** Make debt repayment a regular part of your development process, rather than a one-time effort. This can help prevent the accumulation of new debt over time.

Metrics to Measure Impact on Business

Some of the ways that technical debt can negatively impact a business include:

- ✓ **Reduced efficiency :** Technical debt can make it more difficult for developers to work on a codebase, which can slow down development and reduce efficiency. This can lead to missed deadlines, delays in releasing new features, and a slower time-to-market for new products.
- ✓ **Increased maintenance costs:** Technical debt can also increase the cost of maintaining and updating a codebase, as it may require more time and resources to fix problems or make changes

- **Decreased competitiveness :** If a business's competitors can release new products or features faster, it can reduce the competitiveness of the business. Technical debt can make it more difficult for a business to respond to changing market conditions or customer needs, which can negatively impact its market position.

- **Decreased customer satisfaction:** If a business's products or services are slow or prone to bugs, it can lead to decreased customer satisfaction. This can have a negative impact on the business's reputation and can lead to lost customers.

McKinsey introduced a concept called Tech Debt Score (TDS) to quantify the scale, determine the optimum level and communicate the benefits of reducing tech debt. There are several metrics that can be used to measure the impact of technical debt on a business:

- **Time to market:** Tracking the time it takes for the business to release new products or features can provide insight into the impact of technical debt on the efficiency of the development process.

- **Development velocity:** Measuring the rate at which the development team can deliver new code can help identify any slowdowns that may be caused by technical debt.

- **Maintenance costs:** Tracking the resources (such as time and money) required to maintain and update the codebase can provide insight into the financial impact of technical debt on the business.

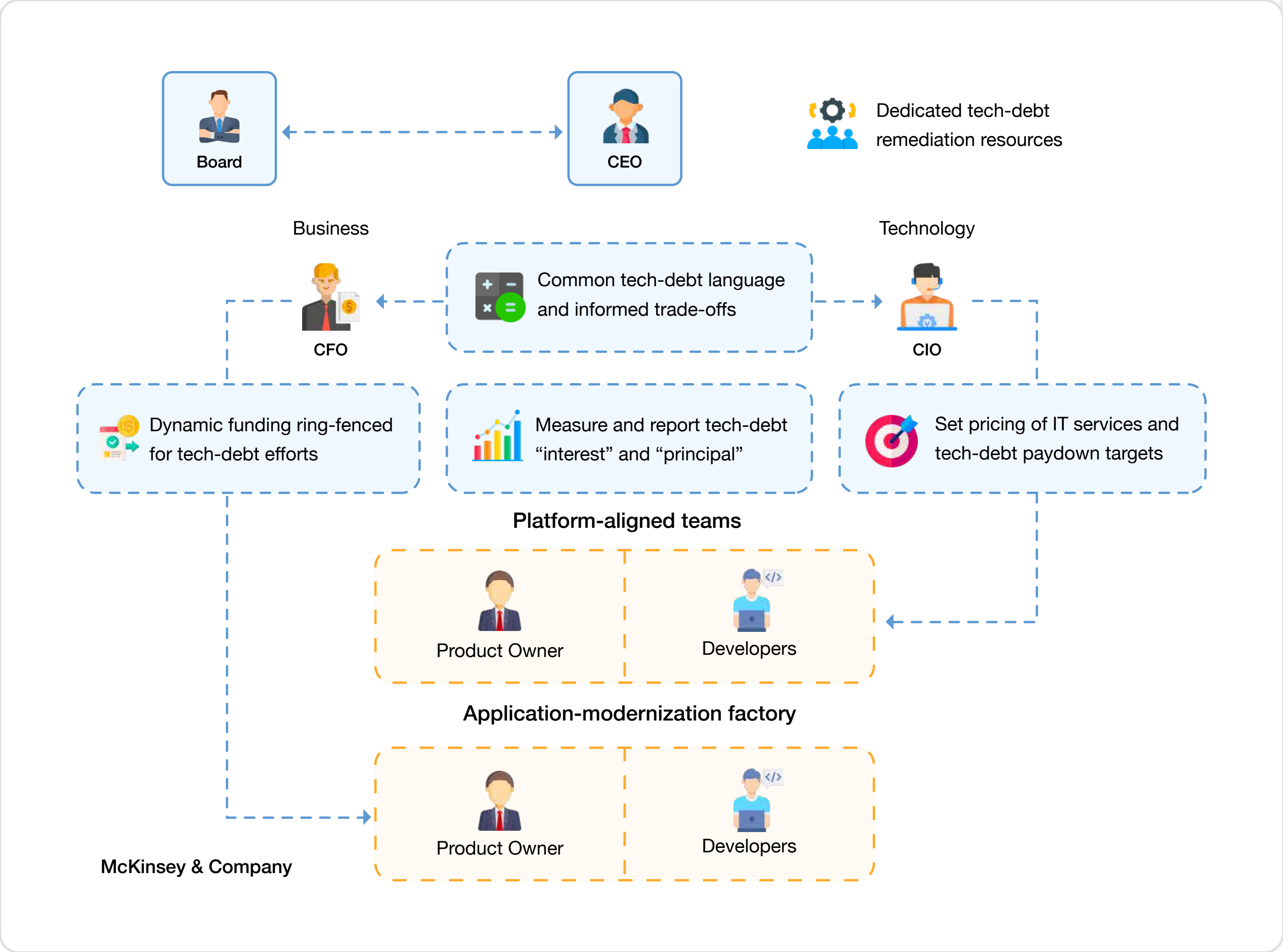
- **Customer satisfaction:** Monitoring customer satisfaction through metrics such as net promoter score (NPS) or customer satisfaction (CSAT) can provide insight into the impact of technical debt on the quality of the business's products or services.

- **Return On Investment (ROI):** Calculating the return on investment for projects that have incurred technical debt can help determine the financial impact of technical debt on the business debt.

- **Codebase quality:** Assessing the quality of the codebase using metrics such as code coverage, code complexity, or defect density can help identify areas of the codebase that may be contributing to technical debt.

McKinsey further created an operating model that can help companies paydown technical debt and build resiliency.

An efficient operating model enables business / technology cooperation on tech-debt paydown.

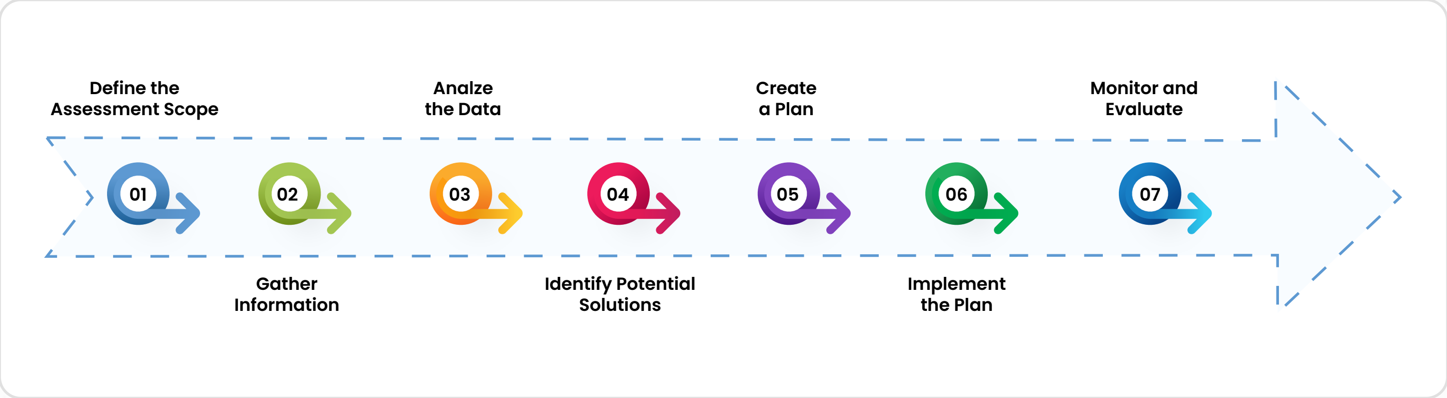

Board
CEO
Dedicated tech-debt remediation resources
Business
Technology
Common tech-debt language and informed trade-offs
CFO
CIO
Dynamic funding ring-fenced for tech-debt efforts
Measure and report tech-debt "interest" and "principal"
Set pricing of IT services and tech-debt paydown targets
Platform-aligned teams
Product Owner
Developers
Application-modernization factory
Product Owner
Developers
McKinsey & Company

Source:

https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/demystifying-digital-dark-matter-a-new-standard-to-tame-technical-debt

Benefits of Technology Assessment

By conducting regular technology assessments, you can identify areas of your system that are contributing to technical debt and address them before they become major problems. Here are the things to consider when you perform technology assessments.



- **Define the scope of the assessment :** Determine which systems and technologies will be included in the assessment and set clear goals for what you hope to achieve.

- **Gather information:** Collect data about the systems and technologies being assessed, including information about their design, implementation, and usage. This might include technical documentation, user manuals, and performance data.

- **Analyze the data:** Use the information you have gathered to identify areas of the system that may be contributing to technical debt or inefficiencies. Look for patterns and trends that may indicate problems or opportunities for improvement.

- **Identify potential solutions:** Based on your analysis, come up with a list of potential solutions to address the problems or inefficiencies you have identified. Consider the pros and cons of each option and consider how they might impact other parts of the system.

- **Create a plan:** Develop a detailed plan for addressing the identified problems or opportunities for improvement. This should include specific steps for implementing the chosen solutions, as well as timelines and resources needed.

- **Implement the plan:** Follow the steps in your plan to implement the chosen solutions. This may involve updating or replacing existing systems, implementing new technologies, or making other changes to the system.

- **Monitor and evaluate:** Monitor the system after the changes have been made to ensure that the desired results have been achieved. Regularly evaluate the system to identify any additional problems or opportunities for improvement.

Conclusion

Modernizing legacy applications is the best approach to reducing technical debt and minimizing the impact of such debt on operational resilience. In 2019, Toyota North America decided to develop an integrated technology management framework that was to help the company:

- Build an inventory of applications by using Application Portfolio Management
- Rank applications based on criticality to business operations
- Determine software currency (how current are the critical software applications)
- Map applications with infrastructure requirements
- Implement disaster recovery for critical applications, data and infrastructure

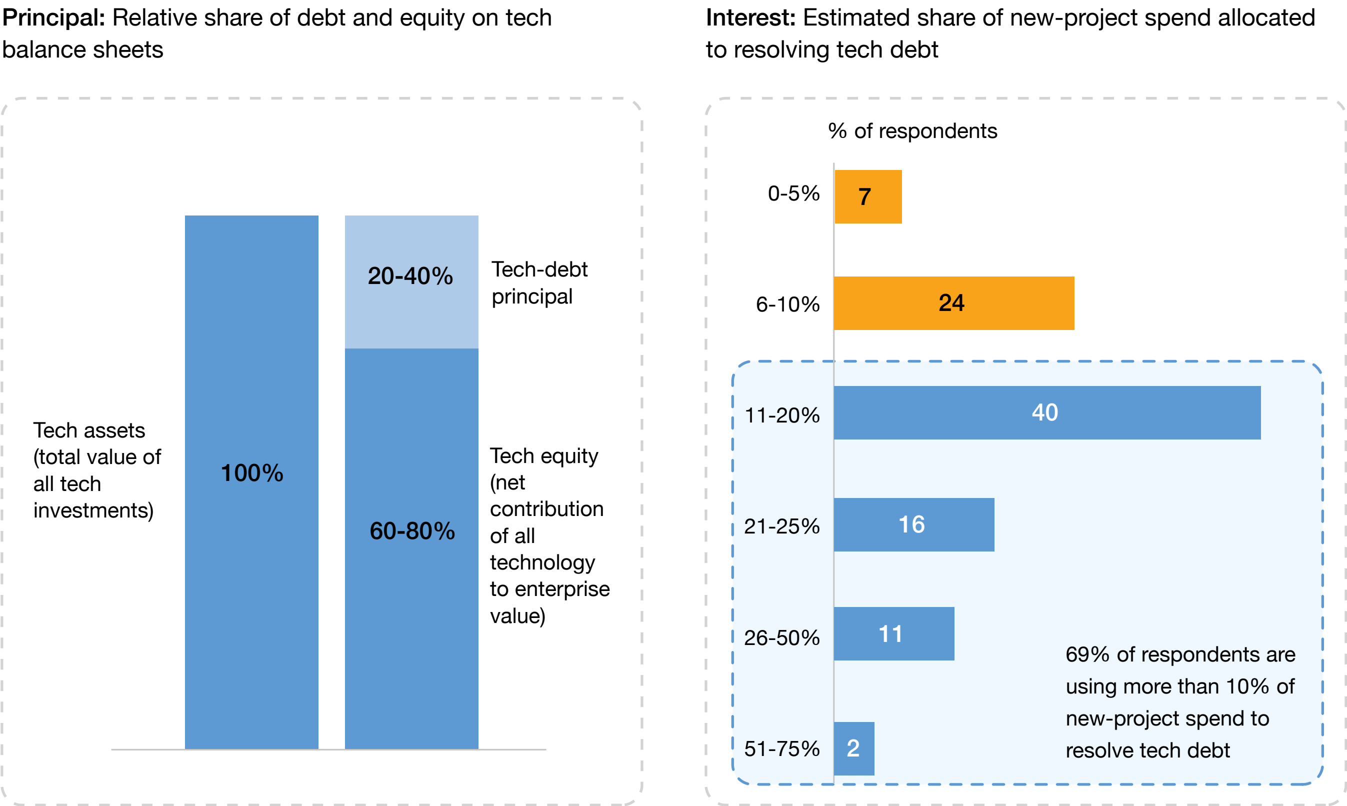
The IT leadership team at Toyota North America ensured that modernization decisions such as code refactoring, architecture changes, programming language changes, and cloud migrations were based on periodic operational risk analysis.

In conclusion, the benefits of a technology assessment far outweigh the cost. Cost of an assessment will depend on the scale and complexity of application portfolio, but it is estimated to range from $10,000-100,000.

McKinsey reported in 2020 that more than 60% of CIOs surveyed reported a rise in technical debt over the preceding three years. 10 to 20% of IT budgets that were supposed to be allocated for new software projects is getting diverted to addressing issues arising out of tech debt. Also, the same study found that they can reduce the developer time paying the tech debt from 75% to 25%.

Tech-debt principal accounts for up to 40 percent of IT balance sheets, while most Companies pay more than 10 percent interest on projects.

CIO estimates of spend on technology debt



Source: Mckinsey survey of tech debt among 50 CIUs, July 2020

McKinsey & Company

If done right, technology assessments and a risk-adjusted roadmap for application modernization will bring down technical debt and allow companies to focus on developing new software that will unlock new revenue opportunities, improve operational efficiency, reduce costs and lead to better profitability.

This article is brought to you by Softura.
Follow us on https://www.linkedin.com/company/softura/.
To learn more about how we can help you modernize applications and reduce tech debt, visit https://www.softura.com/application-modernization-solutions/.

Contact Us at

(844) 791- 0545 | info@softura.com | www.softura.com

Our Offices

Farmington Hills, MI | Chicago, IL | Indianpolis, IN | Pittsburgh, PA | Atlanta, GA | Houston, TX
Tampa Bay, FL | Charlotte, DC | Chennai, IN | Bangalore, IN | Vadodara, IN | Toronto, CA