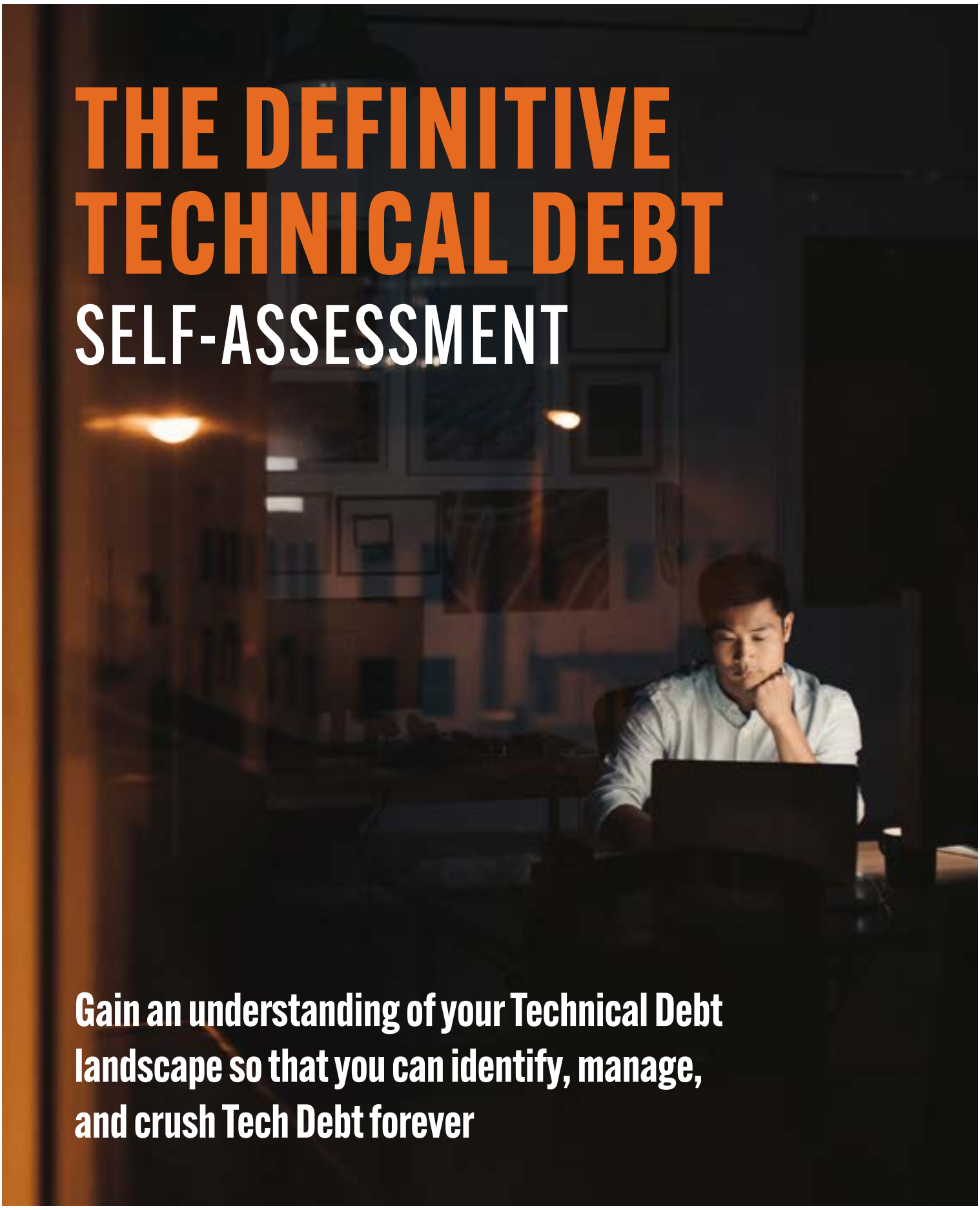
THE DEFINITIVE TECHNICAL DEBT SELF-ASSESSMENT



Gain an understanding of your Technical Debt landscape so that you can identify, manage, and crush Tech Debt forever

EXCEPTIONAL DIFFERENCE®
EMPOWERING THE WORLD'S ELITE ENGINEERS AND LEADERS

exceptionaldifference.com

Contents

About this Assessment

With this assessment guide in 30 minutes, you will be able to:.

1. Understand the kinds of Tech Debt in your environment and the magnitude of its impact.
2. Recognize the enormous costs of Tech Debt – financial, effort, and emotional.
3. Realize that Tech Debt can be managed.
4. See the path to reclaiming your joy of engineering.

In 1992 Ward Cunningham said, “With borrowed money, you can do something sooner than you might otherwise, but then until you pay back that money you’ll be paying interest. Rushing a product out the door to get some experience is a good idea, but you eventually repay that loan by refactoring to reflect your experience.”

Exceptional Difference has helped thousands of engineers across the globe crush technical debt. We have worked in various industries, including defense, financial services, space, information technology, consumer sales systems, navigation, and more, and we have developed a comprehensive approach to understanding and crushing Tech Debt.

We created this assessment as a novel way of categorizing and understanding your Tech Debt.

GOALS	COSTS	TYPES OF DEBT	NEXT STEPS
What are your goals regarding technical debt? Yes, your goal is not to have any. However, the best teams have clear goals about why that is important to them. Take a few minutes to think about why it is important to you.	You will assess the costs of technical debt to your team’s spirit. (Yes, we said spirit. That is just as important as economics!) You will also make an estimate of the tangible costs of technical debt to the organization.	Did you know there are 7 types of technical debt? You will be provided a description of each of the 7 types and guided to assess their impact on you and your organization. You may be surprised by some debt items you have just taken for “business as usual.”	Once you understand the types of technical debt, it will be easier to determine where each exists in your environment. You will also be able to identify where technical debt is injected into your solutions. This will equip you to create a plan to get out of debt and to stay out of debt for good.

How is Technical Debt Sucking the Soul Out of Your Teams?

Do You Spend Your Time in "Create" Or "Waste"?

The technical debt problem is vividly represented in this picture. The green in the picture is the creative side of the work that engineers love to do. It is figuring out the real user needs through user stories. It is building designs to solve complex challenges. It is writing the code that will implement solutions.

The red in this picture is waste. The effort does not add to the value side of the equation. This is time lost to doing rework.

In working with thousands of people in various domains, we studied where engineers spent their time. And although each team is unique in their work and how they do it, we found that all teams share in technical debt problems.

On average, these companies lost 65% of their development time. In other words, their development teams lost all that time to rework and technical debt issues.

Before we look at the economic cost, consider the cost of your team's energy. Which of these is true for you and your team? Check all those that apply:

- Feeling frustrated
- Working late to overcome what should have been much easier
- Feeling plagued by tons of little (and big problems)
- Having to deal too often with "spooky motion at a distance," where you make a change in one part and that mysteriously breaks something in another part of the system
- Making it work because there was no time for a re-design,
- The business side doesn't want us wasting time fixing things that already "work"
- The business side keeps asking for more complex features
- Feeling like most of the time, we are trying to crawl up a muddy hill while people push boulders from the top at us

What is Technical Debt Costing Your Organization?

Consider what technical debt costing your organization.

When you take on financial debt, you pay back that debt with interest. That interest is paid in money.

When you have technical debt, it costs you, your team, and your organization something as well. That something is lost time.

To get started with a plan for addressing technical debt, it is important to first consider how much time the technical debt is costing you and what it will take to pay it off now so you can stop paying with lost time week after week, month after month, and year after year.

Consider this example from one of the teams we worked with. This examples is representative of so many others.

> Pat's team analyzed an item to determine that it would take only three days to refactor that item completely. That sounds small, but it was a piece of *debt that all developers had complained about over the last three years.* They determined that every time they touched this item, it cost them an additional five days because of that tech debt. Five days is no big deal. Why bother?
>
> But, the team typically had to change this item 30 times a year.
>
> In three days, they redesigned the item to removed that debt and discovered they were wrong. It saved them seven days of effort every time they had to work in that item. ***With three days of effort, they found they had saved 210 days for the following year.*** They were excited because they knew they could do amazing things with that time saved.

The ROI on this effort was incredible. And there are many other examples. Some even more dramatic in the amount of engineering time saved. One team invested six people-months in fixing a Gordian Knot (explained later in this guide), ***saving two years of time every year thereafter.***

On the next page, left blank for you, take some time to do your own back-of-the-envelope estimates. This can be challenging without experience. Remember, if you need help, just schedule time with Exceptional Difference for a free strategic assessment session, email: hello@exceptionaldifference.com

Your Analysis

If you could wave a magic wand and all the Tech Debt could disappear, what would that enable you to do?

What is the cost of Tech Debt to your organization?

What happens if you do not fix your Tech Debt?

Which Types of Technical Debt Do You Have?

Following is a quick overview of the major types of things that slow organizations down. Read it through, and we will walk you through some questions about each type to help you better understand the technical debt you and your team may be dealing with.

1. Gordian Knots

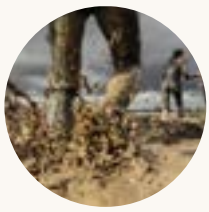
Implementations have highly convoluted and complex structures, like balls of spaghetti. Developers fear making changes because it is hard to make changes and easy to break things.

2. Barnacle Ware

Implementations start with core features (the boat) that work well. Various customers each ask for a special feature (a barnacle) just for them. Soon there are more barnacles than boat.

3. Glut

When the system is choked with too many low-value items. This glut slows development and system performance. Examples include feature glut, application glut, and data glut.



4. Environmental Drag

Work feels like a difficult slog through thick mud. Cumbersome tools, low-value process steps, bad meetings, and other numerous small annoyances accumulate and create drags on speed.

5. The World Moved On...

Developers feel trapped in the past even as they see “better-faster” all around them. Examples include systems that rely on out-of-date technology, or underlying designs that do not match current approaches.

6. Trap Doors

Work that was moving forward well gets derailed by a trap door opening. For example, the foundation on which the system depends has incomplete, incorrect, or even missing parts. Changes that should have be quick end up taking a long time.

7. Skill Debt

The lack of experience in the critical areas that slows everything down. New customer or technology domains can cause experienced people to be rookies. More people could help, but mainly an expertise infusion is needed.

1. Gordian Knots

We call implementations with highly convoluted and complex structures, like spaghetti balls, Gordian Knots. These are intractable problems that can only be solved by cutting the knot, just as Alexander the Great did to the original Gordian knot. Developers fear these parts of the system because making changes is hard and changes often break other parts of the system. This often happens to a core piece of the system. It worked well when it was initially designed, but over time new needs arose and many changes were made. Developers are known to cringe when they have to make more changes inside the mess.

On this page, create a list of the Gordian Knots that exist in your system. Consider how big they are. Consider what types of pain they cause. Contemplate how much pain each one of them is causing you and your teams.

2. Barnacle Ware

Implementations start with core features (the boat) that work well. Various customers each ask for a special feature (a barnacle) just for them. Soon there are more barnacles than boat. When the barnacles start costing you speed, you have a problem.

In some companies, we have seen the customer *specials* (the barnacles) become larger than the core system itself. Furthering the pain, it often turns out that those barnacles eat up engineers' time, but only bring in a small percentage of the revenue.

On this page, create a list of the barnacles that exist in your system. Consider how big they are. Consider what types of pain they cause. Contemplate how much pain each one of them is causing you and your teams.

3. Glut

Glut is when the system is choked with too many low-value items. This glut slows development and system performance. Here are three examples of Glut.

- **Feature Glut**
  - Let's first consider the user's viewpoint. Have you ever used an application that was fun and easy to use? And then, over time, more and more features were added. Did you notice the first ones you liked, but soon the application seemed crowded and harder to use? Has your team created that for some of your users?
  - Now consider it from the developer's viewpoint. Are you maintaining several features that very few customers ever use? Does the maintenance of those features slow you down? Are you maintaining any pieces of the system that are never used by anyone at all?
- **Application Glut.** Many IT teams find they are maintaining *hundreds* of apps. Many are redundant or overlap. Some only have a handful of users. Is this slowing your team down?
- **Data Glut.** One customer we worked with had a customer database with tens of millions of customers and 30 years of history. However, less than 10% of the data was being used. This slowed down every interaction with the database and led to erroneous analysis.

On this page, list types of glut that exist in your system and the impacts these items have on your team.

4. Environmental Drag

Work often feels like a difficult slog through thick mud. The environment slows teams down due to things like cumbersome tools, low-value process steps, and low-value meetings. Often people become so used to the environmental mud, they don't even consider that many of these items can be addressed. These small annoyances accumulate and create drags on speed.

What environmental drags on speed does your organization experience?

5. The World Moves On

Developers feel trapped in the past even as they see "better-faster" all around them.

One example is systems that rely on out-of-date technology. More than one organization we worked with was relying on compilers where the company that created the compiler was out of business. Another example is where the original system designs do not match current approaches the developers are using. The conflict between the approaches causes a significant drag on speed and hinders quality development.

Many times these companies are at risk of going over the technical debt cliff, which is the point at which the system is no longer maintainable.

On this page, list types of "the world moves on" debt that exist in your system and the impacts these items have on your team.

6. Trap Doors

Work that was moving forward often gets derailed by a trap door opening. For example, the system's foundation has incomplete, incorrect, or even missing parts. Changes that should have been quick end up taking a long time.

A trap door is when you think something is a simple change, but then the bottom falls out. Suddenly, you realize this will take longer than you ever expected.

One customer had hard coded a list of products instead of creating a look-up table. This hard coded list appeared over ten times in the code base!

On this page list types of "trap door" debt that exist in your system and the impacts these items have on your team.

7. Skill Debt

The lack of experience in critical areas slows everything down. It helps if your people are extremely talented. But, a new customer domain or technology domain can cause experienced people to be rookies. Ironically, sometimes when organizations fix other types of debt (such as the world moved on with new technology, or environmental drag with new tools), they suddenly find themselves with skill debt. An expertise infusion is needed to accelerate the learning and thus the speed to developing customer value.

On this page, list any skill debt area that exists in your system and the impacts these items have on your team.

There is a Way Out!

Some people who take this assessment find themselves discouraged at this point. Don't be! You've taken an important first step toward eliminating technical debt.

Maybe you've just realized that there is more technical debt in your organizations than you imagined.

Stop and take a breath.

We have worked with many organizations that were once plodding through knee-deep mud.

Some of the teams felt like they were going slower every production cycle – and when we put the measurements in place – we found out they were right.

Know this: Many teams, even those in the deepest mud, have completely overcome their issues with technical debt. The engineers on these teams are realizing the dream of building amazing products, unencumbered by the technical debt.

"Each week I took what I learned and immediately applied it to myself and my team. We now all have so much more control over deadlines and quality." — **Francisco Javier**

At Exceptional Difference, we believe that one person can change a team and one team can change an organization. In fact, that's the only way change happens.

We know that high performing teams are the We know that high performance teams are the difference between success and failure. We also know that exceptional teams don't just happen. They require exceptional engineers, exceptional leaders, and the right culture. Our Exceptional Teams offerings build upon our offerings for engineers and leaders by helping teams achieve a new level of cohesiveness, productivity, and excellence.

At Exceptional Difference, we're committed to empowering elite engineers, teams, and leaders to reduce technical debt so you can reclaim the joy of engineering. Our Technical Debt offerings range from talks and workshops to more in-depth programs such as deep dives, experiences, and highly customized engagements that help you understand the types of technical debt, where they each come from, where they are lurking in your environment, how much they're costing you, and the data required to make decisions you can stand behind with courage and conviction. From there, we will help you identify strategies to remove, manage, and prevent technical debt! We will help you reclaim ownership of your speed to value and foster a culture that crushes tech debt for good!

We Are Here For You

Don't hesitate to reach out to our team for assistance. For decades we have helped thousands of engineers and leaders through the challenges associated with tech debt and other engineering problems. Our mission here is to change the world of engineering for the better so you can do the work you love.

Join us for any of our tech debt talks, workshops, or the Technical Debt Experience. See our website for details.

Julia Mullaney
CEO

Alan Willet
Chief Engineer

David VanEpps
COO

SBA
WOSB CERTIFIED