

# **Transitioning to Leadership: What's a Cookie?**

“Fuck.”

That's exactly how I feel today. I am the first one in the office, and getting in early before the crack of dawn is not something I do very often. The last software build failed, and I have to get it working before everyone gets here. Especially before Jessie gets in. Because once Jessie sees the problem du jour, well then, my day is freaking *over*.

I make a small change and start another software build. It's dark outside. Way too early to be here. I can't believe I started working on this problem before my first cup. But now, I have ten minutes to wait for this build to compile the code into runnable software. Plenty of time to fill the carafe and get a coffee drip going. Enough time to consider what music I will select to keep me inspired and moving. Maybe today it will be the new System of a Down CD because, well, I'm just in that head-banging mood. If I was more at peace, I might put on Pavarotti. Anything to help my concentration.

I think about how the rest of this difficult day will go. There's the 10 a.m. management meeting that I loathe attending, and I ponder how to best prepare for it. What should I report to the other leaders? Should I start with the root cause of the last issue—like I really have a freaking clue what it is—or should I focus on the status of this crazy important, better-hit-the-deadline, our-business-depends-on-it project that we're all working on?

What will the managers ask me this time? Probably the same things they always do. They almost always start by asking, “How’s it going?” Then they get bolder. “When will the application be completed? Does it run fast enough? Do we have to buy more hardware? Did you hire another developer yet?”

I feel like the lead physician at a hospital that’s in the middle of experimental surgery. No one quite understands what’s going on, but they know the procedure is important. From their vantage point, if the tech works, then they can go back to their day-to-day business. So, the last thing I want is to make them nervous. That’ll disrupt the meeting and get them solutioning. If they panic, they’ll run around the office like chickens with their heads cut off, barking orders at everyone. That’s the last thing I want.

This is how leadership meetings run every week, and as the coffee drips, I contemplate what happens to me during them. Why is it that everyone’s eyes are always on me? And why is it that we’re always talking about the underlying technology? Why aren’t we talking about what we’re selling, what customers we’re winning, and why we’re losing business? Why aren’t we going over the financials to be ready for the next board meeting? What the hell are we getting out of all the money we’re spending on marketing?

Most important, how do I answer questions that appease my colleagues? I can’t shut down their questions, or they’ll claim that I am not a team player and am being technically incomprehensible. But if I give them too much information, they’ll dig deeper into the technical details. Their objective is to deflect the conversation off their areas of responsibility and onto another business area. Tech has a target on its back, and me being the youngest in the room makes that target sizeable and easy to hit.

Jessie is sure to zing me on something. She asks for twelve things. I commit to seven and deliver nine. Nope, that’s not fucking good enough. She must deal with the salespeople who want the missing

three that she should have said no to in the first place. She has incredible skills at navigating these meetings and sits next to the most influential leaders. She only shares the good news, and with such elaborate flare, it gets everyone raving. It's not just what she asks; it's when and how she goes about it. She picks the perfect time and gets everyone's attention. Her question missile is laser-guided and almost always aims at me, and I stand no chance. Especially when there are operational issues.

I'm too young, inexperienced, and not very confident. The 27-year-old chief technology officer (CTO) working at a pre-internet-bubble startup is easy pickings when the business isn't moving fast enough. And when the company is doing well, and we're signing up new customers, well then I'm the last in line to get a shitty bonus. Cheers for Sales! Marketing are heroes!

And where is Tech?

Where are the high-fives for the developers, testers, and systems engineers for a job well done? How come no one applauds the product managers for picking the right features or cheers the data analysts who quickly produced the new reporting dashboard?

I get it. The way they see it, there's always something broken, something slow, or something new that at least one executive team member expects. After a release, it always feels like we're out of the kitchen and into the fire.

And then that's when I hear that happy, whirling, swishing sound, bringing me back to the here and now.

Fabulous, the coffee is ready, and I pour myself a mug. Black. Just like in engineering school when we were studying for midterms. Some of those classes are still useful. The statistics, the physics, and the signal processing classes, but there was no way that the antenna theory class would ever be valuable. Why the hell was that a required class? Waste of my damn time. A communications, writing, or marketing class would have been far more useful.

I walk back to my desk, thinking about some of the lessons learned from my grad school days through today: the mistakes I made in pursuing overly elegant technology solutions, debating what should go into minimally viable products (MVPs), the collaboration skills needed to lead agile teams, and how to understand the business through its data. These are lessons people driving transformation need to know, learn, and adapt to their circumstances. And maybe, one day, I will share them with my teams and teach other leaders.

I look at my screen; thank goodness the build is done. Whenever I make a code change, I have to build the software to test it. It's one of those things Jessie and the others don't understand about developing software. I don't just code. I also have to deal with all the infrastructure that makes software run. It's the late 1990s, and as the internet is starting to boom, this means managing the complexities of running a data center, adding servers, and scaling networks. It requires the webserver software and the integration necessary to run the website. And on this day, it means packaging the software to integrate, deploy, and run within the server. This all takes time away from developing the bells and whistles that Jessie and others have on their mile-long list of requirements.

Today I'm trying to debug this nasty defect. How is it that two users can log into the same account, one being the real account owner and the second one getting access illegitimately? It's not their fault, and I've already determined that this isn't someone trying to hack into our login system. This is a software defect, and I have to figure out how two different people using two separate computers and browsers can access the same account.

I think about the problem from a technical perspective. We use cookies to remember the user, and that's the first thing the application checks to match a user to their account. Cookies. Who the hell came up with that term anyway?

I'm trying to focus on the problem, but thinking of the meaning of browser cookies brings me back to another time and place.

Six months ago, I was sitting in one of my first board meetings at the Conde Nast building in New York City. It's brand-new on 42nd Street in the heart of Manhattan, and an architectural standout, especially for Times Square. Just hours earlier, we meet Craig for breakfast in their expansive cafeteria that could live up to *Gourmet* magazine's high standards. Craig is an important board member and a member of "the family" that controls one of the largest media empires. I'm with our chief executive officer (CEO) previewing the primary news and decision points, and we hope Craig will help steer the discussions.

After breakfast, we ride the elevator to the top floor and start our first board meeting after the new round of investment. I'm sitting at the biggest freaking table I have ever seen. There's no way to reach across this table to shake someone's hand, and the room reminds me of the boardroom scene in the movie *Gung Ho*. In the movie, Michael Keaton's character asks his Japanese hosts, "Did you decorate this place yourself?" It's meant to be an ice-breaker, but it doesn't connect with the board and winds up being a disastrous way to start the presentation. I fully expect Brian, our CEO, who is sitting a few seats over from me, to pull out his death stare if I say the wrong thing.

We're working with hundreds of media websites. They publish newspapers, and we help bring their content and data to life on the World Wide Web. It's 1999, and the boom of the internet is in high gear—with nearly 250 million people logging onto it.<sup>1</sup> Napster is growing in popularity, eBay has gone public, and Craigslist incorporates as a for-profit business.

Our board, owners of thousands of newspapers, are antsy but not worried. They have been around a long time and weathered many

storms. This internet thing is just another phase for them. Remember CompuServe and Prodigy? Blips in technology history. People won't abandon their local newspapers and the trust built up from reading the paper daily and weekly over generations.

So they thought.

Daniel, our ad sales leader, is explaining his new product concept, describing how we're going to put ads across all the newspaper websites. Everyone on the board is looking at him like he's Marie Curie trying to explain how she discovered polonium and radium.

The obvious question should have been, "How are you going to get those ads on *our* websites?" But no, one of our more technical board members is one step ahead and asks, "How are you going to track impressions and click-throughs across all of our websites?" To which Daniel replies, "Oh, we'll cookie the user."

That draws a moment of silence, and then Craig, who is usually quiet during these meetings, lifts his head from looking at the print-out of the board presentation and asks, "What's a cookie?"

Instinctively, they all know to look at me. It's the executive radar that pops up when someone talks tech jargon at a business meeting.

I should point out that if I, as the chief technology officer (CTO), had used some form of tech terminology, then their radar would have triggered a devastating verbal missile attack. No one wants the CTO to go deep into technobabble or use overly complex buzzwords that would never appear in a business magazine. If I do such a thing, Brian would wait until after the meeting and give me more than an earful for dropping jargon and making the technology sound unnecessarily complicated.

Even worse is when a board member raises the issue with the CEO about having a CTO who can't talk "the language of the business." One missed word can launch an entire cycle of having to defend yourself from the barrage of MBA-fired warheads who want to

monopolize the jargon and make sure it's their language that dominates the conversation.

Okay, so all eyes are on me. Now, if I were a little more experienced, then I probably wouldn't have been panic-stricken inside. My first "real" board meeting, and I'm asked an off-script question. And not just any question, but a somewhat technical one that is hard to answer without knowing how web browsers and web servers work. Eyes begin to look in my direction as I catch a glaze from Brian sending me a clear message.

He is saying, "This one is on you. And you better not fuck it up."

So, should I respond with the technically correct answer? "A cookie is a key value that a developer can store in the web browser's cache by utilizing the set-cookie header and then retrieving it on subsequent page views. Since web browsers are stateless clients that interface with a web server, setting a cookie is an easy way to retain the state from one page view to another and is commonly used to store user and session identifiers. The browser is smart enough to send cookies only from the domain of the web page the user is viewing."

Yeah, that would have been the right technical answer. In fact, I think the technology team back at the office would have been proud of how I explained a technical concept in such efficient language. But if I answered the question that way to the board of directors, I would be shown the virtual elevator down to the CTO morgue. That is where geeks with ties go when they can't explain technical concepts in simple language.

Here's how I actually answer the question: "A cookie is an identifier. We can track the ad across multiple websites by setting this identifier and having it sent back to our website even if the ad is on your newspaper's website."

I get a look back from Craig. It probably lasts for only three seconds, but it feels like an eternity. In the corner of my eye, I can

see Brian's "oh, shit" face, fearing that we're going to take the meeting off the rails. Two days earlier, when we rehearsed the presentation at the senior leadership team meeting, he kept repeating to us, "We need to get these guys on board," and yes, the board was all men. And then he reminded us how dumbed-down we must make everything.

Craig opens his mouth and looks like he is getting ready to fire off another question. But his head turns, and he looks back over at Daniel to ask, "How do we make money doing this?" It's the right question and why I respect Craig.

Most important, I pass my first test.

And this is how I know the coffee I'm drinking today hasn't set in. I'm moving from one thought to the next. This problem is hard enough, and I need a clue before we meet this morning.

Two users. Two browsers. One cookie ID. How is that possible? How can a user get someone else's cookie ID in their own web browser? I'm not a security expert, and there are some fears that hackers may have infiltrated the website, especially after multiple users report this issue. And those users, Joe and Barbara, are not happy. In the early days of the consumer internet, most web surfers are unaware of privacy and security concerns when registering on websites. All Joe knows is he came back to his newspaper's website, and now the site thinks he is Barbara.

It was a crazy business idea back then, but not stupid enough. We are a Software-as-a-Service (SaaS) company before the invention of that acronym. Back then, the industry called it an Application Service Provider or ASP.

Our mission is to help newspapers go from print to digital, from paper to screens, from scanning paper to searching. We aim to help them save the stronghold newspapers have on classified advertisements. For the first time, newspapers face formidable competition from Craigslist, eBay, Monster.com, Cars.com, and other websites

trying to chip away at their multi-billion-dollar business model. Newspapers can build a website, but they can't easily construct a search engine. Some can send emails to their subscribers, but they can't engineer an intelligent "AdHound" that creates personalized emails based on the cars, jobs, or homes people seek.

So, we build and host it for them. One multi-tenant web architecture hosts newspaper classified ads from thousands of newspapers. It is advanced for that time—a C++ application running off a multi-threaded web server with a Network Service Access Point Identifier (NSAPI) interface. We develop a natural language processing (NLP) engine to pull search terms from classified ads. It's helpful for real estate ads: We can find "BR" and then look one word earlier to see that it is a three-bedroom apartment. For car ads, we can locate Mustang in the ad's text and infer it is a Ford. It is a rudimentary natural language processor that leverages a word tokenizer, a dictionary of terms, and a rules engine. Back then, few people are trying to do this kind of knowledge extraction. *Artificial intelligence* is decades away from being a hyped buzzword that generates multi-million-dollar venture capital funding.

We feed the metadata in key-value pairs along with the text to a search index from AltaVista, and that was the back end to our website—all hosted on DEC AlphaServer 4100s. Of course, you can do key-value pair searching today with Lucene, Solr, and MongoDB, but this was all innovation back then. Most other CTOs at the time were still querying databases with SQL and keyword searching using *like* clauses.

We develop a proprietary tagging and configuration language to produce hundreds of newspaper sites, all with different designs and content configurations. Web page templates are preprocessed server-side and merged with content delivered from the search engine. The application then applies configuration settings specific to each newspaper before sending a completed web page back to the user.

An AdHound is just a simple agent. The user searches and registers the query with a name, and then AdHound emails them new results every day. If you are looking for a red 1969 Mustang under \$10,000, you can save this query and only get results when AdHound finds matches.

Only somehow, Joe Catlover and Barbara Kleinersha have two registrations with the same cookie ID. This means one of two things. One possibility is that someone went into the database and changed the cookie ID for one of these users. While this was indeed possible, it would fail to explain how that cookie ID ends up in their browsers.

So, I pursue the second possibility. Could the system generate the same cookie ID for two different users?

The application that runs user registration uses Common Gateway Interface (CGI), a protocol that allows the webserver to execute a program, pass parameters to it, capture the results, and shut the program down. It isn't a technically efficient process by any means. Still, it is early in the internet days and an easy way to plug proprietary applications into web servers without coding complicated integrations.

The cookie ID it generates aggregates several parameters, including a code for the website the user is on and a time signature. That's all the programmer had instrumented in this "unique" ID. So, for two users to have the same identifier, they would have to register simultaneously on the same newspaper site. The server measures time in microseconds, and it seems like a longshot that two users on the same website would hit "submit" at the same time within 0.000001 seconds of each other.

I keep digging into the detailed manuals on C++, the standard template library, and DEC UNIX to make sure my assumptions are correct.

And there it is. Holy crap, this can't be right. It turns out that when the server is under heavy load, it doesn't update its clock every

microsecond. What? Seriously? The mighty DEC AlphaServer doesn't have enough juice to update its clock? I had to simulate this and see if it was possible. So, I bombard my test web server with thousands of user registration requests and inspect the test database to see what unique registration IDs exist.

```
Select site_id, cookie_id, count(*) as count from t_userRegistrations group  
by site_id, cookie_id having count > 1;
```

The database spits out:

```
news_nyy, 1098165, nyy_102102787, 2  
news_njy, 1098213, njy_102102992, 2
```

The query results show two examples, one from news\_nyy and the other from news\_njy, that have duplicate IDs created: two for nyy\_102102787 and two for njy\_102102992.

And there it is. It is possible. Crap, a small oversight by the developer, and it is possible to compromise a user's account.

The only good news here is that we find the cause and have a way to fix it relatively quickly. Since the user registration application is a CGI, we can append a UNIX process ID and a random number to the cookies generated. We then encrypt the whole string, ensuring that the system will never generate two IDs again, even if the system is overloaded. Nowadays, this technique is known as salting.

We deploy this fix and then null the cookie IDs for the handful of affected users. The change creates a small annoyance, forcing them to log in and reset their password, resulting in the system generating a new and more secure cookie ID.

Coding is relatively easy compared to debugging. Debugging is always challenging but even more complex when issues arise from the impacts of load or volumes of disparate data. Developers quickly code and complete releases when trying to get functionality out the door. They can easily miss key details, especially when they make

what appear to be reasonable assumptions about how the underlying systems perform. In this case, the developer assumed that requesting a time signature to the nearest microsecond would be sufficient to generate a unique ID.

I know for a fact that this developer didn't have the means to test this under load. Load testing tools didn't really exist then, and we launched the application when there was relatively low traffic across our websites. It would have taken much forward-thinking to decide to load test and then identify the duplicate identifiers.

Being a CTO in a growing startup keeps you in the proverbial weeds where you have to dive into code, configuration, and data to understand how things work. It's a swamp, actually. And every time you feel ready to climb out to see the forest from the trees, something crazy and unexpected happens, and today's fire sucks you back into rapid problem-solving. But knowing your way through the weeds is a critical skill for a Digital Trailblazer.

I'm always going in and out of the swamp. I want to see the mess, then get a drone's view of the forest. If you can fly the drone in the sky, operate the airboat across the swamp, and pull out the machete to slash through the weeds, then that's the starting point of some of the transformational tools you need today. But it's just the start. Trust me.

## **Transformative Leadership Requires Working In and Out of the Weeds**

It's roughly ten years later, and I'm at a construction industry conference moderating a panel of chief information officers (CIOs). I am the CIO of a media and analytics company that serves commercial

general and subcontractors and building product manufacturers. I join the leadership team with Engineering News Record (ENR) editors who established this technology conference.

ENR is a premier media institution that published its first issue in April 1917.<sup>2</sup> Every year, this magazine hosts a black-tie awards dinner for leaders in the industry, and executives work hard to be on its top lists or win the Award of Excellence for a top construction project.

This conference is all about the technology construction companies use in the field and back office. There's a mix of emerging use cases showcasing augmented reality/virtual reality (AR/VR) technologies to aid architects during building, lidar for capturing accurate measurements during construction, and other field technologies to improve the productivity and safety of construction workers.

The conference committee agrees to let me moderate a panel of premier construction industry CIOs. Even though I was on the committee that founded this event, it did require some convincing that CIOs belonged on the agenda and that I should interview them. I then take one step further and invite these CIOs to a post-conference council to discuss technical challenges and opportunities. Never mind that I've never formed an executive council and that this is only the second time that I am moderating a panel at a big conference with hundreds of people attending. No worries. I'll get through it.

The panel kicks off fine. I ask questions about CIO priorities. What risks keep them up at night? What new technologies excite them? How do they work with their construction project managers on selecting technologies? It's all going well. Some ups and downs, but the audience looks engaged.

I then ask what I thought should be a simple question.

"So, when do you guys roll up the sleeves and get into the weeds working with project managers, forepersons, or your technology staff?"

Sadly, I call them *guys*. There are no women on the panel, and there are very few at the conference. Construction and technology are two industries poorly represented by minorities and women. I must change my language and thinking—a mental note I make to myself after recognizing my blunder.

My question is open-ended, and they could go anywhere with their responses. It was even on the list of questions I sent them to prepare for the conference, so it shouldn't have been a surprise.

But the room falls silent—dead silence for one, two, three seconds. Five seconds and it feels like an eternity. Since this conference, I've probably moderated hundreds of panels and never had a record-scratching moment like this one.

After a brief panic inside my gut, I pick my dignity off the floor and say, "Okay, no one's taking that one. Let's move on," and I get through the rest of the panel without complications.

But that moment sits with me for the rest of that conference and to this day.

How is it that these CIOs have no answer to this question? If I were on that panel, I'd have dozens of stories to tell. My challenge would be sticking to just one.

At the council meeting later that afternoon, I talk to the CIOs about their priorities. It's my second council meeting, and I come prepared with an agenda and a program. I have about ten CIOs attending and want to make sure their time is well spent. But as one CIO told me, "Isaac, you're working too hard. All we want to do is vent with our peers, without vendors around, and without repercussions."

So, I toss a softball question and let them loose. Some are upgrading their networks to handle their growing data needs better. A few are moving to the latest version of Microsoft Windows and Office while trying to get Microsoft SharePoint to work better for them. One is struggling to provide their legal teams with years of data for

a lawsuit in discovery. I hear about one group experimenting with different security technologies that track who is coming on and off their job sites. We invest an hour hearing the complaints about difficult enterprise resource planning (ERP) upgrades, rogue IT in their organizations, and whether the chief financial officer (CFO) will increase their budgets.

It's 2010ish. No one is talking about *digital transformation* yet. It's not a trending keyword on Google, and the media isn't writing about it yet. Many of us have read Clayton Christensen's books on innovation, but much of the disruption is happening in select industries like media and with startups biting at the fringes. It's not happening in many other industries, including construction. Yet.

I'm irked by the discussion with the CIOs, and I'm still bothered by no one answering my question on getting in the weeds.

Years later, I put it all together.

At that time, these CIOs were only thinking about transitioning their organizations. Disruption hadn't hit the construction industry, and digital transformation was in its infancy. These CIOs were stepping up their games but hadn't gotten into the weeds on transforming—how their companies should use new technologies to drive business impacts, culture changes, growth, efficiencies, and quality differentiators. As one construction CIO recently told me, “All of us spent significant time in the job trailers talking to the workers and working hard to grow technology acceptance and then adoption. At the time, we were not relevant to boards, to Ops, only accounting.”

Because transformation requires picking optimal moments to get your hands dirty, learn how the business runs today at detailed levels, and ask challenging questions, you want subject matter experts to see new opportunities to improve customer experiences or simplify operations and guide them on potential solutions.

What's happening in every industry will change who's on top and who's falling off the disruption cliff. Who are the new competitors

likely to steal customers by offering winning experiences and capabilities? Where is the business using data to competitive advantage, and what data quality issues interfere with data-driven decision-making? Who should organizations partner with to deliver new customer-facing, digitally enabled capabilities? What customers are instrumental and can be early adopters to new business models? What other industries, companies, and leaders can people in your organizations learn from, get inspired by, and adopt their best practices?

These are outside-in examples of getting into the weeds and showcasing what's happening in the outside world that should influence the business's strategic direction. Sure, you can read about it or bring an analyst to the office to provide a digestible synopsis. But true intelligence and insights are best developed by getting out of the office and experiencing things for yourself.

The inside-out view is not about understanding code or improving your department's options. Ideally, you want to speak directly to customers and end-users to understand pain points, identify opportunities to deliver increased value, or discover ways to improve their experiences. You'll want to select stakeholders who are ready to partner on transforming the status quo and challenge how things work today. That partnership gives you the license to learn today's operations and pinpoint inefficient workflows, new ways to collaborate, and opportunities to hyperautomate—ways to use a combination of automation, apps, and machine learning to reduce manual tasks and enhance people's decision-making performance.

Getting into the *business* weeds can be very challenging for operational leaders. Sure, these technologists dive into the networks, systems, and applications hoping that business tools are operational, performing well, and secure. Some are risk-averse CIOs who track metrics like application uptime and the meantime to recover from production issues. These CIOs are more likely to be measured on

cost savings and other operational metrics and are highly unlikely to think about customers, marketing, sales, and revenue growth.

There are analogies with product managers who focus on making incremental improvements to existing product lines. They manage mile-long backlogs of defects and small enhancements, mostly from internal stakeholders or the loudest customers complaining about their specific needs.

I also see similarities with data officers and the myriad data and analytic job roles that report to them. Yes, it's important to address data quality and iteratively improve existing reporting dashboards. It's also incredibly important to address gaps in data policies and security. But today's data and analytics officers must go beyond these basics. They should seek new datasets, consider partnerships, and review new platforms to increase the organization's intelligent capabilities.

Today's product, technology, data, and analytics leaders must aim much higher than the table stakes. Never heard that term? In casinos, it's the minimal bet to sit at the table and play a hand.

Playing to table stakes includes operational leaders who target minimal improvements in key performance indicators and are not taking many innovation bets. Table stake product managers act as administrative funnels and manage all the stakeholders' wish lists without looking at outside and inside opportunities to leapfrog the competition and drive business outcomes. The data officers chasing table stakes may be moving the data governance needle, but they aren't challenging the status quo or bringing in game-changing capabilities. The technology officers keep the lights on and manage risks but do not evaluate emerging technologies' threats and opportunities.

Table stake leaders don't challenge the organizational sacred cows that ground progress toward new business models. They expect that it's another leader's responsibility to drive culture change.

You have the skills to get into the weeds, and you likely have the adventurous spirit to thrash into the unknown. But it can be hard taking your first steps into the murky swamps of your industry or business.

Here's how I go hiking through the swamp and weeds, without a trail paving the path, but with a compass and other tools developed through past experiences.

I claw back to my experiences as a developer, engineer, data analyst, product owner, and architect as a means to get into the messy details buried in architecture, code, system, configuration, data models, and analytics. It's a skill all hands-on specialists must apply differently at the leadership level.

I use it to talk shop with the developers and better understand what's working well and where they face obstacles. I draw my intuitions on selecting technologies based on my skills to rapidly assess the differentiating qualities, ease of use, and scalability. And suppose there's a glaring implementation gap with no one stepping up to fill it. In that case, I just might roll up the sleeves and implement something in a low-code technology, data prep, or data visualization platform.

I talk to customers and go on sales calls. I attend the conferences that customers attend. I ask my peers in other industries what they are investing in and where they experience success.

I get my hands dirty with data. If no dashboard answers my questions, then I quickly develop one. I'll discuss proofs of concept (POCs) with a half dozen data scientist groups to learn and then place a bet to see what some can deliver.

I'll talk to technology partners, agencies, and freelancers. I'll write requests for proposals and sign on to do a handful of pilot projects.

At the same time, getting into the weeds means knowing which areas of the swamp to step into and which ones to avoid. You need

to move fast and develop a plan to get out. Otherwise, you may just sink into the mud.

You don't want to be using these skills to stay in your comfort zone in operations, technology, product, or data. Worse case, this will mark you as a micromanager or someone not letting go of your previous responsibilities. You must venture into the uncharted swamps where you have little direct experience. However, you can still leverage your skills of asking questions, being detail-oriented, listening attentively, learning from diverse people, and making sense of conflicting data.

To grow beyond being a transitional leader and become transformational, you must know when to dive into the technical and operational details, and when to pull up, see the forest from the trees, and propose a vision and plan. You'll need to balance your time between building relationships with colleagues, exploring outside-in perspectives to strengthen these muscles, and selectively choosing when and how to use the inside-out muscles you've developed through your career.

And there are dangers in the swamp. Beyond just sinking, leadership snakes are lurking, ready to take a bite out of you for venturing into their territories. In fertile areas of the business, thick groves of institutional vines will slow down progress in efforts to maintain the status quo.

Then there are many existential threats, largely out of your control, that will either slow you down or require hard pivots. Pandemics, floods, and attacks are examples of events that drive externally driven transformations. It's not about having a plan for these situations. It's about developing the organizational structure, practices, and technologies to help organizations react, respond, and pivot appropriately.

Becoming a Digital Trailblazer and leading transformation is an incredible journey, but it's not for the faint of heart and mind. One

day you'll feel like you are on top of the world, and the next, your colleagues will be coming at you with torches and pitchforks.

I hope you'll be more prepared through my stories. Enjoy the journey.

## **Digital Trailblazer Lessons**

### ***Transitioning to a Leadership Mindset***

As you grow beyond your role from an individual contributor to leadership, you'll face new experiences and situations that require you to build on your knowledge and adapt to new environments.

Debugging applications, chasing customer issues, or fixing complex data problems are things you've done before. But solving them as a leader will require different approaches when you won't have all the background knowledge or the time to learn how everything works. You'll need to be leading and mentoring your staff while influencing colleagues. You may spend some time in the weeds to better understand a problem or opportunity, but don't get stuck in the mud trying to figure everything out. Delegate work to your teams so that they learn firsthand the problems and what types of solutions to recommend.

Other times, you'll be working in areas where you have no or limited experience and must figure out your role and voice in navigating them. Be brave, and don't let mistakes set you back. All leaders have their first experiences presenting to the board of directors, handling difficult situations while delivering presentations, and going on customer calls. Learn from them!

Enjoy the journey. Undoubtedly you will find mentors and remember learning experiences that will help develop your management and leadership skills. I hope the stories and lessons in this book also help mentor you as you blaze trails.

Here are five key takeaways from this chapter:

- 1. Reflect on the skills that got you here won't get you there.** You don't become a leader by perfecting the technical, hands-on skills in areas where you are already an expert or just by building acumen on how to manage these functions. Your technical chops help you get by early in your career, but you will be a lot less hands-on as your responsibility increases. Your knowledge and experience will be critical when directing teams on problem-solving, using data to guide their investigations, and prioritizing where to focus their efforts. But leadership requires contributing to areas that you haven't worked in or directly managed by building skills, believing in yourself, and accepting that you'll make some mistakes. The primary skills you should develop include building trust with people who lead functional groups and asking questions to learn and challenge today's operating practices.
- 2. Avoid solving technical problems with proprietary solutions.** Your prime objective as a digital leader is to solve customer and business problems. That's true for your team as well, and the more they develop proprietary solutions to technical challenges, the more you are creating a technical debt that will require ongoing support. When you have technical problems that aren't business capabilities, it's often better to seek out nonproprietary solutions through commercial technologies, open source platforms, partners, SaaS, web services, and low-code approaches. This is an especially important lesson for product managers who are quick to seek customized methods or data scientists who hard-code algorithms rather than leverage existing models. When you see developers

solving technical challenges, make sure you take the time to ask them if they've first explored third-party solutions.

- 3. Recognize that communicating with the board and executives is a skill that requires practice.** There are some resounding do's and don'ts when communicating with board members and executive teammates. Do answer questions first, then follow up afterward with supporting facts, data, and essential details. Do come prepared to meetings with clear-cut summaries on the status of strategic initiatives because you don't know what board members will ask you. Don't go into technical details, and don't ramble when answering questions. Don't embarrass other executives or surprise them by sharing too much information. Share key decision points and supporting facts before the meeting. Never throw the CEO or your boss under the bus at a board meeting for not supporting your ideas or priorities. More on how to handle working with the strategic leadership team in Chapter 9, but the board meeting is not the place to air out these differences.

These are some of the basics. The art of navigating these meetings requires understanding their protocols and the personal motivations behind participants. Some call this politics, and to some extent, it is. But being successful in these environments is a skill you'll need if you want to grow as a leader.

- 4. Understand the table stakes, then set broader goals.** Go ask your colleagues what they expect of you and your teams. When you do, I sincerely doubt that they'll tell you to keep doing what you're already doing, only better. Iteratively improving your operational metrics, increasing productivity, and addressing issues faster are all yesterday's table stakes. Today, business leaders expect innovation and new ways to

leverage technology and data, and more importantly, they seek Digital Trailblazers who will drive business outcomes. To get started, develop relationships with stakeholders who are likely supporters by scheduling one-on-one brainstorming sessions, sharing articles of mutual interest, or going to lunch when that's an option. Then partner on broader goals such as developing new digital experiences, integrating multiple data sources into real-time analytic systems, or creating a citizen development center of excellence.

**5. Step out of your comfort zone and broaden your perspective by seeking outside-in learning opportunities.**

When a new opportunity or issue presents itself, the typical organizational response, especially from experienced leaders, is through their lenses of known knowns. As a Digital Trailblazer, your goal should be to share ideas and insights from outside your organization. You can learn from others by participating in social media, attending events outside of your industry, or learning from potential technology vendors and partners. When possible, visit customers and observe how they use your products and services. But make sure your learning is two-way and share your insights publicly with peers outside of your organization. You won't get important feedback if you keep your ideas to yourself, and Digital Trailblazers, including introverts, must extend their comfort zones to validate insights and ideas.



If you would like more specifics on these lessons learned and best practices, please visit <https://www.staricio.com/digital-trailblazer/chapter-1>.

## Notes

1. Internetworldstats.com, 2021, "Internet Growth Statistics," <https://www.internetworldstats.com/emarketing.htm>.
2. Andrew G. Wright, Scott Lewis, Debra K. Rubin, and Scott Blair, "A Century of the ENR Banner," *Engineering News-Record*, April 13, 2017, <https://www.enr.com/articles/41841-a-century-of-the-enr-banner>.