

# 1

## The Product Development Challenge

Rather than looking at concepts in the abstract, let's get down and dirty. I want to share a story with you about how product development goes wrong. In uncovering these traps, we'll then be better set up to talk about fixes.

What follows, then, is a kind of case study that highlights problems. And it's not based on a single company. Instead, the particulars are drawn from dozens of real life situations, which I've disguised.

### Key Players

You and I work for Amalgamated Mechanical Incorporated. The company is hot about creating a new medical procedure robot. We're on the reliability team.

It's critical that our robot get out there quickly, because our #1 competitor is on the verge of introducing a similar product. To get a jump on them, our product needs to hit the market in 10 months' time.

According to the program plan, the Accelerated Life Testing (ALT) for the robot's arm should start next week. As far as we know, we expect to receive the arm samples in three months. If the ALT testing begins in three months, however, there's little chance we'll have accurate predictions on how and when the product may wear once it hits the market.

For all practical purposes, even if we provide that test-based prediction for the arm's wear-out failure rate a few months before release, it will be of little value. There's no alarm big enough to sound that would delay release, based on premature wear-out. Even if we discovered that the product wore out, not in the promised five years of normal use but in one solitary year, the leaders would still release on schedule. (After all, they'd reason, we have to beat the competition to market.)

It's been said before by our VP: "We'll release it now and get a fix out there quickly. We already have a punch list for version 2.0."

In many ways, the project has been designed to fail. For me, it feels like trying to stop a freight train that's built up a head of steam. Stepping in front of it creates a mess, and the train still pulls into the station on time.

Half the program's reliability testing was to provide input for program decisions:

- “How confident are we that the arm will reach its life-and-reliability goals?”
- “What's the robot's end-of-life failure mode?”
- “Should we create a preventative maintenance cycle or shorten the robot's promised life to customers?”

This is critical information in a product development program and we don't have it yet.

Why does it always go this way? It's actually made me think about changing disciplines to something other than “reliability engineering”. Why have a career focus that doesn't improve products and is often just a check-the-box nicety?

The product did indeed release on time. The reliability growth (RG) testing showed low statistical confidence in the goal reliability. This is the most critical assembly and we don't believe it will work as it should, and we're going to release it anyway. This is crazy! The ALT testing was never finished, because there was a design change and we didn't receive new arms to test. So we don't know when it'll wear out. How scary is that?

These arms could start failing in large numbers in the customers' hands, because of a predictable wear-out failure mode. Statistically, the majority of the population will fail at this point, with a nice bell curve outlining the full population. More than half of the Failure Mode and Effects Analysis (FMEA) high-risk actions weren't addressed. Some of these actions had “user harm” in the severity ranking.

We released on time. About four months after product release, the field failure rate began to spike. Two specific failure types were dominant. The linear *X* axis bearing and a plunger that penetrates a consumable. Both see high cycles in use, and both were known to be high risk due to changes in the most recent design.

These spiking field failures were the main topic in every Friday steering meeting and hallway conversation. If someone had information, the CEO wanted to hear it. If there were no updates on the root causes and fixes, he yelled about wanting to know what everyone was doing all day.

For a reliability manager, the entire process was depressing. No real value was delivered from our team's work. As a matter of fact, we were usually seen as a nuisance – almost as if we were an outside regulatory organization, but without authority. Something akin to a kid on a Big Wheels pulling over highway motorists and issuing traffic tickets in crayon.

Now that there are high field-failure rates, people are murmuring, “Well, there's no single person to blame for all these failures. . . but. . . aren't you the reliability team? Why did you let this happen?”

As I said, I find myself thinking of abandoning the discipline I love, because it can feel pointless.

Taking a step back, I thought about the program's reliability experience from all the other roles involved. The project managers received bonuses for releasing the product on time. The R&D engineers were promoted and assigned to top-notch programs because of the features they developed. This was all celebrated at a fancy hotel with an offsite party and a band.

OK, that all happened at release. But what happened when the robotic arm assembly began to fail early in life? Surely, that was the moment of reckoning, wasn't it?

This is the team's experience when the failure rate spiked four months after release: they were called together as a "Tiger Team." This means they were borrowed from their new programs, because they were supposedly the only people who could save us: our "heroes!"

During this recovery phase, the Tiger Team got regular facetime with the CEO. Facetime with the CEO is a key element in someday climbing into upper management. Many of us would take a CEO face-to-face over a 20% raise. For the Tiger Team, this experience was pure gold.

Then, when the field issues were solved and the company was saved, there was a celebration with a festive banner and a big ice-cream cake.

As the legendary management leader Edwards Deming said: "One gets a good rating for fighting a fire. The result is visible; can be quantified. If you do it right the first time, you are invisible. You satisfied the requirements. That is your job. Mess it up, and correct it later, you become a hero" [1].

So, in summary, they were rewarded for casting reliability aside to enable meeting only one goal associated with their role: time to market, new features, or cost point. They were then rewarded again for fixing the field problems they themselves created.

Remarkably, the team was doubly incentivized to deliver a product that was unreliable. How could this be? Why did the program's executives engineer things this way? After all, it hurt them most.

But if I'm making it seem like everyone involved in this failed program was rewarded, I'm confusing the issue. People were indeed punished. Who? Those who really wanted to create a product that was reliable. What happens to those people? The next story shows that they have one of two paths.

## Follow the Carrot or Get Out of the Race

The leadership of a large 90-year-old company asked me to evaluate their culture. In my report, I included a story. It had passion, reward, and, most importantly, punishment – all the elements of a Greek tragedy. So I wrote the story and got a response better than I'd ever hoped for. Here's the story:

I began my investigation with the question: "Why is reliability missing from most engineers' work, even though we promote it so assertively as a core value?"

Walking around the company's halls, I saw posters that underlined how seriously they took quality and reliability. These posters bore slogans like "Our customers count on us for reliable products" and "Our product reliability is YOUR legacy."

The hierarchy even jammed the word "reliability" as many times as they could into all their speeches. For instance, at the annual R&D off-site the CEO delivered an 11-minute talk, and used "reliability" eight times. That's almost one "reliability" per minute. I'd already been there long enough to understand the hypocrisy. That's why I was counting.

The company liked to hand out reliability awards. But these awards were largely empty. They rarely included bonus money or anything that resembled actual career growth.

It was easy to see management's true motivation. The late quality guru Philip Crosby said, "An organization is a reflection of its management team." There's no hiding what the

boss truly values. Where this is most obvious is with things like sizable bonuses or meaningful promotions.

At this company, I saw engineers and developers rewarded when their product was on budget and on schedule. There's nothing wrong with handing out rewards for these accomplishments. Unfortunately, these were the only things for which the engineers and developers were rewarded. More notable accomplishments – like excelling at the full set of program objectives – were ignored.

To the team, upper management's reward-incentivized message was clear: "This is what we really value." The next level of management down had no choice, then, but to prioritize these same on-budget, on-schedule metrics above all others.

The written report I would later send to the organization's hierarchy had only two characters, Engineer #1 and Engineer #2. (Those weren't their real names. If they had been, it would have shown some amazing foresight by their parents.) These characters were a composite of the actual engineers on that team.

They differed from each other in a very important way. While they were both good engineers:

- Engineer #1 focused on budget and schedule, because she wanted the bonus and was ambitious enough to crave a promotion. She was attuned to what her management team valued, so she behaved accordingly.
- Engineer #2 followed her innate passion and decided to invest in reliability. Through careful analysis, she learned that her product had a reliability issue that could be resolved in four weeks. Of course, that would push back the product's release date by a month and cost \$150,000 but she knew the fix would later save the organization \$2 million.

From Engineer #2's perspective, when all was said and done her proposed fix would save \$1.85 million, and she should have been rewarded for her thoughtful solution. But you already know where I'm going with this, and so did the leadership team reviewing her suggestion.

The problem with big companies, especially those with impersonal management teams sitting far away from engineering, is that the boss's directive of on budget and on schedule seems frighteningly rigid by the time it reaches the worker bees. The human resources department already has performance appraisal processes drawn up, based on budget and schedule. And because the directive came from so high up, there's not much scope for change at the lower levels.

So Engineer #1, who played ball with the leadership when the program was originally scoped, wins the bonus and a good performance appraisal. And they set an example for the rest of the workforce. They've become a symbol of good behavior.

What also happened, however, is that leadership told Engineer #2 (and everyone else watching) that this reliability stuff is unnecessary, even if it saves the company \$1.85 million.

Engineer #2 will now disappear. If she's really good and prioritizes her professional happiness, she will simply leave and go to another company. If her value system doesn't align with the leadership's value system, and she has options elsewhere (and all the very best engineers do), she'll just disappear. That, or she'll simply turn into Engineer #1.

Anyway, that's what I put into my report (hey, they had paid me up front). These leaders, I wrote, had created a culture "allergic to reliability." If I were forced to be an employee at this company, I'd drop product reliability goals as a way of ensuring I received a paycheck.

The leadership had filled the company with people like Engineer #1, and all the opportunities to save \$1.85 million were never taken.

Engineer #1 got promoted, but profits decreased and times grew tougher. And the organization is now conditioned to fixate on cost and schedule. It reinforces things that don't work even more. The culture gets worse. The talent leaves. Frustration reigns supreme.

## It's Not That I'm Lazy, It's That I Just Don't Care

When the program initially started, the company's leadership talked about reliability as if it were life or death. They demanded "zero failures!" because "today, online reviews appear in seconds, and the market knows whether a product is good or bad instantly."

In the product-specification document, we set specific reliability goals. After hearing the leaders' demands, you'd think any reliability request would be answered with a blank check and permission to jump to the head of the resource allocation line.

Did everyone know they were going to abandon all of these high-minded reliability ideals? Boy, I hope not. That level of deception would be heartbreaking. I'm just going to assume they were as surprised as me when the reliability focus got derailed. But, really, let's be honest: I'm not that surprised.

Having said that, I still feel hopeful that the idea of reliability can take root within an organization. The reason I feel optimistic is that I believe everyone wants to build a product they can be proud of. No one wants to create a product that fails. If we take action that creates a product that's only moderately reliable, it must be due to pressure or objectives that aren't thought through. When teams have had an opportunity to create products with genuinely high reliability, while still achieving the other product goals, I'm certain you'll find nothing but enthusiasm.

That all workers want to create high-quality products was first discussed in Douglas McGregor's 1960 book *The Human Side of Enterprise* [2].

McGregor challenged the universally accepted principle that "workers don't want to work," which he called "Theory X." McGregor said that all workers want to do work that is meaningful and high quality. He called this "Theory Y."

We see symptoms that appear to support the lazy "Theory X" worker. But what commonly occurs are workers being pushed into work that they don't feel is meaningful. They're disconnected from the work's value, so of course there's little incentive to do a good job. This, combined with their not having input in how their role is executed, turns them into cogs in a wheel who only react to short-term positive or negative incentivization.

I believe reliability suffers from this same phenomena. When development teams are asked to do reliability activities in the product development process, they're disconnected from the actual value delivered. So they don't understand how reliability helps them achieve the results they want to achieve.

Sure, it's not as easy as showing them a diagram where A connects to B. It requires that they participate in the selection of the tool to be used. They have to believe that what they're doing is going to have an impact. If they understand how the tools work and select one they believe will have an impact, then they'll have created "a way forward." That's very motivating.

The three steps to making this connection between what they want to accomplish and the methods to get there are to educate, assign goals, and transfer control. The Design for

Reliability (DfR) requires that the full team understand reliability. Reliability can't be designed in if the designers don't understand the practices to measure and improve reliability in all phases of a product's life.

Previous to more expansive reliability education programs, the only reliability input that occurred in the program was from reliability engineers. This will never work if the objective is to have reliability designed in.

If you have kids then you know this dance. When you ask a five-year-old if they have to go to the bathroom, because you see them dancing a bit, they say, "No." This is even though they are about to lose control of their bladder. Sixty seconds later they will ask you to take them to the bathroom. OK, I get it: now it's your idea. Humans want to do things on their own terms from the moment they are born.

One of many issues with reliability engineers solely driving reliability is that the reliability engineer has to stand over the designer's shoulder and force themselves on the design process. Have you ever willingly accepted the input of someone standing over you? Me neither. The other option (usually what occurs) is to have the designers finish the design and then hand it to reliability to "make it reliable." There was an era that produced some evidence that this approach worked, and it's still referred to by old crusty reliability engineers today.

When computing electronics was new, in the 1950s and 1960s, there were a set of common best practices to making it reliable. If a reliability engineer took a new electrical design and then put these practices to the design, it would in fact be more reliable. But this is more akin to a copy editor cleaning up grammar and sentence structure for an author. That interaction isn't going to work with the development of new technology today. Not if you are going to be competitive.

Both of these strategies line up with the management style that was prevalent before McGregor shared his Theory Y. In this application, Theory X would be "designers don't care about reliability." That's not true, they were just being pushed hard on the other product and program factors. It was a management and cultural error that the designers' natural desire to make a high-quality product was not a part of their work. They also knew they had someone to throw it over the wall to make it reliable – so why worry about it? That was someone else's job.

By providing the designers with an understanding of the reliability toolkit, they can design in reliability themselves. It is not just the tools that increase reliability; it's also the tools that measure reliability. It's crazy to attempt reliability work without measuring product reliability during the process. How could you possibly know if you're not making a product reliable enough or, equally as bad, making it too reliable? Too reliable hurts, because the product will cost more than necessary and get to market slower.

The control part of this equation can also be tricky. By putting control of the product's reliability in the design team's hands, we're saying that the designers get to make decisions regarding release date, cost point, and product features. With that much control, if the team feels uncomfortable releasing a product on time, then the company would have to listen. Commonly, the company's concern is that putting this control in the engineer's hands will result in products that are never released. I've heard many managers say, "Engineers will try improving a design for ever. It's their baby and it can always be a better design."

To address this quandary, both the design team and the leaders need to understand the toolkit that reliability has to offer. By measuring performance and then comparing those

measurements to the goals, decisions can be made quantitatively, not by feel. This is critical. The goals come from an agreed-upon balance of time to market, cost point, features, and reliability. It shouldn't be revisited every time resource requests are made. If that's the case, the goal never really was a goal.

When the leaders first set the goals, they likely put a lot of thought into those goals. They weren't random, based on a whim. You'll note I said the leaders have to understand the measurement tools as well. For two reasons. The first is, if they don't know a tool or measurement parameter exists, they can't request it. The second is, they have to trust the outputs from the tool. If the result is not understood, the result is questionable in their mind. Understanding where the value came from and being able to ask questions about how the method was executed are big factors in the credibility of the result in the recipient's mind.

So with the measurement being credible, and the goal being agreed upon by all parties, there is much less to argue about: "At Alpha, our goal is to demonstrate a five-year product life. The recently completed life test shows that up to 15% of a large population of our product may not achieve that life."

Unless we're going to argue about the original goal or the measured results, there isn't much else to do but address the design issues.

The title of this section "It's not that I'm lazy, it's that I just don't care" is a quote from a movie. Not just any movie but the cult classic for anyone who has ever worked in an office, 1999 movie *Office Space*.

The main character works as a software engineer at a company where he is just a cog in a mindless process. He becomes hypnotized and begins to just tell everyone how he honestly feels about everything. Coincidentally, the next day the leadership has invited consultants to come in and find ways to improve the organization and process. Basically, see who they can fire and other ways to make cuts to improve the bottom line. Peter answers their questions with complete honesty. This is the full conversation between the hypnotized employee Peter and the consultant, Bob, interviewing him about his role. You can't tell me you haven't had a moment in your career that this moment does not capture perfectly. I'm grinning just knowing what I am about to write next, from memory.

PETER GIBBONS (SOFTWARE ENGINEER): The thing is, Bob, it's not that I'm lazy, it's that I just don't care.

BOB PORTER (CONSULTANT): Don't. . . don't care?

PETER GIBBONS: It's a problem of motivation, all right? Now if I work my ass off and Initech ships a few extra units, I don't see another dime; so where's the motivation? And here's something else, Bob: I have eight different bosses right now.

BOB SLYDELL: I beg your pardon?

PETER GIBBONS: Eight bosses.

BOB SLYDELL: Eight?

PETER GIBBONS: Eight, Bob. So that means that when I make a mistake I have eight different people coming by to tell me about it. That's my only real motivation: not to be hassled; that, and the fear of losing my job. But you know, Bob, that will only make someone work just hard enough not to get fired."

You're smiling right now, I know it.

## Product-specification Profiles

When the program begins, we formulate the product objectives, which sound like this: “faster robot X axis motion of 10 mm second<sup>-1</sup>, and a retail cost point of \$23 500, and a first-year reliability of 99.9%.”

Then, we feed these objectives into what ultimately becomes what’s called a “product-specification profile.” Until the product hits the market, this specification profile is supposed to be the project’s touchstone. It drives all the decisions.

But how were the product objectives that went into the profile formulated in the first place?

Why, those came from the organization’s business goals!

Really, everything we do at work is driven by the business goals. It might be one degree or five degrees of separation, but at the end of the day all the activity has to line up with the organization’s business goals.

Your product’s reliability, then, needs to be an adhered-to part of your business goals.

Do you see what I mean? Reliability is a part of your brand. What happens to the sales of all your product lines if your brand’s reputation becomes tarnished?

It’s not that every product needs to be 100% reliable, or that reliability is the sole factor in developing your product. You have to be clear about what you’ve decided, and stay true to it. After all, in many markets you’re only as good as your last product.

As I’ve mentioned, a culture built on reliability is one where business and reliability connect. People often think of reliability as being part of the design process only. That’s not true. It’s only half of the story.

Executives at all levels understand how the reliability discipline affects market share. If the new coffee machine their company designed fails, they suffer personally. They’re the ones held accountable.

Let’s say this is the first coffee machine to market with Wi-Fi connectability, that it can email you when your cup of coffee is ready. This should mean it will grab significant market share and make the brand a household name.

But none of this happens, because it immediately has issues when customers try to get that desperately needed caffeine fix.

Uncaffeinated people who are denied coffee are quick to leave bad online reviews. Who cares about a coffee machine that can email you if the machine’s water pump breaks when it is three days old? “Thanks for nothing.”

So how is it even possible that a smart leader would blindly cut back on reliability initiatives when the schedule is tight or a new feature isn’t ready for prime time? The reason is simple: when it comes to reliability, the direct effects on product or program performance investment aren’t apparent – in the short term.

Businesses work on a quarterly system and individuals move to new programs and roles in short timeframes. Very few long-lead actions make it to the top of the daily action list. Most companies can’t keep this high-level perspective when the day’s urgent matters arise. Many that I have seen which hold this far-sighted perspective are privately held by a founder who was an engineer.

This lines up with McGregor’s Theory Y, in that people want to do good work. But this type of situation occurs when “good work” is measured only in short, three-month increments.

An engineer who is also left holding the business accountability long term is the person who we should look to for best practices when long-term growth is the objective. This is one of the positions that seem to find a great balance for all the program and product factors. Let me repeat it to be clear.

An engineer with an accountability for long-term business success. I have proof of this. Some of the most successful companies in the world were founded by engineers who kept the business private. Some of the tools I propose aim to put the leaders in this same mindset.

Executives and owners of companies make decisions based on a significant amount of input from their council, their mid/high level management. If the executives and mid-level leadership's goals are not aligned then this council becomes muddled. This becomes compounded when the executives evaluate the mid-level leaders on the short-term delivery of time to market, cost point, and new features creation. The mid-level managers can't equally support reliability when times get tight and the results of an investment are far down the road.

So how do we fix this? We look at how executives make decisions. The first question is how do they get input/information for their decisions? Many executives work in a dashboard summary manner. The controls and readouts are very similar company to company, which is why it is easy for executives to move between organizations and industries. Reliability simply needs to find a way to format what is important so it can be included on this dashboard. And once created, this connection to reliability functions for executives will be similar to the other critical inputs executives receive day to day. Total Quality Management, The Toyota Way, and Lean Six Sigma have all had a significant impact on how we do business and product programs. They all have found ways to get on that dashboard.

## Product Drivers

It was my first house, and it was a project like many first houses. Originally constructed in 1855, it had seen many caretakers. I was confident I would just be another name on the list and the house would be standing long after I was gone.

Being a first-time homeowner, I needed to begin that tried-and-true ritual of depositing my paychecks directly with Home Depot. Within a short time, I'd be meandering the aisles of the warehouse-size hardware store, with other homeowners, looking confused under the store's fluorescent lights, trying to figure out solutions to the problems that brought us here with little knowledge of how to solve them.

Before long, the cashiers knew me by name and welcomed me the same way the bar staff welcome the barfly Norm in the classic sitcom *Cheers*.

"Adam! How are the kids? Were you able to finish the ceiling?"

In a large hardware store like that, there are two types of people walking around: homeowners and contractors. From a product development perspective, these are two very different types of customers with very different needs.

As a "mortgage poor" person who was working on my own house, not because it was fun but because I was trying to save money, I wanted to buy tools that'd get the job done at the

lowest possible price. Why lowest? Most of the tools I'd buy would be used a few times, and then would sit in my garage. My project was severely underfunded.

The other type of buyer, the contractor, was running a crew on a job site, where the highest operational expense was hourly wage. Having a tool fail would be devastating to the bottom line. It would leave workers sitting around. What that contractor needed to know was that the tool was going to work and not leave them paying idle workers.

So the power drill I'd buy and the power drill the contractor would buy had little in common other than both tools turned drill bits.

A product is defined by many factors. I find that the four primary ones are time to market, features, cost point, and reliability.

Imagine what I was thinking as I stood in front of the power drill display. There were at least a dozen models, and I had a clear ranking in mind as to what was most important to me: #1 was cost, #2 was cost, #3 was reliability.

That contractor standing next to me with the pro-wrestler-size biceps was evaluating these drills using totally different requirements: #1 was reliability, #2 was "This better not break!", #3 was "That quick-release chuck could save some time." Reliability was #1. Time-saving features were #2. Cost point was the last thing on his mind, because the expense of the tool failing was far more costly than the savings he'd enjoy in buying the lowest-priced drill.

Cost was my priority, because every dollar I spent on the drill hurt my bank account. Reliability was at the bottom of my priority list because if the drill broke I could work on something else and replace it the next day or even borrow a neighbor's. Reliability also fell low on my priority list because my duty cycle was a fraction of the contractor's. A design with low reliability used on a low duty cycle produces a similar failure rate as a design with high reliability and a high duty cycle.

I was doing many roles in my "job," and used the drill a total of one to two hours a day, every couple of days. The contractor may have a guy doing drywall construction all day – every day. In one month his drill might be used for 160 hours, while mine was used for 12. My project would be completed in three months, while the contractor works consecutive jobs throughout the year. That takes my yearly usage to 36 hours. The contractor's yearly usage could be over 2000 hours. That's why I can have a cheap drill and experience the same or even better reliability than the contractor purchasing the top-end model.

My whole point here is a drill is a drill, except that it's not. Each drill is designed differently, based on the factors that are most important to its particular market. Success isn't making the drill perfectly reliable. Success is making the right drill for the right customer, even if that means manufacturing a particular drill that has lower reliability.

The degree of a product's reliability should always be a conscious decision.

## Bounding Factors

The design parameters I just talked about I call "Bounding factors." In a sense, they're the measurable factors that guide both your product and your program.

What are the Bounding factors for a product? A design feature or a cost point.

What are the Bounding factors for a program? Time to market or development cost.

The Bounding factors for a product and a program have to be related, because there are tradeoff decisions that affect both.

Here are some common Bounding factors we see in programs today:

- New technology/features
- Cost point
- Time to market
- Reliability
- Serviceability
- Manufacturability with a contract manufacturer
- Marketability.

The four that share the primary balance in most programs are those from the drill example: cost point, time to market, features, and reliability.

It's not possible to turn all of these up to a level of 10 (11, if you're a Spinal Tap fan), because they compete for resources. Extremely high reliability is not in line with quick time to market or ultralow product-development cost. New cutting-edge technology is not in line with low cost point or quick time to market, either. For those combinations, there's a give and take to get to a budget of time and money that works.

When this negotiating between schedule and investigating a reliability issue is done mid-program without program tools, it is often reduced to who makes the best argument at the time. I can think back to many programs where the decision was in favor of the person who was most in favor with the decision maker. "Who do I know best?" "Who do I trust?"

I don't believe the decision maker did this because of favorites. This would be insane, because if the program fails they fail. They do this because they don't have factual information at their fingertips. Without any quantitative information any individual is left with making a decision on counsel and trust.

Unless we can incorporate tools like the Bounding methodologies that will be shared in this book there is no way to expect a leader to make decisions on anything other than listening to trusted counsel.

The Bounding methodology derived its name from the base principle that each factor should "bound" resource and schedule changes to ensure no specific factor is compromised beyond the original product specifications' margins. What is being achieved is that factors are consistently steering the program toward the goals set in the product-specification document.

## Reliability Discipline

*Why* is reliability done in product development programs? This is where we need to start. Without a "why" clearly defined we don't have a foundation to work on. The value in this early discussion is to understand this. Why is reliability test and analysis done? It is done to measure and improve reliability of the product. We usually stop there.

But why are we measuring and improving? This is the question that is left unanswered and we are without a way to make the critical decisions during a program. A great quote on “why” is by a sci-fi character named Merovingian in the movie *The Matrix*:

“Causality. There is no escape from it, we are forever slaves to it. Our only hope, our only peace is to understand it, to understand the why. ‘Why’ is what separates us from them, you from me. ‘Why’ is the only real social power, without it you are powerless.”

Let’s change a few words in that quote to make it specific to our mission:

“Causality. There is no escape from it, we are forever slaves to it. Our only hope, our only chance at program success is to understand it, to understand the why. ‘Why’ is what separates the good program choices from the bad program choices. ‘Why’ is the force that directs product programs, without it we are powerless. Making choices that are reactive, based on fear and blind trust are the ‘why’s’ that make us powerless.”

OK, that got intense. Let’s go to the deepest “why” we can identify first.

The reliability discipline has progressed through several phases of maturity. It was originally approached as a method of identifying areas of risk and used “over design” as a mitigation. Look at any tool made from 1000 BCE to 1950 CE. I have many of my grandfather’s tools, and they still work fine. More than fine, his tin snips could cut 0.5 in. (13 mm) steel cable. The wrench could be used as a hammer, and the drill could be pulled from the rubble of a house fire and used in the reconstruction. Take a look at the photos of his tools next to my modern-day equivalents (Figure 1.1).

But those product designs from many decades ago would be unlikely to survive in today’s market. Looking at the modern-day equivalents next to them the differences are evident even to a non-tool enthusiast.

With rapidly advancing technologies affecting all of our lives, weight and cost quickly became critical to maintaining a competitive product in the market. The reliability engineering discipline took a more formal shape at this point due to a need to have a counterforce that ensured cost and weight didn’t take the product design to a point of being, well, unreliable. The forces for product balance emerged: lightweight, low cost to manufacture, and at a low cost point are now balanced against reliability and developing new technology.

Through these needs, the methods of reliability analysis, test, and design techniques took shape. It was the military that led this initiative initially, simply because it is far worse to have a piece of equipment that will save your life not work than a vacuum or toaster quit at an inconvenient time.

Military customers still often measure reliability in terms of “risk of lost life.” Even when a piece of equipment that is not intended directly for assault or defense (rifles, missiles, shielding, etc.) fails, it may increase the risk of loss of life. Something as simple as a surveillance camera on a tower in a hostile area can result in death if it fails and so needs to be replaced. A soldier doing maintenance on the top of a tower is a target. This places a whole new complexion on designers and engineers talking about a 30% failure rate of a camera in



**Figure 1.1** Grandfather's tools vs my tools.

a meeting back home, when that failure means not just that the camera stops working but that this, in turn, could lead to the death of a soldier.

I won't ever forget that sobering moment at that conference table when a DARPA general made that statement about a project I was working on. The clarity of what was at stake if I didn't succeed at the role I was brought there to do was chilling. When I had walked into the meeting I thought the consequence of a higher failure rate for the security camera was warranty expense and future lost sales. People's lives were at stake if a circuit board joint cracked.

The period for reliability from the 1940s to the 1970s was heavily analytical in nature. It encompassed tools like reliability predictions and specialized tests (Figure 1.2). The predictions were based on historical failure rates of individual components. The specialized testing aimed to predict the wear-out of specific failure modes or to identify the margin of failure of a primary stress. In the 1980s and early 1990s, more advanced techniques of testing like the Highly Accelerated Life Test (HALT) and Accelerated Life Test (ALT) became prevalent. These methods permitted more specific statements of reliability prediction or design improvement input to be made in very compressed timeframes early in the design process.

The 1990s to 2010 was very much characterized by the DfR initiative. This principle is that reliability is "designed in" not "tested in." A fundamental shift with DfR is that reliability practice is intertwined with the full team and design process from start to finish. This was a big shift from the mindset that mechanical engineers do mechanical design,

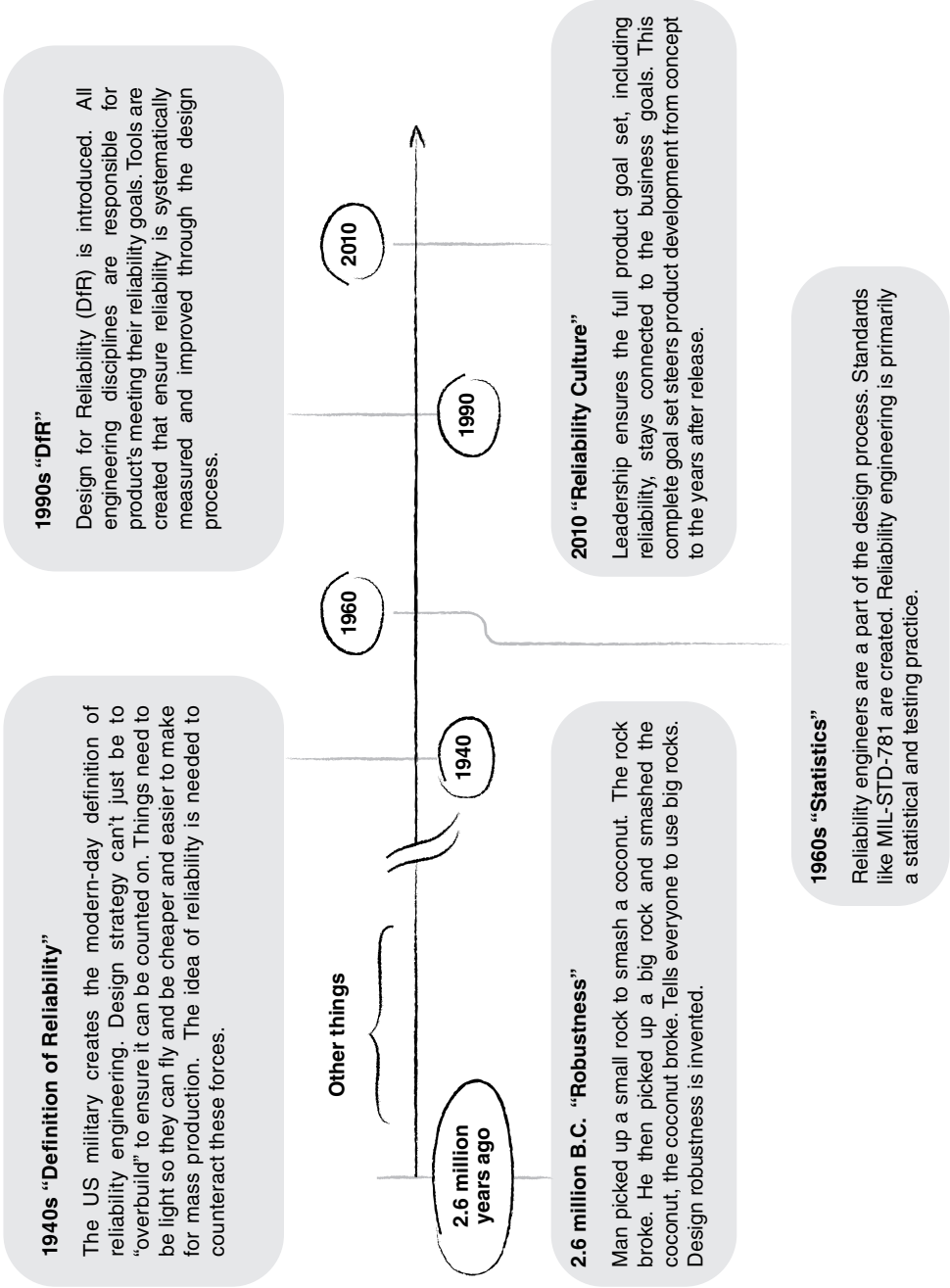


Figure 1.2 Reliability timeline.

electrical engineers do electrical design, and reliability engineers then make it reliable when they are done.

The next phase, today's phase of advancement for the reliability discipline, is reliability culture. This will be the connection of reliability tools techniques and philosophy to the highest levels of business and market objectives in conjunction with DfR. Companies that embrace this next phase of reliability evolution will quickly emerge as the leaders in their markets.

The bottom line in most of our businesses is dollars. The metric that will be used to measure an organization's level of reliability cultural maturity will be dollars. The return on investment of applied reliability tools will be measured in dollars. Companies that do not embrace a culture of product reliability will be ill equipped to compete with those that do. – just as it became impossible to compete without a Total Quality Management process two decades ago. We are on the cusp of placing reliability at the heart not just of the engineering process but of corporate culture. For this to happen, it will be necessary for business leaders to create the correct organizational dynamics and align reliability objectives with a business's financial goals.

Know your target. Make goals and make compromises. Don't commit to high reliability without selecting the sacrifices. Something has to give, be it schedule or new technology development or cost point or very high product development cost. There are no worse words than a leader saying "and it must be highly reliable" or "it will never fail" without discussing the cost of pursuing that reliability goal.

You have to know beforehand whether you are willing to trade reliability for growth of technology or time to market, and by how much. The Mars rover took many years and billions of dollars to create. From a technology standpoint the Mars rover is the equivalent of a high-school robotics science project. It has off the small digital cameras and a small DC motor like in radio controlled cars driving little wheels. There are servomotor-driven arms based on decades-old technology. I build stuff like this in my workshop with my kids.

But what was special about it is it could never fail. It truly was a "This design cannot fail or we are wasting billions of dollars." That was a quantifiable statement. The many years and billions of dollars spent to accomplish that perfect reliability was the cost. It would be a mistake to create your commercial or consumer product with that type of reliability goal. Nobody wants your perfectly reliable flip phone 10 years after the market has moved on to smartphones. So let's figure out what goals you should have for each program and how to correctly structure a program to accomplish them.

## References

1. Edwards Deming, W. (2000). *Out of the Crisis*. Cambridge, MA: MIT Press.
2. McGregor, D. (1960). *The Human Side of Enterprise*. New York: McGraw-Hill.

