

An isometric illustration of a multi-level office building. The building is constructed from grey brick-like blocks. Inside, there are several floors with people working at desks, some with computers and monitors. A unicorn is visible on one of the middle floors. Blue lines, resembling data or network connections, flow through the building. A blue banner with the word 'EXCERPT' is in the top right corner.

EXCERPT

The Unicorn Project

A Novel about Developers,
Digital Disruption, and
Thriving in the Age of Data

Gene Kim

Author of *The Phoenix Project*

EXCERPT

The Unicorn Project

A Novel about Developers,
Digital Disruption, and
Thriving in the Age of Data

Gene Kim

IT Revolution
Portland, OR



Copyright © 2019 by Gene Kim

All rights reserved, for information about permission to reproduce selections from this book, write to Permissions, IT Revolution Press, LLC, 25 NW 23rd Pl, Suite 6314, Portland, OR 97210

First Edition

Printed in the United States of America

24 23 22 21 20 19

1 2 3 4 5 6 7 8 9 10

Book design by Devon Smith Creative

Cover image by eBoy

Cover design by Joy Stauber, Stauber Brand Studio

Author Photograph by Anna Mayer Photography

Library of Congress Catalog-in-Publication Data

Available Upon Request

ISBN: 978-1942788768

eBook ISBN: 978-1942788775

Kindle ISBN: 978-1942788782

Web PDF ISBN: 978-1942788799

Publisher's note: Portions of this book were based on talks and articles by different thinkers and industry leaders with their permission and are included in the references of this book.

For information about special discounts for bulk purchases or for information on booking authors for an event, please visit our website at www.ITRevolution.com.

THE UNICORN PROJECT

Note to the Reader

The Unicorn Project takes place “in the present day,” and is a companion novel to *The Phoenix Project* (which also takes place “in the present day”). The events from both novels take place concurrently, although certain situational elements of *The Unicorn Project* have been altered to account for changes in our industry.

While both books are about Parts Unlimited, *The Unicorn Project* was written to be a standalone book—there is absolutely no need to read or re-read *The Phoenix Project* first! (You may recognize some characters from *The Phoenix Project*—but then again, don’t worry if you don’t!)

Because the two books were written six years apart, there may be some suspension of disbelief required—for example, everyone’s awareness of the Retail Apocalypse and the use of ride-sharing (Uber, Lyft) is much higher now than it was when *The Phoenix Project* was written.

For those who need some concrete waypoints, the characters who appeared in *The Phoenix Project* are indicated as such in the cast of characters, and there is a rough timeline of the two books provided as an endnote (beware, there may be spoilers!).

PROLOGUE

• *Tuesday, September 2*

From: Steve Masters (CEO, Parts Unlimited)
To: All Parts Unlimited Employees
Cc: Dick Landry (CFO, Parts Unlimited),
Laura Beck (VP Human Resources)
Date: 11:50 p.m., September 2
Subject: Payroll Failure

To fellow employees of Parts Unlimited,

Early this morning, several thousand timecards were corrupted due to a technical failure, mostly affecting employees and contractors in our manufacturing facilities and retail stores.

My goal is to ensure that everyone gets paid as soon as possible. Anyone who was underpaid should get a check in the next twenty-four hours.

As CEO, my job is to ensure that we fulfill our obligations to our employees, who make the daily work of this organization possible. Without you, we would not be able to serve our customers, who depend on us to keep their cars running to conduct their daily lives.

I apologize to you and everyone who depends on you for the problems and inconveniences this payroll issue causes. I commit to you that we will provide all necessary help, including communicating with any bill collectors, banks, etc.

At the bottom of this email you will find a list of Frequently Asked Questions from HR and Business Operations. If you are not getting help quickly enough, please email me or call me on my office phone anytime.

In the meantime, our top priority is to understand what factors led to this failure, and we will do whatever it takes to make sure that it doesn't happen again.

Steve Masters,
CEO, Parts Unlimited

From: Chris Allers (VP Dev, Parts Unlimited)
To: All IT Employees
Cc: Bill Palmer (VP IT Ops), Steve Masters (CEO),
Dick Landry (CFO, Parts Unlimited)
Date: 12:30 a.m., September 3
Subject: Corrective actions for the payroll failure

All—

Because of the high-profile nature of the payroll outage, we have conducted a thorough root-cause analysis. We have concluded that it was due to both human error and a technology failure. We have taken decisive actions to ensure that it will not happen again. The person responsible has been reassigned to a role where they can no longer affect production outcomes.

If you have any questions, please email me.

—Chris

Elkhart Grove Herald Times

Parts Unlimited Flubs Paychecks, Local Union Leader Calls Failure ‘Unconscionable’

Automotive parts supplier Parts Unlimited has failed to issue correct paychecks to some of its hourly factory workers, and others haven’t received any compensation for their work, according to a Parts Unlimited internal memo. The company denies that the issue is connected to cash flow problems and instead attributes the error to a payroll system failure.

The once high-flying \$4 billion company has been plagued by flagging revenue and growing losses in recent quarters. These financial woes, which some blame on a failure of upper management, have led to rampant job insecurity among local workers struggling to support their families.

According to the memo, whatever the cause of the payroll failure, employees might have to wait days or weeks to be compensated.

“This is just the latest in a long string of management execution missteps by the company in recent years,” according to Nestor Meyers Chief Industry Analyst Kelly Lawrence.

Parts Unlimited CFO Dick Landry did not return phone calls from the Herald Times requesting comment on the payroll issue, accounting errors and questions of managerial competency.

In a statement issued on behalf of Parts Unlimited, Landry expressed regret at the “glitch,” and vowed that the mistake would not be repeated. The Herald Times will continue to post updates as the story progresses.

PART ONE

September 3–September 18

CHAPTER 1

• Wednesday, September 3

“You’re doing what?” Maxine blurts out, staring in disbelief at Chris, VP of R&D at Parts Unlimited.

Chris smiles weakly from behind his desk. *Even he realizes how absurd he sounds*, Maxine thinks.

“Maxine, I’m really sorry about this. I know it’s a terrible way to come back from vacation, but this payroll outage created an incredible crap storm. The CEO and CFO wanted heads to roll. We agonized about this for days, but I think we came up with a pretty good solution . . . after all, no one is getting fired.”

Maxine slaps the printed copy of his email onto his desk. “You say right here that it was caused by ‘human error and a technology failure.’ And now you say that I’m the ‘human error’? After all that time we spent together deciding how to resolve that compliance finding, you’re placing all the blame on me? What sort of bullshit is this?” She glares at him furiously.

“I know, I know . . . It’s not right,” Chris says, squirming under Maxine’s intense gaze. “Everyone here values your incredible skills and talents and your fantastic contributions to the company over the last eight years—no one actually believes it was your fault. But the payroll issue was front-page news! Dick had to give a quote to keep the unions from filing a grievance! Given all that, I felt like we came up with the best solution in a pretty awful situation.”

“So you blame the person who was on vacation because that person couldn’t defend herself?” Maxine says in disgust. “That’s really admirable, Chris. Which leadership book did you get that from?”

“Come on, Max, you know I’m your biggest fan and biggest defender. In fact, take this as a huge compliment—you have one of the most stellar reputations of anyone in IT,” Chris says.

Blaming someone for a payroll outage is a strange way of appreciating someone, she thinks.

He continues, “*Everyone* knows that this isn’t actually your fault. Just think of this as a vacation—you can work on anything you want, and you won’t have any real responsibilities if you don’t want.”

Maxine is about to respond when she thinks about what she just heard. “Wait, treat exactly *what* like a vacation, Chris?”

“Uh...” Chris stammers, buckling under her stare. Maxine let’s him squirm. As a woman in what remains a largely male dominated profession, she knows her directness might be contributing to Chris’ discomfort, but she will always stand up for herself.

“...I promised Steve and Dick that I’d put you in a role where you couldn’t make any production changes anymore,” Chris says, squirming. “So, uh, effective immediately, you’re moving from the manufacturing plant ERP systems to help with documentation for the Phoenix Project...”

“You’re sending me to...” Maxine can’t breathe. She can’t believe what she’s hearing.

“Look, Max, all you have to do is lie low for four months. Then you can come back and have your choice of any project you want to work on, okay?” Chris says. Smiling weakly, he adds, “See, like a vacation, right?”

“Oh, my God...” she says, finding her voice again. “You’re sending me to the Phoenix Project?!” she nearly yells. Maxine immediately kicks herself for this brief moment of weakness. She takes a deep breath, adjusts her blazer, and pulls herself together.

“This is bullshit, Chris, and you know it!” she says right into his face, pointing her finger at him.

Maxine’s mind races, thinking about what she knows about the Phoenix Project. None of it is good. For years, it’s been the company death-march project, having ensnared hundreds of developers, achieving unprecedented levels of notoriety. Maxine is pretty sure that the reason nothing is going right is simply because they’re not doing anything right.

Despite the Phoenix Project’s obvious failures, it keeps going. With the rise of e-commerce and the decline of physical stores, everyone knows something has to be done to ensure that Parts Unlimited stays relevant in the increasingly digital age.

Parts Unlimited is still one of the largest players in the industry, with nearly a thousand stores across the nation. But there are times when Maxine wonders how the company will fare beyond its hundredth anniversary, which wasn’t that long ago.

The Phoenix Project is supposed to be the solution, the shining hope that will lead the company into the future. It's now three years late (and counting) and \$20 million has disappeared, with nothing to show for it except developer suffering. It stinks of impending failure, which will have grave implications for the company.

"You're going to take one of your best people and exile her to the Phoenix Project because you need a fall guy for the payroll outage?" Maxine says, her frustration boiling over. "This is not a compliment—this is the best way that you can say, 'Screw you, Maxine!' Hell, there's probably nothing in Phoenix that is even worth documenting! Unless it's to document incompetence? This is like labeling all the deck chairs on the *Titanic*. Have I said that this is bullshit already, Chris?"

"I'm sorry, Maxine," Chris says, throwing up his hands. "It's the best I could do for you. Like I said, no one is actually blaming you. Just do your time and it'll all go back to normal soon enough."

Maxine sits, closes her eyes, takes a deep breath, and steeples her hands in front of her, trying to think.

"Okay, okay..." she says. "You need a fall guy. I get it. I can take the blame for this whole fiasco. That's cool, that's cool... that's how business is done at times, right? No hard feelings. Just... put me to work in the cafeteria or in vendor management. I don't care. *Anywhere* but the Phoenix Project."

Listening to herself, Maxine's aware that in less than two minutes she's moved from denial to anger and is now in full-blown bargaining mode. She's pretty sure she's missed a step in the Kübler-Ross grief cycle, but at the moment she can't think of which one.

"Chris," she continues. "I have nothing against documentation. Everyone deserves good documentation. But there are tons of places that need documentation *way* more than Phoenix does. Let me go make a bigger impact somewhere else. Just give me an hour or two to come up with some ideas."

"Look, Maxine. I hired you eight years ago because of your amazing skills and experience. Everyone knows you enable teams to do the impossible with software," Chris says. "That's why I fought for you, and why you've led the software teams that are responsible for all our supply chains and internal manufacturing processes for all twenty-three manufacturing plants. I know how good you are... But, Maxine, I've done everything I can. Unfortunately, the decision has already been made. Just do your time,

don't rock the boat, and come back when everything blows over," he says, looking so remorseful that Maxine actually believes him.

"There are executives being shot left and right, and not just over this fiasco," Chris continues. "The board of directors just stripped Steve Masters of the chairmanship, so now he's just CEO. And both the CIO *and* VP of IT Operations were fired yesterday, no explanations given, so Steve is now acting CIO too. Absolutely *everyone* is worried that there is going to be even more blood in the streets..."

Chris looks to make sure the door is closed and, in a lower voice, says, "And there are rumors of potentially *even bigger and more sweeping changes coming...*"

Chris pauses, as if he might have said too much. He continues, "Look, whenever you're ready, go get yourself set up with Randy, the Phoenix development manager—he's a good guy. Like I said, think of this as a four-month vacation. Seriously, do whatever you think will be helpful. Heck, you don't need to do anything at all. Just keep your head down. Don't rock the boat. And whatever you do, just stay off Steve and Dick's radar. Sound good?"

Maxine squints at Chris as he name-drops Steve Masters and Dick Landry, the CEO and CFO of Parts Unlimited. She sees them every other month during the company Town Halls. How did she go from a two-week vacation seeing the wondrous sights of Kuala Lumpur to having Chris dump all this crap on her?

"Maxine, I'm serious. Just lie low, don't rock the boat, stay clear of outages, and everything will be fine, okay? Just thank your lucky stars you weren't fired for the payroll issue like the two other people were last year," Chris implores.

"Yeah, yeah. Don't rock the boat," she says, standing up. "See you in four months. And *you're welcome* for helping you keep your job. Super classy, Chris."

Chris is actually getting more spineless each year, she thinks, storming out of the room. She considers slamming the door, but instead closes it...decisively. She hears him say, "Please don't rock the boat, Maxine!"

When she's out of sight, she leans against the wall. Tears well up. Suddenly she remembers the missing step in the Kübler-Ross cycle after bargaining: depression.

Maxine slowly makes her way back to her desk. *Her old desk. Where she used to work.*

Maxine can't believe this is happening to her. Trying to counter all the negative self-talk flying through her head, she reminds herself of her qualifications. She knows that for the past twenty-five years, her job has been to bend technology to do her bidding—efficiently, effectively, precisely, with creativity and flair, and most importantly, competence.

She knows she has unmatched real-world experience building systems that run under adverse and even hostile environments. She possesses a fantastic intuition about which technologies are best suited to achieve the mission at hand. She is responsible, meticulous, and careful about her work, and she insists on the same level of excellence and diligence from everyone around her. *After all, dammit, I was one of the most sought after consultants at the top Fortune 50 companies*, Maxine reminds herself.

Maxine stops mid-stride. Even though she is a stickler for details and doing things right, she has learned that mistakes and entropy are a fact of life. She's seen the corrosive effects that a culture of fear creates, where mistakes are routinely punished and scapegoats fired. Punishing failure and "shooting the messenger" only cause people to hide their mistakes, and eventually, all desire to innovate is completely extinguished.

During her consulting days, she could always tell, usually within hours, whether people were afraid to say what they really thought. It drove her crazy when people were careful about how they phrased things, speaking obliquely and going to extreme lengths to avoid using certain *forbidden* words. She hated those engagements and would do everything she could to convince the client to end the project, saving them time, money, and suffering.

She can't believe she's starting to see these red flags at Parts Unlimited.

Maxine thinks, *I expect leaders to buffer their people from all the political and bureaucratic insanity, not throw them into it.*

Only yesterday, she and her family were getting off of a nearly twenty-hour flight back from Kuala Lumpur. When she turned her phone on, it nearly melted from all the incoming messages. While Jake and her two kids went to find food in the airport, she finally got ahold of Chris.

He told her about the payroll failure and filled her in on the mayhem. She listened carefully, but her heart stopped when she heard Chris say "...and we discovered that all the Social Security numbers in the payroll database were corrupted."

She broke out in a cold sweat, her hands tingled, and there was ice in her blood. For what felt like a lifetime, she couldn't breathe. She knew. "It was the tokenization security application, right?"

She cursed loudly. Parents all around herded their young kids away from her on the airport concourse. She heard Chris say, "Yep. And there's going to be hell to pay. Get into the office as soon as you can."

Even now, she's still in awe of the scale of the carnage. Like all engineers, she secretly loves hearing disaster stories... as long as she doesn't have the starring role. "Stupid Chris," she mutters as she thinks about dusting off her résumé, untouched for eight years, and putting out feelers for any job openings.

By the time Maxine reaches her work area, whatever equanimity she had managed to muster is gone. She stops before she walks in. Her armpits are sweaty. She smells them to make sure she doesn't stink of the humiliation she feels. She knows she's being paranoid—she put on so much deodorant this morning her armpits were chalky-white. She was glad she did.

She walks into the work area. Everyone knows she is being reassigned but are trying not to let on. Glenn, who has been her manager for three years, comes up and squeezes her shoulder, a pained expression on his face. He says, "Don't worry, Maxine. You'll be back here before you know it. None of us are happy with the way things went down. A bunch of people wanted to throw you a big party, but I was pretty sure that you wouldn't have wanted to make a big scene," he says.

Maxine says, "Damned right. Thanks, Glenn."

"No problem," he says with a wry smile. "Let me know how I can help, okay?"

With a forced smile, she says, "Come on, it's not like I'm dying or being sent into outer space! I'll be closer to headquarters, which is where all the action is. I'll send updates to all you ignorant villagers who aren't good enough to be in the thick of things!"

"That's the spirit. We'll see you back here in four months if all goes well!" he says, giving her a playful jab. Maxine's brow furrows slightly at the "if all goes well" bit. That was news to her.

As Glenn heads to a meeting, Maxine goes to her desk to start packing up. She picks the most critical things she'll need during her exile: her

carefully configured laptop (she is very picky about keyboards and amount of RAM), pictures of her family, her tablet, and the USB and laptop chargers carefully selected and accumulated over the years, along with the big sign that hangs over them: “DO NOT TOUCH, under penalty of death!”

“Hi, Maxine! Why are you packing up?” she hears someone ask. Looking up, she sees Evelyn, their promising young computer science intern. Maxine recruited her. All summer long, Evelyn has dazzled everyone with how quickly she picks things up. *She’ll have her pick of jobs when she graduates*, Maxine thinks. Which is why all summer long Maxine has been relentlessly selling Parts Unlimited as a great place to work and learn. Which she herself believed, until this morning. *Maybe this isn’t such a great place to work after all.*

“I’ve been temporarily re-assigned to the Phoenix Project,” Maxine says.

“Oh, wow,” Evelyn says. “That’s awful... I’m so sorry!”

You know you’re in real trouble when even the intern feels sorry for you, Maxine thinks.

She leaves the building, carrying her plain cardboard box, alone. She feels like she’s reporting to prison. *Which is basically what the Phoenix Project is*, she tells herself.

It’s a four-mile drive to the corporate headquarter campus. While she drives, she thinks about the pros and cons of staying at the company. Pros: Her husband is a tenured professor, which is why they moved to Elkhart Grove in the first place. Her kids love their schools, friends, and activities.

She loves her work and all the challenges; she loves interacting with the countless and complex business processes that span the entire company—it requires an understanding of the business, incredible problem-solving skills, patience, and the political sophistication to work with sometimes Byzantine and occasionally incomprehensible processes that every large organization seems to have. And the pay and benefits are great.

Cons: The Phoenix Project. Working for Chris. And the feeling that the corporate culture is changing for the worse. *Like how I just got scapegoated for the payroll outage*, she thinks.

Looking around, she sees buildings designed to exude status and success. Parts Unlimited earned that level of prestige by being one of the largest employers in the state, with seven thousand employees. They have

stores in almost every state and millions of loyal customers, although every metric shows those numbers declining.

In the age of Uber and Lyft, the younger generation is more often choosing not to own cars at all, and if they do, they sure don't fix their cars themselves. It doesn't take a strategic genius to realize that the long-term prosperity of the organization requires something new and different.

As she drives deeper into the corporate campus, she can't find Building 5. When she circles around for a third time, she finally sees the sign to the parking lot. Her heart sinks. The building's a dump compared to the others. *It even looks like a prison*, she thinks.

Building 5 used to be a manufacturing plant, just like MRP-8, her "old" building. But where MRP-8 is obviously still the pride of the company, Building 5 is where they dump misbehaving IT people like her and throw away the key.

If the Phoenix Project is the most important and strategic project for the company, don't the teams who work on it deserve a better building? Maxine wonders. But then again, Maxine knows that in most organizations, corporate IT is rarely loved and is often parked in the least attractive properties.

Which is odd. At MRP-8, the ERP technology teams work side-by-side with the plant operations people. They're viewed as partners. They work together, eat together, complain together, and drink together.

On the other hand, corporate IT is usually viewed as ranks of nameless faces whom you call when there's something wrong with your laptop or when you can't print something.

Staring at Building 5, Maxine realizes that as bad as the reputation of the Phoenix Project is, the reality is probably much, much worse.

Everyone tells Maxine that one of her most endearing qualities is her relentless and never-ending optimism. She keeps telling herself that as she walks toward Building 5, carrying the cardboard box full of her belongings.

A bored security guard inspects her badge and recommends she take the elevator, but Maxine chooses to go up the stairs instead. She wishes she had a more cheerful bag to carry all her things in instead of lugging this dumb box around.

As she opens the door, her heart sinks. It's a vast cubicle farm with drab gray partitions separating each work area. The maze of cubicles reminds her of the old computer text game *Zork*—she's already lost in a series of twisty passages, all alike.

It's like all the color has been drained from the building, she thinks. Maxine is reminded of her parents' old color TV set when her brother had fiddled with the brightness, contrast, and color dials to make everything look a sickly gray and green.

On the other hand, Maxine is delighted to see that each desk has two massive LCD screens. She's in the right place—these are developers. The new monitors, open code editors, and the high percentage of people wearing headphones are dead giveaways.

The room is so quiet you could hear a pin drop. It's like a university library. *Or a tomb*, she thinks. It doesn't look like a vibrant space where people work together to solve problems. Creating software should be a collaborative and conversational endeavor—individuals need to interact with each other to create new knowledge and value for the customer.

In the silence, she looks around, feeling even worse about her fate.

"Do you know where I can find Randy?" she asks the person nearest her. He points to the opposite corner of the room without even taking off his headphones.

Walking through the hive of silent cubicles, Maxine sees whiteboards and people huddled in groups, speaking in hushed tones. Along one long wall are enormous Gantt charts easily four feet high and thirty feet across, assembled from what looks like more than forty sheets of paper taped together.

Alongside the Gantt charts are printouts of status reports featuring lots of green, yellow, and red boxes. Standing in front of the charts are people dressed in slacks and collared shirts. Their arms are crossed and they look concerned.

Maxine can almost feel the people mentally trying to compress the bars closer together so that they can hit all those promised dates. *Good luck*, she thinks.

As she walks to the opposite corner where she was told to find Randy, Maxine suddenly smells it: the unmistakable smell of people who have slept in the office. She knows this smell. It's the smell of long hours, inadequate ventilation, and desperation.

In technology, it's almost a cliché. When there's a need to deliver capabilities to the market quickly, to seize a market opportunity, or to catch up with the competition, long hours become endless hours, where it's easier to sleep under your desk than to go home only to come right back. Although long hours are sometimes glorified in popular culture, Maxine views them as a symptom of something going very wrong.

She wonders what is happening: Too many promises to the market? Bad engineering leadership? Bad product leadership? Too much technical debt? Not enough focus on architectures and platforms that enable developers to be productive?

Maxine notices that she is wildly overdressed. She looks down at the suit that she's worn to work for years, realizing that she sticks out like a sore thumb. In this building, T-shirts and shorts far outnumber the collared shirt crowd. And *no one* is wearing a jacket.

Tomorrow, I'm going to leave the jacket at home, she thinks.

She finds Randy in a corner cubicle, typing away and surrounded by huge stacks of paper. Randy is a redhead, wearing the management khaki-land uniform—a collared, striped white shirt and khaki pants. Maxine guesses he's in his late thirties, probably ten years younger than her. Judging from his low body fat, he probably runs every day. But he looks stressed in a way that no amount of running can take away.

He gives her a big smile, standing up to shake her hand. She puts her big cardboard box down and realizes how tired her arms are. As she shakes his hand, he says, "Chris told me about how you ended up here. I'm sorry to hear about all that. But trust me, your reputation precedes you, and we're so excited to have someone of your experience on this team. I know it's not the best use of your skills, but I'll take any help we can get. I think you can make a real difference here."

Maxine forces herself to smile because Randy seems nice enough, even earnest. "Happy to help, Randy. What do you need to get done?" she asks, trying to be equally earnest. She does want to be useful.

"I'm in charge of documentation and builds. In all honesty, things are a mess. We don't have a standard Dev environment that developers can use. It takes months for new developers to do builds on their laptops and

be fully productive. Even our build server is woefully under-documented,” Randy says. “In fact, we’ve had some new contractors on site for weeks, and they can’t even check in code yet. God knows what they’ve actually been doing. We’re still paying them. To do nothing, basically.”

Maxine grimaces. She hates the idea of paying expensive people to just sit around. And these are developers—it deeply offends her sensibilities when willing developers are prevented from contributing.

“Well, I’m happy to help out wherever I can,” she says, surprised at how much she means it. After all, making developers more productive is always super important, even those working on the Phoenix Project in its fiery, meteoric descent.

“Here, I’ll show you where we’ve got you set up,” Randy says.

He leads her past more rows of cubicles, showing her an empty desk, a filing cabinet, and two large monitors connected to a laptop. It’s plainer and smaller than she’d like, she thinks, but it’s fine. Especially since she’ll only be here for a few months. *One way or another, I’ll be out of here soon*, Maxine thinks. *Either my prison term will end or I’ll get another job somewhere else.*

“We got you a standard developer setup, just like any developer who starts at Parts Unlimited,” he says, gesturing at the laptop. “You’ve got your email, network shares, and printers set up with your existing credentials. I’ll send out an introduction email this afternoon. And I’ve assigned Josh to help you get everything set up.”

“That’s great,” Maxine says, smiling. “I’ll take a look at what you have in terms of Dev onboarding and maybe come up with some recommendations. I’d love to get a Phoenix build going on my laptop too.”

“That would be great! Wow, I’m so excited, Maxine,” Randy says. “I never get senior engineers to work on these problems. Any engineers I have that are any good are always poached away by other teams. They’re lured away by feature work that customers see instead of working on boring infrastructure... Now, where is Josh?” he mutters, looking around. “There are so many contractors and consultants running around here that sometimes it’s hard to find the actual employees.”

Just then, a young kid carrying a laptop walks by and sits down at the desk next to them. “Sorry I’m late, Randy. I went to go check on last night’s build failure. Some developer broke the build when they merged their changes in. I’m still looking into it.”

“I’ll help you in a second, Josh. In the meantime, meet Maxine Chambers,” Randy gestures at Maxine.

Maxine does a double-take. He looks barely older than her daughter. In fact, they could be classmates at the same high school. Randy wasn’t kidding when he said he had junior people on his team.

“Maxine is a senior engineer in the company, and she’s been assigned to us for a couple of months. She’s the lead architect for the MRP system. Can you show her what she needs to know to get productive around here?”

“Uh, hi, Ms. Chambers. Nice to meet you,” he says, holding out his hand and looking puzzled. *He’s probably wondering how he ended up being responsible for someone who could be his mom*, she thinks.

“Nice meeting you,” she says, smiling. “Please, just call me Maxine,” she adds, even though it usually irks her when her daughters’ friends call her by her first name. But Josh is a work colleague, and she’s glad to have a native guide who can show her around. *Even if he’s not old enough to drive*, she jokes to herself.

“Okay, let me know if there’s anything you need,” Randy says. “Maxine, I’m looking forward to introducing you to the rest of the team. Our first staff meeting is next week.”

Randy turns to Josh. “Tell me more about the build failures.”

Maxine listens. All those stories about caveman technical practices in the Phoenix Project are actually true. She’s learned over her entire career that when people can’t get their builds going consistently, disaster is usually right around the corner.

She looks around at the entire floor. Over a hundred developers are typing away, working on their little piece of the system on their laptops. Without constant feedback from a centralized build, integration, and test system, they really have no idea what will happen when all their work is merged with everyone else’s.

Josh spins his chair around to Maxine. “Mrs. Chambers, I’ve got to go show Randy something, but I just emailed you what we’ve got in terms of documentation for new developers—there are wiki pages where I’ve assembled all of the release notes we’ve written and the documentation from the development teams. There’s also links to the stuff we know we need to write. Hopefully that will get you started?”

Maxine gives him a thumbs up. As they leave, she logs in with her new laptop and is able to get in and open her email, miraculously working the

first time. But before looking at what Josh sent, she pokes around to see what else is on her new laptop.

Immediately, she is mystified. She finds links to HR systems, network shares to company resources, links to the expense reporting system, payroll, timecard systems... She finds Microsoft Word and Excel and the rest of the Office suite.

She frowns. *This is fine for someone in finance*, she thinks, *but not a developer*. There are no developer tools or code editors or source control managers installed. Opening up a terminal window she confirms that there aren't any compilers, Docker, Git... nothing. Not even Visio or OmniGraffle!

Holy cow! What do they actually expect new developers to do? Read emails and write memos?

When you hire a plumber or a carpenter, you expect them to bring their own tools. But in a software organization with more than one developer, the entire team uses common tools to be productive. Apparently here on the Phoenix Project, the toolbox is empty.

She opens her email to see what Josh sent. It takes her to an internal wiki page, a tool many engineers use to collaborate on documentation. She tries to scroll up and down the wiki page, but the document is so short there isn't even a scroll bar.

She stares at the nearly empty screen for several long moments. *Screw you, Chris*, she thinks.

Driven by morbid curiosity, Maxine spends the next half hour digging. She clicks around and finds only a handful of documents. She reads PowerPoint slides with architecture diagrams, lots of meeting notes and Agile sprint retrospectives, and a three-year-old product management requirements document. She is excited when she finds tantalizing references to some test plans, but when she clicks on the links, she is prompted by an authentication screen asking for her login and password.

Apparently she needs access to the QA servers.

She opens a new note file on her laptop and types a note to herself to find someone who can give her access.

Giving up on documentation for the moment, she decides to find the source code repositories. *Developers write code, and code goes into source*

control repositories. There are developers working on Phoenix, ergo, there must be a Phoenix source code repo around here somewhere, she thinks.

To her surprise, despite almost ten minutes of searching, she can't find it. She adds to her notes:

Find Phoenix source code repo.

She finds links to internal SharePoint documentation servers, which may have more clues, but she doesn't have accounts on those servers either.

She types another note:

Get access to DEVP-101 SharePoint server.

For the next hour, so it goes—Search. Nothing. Search. Nothing. Search. Click. Authentication screen. Click. Authentication screen.

Each time, she adds more notes to her growing list:

Get access to QA-103 SharePoint Server.

Get access to PUL-QA-PHOENIX network share.

Get access to PUL-DEV-PHOENIX network share.

She adds more notes and to-dos, accumulating a list of more user accounts that she needs, adding the QA wiki server, the performance engineering wiki server, the mobile app team wiki, and a bunch of other groups with acronyms she doesn't recognize.

She needs network credentials. She needs installers for all the tools that are mentioned. She needs license keys.

Maxine looks at her watch and is surprised to see that it's nearly one o'clock. She's achieved nothing in two hours except document thirty-two things she *needs*. And she still doesn't know where the development tools or the source code repositories are.

If the Phoenix development setup were a product, it would be the worst product ever.

And now she needs food. She looks around, and seeing the nearly empty floor, realizes that she missed the lunch rush.

It would have been nice if she had followed them, but she had been too engrossed in digging through the labyrinth of Phoenix docs. Now she doesn't know where people find food. She wonders if she should add that to her list too.

Right after "update and send out my résumé."

From: Alan Perez (Operating Partner, Wayne-Yokohama Equity Partners)
To: Steve Masters (CEO, Parts Unlimited)
Cc: Dick Landry (CFO, Parts Unlimited),
Sarah Moulton (SVP of Retail Operations),
Bob Strauss (Board Chair, Parts Unlimited)
Date: 6:07 a.m., September 4
Subject: Go Forward Options, January Board Session
CONFIDENTIAL

Steve,

Good seeing you two days ago in Elkhart Grove. As a newly elected board director, I've been learning a lot and appreciate the time being invested by the management team to get me up to speed. I've been especially impressed with Dick and Sarah (CFO and SVP Marketing, respectively).

Although I'm new, it's clear that Parts Unlimited's failed efforts to increase shareholder value have raised questions of confidence and created need for action. We must work together to break the string of broken promises repeated quarter after quarter.

Given how essential software is to your plans, your decision to replace your CIO and VP of IT Operations seems proper—hopefully this will restore accountability and increase urgency in execution.

To reiterate my motivation for reviewing strategic options at the board level: revenue growth isn't the only way to reward shareholders—we've

put so much focus into forcing Parts Unlimited to become a “digital company” that I believe we’ve lost sight of low risk ways to unlock value, such as restructuring the company and divesting non-core, poor-performing assets. These are just two obvious ways to increase profitability, which increases shareholder value and provides working capital for transformation.

We need to quickly assemble options for the board to review and consider. Given how much time management is spending on the current strategy, the board chair asked me to work with a few key members of the executive team to generate options for the board to discuss. I will work with Dick and Sarah, given their tenure and breadth of experience at the company. We’ll have bi-weekly calls to discuss and assess ideas, and we’ll be ready to present strategic options to the entire board in January.

Our firm bought a significant interest in Parts Unlimited because we believe there’s considerable shareholder value that can be unlocked here. I look forward to a productive working relationship and improved outcomes for Parts Unlimited that we can all be proud of.

Sincerely, Alan

CHAPTER 2

• *Friday, September 5th*

Maxine scans her to-do list, slowly shaking her head in frustration. It's been two days and she is determined to perform a Phoenix build on her laptop, like any new developer should be able to do. This has become her mission. But according to her list, there are over a hundred items that she's missing, and no one seems to know where to find them.

She has done nothing on her list. Except for updating and sending out her résumé. Many friends replied to her right away, promising to look for positions she might be interested in.

Maxine asked her guide, Josh, about all those missing build items, but he didn't know anything about them. The build team used to know these things, but the details are either out-of-date or missing entirely, the knowledge scattered across the entire organization.

She is frustrated, every turn she takes leads to a dead end. There is nothing fun about this challenge. What she is doing, she's pretty sure, is the exact opposite of fun.

She's an engineer at heart, and she loves challenges and solving problems. She's been exiled smack in the middle of probably the most important project in the entire company's history. And somewhere, there's code—almost certainly millions of lines of code written by hundreds of developers over nearly three years. But she can't find any of it.

Maxine loves coding and she's awesome at it. But she knows that there's something even more important than code: the systems that enable developers to be productive, so that they can write high-quality code quickly and safely, freeing themselves from all the things that prevent them from solving important business problems.

Which seems to be completely missing here. Maxine is one of the best in the game, but after four days she still has almost nothing to show for it. Just endless clicking around, reading documents, opening tickets, scheduling meetings with people to get things she needs, trapped in the worst scavenger hunt ever.

For a moment, Maxine wonders if she's the only person having this problem. But she sees developers all around her struggling, so she quickly pushes away any feelings of self-doubt.

Maxine *knows* her kung fu is amazing. Many times in her career she's had to solve problems that seemed hopeless and impossible. Often in the middle of the night. Sometimes without any documentation or source code. One of her most famous escapades is still known as the "Maxine Post-Holiday Save," where all the in-store systems that handled refunds crashed spectacularly on the Friday after Christmas. It's one of the busiest shopping days of the year as people come in to return gifts from loved ones so they can buy something they actually want.

With her team, Maxine worked into the wee hours of Saturday morning to fix a multi-threading deadlock in a database vendor's ODBC driver. She had to manually disassemble the vendor library and then generate a binary patch. By hand.

Everyone said it couldn't be done. But she pulled it off, to the amazement of the scores of people who had worked that outage for over seven hours. The database vendor professional services team was in awe and immediately offered her a job, which she politely declined.

The legends about her kept growing after that. She's classically trained as a developer, and in her career, she's written software to stitch together panoramic graphic images and chip layout algorithms for CAD/CAM applications, back-end servers for massive multi-user games, and, most recently, the ordering, replenishment, and scheduling processes that orchestrate thousands of suppliers into a plant production schedule for their MRP systems.

She routinely lives in the world of NP-complete problems that are so difficult to solve they can take more than polynomial time to complete. She loves the *Papers We Love* series, revisiting her favorite academic papers from mathematics and computer science.

But she has never seen her job as just writing application code, working only pre-deployment. In production, when theory meets reality, she's fixed wildly misbehaving middleware servers, overloaded message buses, intermittent failures in RAID disk arrays, and core switches that somehow kept flipping into half-duplex mode.

She's fixed technology components that were spilling out their guts in the middle of the night, having filled up every disk and log server, making

it impossible for teams to understand what was actually happening. She led the effort to systematically isolate, diagnose, and restore those services based on decades of intuition and countless production battles.

She's deciphered stack traces on application servers that were literally on fire, racing to get them safely backed up before the flooding water, halon extinguishers, and emergency power shutdowns destroyed everything.

But deep down, she's a developer. She's a developer who loves functional programming because she knows that pure functions and composability are better tools to think with. She eschews imperative programming in favor of declarative modes of thinking. She despises and has a healthy fear of state mutation and non-referential transparency. She favors the lambda calculus over Turing machines because of their mathematical purity. She loves LISPs because she loves her code as data and vice versa.

But hers is not merely a theoretical vocation—she loves nothing more than getting her hands dirty, creating business value where none thought it could be extracted, applying the strangler pattern to dismantle decades-old code monoliths and replacing them safely, confidently, and brilliantly.

She is still the only person who knows every keyboard shortcut from *vi* to the latest, greatest editors. But she is never ashamed to tell anyone that she still needs to look up nearly every command line option for Git—because Git can be scary and hard! What other tool uses SHA-1 hashes as part of its UI?

And yet, as awesome as she is, at the height of her soaring powers and skills honed over decades, here she sits in the middle of the Phoenix Project unable to do a Phoenix build, even after two days. She found where two of the four source code repositories are, and she's found the three installers for some of the proprietary source code management (SCM) tools and compilers.

However, she is still waiting for license keys for the SCM, and she doesn't know who to ask to get license keys for the two other build tools. She needs credentials for three network shares and five SharePoints, and no one knows where to get the ten mysterious configuration files mentioned in the documentation. When she emailed the person who wrote the docs, it bounced. They had long left the organization.

She is stuck. No one responds quickly to her emails, her tickets, or her voicemails. She's asked Randy to help, to escalate her requests, but everyone says it will take a couple of days because they're so busy.

Of course, Maxine never just takes “no” for an answer. She has made it her mission to do whatever it takes to get a build running. She’s hunted down almost all of the people who have promised her something. She’s found out where they sit and has pestered them, even camping out at their desks, willing to stay there until they get her what she needs.

Sometimes she got what she needed: a URL, a SharePoint document, a license key, a configuration file. But more often than not, the person she hunted down didn’t have what she needed—they would have to ask somebody else, so they would open up a ticket on Maxine’s behalf. And now they were both waiting.

Sometimes, they had a promising lead or clue on who or where Maxine needed to go to next. Most times, though, it was just a dead end, and she was right back where she started.

Trying to get a Phoenix build going is like playing *Legend of Zelda*, if it were written by a sadist, forcing her to adventure far and wide to find hidden keys scattered across the kingdom and given only measly clues from uncaring NPCs. But when you finally finish the level, you can’t actually play the next level—you have to mail paper coupons to the manufacturer and wait weeks to get the activation codes.

If this really were a video game, Maxine would have already quit, because this game sucks. But Phoenix isn’t a game—Phoenix is important, and Maxine never quits or abandons important things.

Maxine sits at her desk looking at the calendar she’s printed out and pinned to the wall.

She turns back to her computer and runs her finger down her ever-growing list of to-dos again—each item a dependency she requires to get her build going.

She just added two more SharePoint credentials she needs to get from two different Dev managers who, for some reason, run their own Active Directory domains. They’re rumored to contain some critical build documentation with some of the information she seeks.

Randy sent her a ton of Word docs, Visio diagrams, and marketing PowerPoint presentations, which she quickly skims for clues. They may be helpful to marketing people and architects, she supposes, but she’s an

engineer. She doesn't want to see brochures for the car they've promised to build—she wants to see the engineering plans and the actual parts that they're going to assemble the car from.

These documents might be useful to someone, so she posted them on the wiki. Moments later, someone she doesn't know asks her to take them down because they might contain confidential information.

Looking further down her to-do list, she reads:

Find someone who can give me access to Dev or Test environments.

These were referenced in some of the documentation she read yesterday, but she has no idea who to ask to get access.

She has crossed off one item:

Get account for integration test environment.

This was less satisfying than she had hoped. She poked around the environment for two hours, trying to gain an understanding of the giant application. But in the end, she found it too bewildering—it was like trying to picture the layout of an enormous building by crawling around air ducts without a map or a flashlight.

She types out a new to-do:

Find someone who's actually doing integration testing so I can shoulder-surf while they work.

Watching someone use the Phoenix applications might help orient her. She's baffled that no one knows of an actual person who uses Phoenix. *Just who are they building all this code for?*

Scanning her to-do list again, she confirms that she in fact has nothing to actually do—she has already pestered everybody today and now she's just waiting for people to (not) get back to her.

It's Friday, 1:32 p.m. Four and a half hours to go until five, when she can finally leave the building. She tries hard not to sigh again.

She looks at her to-do list. She looks at the clock.

She looks at her nails, thinking that she needs a manicure.

She gets up from her desk with her coffee mug and walks to the kitchen,

passing by groups of people wearing hoodies, huddled together, talking in hushed urgency. Just to have something to do, she pours herself another cup of coffee. She looks down at her mug and realizes that she's already had five cups today, to satisfy the need to be *doing something*. She pours her coffee down the drain.

Along with her ever expanding to-do list, Maxine has kept a daily work diary on her personal laptop for the last decade. In it, she tracks everything she's worked on, how much time she spent on it, any interesting lessons she learned from it, and a list of things to never do again (most recently, "Don't waste time trying to escape spaces in file names in Makefiles—it's too difficult. Use anything else instead.").

She stares at her huge to-do list and her recent work diary entries in disbelief. She has never met a system she couldn't beat. *Is it possible that the utterly mediocre Phoenix Project, totally incapable of anything, is actually defeating me? Over my dead body*, she vows silently, then turns back to her work diary entries.

WEDNESDAY

4 p.m.: Waited around for Josh this afternoon who was supposed to walk me through his setup so I can replicate it. He's dealing with more nightly build problems.

I have a ticket to get access to build server, but was told by security that I need authorization from my manager. Sent email to Randy.

I'm reading every developer design doc I can find, but they're all starting to look the same to me. I want to see the source code, not read design docs.

4:30 p.m.: In one of the design docs I found the most succinct description of Phoenix: "Project Phoenix will close close the gap with the competition, allowing our customers to do the same things online that they can do in our 900 stores. We will finally have a single view of our customers, so in-store employees can see their preferences and order history, and enable more effective cross-channel promotion."

The scope of Phoenix is a little frightening. It needs to talk to hundreds of other applications across the enterprise. What could go wrong?

5 p.m.: Calling it a day. Chris stopped by, reminding me to not rock the boat or call too much attention to myself. And to not deploy anything into production.

Yeah, yeah. Sheesh. I can't even get a build going or log into network shares. How would I be able to push anything into production?

Bored out of my skull. Going home to play with new puppy.

THURSDAY

9:30 a.m.: Yes! They've given me accounts on a couple more wikis. I'm eager to dig in. This is progress, right?

10 a.m.: Seriously? This is it? I found some QA docs, but this can't be all of it, right? Where are all the test plans? Where are the automated test scripts?

12 p.m.: Okay, I met William, the QA director. Seems like a nice guy. We were able to meet just long enough to get me user accounts for their network share. Millions of Word docs filled with manual test plans.

I emailed William asking whether I can meet some of his test team. How do they execute all these tests? Seems like they need a small army. And where do they put the test results? I'm on his calendar. In two weeks. Madness.

3 p.m.: I found out where the big daily project stand-up is held: it's 8 a.m. by the whiteboards. I missed it today, but I won't miss it tomorrow.

5 p.m.: I've gotten so little done in two days. Everything I try to do requires an email, a ticket, or trying to find someone. I'm now resorting to asking them out for coffee. Maybe I'll get more responses.

FRIDAY

10 a.m.: The “15-minute stand-up meeting” went for almost 90 minutes because of all the emergencies. I don’t know how I missed this meeting yesterday—seems hard to miss because of all the yelling. Wow.

OMG. Almost no one else can build Phoenix on their laptops, either. They’re supposed to deploy this into production in TWO WEEKS! (No one is worried. Crazy. They think it will be delayed again.)

If I were in their shoes, I’d be losing my shit. Oh, well.

2 p.m.: I found a bunch of contractor developers brought in two months ago. They can’t do builds, either. Shocking. I took them out to lunch. What a disappointment. They know even less than I do. At least the salad was okay.

I shared everything I know with them, which they were extremely grateful for. Always good to give more than you get—you never know who can help you in the future. Networking matters.

Note to self: I must curb my coffee intake. Must have drunk 7 cups yesterday. This is not good—I think I’m getting heart palpitations.

At 4:45, Maxine packs up her things. There’s no chance anyone will get her anything this late on a Friday.

She almost makes it to the staircase when she runs into Randy.

“Hi, Maxine. Bummer we couldn’t get further on the Dev environment. I escalated a bunch of issues and will make some phone calls before I leave today.”

Maxine shrugs. “Thanks. I hope that lights a fire under some people.”

“Whatever it takes, right?” Randy smiles. “Uh, I have one more thing that I need?”

Uh oh, thinks Maxine. But she says, “Sure, what’s up?”

“Umm. Everyone on the Phoenix Project is required to submit time-cards,” Randy says. “We’ve got to show utilization levels or the project management people take our people away. I sent you the link to our

time-carding system. Could you fill it out before you leave? It should only take a couple of minutes.”

He looks both ways before whispering, “I especially need *your* hours because when it comes to budgeting next year, it will help me backfill your position.”

“No problem at all, Randy. I’ll take care of it right now before I leave,” Maxine says agreeably, but she is not happy. She understands the budgeting games that need to be played, but that’s not what bothers her. Instead, it’s that she already knows exactly what she worked on all week, because she keeps such meticulous notes. Absolutely zero real outcomes accomplished. Zilch. Nothing. Nada.

Back at her desk, she logs into the time-carding system. By her name are hundreds of project codes. They’re not the project names. Instead, they’re project codes that all look like airline reservation numbers—ten characters long with capital letters.

Looking at Randy’s email, she copies the project code he gave her (PPX423-94-10) into the field and then dutifully puts in eight hours into each field from Wednesday to Friday and hits submit. She frowns. It won’t let her submit until she describes what she did each day.

Maxine groans. She writes something for each day, basically saying some variant of “Working on Phoenix builds but waiting for entire universe to get me something.” She spends five minutes modifying the text so that each entry is sufficiently different from each other.

It felt bad enough to be sitting on her butt getting so little done this week, despite her very best efforts, but it feels far worse to have to lie about it in writing.

Over the weekend, Maxine continually scans her phone for updates on her tickets, but only sees them being transferred from one person to another. When her husband, Jake, asks her why she is brooding, she refuses to admit that it’s because of the timecard she filled out—it was like rubbing salt in the wound of non-productiveness. She allows herself to be distracted by their new puppy, Waffles, and thoroughly enjoys seeing her kids play with him.

By Monday morning, Maxine has successfully convinced herself to be cheerful, upbeat, and optimistic as she files into the big auditorium

for the Town Hall that the CEO of the company, Steve Masters, hosts every other month. She's enjoyed attending these ever since she joined the company. Her first one made a big impression on her because it was the first time she had seen executives directly address an entire company, taking questions from any of the nearly seven thousand employees.

Steve usually presents with Dick, the CFO. About a year ago, Steve also started co-presenting with Sarah Moulton, the SVP of retail operations. She has profit-and-loss responsibility for retail, the second largest business unit generating over \$700 million in revenue per year. While Steve and Dick exude a certain amount of trust and authenticity, Sarah seems less trustable and believable. Last year, she had a different pitch every Town Hall, promising a totally different transformation than what had been presented before, causing lots of confusion, organizational whiplash, and eventually ridicule.

Maxine sees Steve preparing off stage, writing last-minute notes on a folded sheet of paper. Someone hands him a microphone, and he walks onto the stage to polite applause. "Good morning, everyone. Thank you so much for joining us today. This is sixty-sixth Town Hall that I've had the privilege of hosting.

"As you know, for almost a century, our mission has been to help our hard-working customers keep their cars running so that they can conduct their daily lives. For most of our customers, that means driving to work so they can collect a paycheck, take their kids to school, and take care of their loved ones. Parts Unlimited helps our customers in many ways. We are one of the world's most admired manufacturing organizations, making the high-quality and affordable parts that our customers need to keep their cars running. We also have over seven thousand world-class employees directly helping our customers in nearly one thousand stores around this great nation. We are often the only thing that keeps their cars out of expensive service stations."

Maxine has heard this almost fifty times from Steve at these sessions—it's obviously important to him to remind everyone who their customers really are. When things go wrong with Maxine's car, she usually takes it to her car dealership because it's still under warranty. But the vast majority of their customers don't have that luxury. Their cars are older, sometimes older than her kids—in fact, their customers may be driving the same make, model, and year of the car she drove as a teenager. They

often have little discretionary income. When something goes wrong with their car, it can wipe out whatever savings they have (if any). And when their car is at a repair shop, they not only deplete their savings but they also can't drive to work. And that means they can't provide for their families.

Maxine appreciates these reminders about their customers—when engineers think of “the customer” in the abstract instead of as a real person, you rarely get the right outcomes.

Steve continues, “For almost a century, that mission has remained unchanged, although the business environment certainly has. On the manufacturing side, we now have fierce overseas competitors that undercut our prices. On the retail side, our competitors have opened up thousands of stores in the very same markets we serve.

“We are also in a time of incredible economic disruption. Amazon and the other e-commerce giants are reshaping our economy. Some of the most famous retailers that many of us grew up with are going out of business, such as Toys“R”Us, Blockbuster, and Borders. Just down the road here from corporate headquarters, many of us drive past that space where the old Blockbuster used to be, and that space has remained vacant for over a decade.

“We are not immune to this. Our same-store sales are continuing to decline. Many of our customers would rather order their windshield wipers from their phone from someone else than go into one of our physical stores and talk to someone.

“But I believe people don't want just automotive parts; they want help from people they trust. And that's why our store associates are so important. That's why we invest so much in training. And the Phoenix Project will allow us to bring that expertise and trust to our customers in the channels they want to use, whether it's our physical stores or online.

“Sarah will talk about the progress of the Phoenix Project later, and how it supports the three metrics I care most about: employee engagement, customer satisfaction, and cash flow. If all our employees are excited to come to work each day, and if we're delighting our customers through constant innovation and great service, cash flow will take care of itself.

“But before we go through our top annual goals, let me first talk about something that is probably on all your minds,” Steve pauses for

several moments. “Recently, I sent out an email that Bob Strauss is taking over as chairman of Parts Unlimited. As many of you know, I’ve been here for eleven years, and for the first eight of those, I had the privilege of working for Bob. He was the person who hired me back when I was head of sales at another manufacturer. I’ll always be grateful to Bob for giving me a chance to be COO of this company and for mentoring me over the years. When he retired, I took over for him as CEO and chairman.

“Effective last week, the board of directors has re-appointed Bob to be board chairman,” Steve says, his voice starting to quaver. Maxine watches with amazement as he wipes a tear from his eye. “Of course, I support this move and I look forward to working with Bob again. I’ve asked Bob to come out and share some words with us and tell us what it means for the company.”

Until this moment, Maxine hadn’t realized how much of a setback this was for Steve. She had heard that it was a demotion, but to be honest, she didn’t really understand or care much about these types of changes at the executive level. Executives came and went, often without much impact on her and her daily work. But she’s riveted by the drama unfolding in front of her.

A slightly stooped older man with white hair and a wry smile walks on stage and stands next to Steve.

“Hi, everyone. It’s great to be here in front of you after so many years. I even see some familiar faces, which makes me very happy. For those of you who don’t know me, my name is Bob Strauss. I was CEO of this company for fifteen years, back when dinosaurs roamed the earth. And even before that, I was an employee at this great company for nearly thirty years. As Steve mentioned, it was with great hope and pride that I recruited him away from another company many years ago.

“Since retiring, I’ve continued to serve on the board of directors. The job of the board is very simple: to represent the interests of the company shareholders, which includes almost all of you. We want to ensure that the company’s future is secure. If you have a pension or have been part of our employee retirement stock purchase plan, this is probably as important to you as it is to me.

“We do this primarily by keeping company executives accountable, by hiring, and, umm, occasionally firing the CEO,” he says plainly. Maxine’s

breath catches slightly—until that moment, Bob seemed like a friendly grandfather. Apparently, he has a stricter side.

“Just by looking at the stock price, you know that the markets don’t think we are performing as well as we should be. When our company’s stock price goes down while our competitors’ shares are going up, something has to change.

“I like to think that companies have two modes of operations: peacetime and wartime. Peacetime is when things are going well. This is when we are growing as a company and can continue business as usual. During these times, the CEO is often also the board chairman. However, wartime is when the company is in crisis, when it is shrinking or at risk of disappearing entirely, like what’s happening to us now.

“During wartime, it’s about finding ways to avoid extinction. And during wartime, the board will often split the roles of CEO and chairman.” Bob pauses, squinting into the bright lights, looking across the entirely silent audience. “I want everyone to know that I have complete confidence in Steve and his leadership. And if all goes well, we’ll figure out how to get him the chairmanship again so I can go back into retirement where I belong.” The crowd laughs nervously as Bob waves and makes his exit.

Steve steps up to the front of the stage and says, “Could everyone give a round of applause for Bob Strauss?”

After a muted applause, Steve resumes, “The company goals this year were to stabilize our business. Our manufacturing operation makes up two-thirds of our revenue, which has remained flat but still profitable. This has been the mainstay of our business for almost a century, and we’ve been able to fend off our very fierce Asian competitors.

“However, our retail operations continue to underperform. Our revenue is nearly five percent lower than last year,” he says. “Our biggest quarter is still coming up, so there’s hope. But hope alone is not a strategy, and you can see how Wall Street has reacted to our performance so far. However, I remain confident that the Phoenix Project will help us adapt to these new market conditions.

“So, without further ado, I’ll turn it over to Sarah Moulton, our SVP of retail operations, to describe why the Phoenix Project is so important to the future of the company.”

Sarah walks onto the stage wearing a strikingly beautiful royal blue business suit. Whatever Maxine’s opinion of Sarah, she grants that Sarah

always looks fabulous. In fact, she would look right at home on the cover of *Fortune*—intelligent, aggressive, and ambitious.

“As Steve and Bob mentioned,” Sarah begins, “we are in a time of incredible digital disruption in retail. Even *our* customers order online and through their phones. The goal of the Phoenix Project is to enable our customers to order however they want, whether it be online, in our stores, or even through our channel partners. And wherever they order, they should be able to have their product delivered to their homes or to pick it up in one of stores.

“This is what we’ve been trying to do for years. Right now, our stores are still in the dark ages. That was Parts Unlimited 1.0. The Phoenix Project will create Parts Unlimited 2.0. There are so many efficiencies we can create to help us compete against the e-commerce giants, but we must innovate and be agile. In order to remain relevant, people need to view us as a market leader creating new business models—what worked for our first century may not work for our second.”

As always, there’s some validity to what Sarah says, Maxine grudgingly acknowledges, but she can be so condescending.

“The Phoenix Project is the most important initiative for our company, and we’re betting our survival on it. We’ve spent nearly \$20 million on this project over three years, and customers still haven’t seen any value,” she continues. “I’ve decided that it’s time for us to finally get in the game. We will be launching the Phoenix Project later this month. No more delays. No more postponements.”

Maxine hears an audible gasp from the whole audience and a loud buzz of urgent murmuring. Sarah continues, “This will finally give us parity with the competition, and we will be poised to regain market share.”

Maxine sighs in frustration. She understands Sarah’s urgency, but it doesn’t change the fact that there are over a hundred developers who are nowhere near as productive as they should be, struggling to perform routine builds, spending too much time in meetings, or waiting for things they need. Sarah’s speech sounds like listening to a general tell you how important winning the war is and then finding out that all the soldiers have been stuck in port for three years.

On the other hand, at least Sarah hasn’t pitched something completely new today.

Steve thanks Sarah and then quickly reviews the company financials, and an injury that happened in one of the manufacturing plants last month. He talks about Hannah, who had her finger crushed by a stamping machine, and how they've replaced that machine with one that has a sensor preventing the plates from closing when anyone is in the danger area. He applauds the team for not waiting for budget to act on this, "Remember, safety is a precondition of work."

Maxine loves these report-outs and has always been impressed and emotionally moved by how much Steve cares about employee safety.

He says, "That almost concludes our report-out. We have about fifteen minutes for questions and answers."

Maxine's attention wanders as people ask Steve questions about the revenue forecast, the performance of the physical stores, the recent issues in manufacturing... But when someone asks about the payroll outage, she's jolted alert before shrinking back into her seat while straining to hear every word.

"I apologize to everyone who was affected by this," Steve replies. "I understand how disruptive this was to everyone, and rest assured that we've taken very specific actions to make sure that it never happens again. It was a combination of technical problems and human error, and we think we've remedied both of them."

Maxine closes her eyes, feeling her cheeks turn bright red, hoping no one is looking at her. She can't see how her exile to the Phoenix Project could possibly be considered a remedy.

CHAPTER 3

• *Monday, September 8*

After the Town Hall, Maxine returns to her desk. She looks at her calendar. It's day four of her imprisonment and her quest to perform a Phoenix build, but it feels like it's been nearly a year, the hours passing like molasses.

She gets a notification on her phone, startling her back to reality:

Phoenix Project: stakeholder status update (starting in 15 minutes).

This is a new meeting for her. To further her quest, she has asked everyone to feel free to invite her to any meetings. It beats sitting at her desk, and she's still trying to get the lay of the land. She's hoping to find someone who can get her some of the things she needs. She's been careful to avoid getting assigned any action items or to volunteer to work on fun-sounding features—she cannot be distracted from the Phoenix build.

Everyone around here thinks features are important, because they can see them in their app, on the web page, or in the API. But no one seems to realize how important the build process is. Developers cannot be productive without a great build, integration, and test process.

She arrives early and is surprised that there's no room except in the back. She stands against the wall with five other people. Looking around, her eyes widen—all the movers and shakers in the company are here. Maxine smiles when she sees that Kirsten Fingle is leading the meeting. She heads up the Project Management Office. Maxine loved working with her when she was supporting a major program that had several of Kirsten's project manager ninjas assigned to it—they were typically reserved for the most important projects that required lots of coordination across many groups within the company. They're aces at making things happen. They can escalate and resolve issues quickly, often with a single text message.

At the front of the room is Chris, who gives her a terse nod—he oversees the efforts of over two hundred developers and QA people,

dominated by the Phoenix Project. Chris is glaring at someone across the table from him who looks like Ed Harris from *Apollo 13*. When she quietly asks the person next to her who he is, he responds, “Bill Palmer, the new VP of IT Operations. Promoted last week after the big executive purge.”

Great, Maxine thinks. But she enjoys seeing the seniority of the people in the room. It’s like being up on the bridge of the starship *Enterprise* and watching the officers interact with each other.

She enjoys the first fifteen minutes of the meeting. It’s chaos. Everyone is trying to decipher what exactly Sarah meant at the Town Hall when she said that the launch was “later this month.” Kirsten says emphatically, “The date is still being negotiated and nothing specific has been communicated to me yet.” *Could it really be another false alarm?* Maxine thinks, incredulous.

Maxine guesses that there is extra urgency as they review the business priorities, the top issues needing escalation and attention, and the priorities that need deconflicting. She doesn’t know what all the acronyms mean, but she adds the ones that she thinks are actually important to her list and leaves out the insipid corporate argle-bargle.

As the meeting drones on and against all her expectations, she grows bored as the focus turns toward meaningless minutia, passionately pushed by... she honestly has no idea. *Does OEP stand for “Order Entry Protocol” or “Order Entry Program”? Or were they talking about the OPA again? Or maybe they were the same thing? Do I really care?*

Forty minutes later, her eyes are glazed over—it’s the “task status” phase of the meeting, and Maxine has lost all interest. If she had anything else to work on, she would have left by now.

Her feet hurt from standing so long, and she is reconsidering her decision to stay in the meeting when she hears someone complaining about how long they’ve been waiting for something they need. She smirks as she thinks, *Join the club. That’s what I do all day long.*

One of the Dev managers responds from the “junior people” side of the table. “Yes, we’re definitely behind, but we have a couple of new developers starting this week to help, and they should be up to speed and productive in a week or two.”

Ha. I’m really good at this stuff and I’ve made almost no progress, she thinks, looking at the floor. She smirks to herself. *Good luck, chumps.*

There is a long, awkward silence in the room. Maxine looks up. To her horror, everyone is looking at her—she realizes that she must have said something aloud.

She looks at Chris, who has a stunned expression on his face and is wildly gesturing with his hands at her in a “no, no, no” kind of way.

From the front of the room, Kirsten quickly says, “Great to see you, Maxine! I had no idea you were on Phoenix. We’re glad to have someone of your experience helping this effort—you couldn’t have shown up at a better time!”

Chris buries his face in his hands. If Maxine hadn’t already been standing against the wall, she would have been backpedaling. Mimicking Chris, she waves her hands in front of her. “No, no, no... Sorry, I’ve only been here a few days. You’re all doing an amazing job. Please carry on—I’m just here to help with documentation and builds.”

Kirsten, with the earnestness that makes her so effective, doesn’t let it go. She leans forward. “No, really. I think you said, ‘Good luck, chumps.’ I’m always interested in your perspective, given your extensive success in plant operations. I’d love to better understand what made you laugh.”

“I’m sorry I laughed,” she starts. “It’s just that I’ve done nothing except try to get Phoenix builds going on my laptop since last Wednesday, and I’m basically nowhere. I’m waiting for credentials, license keys, environments, configuration files, documentation, you name it—I know everyone has a lot on their plate and I know that Phoenix is such a large application that getting all the pieces together to do a build must be a pretty immense undertaking, but if we all want our developers to be productive, they need to be able to perform builds on Day One. Ideally, they should be writing code in a production-like environment so they can get fast feedback on whether the code they write works with the system as a whole. After days of trying, I still don’t have anything resembling the whole system—I have a box of subassemblies, with a whole bunch of missing parts. And I’m really, really good at this stuff.”

She looks around the room and gives a half-hearted shrug to Chris. She really needed to get that off her chest. Chris looks aghast.

“I’m just hoping these new engineers that you’ve been hiring have better luck than me, that’s all,” she concludes quickly.

There’s a long awkward silence. Randy nods emphatically and crosses his arms, looking smug. Someone from across the table laughs

loudly. “She is so right! They’ll need a lot more than luck! Getting a Dev environment around here is like going to the local DMV to renew your driver’s license—take a number, fill out a bunch of forms, and wait. Hell, I can get a driver’s license in a day...it’s more like trying to get a permit to start a new construction project—no one knows how long it’ll take.”

Half the people in the room laugh unkindly, while the other half of the room is clearly offended.

Maxine looks at the wise-cracking person who spoke—he’s about forty-five, slightly overweight in an “ex-athlete” sort of way. He’s square-jawed, oddly clean-shaven, with big, square glasses. He’s wearing a skateboarding T-shirt, and his face has a permanent scowl.

Based on his crankiness, Maxine bets that he’s a senior developer—being stuck in an environment like Phoenix for a long period of time must take a toll on people.

Someone from the front of the room starts responding—she recognizes William, the *super nice* Director of QA, who has gone *out of his way* to help her. “Look,” he says, “our teams are getting further behind on testing, so we all agreed that in order to hit our dates we would deprioritize environment work—shipping fully tested features would take priority. We all knew that this would increase the lead times for getting environments to our teams. Trust me, my teams are being hit just as hard as yours—QA needs environments to test in too.”

The cranky developer immediately responds, “William, you got suckered. That was a terrible decision. This is a disaster. Maxine is right—developers need environments to be productive. You should have an entire *team* of people assigned to fix the environment creation process. I’m on three projects that need staging environments, all of which have been waiting months. In fact, this is so important, I’d like to volunteer to help,” he says.

“Denied,” says Chris wearily from the front of the room. “Stay in your lane, Dave. We need you focused on features.”

William says, “Wait, wait...I’ll have you know that we’re not actually the bottleneck for environments—we have several environments that are ready to go, but we still need login accounts from Security and storage and mount points from Ops. I’ve escalated it but haven’t heard anything yet.”

Chris points a finger accusingly at Bill and turns to Kirsten, “I need help on escalating our needs to Operations.”

Bill quickly responds. “If we’re the bottleneck, I need to know. Let’s figure out how to get William what he needs.”

Kirsten nods, appearing slightly exasperated. Maxine assumes it’s because more and more dependencies are surfacing. “Yes, good idea, Bill. Alright, let’s move to the next milestone on the list.”

As Kirsten talks, Chris turns to look at Maxine, his expression screaming, *Which part of “lie low” did you not understand, Maxine?* Maxine mouths the word sorry.

Out of the corner of her eye, she sees a younger man kneel by Kirsten, whispering in her ear while gesturing at Maxine. Instead of wearing khaki pants, he’s wearing jeans and is holding a black Moleskine notebook.

Kirsten nods and smiles at him, points at Maxine, and whispers a couple of sentences in return. The young man nods, furiously taking notes.

Maxine decides to make a beeline for the door, leaving as quickly as possible before she does something else stupid.

She makes it into the cool hallway, relieved to be out of that hot, stuffy room. She heads to the kitchen, where it is even cooler. She’s thinking about getting a mug of coffee, maybe her fifth today, when she hears someone say behind her, “Hello, you must be Maxine!”

She turns around. It’s the young man from the meeting who was talking to Kirsten. He smiles broadly and extends his hand, saying, “Hi, I’m Kurt. I’m one of the QA managers who works for William. I heard in the meeting that you need license keys and environments and a bunch of other things to get a build running? I think I can help.”

For a moment, Maxine just stares back at him, not sure if she heard him correctly. For days, her life has been to search every nook and cranny for the components needed to build Phoenix. For days, she’s been submitting ticket after ticket into an uncaring and faceless bureaucracy. She’s stunned that someone actually seems to want to help her.

Maxine catches herself staring at Kurt’s outstretched hand, snaps back to reality, and shakes it. “Nice meeting you. I’m Maxine, and, yes, I’ll take any help I can get to get a Phoenix build going!”

She adds, “I hope that I didn’t step on anyone’s toes back there. I’m sure everyone is doing their best, you know, given everything that is going on...”

He smiles even more broadly, pointing his thumb back toward the conference room that they were in. “Those folks? Don’t worry about it. They’re in such deep trouble that they’re all covering their asses and

throwing each other under the bus. I doubt they'll even remember what you said by the end of the day."

Maxine laughs, but Kurt is all business. "So, you need to get Phoenix builds going. How far have you gotten and what do you still need?"

Maxine slumps. "Not nearly as far as I want, and it's not for a lack of trying." She describes in considerable detail what she's done so far and all the steps that still remain. She opens her checklist on her tablet, showing him all the open to-dos, pointing out everything she's waiting for.

"Wow, most people give up long before getting as far as you have," Kurt says. "May I see that?" he adds, gesturing to her tablet.

"Sure thing," she says, handing it to him. Kurt runs his finger down the list, nodding and appearing to compare it with another list in his head.

"No problem, I think I can get you almost all of these things," he says. And with a smile, he adds, "I'll throw in a couple other things that I'm guessing you'll need later. Don't worry, you couldn't have known. We had to learn the hard way too. No one around here documents the build environment very well."

Kurt takes a picture of her list with his phone and gives it back to her. "You'll hear from me in a day or two," he says. "The Phoenix Project is in the Stone Age. We've got hundreds of developers and QA people working on this project, and most can only build their portion of the code base. They're not building the whole system, let alone testing it on any regular basis. I keep pointing this out to the powers that be, but they tell me they have everything under control."

He looks pointedly at her. "You wouldn't put up with that back in your old MRP group supporting the manufacturing plant, right?" he asks.

"No way," she responds quickly. "It's like that guy said in the meeting—developers need a system where they can get fast and continual feedback on the quality of their work. If you don't find problems quickly, you end up finding them months later. By then, the problem is lost in all the other changes that every other developer made, so the link between cause and effect disappears without a trace. That's no way to run any project."

Kurt nods. "And yet, here we are, running the Phoenix Project, the most important project in the company, like we would have run a program in the 1970s. Developers coding all day and only integrating their changes and testing at the end of the project. What could go wrong?" he adds with smirk. "They keep telling me these decisions are above my pay grade."

They both laugh.

Kurt doesn't seem bitter or cynical. He radiates a good-natured vibe with an easy acceptance of the way the world works. He continues, "I envy how much your manufacturing team got done and how many platforms you supported. We've got ten times as many people on Phoenix, but I suspect your old team gets a lot more done than we do."

Maxine nods. She definitely misses her old team.

"Oh, and by the way, there's a rumor that might interest you," Kurt says, looking around as if afraid of being overheard. "Word is that Sarah pushed for Phoenix to launch this week, and that Steve just approved it. All hell is about to break loose. Let me know if you want to tag along as they assemble the release team. That'll be super fun to watch."

After that strange interaction, Maxine sits back at her desk and realizes that she is waiting again. Absentmindedly, she looks at a quote she always has taped to her desk from one of her favorite Dr. Seuss books, *Oh, the Places You'll Go*.

The book describes the dreaded Waiting Place, where people wait for the fish to bite, wind to fly a kite, for Uncle Jake, for pots to boil, or a better break... Everyone is just waiting.

NO!
That's not for you!
Somehow you'll escape
all that waiting and staying.
You'll find the bright places
where Boom Bands are playing.

Everyone on the Phoenix Project is stuck in the Waiting Place, and she is determined to rescue everyone from it.

It's 11:45 a.m. Maxine looks at her calendar. It's only day four of her forced exile. While she hadn't hear back from Kurt, she did manage to get access to the third of four source code repositories. Today, she decides that she cannot wait for other people any longer.

She is going to build something.

Over the next four hours, she tries running every Makefile, maven POM, bundler, pip, ant, build.sh, build.psh, and anything resembling a build script she can find. Most fail immediately when she runs them. Some barf out alarmingly long error messages.

She pores through all the error logs, scanning for any clues on how to get something to actually run, sifting through all the poo for anything resembling a peanut—laborious and unpleasant work. She identifies at least twenty missing dependencies or executables that she needs. Several times she asks around to see if anyone knows where to get these things, opening tickets and sending out emails, but no one knows. She spends three hours searching around the internet for clues, pouring through Google and Stack Overflow.

In a moment of very poor judgment, she decides to try building some of the missing components from scratch, from similar sounding components that she finds on GitHub. Five hours later she's in a terrible mood—exhausted, frustrated, irritated, and absolutely positive that she has just wasted an entire day going down rabbit holes she never should have gone down.

Basically, she tried to forge missing engine parts by melting down aluminum cans. *That was really dumb, Maxine*, she thinks.

That night when she gets home, she realizes that she has brought all the frustration from work with her. She warns her husband and kids that she is currently incapable of conversation and finds two mini-bottles of Veuve Cliquot rosé in the fridge. When her teenage kids see her, they know immediately to avoid her. She is wearing her “Mom is in a super bad mood” face.

While they prepare dinner, she crawls into bed and watches movies.

What an utter waste of a day, she fumes.

She reflects upon the difference between a great day and a bad day. A great day is when she's solving an important business problem. Time flies by because she's so focused and loving the work. She's in a state of flow, to the point where it doesn't feel like work at all.

A bad day is spent banging her head against a monitor, scouring the internet for things she doesn't even want to learn but needs to in order to solve the problem at hand. As she falls asleep, she tries not to think about how much of her life is spent searching the internet for how to make error messages go away.

It's a new day. Fresh from a good night's sleep, she sits at her desk, intent on *not* repeating yesterday's mistake. Just because she felt busy doesn't mean that she actually did anything meaningful. In a terminal window, she pulls up her work from yesterday and, without looking at it, deletes it all.

Then she pulls up all of her open tickets in the ticketing system. She refuses to feel powerless, at the mercy of distant powers, trapped inside a cold bureaucracy actively impeding her goals, aspirations, desires, and needs.

Maxine has a long and complicated history with ticketing systems, both good and bad.

Last year, she funded a Kickstarter project for a mug that promised to keep her coffee or tea at whatever temperature she wanted for hours. It even had a Bluetooth connection so that she could see and set the temperature of her drink from her phone. She loved the idea of it and quickly paid five hundred dollars to help the inventor.

She was thrilled every time she got a notification: when the inventor reached her funding goals, when a manufacturer was selected, when the first production run started, and, most importantly, when her mug had been shipped. It was so satisfying to be a part of the collective journey and to eventually hold one of the first five hundred mugs made.

The Dev ticketing system felt totally different. It was the opposite of the joy and anticipation she had felt with her magic mug. Instead, it reminded her of the horrific experience of getting her first high-speed DSL broadband package working back in the 1990s. Although she received the DSL modem right away, she then had to deal with the internet service reseller (who sold her the DSL service) and the phone company (who owned the copper wires to her house).

Whoever did the installation at her house must have screwed up because nothing worked—and when she called either organization, they told her that it was the other's responsibility to fix it. Sometimes they could find a ticket related to her previous conversations, sometimes not. She was trapped in a cruel, uncaring, Kafka-esque bureaucracy. For four weeks, her awesome DSL modem did nothing except flash a red blinking light. It was as useless as a brick, and she had countless open tickets with both organizations.

One day, Maxine decided to take an entire day off of work just to tackle activating her DSL line. After three hours, she finally managed to talk her

way up the chain to a Level 3 escalation support person who had access to both ticketing systems. He was amazing and obviously incredibly savvy, able to thread Maxine's request to the right department, using the right keywords to get the two supervisors from both companies on the phone to line up all the necessary work. An hour later, she finally had her 64 Kbps broadband connection working.

Even decades later, she remembers how grateful she was to that support person. She had told him, "I'd love to talk to someone that matters about what an amazing job you did and how grateful I am for your help." She happily stayed on hold for ten more minutes to talk to a supervisor and then spent ten minutes gushing in great detail about all the help that she had received.

It was so important to Maxine to describe how extraordinary and even heroic everything that Level 3 support rep did and how much she valued his help. Maxine was gratified to hear that he was being considered for a promotion and that her phone call likely sealed the deal.

Afterward, Maxine spent an hour watching the blinking green light and savoring the blazingly fast download speeds.

Remembering that savvy support rep, Maxine reminds herself that she loves solving problems, that she loves challenges, and how important her work is. It will help every developer become more productive.

Taking a deep breath, she summons the relentless optimism she's been accused of having, and carefully scans her email for any new ticket status changes. She ignores all the team status updates, except for the ones where people are yelling at each other in ALL CAPS. She wants to know who the hotheads are so she can avoid them.

Continuing to scroll, Maxine's heart leaps when she sees the subject line:

Notification: change to ticket #46132: Phoenix Dev environment.

Is her Dev environment finally ready?

She tries not to get excited, because she's been fooled before. Twice yesterday she got a notification, but it was only for the most minor status

changes on her ticket. The first was when someone finally looked at the ticket; the second was when it was reassigned to someone else.

Maxine clicks on the link in the email, which pulls up the entire history of the ticket on her browser. She squints and leans closer to her screen. She opened the ticket six days ago (if she counts the weekend), and there have been seven status changes as different people worked on her environment request. As of 8:07 this morning, the ticket is marked closed.

She hoots loudly. At long last, the Ops people are done! She is in the build business!

But she's confused. Where's her environment? How does she log in? What's the IP address? Where are the login credentials?

She scrolls to the very bottom to the notes and comments, reading what each person typed in as they were worked on her ticket. It got bounced from Bob, to Sarah, to Terry, back to Sarah, then finally to Derek. At the very bottom of the notes, Derek wrote:

To get a Dev environment we need approval from your manager. The correct process is documented below. Closing ticket.

Maxine's face turns hot and bright red.

Derek closed my ticket?! After all that waiting, he just closed my ticket because I didn't have an approval from a manager?

Who the hell is Derek?! Maxine yells to herself.

She moves her cursor around the ticketing screen, trying to find anything to click on. But the only clickable link is the policy document that Derek provided. She can't find any way to find out who Derek is or how to contact him. She finds a button to reopen the ticket, but it's grayed out.

Thanks a lot, Derek, you shithead, thinks Maxine angrily.

Fuming, she realizes she needs a break. She stomps out of the building and sits on a bench in front of the office, she takes a deep breath, closes her eyes, and counts to fifty. Then she walks back into the office and sits down at her desk.

She clicks the button to open a new ticket. When the blank ticket appears, with the countless fields that need to be filled in, she almost gives up and goes home. Almost. Instead, she forces herself to smile and summons up her friendliest self:

Hi, Derek. Thank you so much for working on my environment, which we badly need for the Phoenix Project—please see Ticket #46132 (link below). I'm assuming I can paste my manager's approval (Randy Keyes) in this ticket? I'll get an email from Randy with his approval for you in the next 30 minutes. Can I call you to make sure this gets processed today?

She clicks the Submit button, writes a short email to Randy asking him to okay the creation of a Dev environment, and runs to his desk. She's relieved to see that he's there and not in a meeting. She tells him what she needs and then stands over him as he sends a reply that merely says "Approved."

By the time she gets back to her desk, she feels a sense of grim determination and relentless focus. She will do whatever it takes to get her Dev environment. She sits down, copies and pastes Randy's approval from her email into the service desk ticket, and adds the note:

Derek, thanks so much. Here's my manager's approval. Can I still get this environment today?

She hits submit.

She pulls up the corporate phone directory and scans for all the Dereks in the IT organization. There's three. The one in the helpdesk department is the most promising.

She emails him a nice, friendly note, CCing Randy, to thank him in advance for all his help and to let him know how grateful she'll be to get builds going for Phoenix, practically begging him not to put her ticket at the bottom of the queue where it will take another week to process.

She hits send. Five seconds later:

This is an automated response—please put all service desk related tasks into the service desk. I will do my best to read all my emails and respond in 72 hours. If this is an emergency, please call this number . . .

She curses. She imagines Derek sitting with his feet up on his desk, guffawing at her misery. She prints out everything related to Ticket #46132, her emails, and the names of the three Dereks, looking up

where each one of them sits. Helpdesk Derek is two buildings over, lower level.

She steps out of the elevator on Derek's floor and sees rows of small cubicles with people wearing headsets in front of computers. There are no windows. The ceilings are surprisingly low. She can hear the electrical buzz of florescent lights. It's oddly quiet. *There should be more fans running to make the air less stale*, she thinks. She hears people typing and a few people talking politely to people on the phone.

Seeing all this, she feels suddenly very, very guilty about her previous anger toward Derek. She asks someone where he sits. After walking through the maze of cubicles, she finally sees Derek's nameplate, printed by an ink-jet printer without enough ink. Then she sees Derek.

He's not a hardened bureaucrat at all. Instead, he's in his early twenties. He's Asian, with a remarkably earnest expression as he scans something on a small LCD screen. Maxine has had laptops with bigger screens than this budget PC setup. She feels even worse about all the bad things she's thought about him.

There are no extra chairs, so Maxine kneels down beside him. "Hi, Derek," she says in her most friendly voice. "I'm Maxine, the person who submitted the ticket about the Dev environment last week. You closed it this morning because I didn't have a manager's approval. I just got it. And because you closed the ticket, I had to open a new one. I'm wondering if you can help me get this through the system."

"Oh, golly, sorry about closing the ticket. I'm new to all of this!" Derek replies earnestly, obviously distraught that he might have screwed up. "All I know is that I'm supposed to make sure that certain requests that need approvals have them. I can't reopen tickets. Only supervisors can do that. And all new tickets go into the queue where they get assigned to the next open person. Maybe my supervisor can help?"

Maxine slumps, dreading what might be ahead. But looking around at the sea of people, she realizes that if she doesn't get this straightened out now, she will never get her Dev environment.

"Absolutely. That would be great, Derek." He smiles and they walk toward one of the outer offices.

Over the next fifteen minutes, Maxine watches Derek's supervisor expertly navigate through the extensive ticket trail, examining its history. In the time since Maxine left her desk, someone named Samantha has already

closed her new ticket, pointing out that approvals cannot be submitted in the “Notes” field.

Maxine refuses to lose her cool. These people are trying to help her. The supervisor apologizes about how inconvenient this is. She merges Maxine’s two tickets together, puts Randy’s name in the approver field, and resubmits the ticket. “Now Randy just needs to hit one button in the tool and you’re good to go! Sorry we can’t actually authorize requests—only designated managers can.”

“Can he approve it from his phone right now?” she asks, with forced cheer. Apparently not—the helpdesk tool was written before smartphones existed and mobile phones were probably still the size of suitcases capable of showing only seven LED digits.

Maxine sighs, but she thanks them effusively because she is certain she is finally close to achieving her goal. As she turns to leave, Derek asks tentatively, “Do you mind if I ask a stupid question?”

“Of course. There are no stupid questions. Fire away,” she smiles, trying not to look manic.

“What’s a Dev environment? I’ve dealt with laptop issues, password resets, and things like that. But I’ve never heard of an ‘environment’ before.”

And there it is, Maxine thinks, thoroughly humbled. This week’s lesson on patience, kindness, and empathy, brought to you by Derek and the helpdesk department.

Maxine is proud that she has earned a reputation for being level-headed, compassionate, and having empathy for others. But right now, she feels like she demonstrated none of those things. Is being assigned to the Phoenix Project making her a bad person?

She realizes how misdirected her anger at Derek was. This poor guy didn’t hold anything against her because she was a developer. He didn’t even know what she was asking for, let alone understand how important it was. In his inexperience, he had merely closed her ticket by following the rules set for him. He was only trying to do his job the best he’d been shown how.

Maxine returns to her desk two hours later. She had taken Derek and his supervisor to lunch—out of appreciation for their help and to atone for thinking the worst of Derek. She got the chance to explain the world

of development to him, and his earnest curiosity was infectious. She described all the exciting career tracks available for technical people outside of the helpdesk, hoping it might encourage him to explore some of the options available to him.

She goes to find Randy to make sure he approves her request. He's not at his desk. She calls him on her cell phone right away.

"I can't approve it until I'm back at my desk," Randy tells her. "I absolutely promise that I'll approve it when I'm out of meetings. It'll be done before five o'clock."

Maxine goes back to her desk, feeling conflicted. She understands the need for automated workflows. On the manufacturing plant floor, the MRP systems she wrote control everything that thousands of people do virtually every minute of the day. You can't manufacture products in large volumes, costing thousands of dollars, without a rigorous process.

Even the helpdesk process, whether here or at the organization she had to deal with to get her DSL model installed decades ago, is a pretty good way to provide consistent customer service most of the time, even when it's delivered through thousands of call center staff.

So why does the ticketing system here feel so awful? We're all part of Parts Unlimited, so why does everything feel like I'm dealing with a government bureaucracy or an uncaring vendor? Maxine ponders. *Maybe it's because when friends do favors for friends, we don't require them to open a ticket first.*

The next day, Maxine sees that Randy delivered as promised—he approved the Dev environment ticket, but it was too late for anyone to start work on it.

Despite this breakthrough, Maxine is still waiting for a Dev environment. Disappointed, she wanders aimlessly from meeting to meeting, not wanting to be idle at her desk.

Killing time in the kitchen waiting for another pot of coffee to brew, her phone buzzes. Screenful after screenful of email notifications about changes to Ticket #46132 appear. A request to get a virtual machine assigned to the distributed systems group, a request for storage from another group, an IP address from yet another group, a network mount point from another group, application installs from three more groups...

Maxine whoops in delight—progress, at last! Santa just mobilized his magical elf army to start building the Dev environment that she so desperately needs. The cavalry is finally coming!

Exhilarated, she reads through all the tickets. So much work is being dispatched to what appears to be the entire Ops organization. Maxine suddenly becomes alarmed at just *how many people* are required to create one environment.

She works at her desk, planning out what she'll do first with her Dev environment, when her phone starts buzzing incessantly. Opening her emails, her jaw drops at the forty notifications in her inbox. At the top of her screen are a flood of new notifications marking all her tickets as closed.

"No, no, no," she moans, starting to trace through the ticket history from the beginning. She sees that the user accounts were created, the mount points configured, but then she sees a note from one of the storage administrators:

I'm sorry to have to close your ticket. Believe it or not, we've been out of storage space for the last three months. We have a big order for more storage that we can't expedite until January, and worse, all the controllers are already at capacity. Purchasing says we can only order twice per year to get the best quantity discounts with our vendors. You are near the top of the list, so we'll get this scheduled for February.

Maxine blinks.

It's September.

Phoenix is the most important project in the company. They've spent \$20 million over three years. And yet, here she is, trying to help, and they won't spend \$5,000 on more disk space. And now she won't get a Dev environment for five months! She buries her head in her hands and silently screams down at her keyboard.

Completely defeated, she takes another walk, to nowhere in particular. It's two thirty p.m. None of the meetings on her calendar seem interesting anymore. It's just people complaining about waiting. Waiting for something. Waiting for someone. Everyone is just waiting. And she wants no part of it right now.

She returns to her desk, picks up her jacket and laptop bag, and decides to leave. She doesn't know where she'll go, but she just can't stay here today.

It isn't until she's behind the steering wheel that she decides to drop by her kids' old school. Not to see her kids—they're at that age where they don't want to be seen with their parents anymore. No, she's going to the lower school where the fifth- and sixth-graders meet for after-school activities. She is proud of having helped the teachers create a coding club three years ago, which has become wildly popular. And she is delighted they've found so many students who are having fun with science, technology, engineering, and math before they go into high school.

Maxine believes learning these skills is incredibly important, because coding is a proficiency that every profession is likely to need in the next decade. It won't be just for developers anymore.

She walks into the classroom and immediately sees Maia and Paige, two of her favorite kids to work with. They're best friends but also fierce competitors, sometimes even archrivals. They're both smart, ambitious, and have a gift for problem-solving.

This is the first time this school year that Maxine has visited. She's surprised at how much older everyone looks and how much their coding skills have progressed. Some are writing what look like games in JavaScript, some are working on web servers, and two students look like they're writing mobile phone apps.

She spends the next hour learning what each group is working on, laughing in delight as they show off what they've created, and loving it when they ask her questions. When Maia and Paige ask her to help solve a problem, she quickly pulls up a chair.

They're trying to complete a classroom exercise to compute the mean, mode, and interquartile ranges of an array of numbers in Python. She immediately sees they've made the same indentation mistake over and over again.

Of course, when they try to run their program, the Python parser vomits at all the indentation errors. They've clearly been struggling to figure out what exactly they did wrong, trying everything they can to make the errors go away.

"May I make a suggestion?" Maxine asks, taking a more active role.

“Of course, Mrs. Chambers,” Maia says. Maxine sighs, still not sure how she wants to be addressed by teenagers.

Maxine explains to them how Python indentation works and how it differs from most other programming languages. “But whatever language you’re using, the most important thing is to run your program all the time,” she says. “When I’m doing something for the first time, I run my program every time I change *anything*, just to make sure it still compiles and runs. That way, I don’t make the same mistake for hours without knowing. Better to catch the mistake the first time you make it, right?”

She directs them on how to correctly line up some of the indentation. “Let’s see if that fixes the first error...”

She scans the buttons on their editor. “It looks like you can run your program just by hitting Control-Enter. Ah, looks like there’s one more little change needed. Yep, you’ve now fixed that first error. Now fix up the next error until you’re back into a working state. If you keep checking after each small change, you’ll never have something big you need to fix...”

Reflecting on what she said, she adds, “One of the really nice things about running your program frequently is that you get to see it running, which is fun, and that’s what programming is all about.”

The girls smile in understanding and stomp out the rest of their errors in quick succession. Maxine grins seeing them use the keyboard accelerator she showed them.

The girls are smiling because their program is actually running now. Looking at the output, Maia notices that something seems wrong. Their computed average is way off.

“Hmm...I think this is an ‘off by one’ error,” Maxine says. “This is one of the most common errors that developers make. It often happens when we are looping through every element in an array, and we miscompute which one is the last element. And that’s what’s happening here—we missed the last element...and believe it or not, when you accidentally process one element too many, that can cause the program to crash or even be exploited by a hacker.”

Maxine can’t stop smiling. She’s so excited to share this lesson, because over the decades, she’s learned that state mutation and looping is so very, very dangerous and difficult to get right. That crashing ODBC database driver she fixed in the middle of the night a decade ago, which made her so notorious, was caused by this type of problem.

This is why Maxine is so dedicated to applying functional programming principles everywhere. Learning Clojure, her favorite programming language, was the most difficult thing she had ever done, because it entirely removes the ability to change (or mutate) variables. Without doubt, it's been the one of her most rewarding learnings, because she's found that about ninety-five percent of the errors she used to make (like the ones the girls just made) have disappeared entirely.

Functional programming is truly a better tool to think with.

"You want to see something cool?" Maxine asks the girls. When they nod, she says, "Here's what I would do. It looks a little strange, but you can get rid of loops entirely by using iterators—it's an easier, and much safer, way to write a loop."

She scans the Python documentation she finds on the internet, and types one line of code in their editor, hitting Control-Enter a couple of times and converging on the right answer.

"Voila! Look at this. It does the same thing as what you wrote, but with no loops or conditional logic, such as checking for the end of the array. In fact, it's only one line of code, with no risk of an 'off-by-one' mistake!" she says, actually feeling proud of what she just wrote.

Maxine is rewarded by "oohs" and "ahs" as the girls' eyes widen at what Maxine shows them. Maxine is pleased, because even this small exercise has banished some unnecessary complexity from the world. It might save the girls from decades of frustration and make the world a safer place.

She spends the next hour floating between the teams, having fun watching kids solve problems and teaching them little tricks here and there to make them more productive and joyful. When the kids adjourn, packing up their huge, heavy backpacks, Maxine realizes that she's in a very good mood.

The joy of teaching people something that they want to learn is awesome. Plus, these are really great kids. She thinks about how easy everything is here. You hit Control-Enter and the program builds and runs. If there is an error, you fix it, hit a key, and try again.

In her current work hellhole, it's the opposite. She still can't build any part of the Phoenix system. Somehow builds have ceased being a part of everyone's daily work.

Maia and Paige made the same indentation mistake for a half hour. At Parts Unlimited, there are a hundred developers probably making bigger

mistakes, and it'll be months before they realize they've done something wrong.

Everyone waves goodbye to Maxine, thanking her for all her help. She climbs into her car and slumps in her seat. To her surprise, she feels sad and dispirited—at work, there is none of the joy and learning that she just experienced here. She wonders if this is how everyone on the Phoenix Project feels all the time.

Maxine is about to start her car when her phone vibrates. It's a text message from one of her collaborators on an open-source project she wrote to help her with personal task management. She started this project over five years ago to help her keep awesome work diaries. She's always been a manic record-keeper of how she spends her time.

Initially, it was just to help her be more productive, to triage her incoming tasks from email, Trello, Slack, Twitter, reading lists, and what seemed like a gazillion other places where work is generated. Her app let's her easily push work into GitHub, JIRA, Trello, and the many other tools where she interacts with other people and teams.

Over the years, she's used this program every day to help her run most of her professional and personal life. It's where she sends all her tasks. It's her master inbox, where she can see all her work and move work between all the systems that she works with.

Many other people use her application too, and some have written adapters to other tools they need to connect to. She's constantly amazed that thousands of people around the world use it every day, with over twenty active contributors writing code for it.

She looks at the new pull request from the text message—someone has created a new adapter to the task manager. The proposed change looks fantastic. She's wanted to do it for years. Their change is quite clever, and she approves of the way the automated tests were written to show that his change works without breaking anything else. He also documented his work, writing several paragraphs on what he did and why. She approves of the way he's turned it into a tutorial, so others can do something similar.

She loves seeing other people's ingenuity and their willingness to make the app better. As the project owner, she sees it as her primary responsibility to ensure that any contributor can be productive.

A couple years ago, there were over twenty active pull requests, but for a variety of reasons, she couldn't merge them in—sometimes the changes were conflicting, sometimes her API couldn't quite accommodate what they needed. She knows it's dispiriting when you submit a change to someone's project, but no one ever looks at it or they tell you that it can't be integrated. If that happens enough, people eventually give up or fork your project and splinter the community.

So, when this started happening to her project, she spent every evening for weeks rearchitecting her system so that people could quickly, easily, and safely make the changes they wanted. It took a lot of effort, and she personally rewrote every pull request so that the contributors didn't have to redo their work. But everyone was delighted and grateful when their changes made it in—but not as delighted as Maxine.

Maxine knows that agility is never free. Over time, without this type of investment, software often becomes more and more difficult to change. There are exceptions, like floating-point math libraries that haven't changed in forty years—they don't need to change, because math doesn't change.

But in almost every other domain, especially when you have customers, change is a fact of life. A healthy software system is one that you can change at the speed you need, where people can contribute easily, without jumping through hoops. This is how you make a project that's fun and worthwhile contributing to, and where you often find the most vibrant communities.

She drives home and is delighted that her husband has already taken care of dinner. She regales her kids about her last-minute decision to visit their old middle school and the exciting new generation of geeks.

When they disappear to do their homework, she grabs her laptop and brings up the exciting new pull request. She pulls the code in and spins up the new version on her laptop. She logs in and clicks around, testing some of the corner-cases to make sure he got the details right.

Smiling, she brings up the pull request in her browser and clicks a button, which merges it into the code base. She writes a thank you note to the submitter, complimenting him on his cleverness and initiative.

Before hitting send, she notices something he wrote: "Maxine, I'd throw you a huge party if you could display a desktop notification whenever someone modifies this property..."

Good idea, she thinks. She pulls up her code editor, and in the next fifteen minutes takes a stab at implementing this idea. When it works the

first time, she grins and laughs out loud, clapping her hands in gleeful selfcongratulation. She's in a great mood. You can do so much with so little effort because of all these miracles of technology.

She resumes her note to the submitter:

Again, really nice work. I'm sure everyone is going to love it as much as I do. Thank you! (And I just added your notification feature. Your offer for throwing me a party is accepted.)

Hitting send, she wonders if the universe is sending her a message. Her afternoon with the middle schoolers and the ease with which she added functionality to her application (which is years older than the Phoenix Project) shows her what coding *should* feel like.

She is able to build things with focus, flow, and joy. She had fast feedback in her work. People were able to do what they wanted without being dependent on scores of other people. This is what great architecture enables.

She has been exiled to the most strategic initiative of the company, on which the survival of the entire company hinges. And yet, hundreds of engineers are paralyzed, unable to do what needs to be done.

In that moment, Maxine decides she must bring this level of productivity that she's helped create for middle-schoolers and her open-source project to the Phoenix Project, even if it means personal suffering in the short term.

CHAPTER 4

• *Thursday, September 11*

The next morning, Maxine still feels triumphant from yesterday's numerous victories. But as Kurt predicted, everyone is freaking out. To everyone's shock and disbelief, the launch was not going to be called off or delayed. Instead, the Phoenix Project is going to be launched tomorrow at five.

Captain Kirk apparently hit the warp speed button, despite Engineer Scotty telling him the dilithium crystals were about to blow. So, no boring status meetings today. Instead, every meeting is in a genuine shitstorm, with people on the verge of panic. One meeting quickly devolved into bedlam and pandemonium, full of questions, objections, and shocked disbelief. People were furiously typing away on their phones and laptops, and a third of people in the room were making phone calls. It was like watching an old movie from the '40s, with reporters racing out of the courtroom to the payphones or back to their offices, frantic to break the news first.

Maxine turns to the person next to her, loudly yelling, "Has Phoenix ever been deployed into production before?!"

"Nope," he yells back.

"Is there a release team yet?" Maxine asks.

"Nope. Chris, Kirsten, and Bill are mustering up a formal release team today, but I have no idea who's going to be in charge," he responds and mimics biting his fingernails in nervousness and fear.

Maxine looks back at him, speechless.

Maxine does not delight in the suffering of others, but watching the fireworks surrounding Phoenix is much more exciting than waiting for people to work on her tickets. She groans, realizing that given this crisis no one will be able to work on her tickets now.

Later that morning, Chris announces that William, the QA director, is in charge of the release team. His goal: get everything into a releasable state and to coordinate with Ops, who were blindsided too.

Poor bastard. She knows that they are in big trouble. The Phoenix developers can't even merge their own code together without accidentally leaving parts of it behind or blowing up the build. Pulling off a successful production deployment seems wildly optimistic. *Or plain bat-shit crazy,* she thinks.

"William, when is your release team meeting?" Maxine asks him as he jogs by. She runs to keep up. "Can I help?"

"First meeting is in one hour. We need all the help we can get," he says, not even breaking stride. Maxine is delighted. Finally, a chance to actually use her skills and experience.

This will be an interesting meeting, she thinks. Maxine has seen how Dev and Ops interact around Phoenix. Instead of acting like an actual team, they act more like sovereign states on the brink of war, with diplomats trying to patch together an uneasy peace, complete with embassies, protocols, and official formalities. Even scheduling a meeting between these two groups requires a summit and lawyers present.

Regardless, she's excited to be in the game. In a perverse way, this is the most fun she's had on the Phoenix Project so far. She realizes she's grinning from ear to ear. *Does this make me a bad person?* she wonders. She grins again, not caring.

Despite trying to arrive early, Maxine is late to the war room. They had to move the meeting twice because the crowd kept outgrowing the space.

It's fifteen degrees hotter in the room than in the hallway, and the air is stale. Nearly fifty people are crammed into a space designed for half that number. She sees Chris, Kirsten, William, and a bunch of the lead developers and managers. Kurt, sitting next to William, waves to her.

On the other side of the table is Bill Palmer, surrounded by a phalanx of faces she doesn't recognize. She notices that there's something... different about them.

The largest one of them to the left of Bill has his arms crossed and a huge, unhappy scowl on his face. He shakes his head in disbelief. "What is wrong with you people? You're telling me you don't know how many Windows servers you need, on top of the handful of Linux servers... Tell me again, how many exactly is a 'handful of servers?' Is that in metric or

imperial handfuls? While we're at it, you have any Kumquat boxes you need, or perhaps a Tandem?"

Flanking him are a woman and a younger man. The way they snicker makes Maxine immediately think of Crabbe and Goyle, the two mean-spirited goons who were best friends with Malfoy, Harry Potter's rival from Slytherin house.

"Uhh..." says one of the Dev managers. "Actually, there is *one* component that can only run on Kumquat servers. It's an extension we had to build off of the existing message bus. But it's only a small modification. It shouldn't cause any problems, and it should add negligible load..."

Maxine hears groans from around the room, and not just from the Slytherins on the opposing side of the table. The younger man sitting next to the large man, whom Maxine is already thinking of as Big Malfoy, sighs. "Technically, there's nothing wrong with Kumquats—we have over a decade of experience running production workloads on them, and we understand their characteristics pretty well. The problem is that the reboot time for that cluster is almost eight hours. We just need to be careful about anything that could involve restarts, like security patching. I'm concerned that certain changes will require multiple reboots, which could mean a day of downtime... or that they'll never come back at all..."

These are all the Ops people, Maxine realizes. No wonder she hasn't seen them around.

"Wes, trust me, we are as afraid of that scenario as you are," the Dev manager replies from across the table. "We've been trying for three years to get this application re-platformed, but it always takes a backseat to something more important."

"Yeah, you developers always make sure features take priority, and you never clean up all the technical debt you create... typical crap," Big Malfoy says, gesturing angrily.

Bill says to Big Malfoy, without even turning his head, "Stow it, Wes. Work the problem. Stay focused."

"Yeah, yeah. Got it, boss," Wes (Big Malfoy) says. "Handful of Linux servers, handful of Windows servers, and one Kumquat server. Got it. Now who can define a 'handful?'"

Maxine watches all the Dev managers put their heads together, tabulating the compute needs for each of their components. It's clear they're just going on gut, not any sort of thoughtful capacity-planning exercise.

Maxine realizes that this release is in even more trouble than she thought. The developers still haven't merged all their code together. And they haven't defined the production environment that the application needs to run in—describing your environment in “handfuls” definitely does not cut it.

Raising her voice, she asks, “How many transactions per second are we expecting for product displays and orders? And how many transactions per second are the current builds capable of handling right now? That will tell us how many servers we need for the horizontally scalable portions, as well as how far we're off for the vertically scaled components, like the database.”

The room falls silent. Everyone turns toward Maxine. They seem startled by her common sense question. The woman sitting to the left of Wes says, “Thank you! That is precisely what we need to know!”

Maxine gives a small nod and winks.

Chris stands up. “This is the highest publicity release in the entire company's history. Marketing has pulled out all the stops. They're going to spend almost a million dollars getting the word out about the Phoenix launch. All the store managers have been given instructions to tell every customer to download the app and hit the website Saturday—they're even having contests to see which stores register the most new mobile customers. They're hitting all the industry and business press. They're trying to get either Sarah or Steve on all the news shows—even *Good Morning America*.

“Here are the best calculations that I've been able to get from Marketing,” Chris continues, flipping through his notebook. “Expect one million people to come banging on the Parts Unlimited website and mobile apps. If all goes well, we should be prepared to sustain at least two hundred orders per second.”

Maxine hears mutters and curses from all around the room.

Wes scans the room and finally turns to Chris, all sense of jocularitas gone. “Okay, that's good to know.” He gestures at Maxine, “Our smart architect just asked how many transactions Phoenix can handle right now. Well?”

Chris looks to William, who pulls out a printout. “Hot off the press from this morning. In our tests, Phoenix currently handles about five transactions per second. Anything over that causes the database clients to

start crashing due to timeouts, including the mobile apps... I think we're missing a bunch of database indexes, but we haven't figured out which ones yet..."

William looks up. "It's very, very not good, Chris."

Wes sits in stunned silence for a moment. And then in a blunt, world-weary voice, he says to Chris, "We're not going to make it, are we?"

No one says anything. Eventually, Bill asks, "Wes, what help do you need?"

"...I don't even know," he replies. "Maybe just give the teams some air cover so they can stay focused."

At that moment, Maxine hears a loud voice from the doorway. "For the survival of Parts Unlimited, we have to make this work, so of course we're going to make it."

Oh no, Maxine thinks. It's Sarah Moulton.

She's dressed in a bright, expensive looking yellow suit, and her face is so radiant that Maxine wonders how it's even possible. The fluorescent lights in the office usually make people look ghoulish and devoid of color. Maxine wonders if maybe she adds radium to her makeup to make herself glow like a 1950s bedside clock. Sarah has a certain dangerous glamour about her, and everyone in the room seems similarly rapt.

"We are in a market that is shrinking, with fierce competitors taking market share away from us," Sarah says. "Not to mention tech giants like Amazon and twenty new startups that are coming in to disrupt this entire category. As Steve said at the Town Hall, we've had three years to prepare for this. Now it's time for us to go to war and defend what is rightfully ours."

She looks around the room, scanning for signs of resistance or rebellion. "This is the strategy the executives of this company have decided on. Anyone have a problem with that?" she challenges.

Incredibly, Maxine hears herself laughing. Horrified, she covers her mouth. *Keep it together, Maxine!* Quickly, she wipes all expression from her face, like a student caught doing something bad in high school. *Since when have you ever cared what people in authority think about you?* she wonders.

Ever since Chris warned me to keep my head down, she realizes. Maxine forces herself to look calmly at Sarah with her best Lieutenant Saavik expression, radiating only cool, dispassionate logic.

“Something seem particularly funny to you...umm, sorry, what is your name?” Sarah asks, looking at Maxine coolly.

“Maxine,” she replies, calmly. “I was laughing because you were talking about *why* you think Phoenix is important. But in this room, we’re just trying to figure out *how* to get Phoenix deployed.”

“Which is not going terribly well, either,” Wes mutters loudly to some nervous laughter.

“I can see that some of you have not bought into our mission,” Sarah says, appraising everyone in the room. “Well, as I mentioned in the Town Hall, the skills that got us here are not necessarily the same skills that will take us to where we need to go. As leaders, we need to figure out if we have the right people on the bus. I’ll be sure to keep Steve apprised. I know this release is personally very important to him.”

Upon hearing Steve’s name, Chris looks at Maxine with disbelief and then covers his face with both hands. *Nice job keeping a low profile*, Maxine tells herself.

“Okay, Sarah, that’s enough,” Bill says, standing up. “Let’s go apprise Steve about some of these problems and let the team figure out how to execute the release. We’re only getting in the way here.”

“Yes, Steve needs to hear about this,” she says. Sarah turns to leave, but then looks back at Maxine. “I like that you say what you think. If you’re available sometime this week, let’s get lunch. I’d like to get to know you better.”

What the... Maxine freezes like a deer in headlights.

“As women, we really need to stick together, don’t we?” Sarah says with a wink.

With a frozen smile, Maxine says, “Uhh...thank you—I...I’d love to.” Immediately, she hates herself, embarrassed that so many people just witnessed her lying so baldly.

“Let’s make it so,” Sarah replies with a warm smile. “And if you need someone to mentor you, I’d be happy to.” She looks at her phone and says, “That’s Steve. He needs something from me. I’ll leave you to it. Remember, we all need optimism.”

When Sarah is gone, Maxine lets out a long breath, not quite believing what happened. She knows how important it is to have a great network, able to find people who can help get important things done. But she’s not terribly excited to be associated with Sarah, no matter how influential she is. Maxine is very picky about who she associates with.

For the next hour, Maxine drifts between the various groups as the huge release team tries to fully understand what is required to support the Phoenix launch. There are at least twelve different technology stacks that need to be deployed, more than Maxine had estimated during her build archaeology.

She knew of the various application servers on Windows and Linux, and the front-end applications that run on the web, but she totally forgot about the two mobile applications (one for the iPhone and one for Android), and those all collectively hit at least ten different back-end systems from across the business, all of which required changes in order to accommodate Phoenix.

She had also forgotten that when you throw the Operations teams into the mix, the number of teams involved more than doubles, because getting all those applications running in production requires the server administration teams, virtualization teams, cloud teams, storage teams, networking teams...

All this reminds Maxine of why production deployments are some of the most complex activities of any technology organization, because they require so much coordination between so many different parts of the organization. And Phoenix wasn't just any deployment—it was designed to change how almost every part of the organization interacts with the customer.

The more Maxine hears, the worse she feels. It seems impossible that they can get everything right the first time with this many moving parts. Getting an environment required Maxine to open up scores of tickets, and she still wasn't successful. She's guessing that deploying Phoenix will require hundreds, or maybe even thousands, of tickets.

The project manager in the group she's sitting with says, "Won't we need a bunch of firewall changes too? Not just to external traffic. I don't think some of these systems have ever talked to each other..."

Maxine raises an eyebrow. She hears more groans around the group. "Oh, great. The firewall teams usually need at least four weeks to get change requests through," says the woman who Maxine learns is Patty. "You think our change management process can be slow? We're speed demons compared to Information Security."

Suddenly, Maxine hears a door slam open behind her, and Patty looks up. "Well, speak of the devil. Here's John, our chief information security officer. This should be fun..." she says.

John is in his late thirties. He's about twenty pounds overweight, but his clothes are still baggy on him. Like in an old Western, John is flanked by two people—one male and one female engineer, who looks vaguely familiar. "At last, I've found you all," John sneers, looking around as if he were a sheriff who had hunted down a group of outlaws. "I'm here about this mad plan to deploy the Phoenix application. This deployment will only happen over my dead body."

The woman behind John suddenly looks embarrassed, as if she's seen John say this before. John continues, "The Phoenix Project has millions of lines of new code, and we cannot responsibly deploy it without my team testing it for vulnerabilities. We just came out of a very, very interesting meeting with the auditors, and trust me, they aren't going to take it very kindly if we put something into production that jeopardizes our compliance posture."

"I have it on *pretty* good authority that the CIO and VP of IT Operations were just fired over some compliance audit findings that were no longer tolerable," John continues. "Let that be a warning to you that compliance is not just a moral obligation or a set of contractual obligations... it's also the law."

Maxine wonders how many times John's rehearsed that line. *It's a pretty good line*, she acknowledges.

Kirsten says from the front of the room, "As you know, the decision to ship Phoenix came straight from the top—Steve Masters, the CEO, and Sarah Moulton, the SVP of retail operations. In fact, Sarah was just here, reminding us of that. The release is scheduled to start at five tomorrow so that everything is live when stores open on Saturday morning."

"That's what you think, Kirsten," John says. "I'm going to go speak with Steve right now. Rest assured that I *will* stop this madness."

He turns to Wes. "You were there at the meeting with the auditors—tell them how serious this is and why there is no way the production release can happen tomorrow!"

Wes quickly replies, "No—leave me out of this, John. That train has left the station, and you can't put the toothpaste back in the tube. The only thing we can do is figure out how to keep this rocket from blowing up on the launch pad and killing us all. Pardon the mixed metaphor," he says with a loud laugh, looking around the room to see who's with him.

"Or was that a simile?" Wes asks suddenly, with a puzzled look on his face.

The woman behind John says in a deadpan voice. “It’s a metaphor, Wes. When you say something ‘is a pile of crap,’ that’s a metaphor. When you say that something ‘is like a pile of crap,’ that is a simile.”

“Thanks, Shannon,” he says with a big smile. “I’ve always gotten those confused.”

John glares at Shannon, and then says angrily to Wes, “I will *not* leave you out of this, Wes. It is your moral responsibility to stop this release!” He turns to the whole room. “It is all of your moral responsibility to stop this release! You all know where I stand—as I said, this release will be deployed over my dead body.”

Wes mumbles, “We can always hope.”

Maxine hears some nervous giggles as John and his posse leave. Kirsten stands up, looking a bit uncomfortable. “Well, I should take a moment to say that we made a commitment to deploy Phoenix on Friday. But if any of you feel you have a, umm, moral obligation to not participate in this release, please let me know.”

Wes chortles. “Kirsten, going down this path is almost certainly the stupidest thing I’ve seen in my entire career . . . but to support the team, I promise we’ll all do what we can.” With an air of weary exhaustion and resignation, he continues, “Let’s just get it over with.”

Maxine looks around, thinking about the sudden, surreal appearance of Sarah and then John. She’s reminded of *Redshirts* by John Scalzi and Wil Wheaton, a funny book loosely based on a *Star Trek*-like universe. It’s written from the perspective of one of the redshirts, the nameless low-ranking characters wandering in the background of the show, who learns that interacting with any of the bridge officers is bad news. Whoever is chosen to beam down to the planet with the officers is doomed to die in bizarre ways: an Alteran blood worm, mind virus, carnivorous plants, an errant Klingon disruptor blast. In the book, the redshirts plant sensors everywhere to detect when the equivalents of Captain Kirk or Commander Spock come below deck so that they can hide.

She is disheartened by how Parts Unlimited executives, the bridge officers, are so disconnected from the daily work of the “redshirts” in the technology organization. It was not helpful for Sarah to remind everyone of how “saving the universe” depends on Phoenix. And it was not helpful for John to appeal to their “moral sense of correctness.”

We all know the threat the company faces is real, she thinks. The job of the bridge crew is to ensure the company strategy is viable, not to remind them of the strategy or to micromanage everyone to death. Their job should be to ensure everyone can get their work done.

How did this all come to be?

Maxine drags herself back to her desk with a sandwich, exhausted from the endless Phoenix release meetings, surrounded by everyone who has similarly been sucked into the launch vortex. Oddly, she also sees some people happily working at their desks, as if it were just a regular day.

Curious, she asks one of the them why he doesn't seem very worried. He replies with a puzzled look, "I'm a developer—I work on features. I give them to QA and Ops to test and deploy. Then I work on the features for the next release. That keeps me plenty busy."

Maxine leaves, boggling at what he said. She has never in her career abdicated all testing and deployment to someone else. *How can you create anything of value if you don't have feedback on how it's used?* she thinks.

When she gets to her desk, Kurt is there with a black three-ring binder. Seeing her, he flashes a big smile. "I have a present for you!"

It's an eighty-page document full of tabs. Just scanning the section headings makes her heart leap—they're the painstakingly assembled Phoenix build instructions, complete with links to documents, license keys, step-by-step tutorials, and even links to a bunch of videos. One is titled "Getting your uberjar to run in our (very) crazy, screwed up production web cluster (8 min)," and another is "How to monitor your apps despite our Ops groups (12 min)."

She sees twenty-character hexadecimal strings of activation codes and license keys. She sees user names and temporary passwords for network shares. Best of all, there's a link to a four-node virtual machine cluster *with* administrative access! That means Maxine will be able to do whatever she wants without having to fill out another service desk ticket!

She's speechless. She feels her eyes tearing up. *Over license keys?*

She wonders for a moment whether she has lost all sense of perspective. But after being stuck inside the Phoenix Project, having someone actually care about what she needs is...so unexpected and so utterly appreciated.

Maxine is reminded of when she and her family volunteered for a day to help new refugee families. She remembers how her then ten- and eight-year-old kids reacted when families cried when they were given food, soap, and laundry detergent.

There is nothing so rewarding as providing something to someone who really needs your help. She needed help and she received it.

Elated, Maxine flips through the document. She sees a long list of Windows registry keys that need to be set. “Don’t worry, Maxine,” Kurt says, politely ignoring her emotional reaction. “You have the electronic version of this in your inbox, so you can copy and paste everything.”

With a twinkle in his eye, he adds, “And there’s a link to a wiki page where you can incorporate any notes if we missed something. There’s a bunch of people who really appreciate your work. We’ve been trying to crack the Phoenix build puzzle for months! But we’ve never been able to work on this full-time. Your notes helped us put all the pieces together. This saved us months of work!”

Maxine’s brow furrows. She has no idea what Kurt is talking about, but she doesn’t even care. “Thank you so much! I can’t tell you how much this means to me. How can I ever repay you for this?”

“Anything to help another hard-working engineer trying to help other engineers be productive,” Kurt says, laughing. But with a serious look on his face, he adds, “If you want to meet the people who made all this happen, despite considerable adversity and huge obstacles that typically prevent feats of greatness like this, come to the Dockside Bar tonight at five. We meet there on Thursdays.”

“Wait, hang on a second,” Maxine says, suddenly suspicious. “If this all works, how come everyone isn’t using it?”

“That is a great question, with some very surprising answers,” Kurt says. “The short version is the ‘official build team’ hasn’t exactly authorized these. They seem to view our efforts as a nuisance, or worse, as competition. Which, on the eve of the biggest and potentially most risky application launch in the history of the company, sure does seem odd, doesn’t it?”

“But by all means, if you like what we’ve done, feel free to share it with anyone who needs it. I can explain more tonight. Please, try to join us at five—there’s a bunch of people who are dying to meet you!” he says. “And good luck with the build!”

Maxine opens up a terminal window on her laptop and starts following the instructions Kurt gave her. Her excitement grows when she realizes that this might be an actual, working Dev environment.

She's exuberant when she's able to log in and type "make" on a command line, which starts streaming screenfuls of happy output onto her screen.

She's delighted as she sees files getting compiled, binaries getting linked, programs being copied, build tools being installed and run... the output keeps going, and going, and going...

Amazingly, things are still building for ten more minutes...fifteen minutes...thirty minutes...she's relieved as it keeps going without an error, but starts to become alarmed at the size of the Phoenix build. It's *huge*.

Forty-five minutes later, she can't hold off going to the bathroom any longer; she was too afraid she was going to miss something if she stepped away. She hurries there and back and is relieved to see that the build hasn't failed, still generating endless output in her terminal window.

She scrolls through the history to see if she missed anything interesting. She decides to skip the next release team meeting just so she can watch the continuing build, which seems a bit irresponsible, but she knows that having a great build process is key to having a good code deployment and release process. And maybe with the help of these mysterious benefactors, she's on the verge of finally conquering the Phoenix build.

The build output is hypnotic and educational, because she's seeing some components of Phoenix for the first time. There's Java JAR files, .NET binaries, Python and Ruby scripts, and lots and lots of bash scripts.

Wait, is that a remote shell and installer that popped up? Before she can figure out what it is, the window is gone. Maxine's awe and concern at the size and variety of Phoenix continues to grow.

She's about to scroll back further when she sees Eclipse being downloaded from somewhere. *What in the world?* she thinks. Twenty minutes later, she could have sworn that she saw an InstallShield installer, but she knows she's getting tired and might be imagining things.

Honestly, after another hour of watching build output, she's having trouble staying focused on the screen. But she can definitely see the

different personalities and tech stacks of the different teams working on Phoenix. She had no idea there were so many.

This is crazy, she thinks. There can't be this many teams working on Phoenix, right? And she wonders how any one person could possibly understand the system as a whole, especially when it's built from so many different technology stacks.

Maxine isn't usually a fan of rigid standardization, but she's not a fan of everyone getting to choose whatever they fancy in the moment. Each decision is a commitment to support it for years or even decades—these are decisions that go far beyond just one team.

Like most developers, she's very superstitious that if she stops watching the build, it will fail. Finally, nearly three hours after she started the build, she sees the scrolling output from her build window stop. Her heart falls when she reads:

```
builder: ERROR: missing file: credentials.yaml
```

Damn! She's guessing that she needs a login credential that she doesn't have.

She texts Kurt and he quickly replies:

```
Ah, yes. For that you need to open a ticket to get your login tied to your  
ActiveDirectory account. Only Susan can issue those. Contact info coming.
```

Instead of emailing Susan, Maxine goes to Susan's desk and learns that this missing file contains a cryptographic certificate that comes from some distant security group. Susan searches through years of emails to find how to get a new one. When she finds it, Maxine takes a picture of the email address with her phone.

She is so close to getting a Phoenix build going!

CHAPTER 5

• *Thursday, September 11*

Still on a high after getting so far on the Phoenix build, Maxine hops in her car to make the five-minute drive to the Dockside parking lot, right on time for Kurt's mysterious meeting.

She suspects that the shiny new Lexus IS300 in the parking lot is Kurt's. She doubts it's the Datsun 300 she parked next to. It's surprising that the meeting is at Dockside. It's not one of the usual hangouts for technology people, but she knows it's a longtime favorite for many of the factory workers.

Maxine asked some people about Kurt that afternoon. Three people gave her enthusiastic endorsements, describing how competent and helpful he was. One development manager in her old group called him one of the smartest people in the entire technology organization. Curiously, one of her colleagues texted her:

Kurt? He's not the sharpest knife in the drawer, which is why he's stuck in QA. He's also really nosy. Why do you ask?

This made Maxine even more curious. *What exactly is Kurt up to?* His gift of the binder probably saved her months of waiting. But what is his motivation? He clearly has an inside track on getting things people need. She's pretty sure he isn't pilfering corporate resources—and even if he was, why would he give them to her?

As she walks in the door, she's immediately hit by the smell of hops. She hasn't been here in years. She's relieved to see that it's much cleaner and brighter than she remembers. There's no longer sawdust on the floor, and it's more spacious than it seems from the outside.

The bar is half-full, but it's loud—maybe because of the cleanly swept cement floor.

Seeing her, Kurt smiles and waves her over to a group of tables on the far side of the room by some booths. "Hey, everyone, meet Maxine,

the newest member of the Rebellion if I can help it. She's the person that I've been telling you all about."

She immediately recognizes the cranky developer who backed her up in the Phoenix status meeting about environments and is startled to see the petite woman named Shannon who was with John earlier today. There's another man in his late thirties sitting next to someone who looks out of place—he's in his fifties and wearing a bowling shirt. Next to him is Brent, who she also saw in the Phoenix release meeting. He and Shannon are the youngest people at the table.

Everyone has an open laptop in front of them. Suddenly, she wishes she had hers with her—she'd gotten out of the habit of carrying it around lately because she's had so little to work on.

"You remember Dave?" says Kurt, gesturing at the cranky developer. "He's one of the Dev team leads. He complains a lot, but he's probably banging the drum loudest on the need to pay down technical debt and modernize our architecture, platforms, and practices."

Kurt laughs. "The reason Dave is so good is that he never asks for permission!"

Cranky Dave raises his glass at Maxine with the smallest of smiles, as if smiling causes him physical pain, then takes a sip of his beer. Up close, he looks older than her. "Breaking the rules is the only way anyone can get anything done around here," he grumbles. "Can't do anything without twenty meetings." Cranky Dave pauses. "You know, that's the best compliment Kurt has ever given me. You've probably noticed that he's running his own black market inside the company, right?"

Kurt laughs, clearly not bothered by the characterization. "I'm just trying to solve people's problems. If I'm guilty of anything, it's that I care too much about the success of Phoenix, and even the whole company, to allow bureaucracy to kill it! And if that's a crime, I plead guilty! It's a pity no one will ever give us a medal for the great work we do. The satisfaction of helping people must be reward enough, right?"

Everyone groans, and someone from across the table hollers out, "Good one, Kurt."

Ignoring the banter, Kurt points to the man in his late thirties who is wearing a funny vendor T-shirt. "This is Adam, one of my test engineers. But don't let his title fool you—he's a developer at heart, and he's also one of the best infrastructure people I've ever met.

“You can thank him for all those virtual machines and pre-built containers you got—he built them all. And that’s only a fraction of what Adam does. His day job is helping automate a big chunk of the legacy test suite that we inherited from an outsourcer.”

Adam smiles sheepishly. “Actually, Brent over there did most of the work,” he says. “He’s an ace at infrastructure. We’ve been working together to try to automate environment creation for over a year. It’s been a tough road, working evenings and weekends, because it’s not officially sanctioned. Despite all the dead ends and cul-de-sacs, we’re proud of what we’ve been able to achieve.”

“Your build notes were awesome, Maxine. Brent here almost fell over and died when he was reading through them. He’d been trying to piece that together for months,” Adam says.

Brent smiles at Maxine. “That was amazing detective work, Maxine. Documenting all those environment variables was so helpful!”

“Let us know how the environment works for you,” Adam continues. “It’s such a pain to get things from Operations through normal channels, so we scraped together enough hardware to build a cluster big enough to support a couple of teams. Now you can get an environment on demand, without needing to open up a ticket.”

Maxine blurts out, “Wow, thank you so much. The environment worked! I got three hours into the Phoenix build with it until it failed because of a missing certificate.”

“Wow! That’s amazing,” Brent says.

“So where did all that hardware come from if not from Operations?” Maxine asks.

Adam smirks. “Kurt has his ways—a little from here, a little from there, you know? Kurt keeps saying that it’s best not to ask where it comes from... I’ve always suspected there are a bunch of people missing entire server clusters if they’d bother to check.”

Kurt feigns a hurt expression. “Server-hoarding is a huge problem,” he says. “Because it takes so long for Ops to get anything to us, people always ask for way more than they need. And that makes Ops’ job harder and lengthens the lead times for everyone else, making the shortages even worse! It’s like being in the old Soviet Union, where you have to wait in line for everything. You could say we’re creating a secondary market to ensure that some of those unused environments go where they’re needed most.”

You know, to ameliorate the mismatch between supply and demand,” he says.

Cranky Dave mutters, “Don’t get him started,” rolling his eyes as Kurt lectures like a professor.

Adam adds, “But Dave is right—Kurt *is* running a black market.”

“Pay no attention to them, Maxine,” Kurt continues. “Next down the table, we have Shannon, a security engineer who works on building automated security tools. She spent nearly five years in the data warehouse team before that. She’s currently working with Brent, experimenting with some machine-learning and data visualization toolkits and standing up some big data infrastructure, trying to get ahead of some of the marketing initiatives that we know are coming. You probably remember her from the full-scale red-team exercises that she ran last year.”

Maxine smiles. That’s why she looked so familiar. She definitely remembers—it was the first time she had been the target of a no-holds-barred penetration test. They had tried to plant malware by getting physical access to the manufacturing facilities, sending emails with malicious links, pretending to be company executives, and, in one case, one of their most critical vendors.

She had been very impressed. *It takes a lot of balls to run those types of exercises*, she thinks. Maxine remembers one person being fired for trying to do one because he made a bunch of people look bad.

Shannon looks up from her laptop and says, “Nice meeting you, Maxine. I remember your group. You were one of the best-prepared in the whole company. I was very impressed that everyone in your division knew not to click on links in emails, no matter how official they looked. Someone did a great job training everyone.”

Maxine nods with respect, saying, “Nice meeting you, Shannon. We spent weeks fixing the problems you all found. Nice work.”

Shannon looks back down at her laptop and types something. Suddenly, she looks up at everyone and says, “Oh, by the way, sorry about that episode with John. He’s such a tool. But he’s my boss.”

Everyone laughs, and several people imitate Shannon’s expression from earlier today.

“Up next is the aforementioned Brent, who has his hands in everything infrastructure related,” Kurt continues. “If it connects to AC power, Brent has probably mastered it. Networks, storage, compute, databases.

But he's not just good with a screwdriver, he's always on the frontier of automation. Unfortunately, he's so good at what he does, everyone seems to have him on speed-dial. And he's on pager duty way too often, which we're trying to fix."

Brent merely shrugs his shoulders. Suddenly, the camera flash on his phone flickers and notifications flood his screen. He picks up his phone and mutters, "Dammit, another outage call. I probably need to jump on this." He drains the rest of his beer and starts dialing his phone.

"Yeah, that's a real problem," says Kurt, watching Brent walk away. "We've got to bring some sanity to his work life. He's brilliant, but because of the way people dump things on him, he hasn't been able to go on vacation without a pager for years..."

He pauses. "In the meantime, last but not least is Dwayne," says Kurt, gesturing to the oldest person at the table. He's not only dressed differently than everyone, his laptop is different too—it's a beast with a massive screen. "He's a senior database and storage engineer from Ops and was the person who brought Brent into this group. They conspire all the time to find better ways to manage infrastructure."

Maxine smiles. To most people on the Phoenix Project, centralized Ops are merely the people on the other side of a ticket. They're the people everyone is always complaining about. But clearly Kurt and this motley crew have a different way of working, bypassing the normal organizational lines of communication, however informal.

Dwayne reaches across the table, extending his hand. "Great to meet you, Maxine."

Maxine realizes that Dwayne is wearing an actual bowling shirt, complete with his initials sewn on it, "DM," and a faded mustard stain right next to them.

"Dwayne has been trying to get automation initiatives going for years, but he and Brent always get shot down," Kurt continues. "So, they've been helping Adam build up our own infrastructure instead. He knows almost everyone in Operations, and he can usually get them to do anything. Like earlier this week when we needed a firewall port opened between two internal networks. Dwayne made that happen."

"All in a day's work," Dwayne says with a friendly smile. "But to be fair, Kurt is really the master of getting the impossible done... I'm just learning from him!"

Maxine is certain Dwayne is exaggerating. Dwayne looks like he's in his mid-fifties. Just how much could he learn from a young guy like Kurt?

Kurt leans back in his chair, arms spread out. "Maxine, your work cracking the code of the Phoenix builds has impressed us all. We are in awe of the technical and social skills you displayed to successfully hunt down almost all the pieces of the environment, which required incredible perseverance, focus, and never taking 'no' for an answer!"

Confused, Maxine looks around, but she sees everyone nodding at her, genuinely impressed at her work. Kurt continues, "We invite you to be a part of the inner-circle of the 'Rebellion.' We're recruiting the best and brightest engineers in the organization, training and preparing in secret for the right time to overthrow the Empire, the ancient, powerful, and unjust order that definitely needs to be toppled."

Everyone chuckles, and Cranky Dave raises his glass, shouting with a laugh, "To the overthrow of the Empire!"

Confused, Maxine looks around the table. These are people from Dev, QA, Security, and Ops—a very unlikely group of people to be socializing, let alone working together. And she notices that everyone has a small sticker of the Rebel Alliance from Star Wars on their laptop, just like the X-Wing pilots wore on their helmets. She grins at their subtle but subversive badges of solidarity.

Seeing Maxine toast with an empty hand, Kurt jumps up. "What do you want to drink?"

"A pinot noir, please."

Kurt nods and heads toward the bar, but before he can take three steps, a tall and somewhat overweight man with graying hair walks up to him and gives him a big hug. In a loud and boisterous voice, he says, "Kurt! Good seeing you again, my young friend. What do you need?"

Noting the attention that Kurt's group gets from the bar staff, Maxine guesses they must come here often. She smiles. For the first time since her move to the Phoenix Project, she feels like she's in the company of kindred spirits.

"Who are you people? Why are you all here? What are you possibly trying to achieve?" she asks quickly, while Kurt is at the bar.

Everyone laughs. Dwayne says, "As you know, we're a huge Kumquat database shop, which is what I cut my teeth on. I want to migrate us to MySQL and open-source databases wherever we can, because I'm tired of

sending millions of dollars each year to an abusive vendor. We're figuring out how to engineer our way there."

Looking around, he says to everyone, "Other companies have done this already. I think that anyone who is still paying Kumquat database maintenance fees is simply too dumb to migrate off it."

Maxine nods in approval. "Smart thinking! We've saved millions of dollars in my old group doing this, which we can now spend on innovation and other things the business needs. And it's been fun. But why this crusade for open-source software?"

"I'll tell you why," Adam says. "For almost five years, back when I was in Operations, I had a team that kept getting pager alerts at two in the morning for some middleware we used. In almost every case, it was because of their database driver. I was the guy who had to generate a binary driver patch! After all that work, the problems started happening again six months later, because when the vendor released their patches, they didn't integrate my fixes into their code. Next thing you know, we're all up at two a.m. doing the same thing over again."

Maxine is impressed. *Adam has great kung fu too. And so does everyone else here.*

Cranky Dave frowns. "I've been at Parts Unlimited for almost five years, and I can't believe how the bureaucracy and silos have taken over. You can't do anything without first convincing a bunch of steering committees and architects or having to fill out a bunch of forms or work with three or four different teams who each have their own priorities. Everything is by committee. No one can make decisions, and implementing even the smallest thing seems to require consensus from everyone. Almost everything I need to do, I have to go up two levels, over two levels, and down two levels just to talk with a fellow engineer!"

"The Square!" cries out Adam, and everyone laughs.

Dwayne chimes in. "In Ops, we often have to do the return path—up, over, down, and then back up, over, and down before two engineers can finally work together to get something done."

"I want to bring back the days when a developer could actually create value for someone who cares, easily and quickly," Cranky Dave says. "I want to build and maintain something for the long haul, instead of shipping the 'feature of the day' and dragging all this technical debt around."

Cranky Dave is on a roll. “This company is run by a bunch of executives with no clue about technology, and project managers who want us to follow a bunch of arcane processes. I’ll scream at the next one who wants me to write a Product Requirements Document.”

“The PRD!” everyone shouts, laughing. Maxine raises her eyebrows. Those made sense decades ago, when you wanted written justification before you wasted a bunch of developers’ time. But now you can prototype most features in the time it takes to even write one page of a PRD. One team can now build things that used to require hundreds of people.

Kurt sits next to Maxine, putting a glass of red wine in front of her. “We’re like the redshirts in *Star Trek* who actually get the real work done.”

“I was literally thinking that earlier,” Maxine says, smiling.

“Right? You’ve seen firsthand the reality bubble the bridge crew is in,” Kurt says. “They know the Phoenix Project is important, and yet they couldn’t have come up with a worse way to organize everyone to achieve it. They outsourced IT, brought it back in, outsourced one piece, maybe two pieces, shuffled them around... In many areas, we’re organized as if we’re still outsourced, and nothing can get done without permission from three or four levels of management.”

“Kurt’s right,” says Cranky Dave. “We’re just another cost center, little cogs in a big machine that can be easily outsourced to some random corner of the globe. We’re viewed as replaceable and fungible.”

“That’s why I’m here, Maxine,” Shannon says. “We could build a world-class technology organization and create an engineering culture. That’s how we survive and innovate for our customers. And my dream is that *everyone* is the custodian of company data. It’s not just the job of one department.

“In Steve’s Town Hall, he talked about how we’re being disrupted and how we need to compete with the e-commerce giants,” she says. “Well, we can only win by innovating and understanding our customers, which we can only do by mastering data. I think the capabilities we’re building are the future of the company.”

Everyone cheers and hoists their glasses.

After everyone is done toasting each other, Dwayne turns to Kurt and asks, “So, how did the meeting go with your boss? You said you pitched William on funding an automated testing pilot.”

Everyone leans in.

“You know, I really thought he was going to go for it. I had testimonials from two of the Dev managers and a product owner about how great it would be. One of them had this great line: ‘Without automated testing, the more code we write, the more money it takes for us to test.’ Ha! I really thought that would scare the pants off of William!” Maxine can feel the mood deflate around the table.

“Don’t keep us in suspense, Kurt. What did he say?” prompts Dwayne.

“Son, let me explain something to you,” Kurt says, in a shockingly good impersonation of William. “You’re young. You clearly don’t understand how this game works. We’re QA. We protect the organization from developers. It sounds to me like you’ve been hanging around too many of them. Do not trust them. Do not get chummy with them. You give developers an inch, and they’ll take a mile.”

Maxine laughs at Kurt’s uncanny impression.

“Son, you’re a pretty good QA manager with a half-million-dollar budget.” Kurt’s on a roll. “If you do your job well, you can be like me with a three-million-dollar budget. And if I do my job well, then I’ll get promoted and have a \$20 million budget. You go around automating your QA, your budget shrinks instead of grows. I’m not saying you’re stupid, son, but you sure don’t seem to understand how this game works.”

Maxine laughs with everyone else. She is sure Kurt is exaggerating.

“William is like a union leader, not a business leader,” Shannon says. “He only cares about growing his union membership dues, not about what’s right for the business. You see the same thing inside Ops and even Infosec.”

A frown crosses Dwayne’s genial face. “Trust me, it’s way, way worse in Ops. At least Development is seen as a profit center. In Ops, we’re a cost center. The only way to fund infrastructure is through new projects. If you don’t find new funding sources, you’re screwed. And if you don’t spend your whole budget, they’ll take the money away from you next year.”

“Ah, the project funding model... Another big problem here at Parts Unlimited...” Kurt says, as everyone groans in agreement.

“So, what’s your plan now, Kurt?” Dwayne asks.

“Don’t worry, Dwayne. I’ve got another plan,” Kurt says, confidently. “We’re going to lie low and keep doing what we’re doing, looking for new potential customers and recruits. We keep our eyes and ears open for opportunities to get in the game.”

“Oh, that’s a great plan, Kurt,” Dwayne says, rolling his eyes. “We hang out at a bar, complain, and drink beer. Brilliant.”

Dwayne leans over to Maxine, explaining, “It’s actually not that crazy. It’s like in that movie *Brazil*, where the number-one fugitive is the rogue air conditioner repairman who fixes people’s air conditioners because Central Services never gets around to it. That’s us. We’re always on the lookout for places we can help. It’s a great way to make friends and find potential new recruits for the Rebellion.”

“What?” she says in disbelief. “That can’t work, can it?”

“Well, it’s how we got you here, isn’t it?” Dwayne says with a big smile.

“I’m working all the angles,” Kurt continues. “I’m even thinking about asking William if I can have a meeting with him and his boss, Chris. I’d tell William that it’s really important to me that Chris hears my proposal and that I want him there.”

Wow, Maxine thinks. *That’s pretty gutsy, maybe savvy, and probably fatal.*

“I’ll keep you posted,” Kurt says. “Okay, who has new information or intelligence to share?”

Shannon updates everyone on a nascent data analytics group in Marketing she’s been working with and how she’s setting up a meeting between them and Kurt. “They’re working on a bunch of projects to increase customer promotion conversion rates, and boy, they really need help. They’re not even using version control! They’re struggling with basic data engineering problems, and they’re still trying to get what they need from the data warehouse people,” she says, visibly bothered by their suffering. Kurt quickly pulls out an org chart on his laptop.

He asks her, “Another data analytics project? Who’s funding it? How much budget do they have? Who’s leading it?” As she talks, he takes notes.

When it’s his turn, Dwayne says, “I’ve got bad news. The Phoenix release caught everyone in Ops flatfooted—no one had it on their radar until last week. No budget was assigned to support it. Everyone’s scrambling to find enough compute and storage infrastructure. This is the biggest launch we’ve done in almost twenty years, and everything we need, we don’t have enough of. It’s bad.”

“Holy shit,” says Adam.

“Yep,” Dwayne says. “I’ve been trying to tell everyone for months, but no one cared. Well, now they do, and everyone’s dropping everything to

support the Phoenix launch. Today, I heard someone trying to work with procurement so they can break the rules and order outside of the annual ordering process.”

Despite the crisis, bean counters are still bean counters, Maxine thinks.

“Everyone is still scrambling to get environments ready for the release tomorrow,” Dwayne says. “No one has any build specifications that Dev and Ops both agree on. I gave them the ones we wrote, and they pounced on them and started using them right away. But still, this release is going to go real bad, real fast.”

“I think you’re right,” Maxine says. “I’m really, really good at this stuff, and I spent nearly a week trying to get a Phoenix build going. If it weren’t for the environment that Kurt gave me, I’d still be at square one. With the release team only starting today and the launch tomorrow, they are in big trouble.”

Kurt leans forward, a serious look on his face. “Tell me more.”

Suddenly, Maxine realizes why she was invited and that Kurt is no dummy after all.

Over the next twenty minutes, Maxine describes her experiences, reading from her work diary, which she can access from her phone. She mentally kicks herself again for not bringing her laptop. Everyone takes notes, especially Brent when he returns. He and Adam pepper her with questions as if she were a captured secret agent being debriefed by the CIA. Everyone’s interested in how she was able to piece together the Phoenix build puzzle faster than anyone else had done. They ask lots of questions about who she talked to, what teams they were on, where she got stuck, and so forth.

“That’s really impressive, Maxine,” says Cranky Dave. “Years ago I put together a build server that my team could use on a daily basis. But that was when Phoenix only had two teams; now we have over twenty. The build team is completely out of their league, with people who, I’m sorry to say, are the people who didn’t have enough experience to be application developers.”

Adam says, “We’re really close now. I think we’re down to just one missing signed certificate for the payment processing service.”

“He’s right,” Brent says. “Maxine, can you show me the build logs? I bet we can create that certificates ourselves—it wouldn’t actually be valid, but it would be good enough for a Dev or Test environment.”

Maxine curses, mentally picturing her laptop still on her desk. “I can show you first thing tomorrow,” she sighs.

“This is great, people. Here’s what we still need: we need an automated way to create environments and perform code builds,” Kurt says, counting off on his fingers. “We need some way to automate those tests and some automated way to get those builds deployed into production. We need builds so that developers can actually do their work.

“So, who’s willing to volunteer some of their time to help Maxine get those Phoenix builds going?” Kurt asks. To Maxine’s surprise, all hands shoot up.

“Maxine, would you be able to lead this effort, with the help of any or all of these willing and talented volunteers?” Kurt asks.

Maxine is overwhelmed by the sudden support of all these people. Last week, she was unable to get help from anyone and was thinking about interviewing at other places. Suddenly, she’s not so sure.

She takes a moment to collect herself and says, “Yes, I’d love to. Thank you, everyone. I look forward to working with you all.”

Maxine is excited. She’s genuinely amazed at what this group has been doing and that she’s been chosen to help. *I’ve finally found my tribe*, she thinks. *And this is what an effective network is all about—when you can assemble a group of motivated people to solve a big problem, even though the team looks nothing like the official org chart.*

I’m pretty sure I’ll learn and achieve more with this group than I would by having lunch with Sarah, she thinks. She wonders if she’s being small-minded and petty. She still wonders if she should take the meeting or just wait for Sarah to forget about her.

“Excellent! Let me know if you need anything from me,” Kurt says to the table. To Maxine, he says, “We try to meet every week. We typically have only two agenda items. First, we share intelligence on who needs help and other people to potentially recruit. After that, we usually share about something we’ve learned lately or new technologies that we think could change the game here at Parts Unlimited. I propose we add a third agenda item, which is discussing the progress of Phoenix builds, yes?”

Everyone nods.

Kurt looks at his watch. “Folks, one more thing before we adjourn. I’m starting a betting pool on when the release team will have the Phoenix application successfully running in production.”

The most optimistic bet comes from Cranky Dave, who guesses Saturday at two a.m., fully eight hours after the deployment starts. Most bets are scattered between three and nine a.m., with Maxine betting six a.m.

“After all,” she says, “the in-store point of sales systems need to be up by eight on Saturday morning.”

To everyone’s surprise, Dwayne bets Sunday evening, “You people have no idea how unprepared we really are for this release—this one will go down in the record books.”

From: Alan Perez (Operating Partner, Wayne-Yokohama Equity Partners)
To: Dick Landry (CFO, Parts Unlimited), Sarah Moulton (SVP of Retail Operations)
Cc: Steve Masters (CEO, Parts Unlimited), Bob Strauss (Board Chair, Parts Unlimited)
Date: 3:15 p.m., September 11
Subject: Maximizing Shareholder Value **CONFIDENTIAL**

Sarah and Dick,

Thanks for the call today, and for walking me through the strategy and the Phoenix Project. I agree that an omni-channel strategy is required for any retailer to survive these days, especially given the e-commerce threat. And selling products manufactured in-house with low cost of sales is intriguing.

However, I’m concerned at how much cash you’ve diverted from Manufacturing (\$20MM) to invest in Retail over the last three years, with no obvious return. The question becomes what return you could have gotten if this were invested elsewhere in the business or just returned to shareholders. As of right now, investing in lottery tickets would have made more economic sense.

Stories about innovation and omni-channel are nice, but the board needs more than stories and PowerPoint slides.

Good luck with the Phoenix release tomorrow. I know a lot rides on it.

—Alan

Copyrighted Excerpt
Not for Duplication

CHAPTER 6

• *Friday, September 12*

Friday goes by in a blur for Maxine as the emergency release preparations continue. She sees endless mayhem as Dev, QA, and Ops try to line up hundreds of moving pieces for the deployment. *Dwayne was right*, she thinks. And it's too late to change her bet to Sunday in the betting pool.

At five p.m. the release starts on schedule. There are rumors of last-ditch attempts to call it off, because William, Chris, and Bill are nowhere to be seen. These hopes are crushed when an email comes out from Sarah and Steve, making it very clear that the release is to proceed as scheduled.

Maxine is still at the office at ten that evening. By now, there's a sense of genuine panic that things are going very, very wrong. So spectacularly wrong that even Dwayne, who was the most pessimistic in the Phoenix release betting pool, mutters to Maxine, "This is going worse than I thought it would."

That's when Maxine becomes genuinely frightened.

By midnight, it's clear that a database migration is going to take five hours to complete instead of five minutes, with no way to stop it or restart it. Maxine tries to be helpful, but she isn't familiar enough with the Phoenix systems to know where she would be the most useful.

In contrast, Brent is being pulled every which way, needed for just about every problem, from the huge database meltdown in progress to helping people fix their configuration files. Seeing this, Maxine organizes a team to play goalie, protecting Brent from interruptions and fielding problems that don't require him.

Maxine notices something else. There must be two hundred people responsible for some portion of the release, but for most of them, it's only about five minutes of work. So, they have to wait around for hours to perform their little part in this excruciatingly long, complex, and dangerous operation. The rest of their time is spent watching and . . . waiting.

Even in the middle of this crisis, people are just sitting around, waiting.

By two a.m. everyone realizes there is a very real risk that they are going to break every point-of-sale register in every one of the nearly

thousand stores, knocking Parts Unlimited back into the Stone Age. And with all the promotion that Marketing has been doing, the stores will be filled with angry customers unable to buy what they were promised.

Brent asks her to join a SWAT team to figure out how to speed up the database queries, still nearly a thousand times slower than they need to be in order to handle the expected load when stores open up later that morning.

For hours, she works with a bunch of Phoenix developers and Ops DBAs with her IDE and browser open. They are stunned when they discover that clicking the product category drop-down box floods the database with 8,000 SQL queries.

They are still working on fixing this when Wes pokes his head in the room, “Brent, we’ve got a problem.”

“I’m busy, Wes,” Brent replies, not even looking up from his laptop. “No, this is serious,” Wes says. “The prices have disappeared from at least half of our products on the e-commerce site and mobile apps. Where the price should be displayed, either nothing shows up or it says ‘null.’ Screenshots are in the #launch channel.”

Maxine blanches, pulling up the screenshot. *This is much more serious than slow database queries*, she thinks.

“Dammit, I bet it’s another bad upload from the pricing team,” Brent says after staring at his screen for several moments. Maxine leans over as Brent pulls up various administrative screens and product tables—some are inside of Phoenix and others on systems she doesn’t recognize.

Maxine takes notes as Brent pulls up log files, runs SQL queries against a production database, pulls up more tables in various applications... Only when he opens up a terminal window and logs into a server does Maxine ask, “What are you doing now?”

“I need to inspect the CSV file that they uploaded into the app,” he says. “I think I can find one in the temporary directory on one of the application servers.” Maxine nods.

When Brent squints at his screen, Maxine does as well. It’s a comma-separated text file with column names in the first row—product SKU, wholesale price, list price, sale promotion price, promotion start date... “It looks fine,” Brent mutters.

Maxine agrees. She says, “Can you copy that file into the chat room? I’d like to take a look at it.”

“Good idea,” he says. She imports it into Excel and several other of her favorite tools. It looks fine.

While Wes tries to get one of the development managers on the phone, Brent tries to figure out what is going wrong. It’s almost thirty minutes later when he curses. “I can’t believe it. It’s a BOM!”

Seeing Maxine’s confused expression, he says, “A byte-order mark!”

“No way,” mutters Maxine, pulling up the file again, this time in a binary file editor. She stares at her screen, stunned that she missed it. A BOM is an invisible first character that some programs put in a CSV file to indicate whether it’s big-endian or little-endian. She’s been bitten by this before.

Years ago, a colleague gave her a file exported from the SPSS statistical analysis application, and she spent half a day trying to figure out why her application couldn’t load it as expected. She finally discovered that the file had a BOM, which got interpreted as part of the first column name, which caused all her programs to fail. *Which is almost certainly what is happening here*, she thinks.

Any intellectual satisfaction she feels at understanding this particular puzzle quickly disappears. She asks Brent, “This has happened before?”

“You have no idea,” Brent says, rolling his eyes. “Different problem every time, depending on who generated the file. The most common problems lately are zero-length files, or files with no rows in them. And it’s not just the pricing team—we have data problems like this all over the place.”

Maxine is appalled. The first thing she would have done right away is write some automated tests to ensure that all input files are correctly formed before they allow them to corrupt their production database, and that the correct number of rows are actually in the file.

“Let me guess. You’re the only one who knows how to correct these bad uploads?” Maxine asks.

“Yep,” she hears Wes say from behind her. “All roads lead to Brent.” Maxine jots down more notes, determined to investigate this and do something about it later.

It’s almost two hours later before the pricing tables are corrected. Because of what Brent said, Maxine double-checks the file and is certain that it’s

missing a significant number of product entries. And because the pricing team wasn't part of the release, no one knows how to get a hold of them in the middle of the night (or early morning as it seems to be). Maxine adds some more things to her list of things that she'll insist on building so that this won't happen again.

At seven a.m., Maxine rejoins the database team. They're still working on speeding up queries—but it's too late. An announcement is made that stores are beginning to open on the East Coast.

The Phoenix release is still nowhere near complete. "We're fourteen hours into the launch, and the missile is still stuck in the tube," Dwayne says glumly.

Maxine doesn't know whether to laugh, smirk, or throw up—when missiles are stuck in the launch tube, it's a very dangerous scenario, because at that point the missile is already armed and too dangerous to approach.

At eight a.m., they are still hours away from having a working point-of-sale system. Sarah and her team are forced to train every store manager on how to use carbon paper imprints, and some stores are forced to only accept cash or personal checks.

For Maxine, the rest of Saturday goes by in a blur. She's unable to go home. The Phoenix rollout was more than just a spectacular outage... it was the most amazing example of production data loss Maxine has ever seen.

Somehow, they managed to corrupt incoming customer orders. Tens of thousands of customer orders were lost, and an equal number of customer orders were somehow duplicated—sometimes three or four times. Hundreds of order administrators and accountants were mobilized, reconciling database entries against paper order slips being emailed or faxed from stores.

Shannon texts everyone in the Rebellion, horrified that boxes of customer credit card numbers are being transmitted in the clear—but in the grand scheme of things, it's just another blip in the Phoenix disaster.

At three p.m., Kurt texts everyone:

Not to put light on this big pile of suck, but Dwayne wins the betting pool. Congratulations, Dwayne.

Dwayne replies:

Not worth it! FUUUUUUUUUUUU . . .

He posts an image of a burning tire fire.

By Saturday night, Maxine finally manages to go home and sleep for six hours before coming back to the office. *Dwayne was right, this will go down in the record books*, she thinks glumly.

On Monday morning, Maxine is shocked to see her reflection in a mirror. She looks like crap, just like everyone around her—bags under her eyes, hair stringy. Long gone are her carefully pressed blazers. Now it's jeans and a wrinkled jacket to cover up a stain on her equally wrinkled blouse. Today she doesn't look classy. Like everyone else, she looks like she's recovering from a hangover, having slept in her outfit from the night before.

Since Saturday morning, their e-commerce site has been continually crashing under the unprecedented levels of customer traffic. In a status update meeting, Sarah crowed about what a great job Marketing did promoting Phoenix, then demanded that IT pull their weight.

"She's unbelievable," Shannon mutters. "She created this whole disaster! Is anyone ever going to call her on this?" Maxine just shrugs.

The carnage is unbelievable. Most of the in-store systems are still down—not just the point-of-sale registers, but nearly all of the back-office applications that support the in-store staff.

For reasons that continue to mystify everyone, even the corporate website and email servers are having problems, further hampering their ability to get critical information to people who need it—not everyone has access to the developer chat rooms.

In situations like this, technology failures cascade through the organization, like water flooding through a sinking submarine.

Trying to stay alert, Maxine goes to get more coffee from the kitchen. Dwayne's there doing the same thing. They nod at each other, and he says, "Did you hear we have hundreds of people who can't even get into their buildings because their keycards won't work?"

"What?!" Maxine exclaims, exhausted but laughing. She says, "I was just talking to someone who's trying to figure out why a bunch of batch

jobs aren't running. He's even saying payroll might be delayed again—umm, I'll leave that to other people to fix," she concludes with a small laugh.

"Huh," he muses. "I wonder if we managed to knock out an interface to an HR application. That might explain these strange errors. We managed to screw up everything else."

All day during the recovery efforts, she hears questions like: Why are all those transactions failing? Where are they failing? How did it get into that state? Of the three ideas that might fix the problem, which one should we try? Will it make it worse? We think we fixed it, but is it really fixed?

Once again, Maxine's sensibilities are offended by how entangled all these systems are with each other. It's so difficult to understand any part of the system in isolation.

At times, it was difficult not to feel panicked. Earlier in the day, it looked like the Parts Unlimited e-commerce site was being attacked by an external party actively stealing credit card numbers. It took over an hour for Shannon and the security team to send out an email concluding that it was an application error—if someone refreshed the shopping cart at the wrong time, the full credit card number and three-digit CVV code of a random customer was displayed.

The good news was that it wasn't an external hack. The bad news was that it was a genuine cardholder data exposure event and likely another reason to be front-page news. All the attention and ridicule exploding on social media only added to everyone's stress.

Taking a break, Maxine walks back to her desk. She sees the developer who was so unconcerned with the release last week. He's wearing fresh clothes and appears to be well-rested.

"Rough weekend, I'm guessing?" he says to Maxine.

Maxine stares at him, speechless. He's still working on features for the next release. The only big change for him is that all his meetings have been canceled because most people have been sucked into the Phoenix crisis.

He turns back to his screen to work on his piece of the puzzle, not caring that none of the pieces actually fit together. Or that the entire puzzle has caught on fire over the weekend, along with the house and the entire neighborhood.

From: Alan Perez (Operating Partner, Wayne-Yokohama Equity Partners)
To: Dick Landry (CFO), Sarah Moulton (SVP of Retail Operations)
Cc: Steve Masters (CEO), Bob Strauss (Board Chair)
Date: 8:15 a.m., September 15
Subject: Phoenix Release **CONFIDENTIAL**

Sarah and Dick,

I've been reading the news headlines about the Phoenix release. Not a great start. Again, I question whether software is a competency Parts Unlimited can create. Maybe we explore outsourcing IT?

Sarah, you mentioned the large number of developers you've contracted to help. How long until they are fully contributing? When you grow a sales team, it takes time for new salespeople to carry full quota capacity. Can new developers really be onboarded fast enough to make a difference? Or are we just throwing good money after bad?

Sincerely, Alan

From: Sarah Moulton (SVP, Retail Operations)
To: All IT Employees
Cc: All Company Executives
Date: 10:15 a.m., September 15
Subject: New production change policy

Thank you for all your hard work helping deliver Phoenix to our customers. This is a badly needed step for us to regain parity in the marketplace.

However, due to the harm that we did to our customers because of unanticipated problems caused by poor judgment exercised by certain

members of the IT organization, all production changes must be approved by me, as well as Chris Allers and Bill Palmer.

Changes made without approval will result in disciplinary action.

Thank you, Sarah Moulton

Maxine reads the email from Sarah. There's a new, maybe even sinister, dynamic creeping into the Phoenix Project. In each of the outage calls and crisis management meetings, senior leaders seem to be going out of their way to posture about how they did their job but other people didn't do *their* jobs, sometimes subtly, sometimes very blatantly.

While the redshirts battle to contain the raging engine fire that is threatening the entire ship, the bridge officers continue to cover their asses, Maxine observes. Some are even using the disaster to their political advantage, often to punish individual engineers or entire departments for supposed dereliction of duty.

Apparently, no one in IT leadership is safe—Maxine hears whispers that both Chris and Bill, as the heads of Dev and Ops, are in jeopardy of being fired, and there are rumors of all of IT being outsourced again. However, most believe William, as head of QA, is most likely to be axed.

Which is bullshit, thinks Maxine. *William was assigned to head up the release team less than twenty-four hours before the release! No one can get fired for trying to avert a disaster, right?*

"It's like the TV show *Survivor*," says Shannon. "All the technology executives are just trying to last one more episode. Everyone is freaking out. Steve has been demoted, and Sarah is trying to convince everyone that she can save the company."

Later that afternoon, Brent invites Maxine to join a meeting. "We've got nearly sixty thousand erroneous and/or duplicate orders in the database, and we've got to fix them so that the finance people can get accurate revenue reports."

Maxine helps the group wrangle the problem for an hour. At the end, once they find a solution, one of the Marketing managers says, "This is above my paygrade. Sarah is super-sensitive about changes right now. I've got to get her approval."

Ah, the Square in action, just like Cranky Dave described. But now, decisions that might have needed only to go “up and over one” now have to go “up and over two.” Now, all product managers need to run everything by Sarah. Someone mutters, “Don’t hold your breath—she never responds right away.”

Great, Maxine thinks. Sarah has effectively paralyzed everyone in this room even further.

Throughout the day, all decisions and escalations quickly grind to a standstill, even for emergencies, which Maxine didn’t expect. She discovers why: every manager insists on being a part of the communication plan. Why? They want to hear any bad news first, so they don’t appear out of touch and can massage any messages up the chain.

Maxine is sharing this observation with Kurt when his phone buzzes. Seeing his sour expression, she asks, “What’s up?”

“It’s Sarah,” he says. “She says she’s getting conflicting information from Wes and me about the corrupted order data. I need to spend thirty minutes explaining it to her when I’ve got two actual emergencies going on.”

Kurt storms off before she can even wish him good luck. Maxine shakes her head. The lack of trust and too much information flowing around is causing things to go slower and slower.

On Tuesday, Maxine joins a meeting led by Wes about more mysterious, intermittent outages for both the e-commerce site and the point-of-sale systems.

Sarah has been sending out emails, sometimes in all caps, reminding people how important this is. But everyone already knows how important this is—processing orders is one of the most important functions for any retailer.

The room is almost empty, even though this is a Sev 1 outage.

Apparently, everyone has had to go home sick. The Phoenix release forced people to work long hours together in close proximity all day and night, and with little sleep. Now everyone is dropping like flies. Of the people needed on this call, no one is healthy enough to be in the office. In fact, only two people are healthy enough to even be on the conference line.

Maxine looks up when she hears Sarah shouting, “What can you do about this? Who can fix this? Our store managers need our help! Don’t people realize how important this is?”

Maxine stares at Sarah in disbelief, noting that she looks tired, not her usual immaculate self. Even Sarah isn’t escaping the Phoenix carnage completely unscathed, despite her Teflon-like ability to avoid getting blamed for nearly anything in her three-year tenure at Parts Unlimited.

Wes throws up his hands. “What can we do about it? Nothing. The entire application support team is out sick. Brent just went home sick. The DBAs are out sick. Even though we’ve got the supremely competent Maxine here, she’s like me—we don’t know enough about the service to do anything except reboot the systems, which is what the support teams are already doing.”

Maxine sees that Wes is sick too—he’s congested and looks terrible. Bags underneath his red eyes, hoarse voice . . . she suddenly wonders if she looks as bad as he does.

“This is not acceptable, Wes,” says Sarah. “The business depends on us. The store managers depend on us. We need to do something!”

“Well, these were the risks we warned you about when you proposed proceeding with the Phoenix launch—but you emailed saying that we ‘need to break some eggs to make omelets,’ right? We’re doing everything we can, but unless you want to help reboot some servers, I’m telling you there’s nothing we can do.”

Wes continues, “But here’s something that we should talk about: How do we keep our people healthy enough so they can actually do their jobs? And how do we keep them happy enough so they don’t quit? Chris says two of his key engineers quit in the last week. I’ve lost two people on the Ops side too, and there’s a good chance I may lose three more. Who knows how many more are actively looking?

“And when that happens, we will truly be up shit creek, because then we’ll have empty meetings like this all the time,” Wes says with a half-hearted laugh that turns into coughs.

He grabs his laptop and starts walking out the door. Before he leaves, he says, “Sarah, I know you think it’s strange that we have no one left on the bench to solve this important problem, but that’s the way it is. If you want to help, learn to be a doctor or learn some middleware. In the meantime, just stay out the way because we’re doing our best.”

Maxine likes the way Wes rolls—he's fearless and he always says what he thinks.

She makes a mental note to ask the Rebellion about recruiting Wes.

Thinking about the Rebellion, she realizes how important that group is. To her, it's a beacon of hope. Maxine knows she may be manic and loopy from lack of sleep, but the Rebellion has assembled some of the best engineers in the company. And they could liberate everybody from...from...all of this.

We need to keep the Rebellion together and keep this important work going, she thinks.

She texts Kurt right away:

No matter what, we cannot cancel our Dockside meeting on Thursday.

His reply shows up right away.

Great minds think alike. In fact, I have a surprise for everyone. See you in two days!

By Thursday, things have stabilized substantially. The most glaring defects and performance problems in Phoenix have been fixed. And it helps that customer traffic is way, way down. Who wants to go to a store or website that can't take orders? The result is that it's no longer necessary for everyone to work all night. Maxine slept in until ten this morning. As she was driving into work, she realized how much she was looking forward to the Dockside meeting that evening.

As promised, Kurt texted everyone in the Rebellion:

I'll be a little late. Dwayne and Maxine, please run the standard agenda, including the Phoenix environment build. I will be bringing a very special guest.

Maxine is pretty sure everyone will be there tonight.

But despite getting some sleep, she doesn't feel well. She desperately hopes she is not getting whatever illness decimated her fellow co-workers. Despite that, she is very glad to be working on the Phoenix builds again.

That evening, when she arrives at the Dockside, Maxine's excited to see everyone. She wants to find out how to get a Rebellion sticker for her laptop and to trade war stories. She's surprised to see that everyone looks angry and dejected.

Throwing her jacket over the back of a chair, she says cheerily, "Hi, everyone! What's got everyone so grouchy?"

Dwayne looks at her. "Read the email that was just sent out. They fired William."

From: Chris Allers (VP Development)
To: All IT Employees
Date: 4:58 p.m., September 18
Subject: Personnel changes

Effective immediately, Peter Kilpatrick (Front-End Dev Manager) will be leaving the company, and William Mason (QA Director) will be on a leave of absence. We especially appreciate all their contributions.

Please direct all front-end Dev emails to Randy and all QA-related emails to me.

Thank you, Chris

Maxine slumps as she reads the message. The witch hunt has begun. Adam shakes his head angrily. "I wasn't a huge fan of William," he says, "but to blame him for everything is wrong."

In Chris' email there's no mention about his own culpability in the Phoenix disaster. And even though Maxine doesn't believe in punishment or scapegoating, it's doubly unfair that all the blame is being put on the technology organization, and no one from the business or product side is being held accountable.

Cranky Dave looks up from his phone, disgusted. "Ditto for Peter—he was just doing what the business managers demanded. What a complete shit show."

"This is so wrong," Shannon mutters. "I don't suppose it would help to write a petition or anything, right? You know, lodge our protest about their firing?"

Adam says, "No one who matters is being held responsible! We should..."

He suddenly stops talking, staring slack-jawed at something behind Maxine. "Holy shit..." he finally says. Everyone next to Adam is also looking shocked at whatever is behind her.

Maxine turns around and sees Kurt walking through the entrance.

Next to him is Kirsten, the director of project management.

"My God," Maxine hears Adam say. He looks frightened, closing his laptop and standing up, as if he's going to flee the scene.

"Oh, for Chrissakes, sit back down, Adam," says Maxine. "This isn't like the secret police showing up. Not one of us has done anything wrong—have some dignity."

Cranky Dave laughs nervously, but like everyone else, he's already closed his laptop, as if he has something to hide.

Kirsten is wearing a fancy blazer, two steps up from Maxine's usual casual business garb and a full four steps up from the hoodies, T-shirts, and bowling shirts worn by the other engineers around the table. People in the bar are staring, clearly wondering who invited the management suit here.

Maxine knows that she looks slightly out of place at the Dockside, but wow, Kirsten looks *way* out of place, like she was on the way to an event for senior law partners but had a flat tire while driving by with a dead cell phone and had to come in to find help.

Looking around, Kurt smiles and says, "For those of you who don't know Kirsten, she leads Project Management, which is undoubtedly the most trusted organization at Parts Unlimited, despite their association with us technology people." Kurt laughs. "All of the most important company initiatives go through Kirsten and her project management clerics, and she routinely briefs Dick Landry, our CFO, on how they're going."

This is true, Maxine thinks. Kirsten is truly the high priestess of order and discipline. She assigns the score of red, yellow, or green to each major initiative of the organization, which can have career-catapulting or career-ending consequences for the people involved. Besides Sarah and the VP of sales, Kirsten is the person most mentioned by the CFO in his Town Halls.

Sitting, Kirsten pours herself a beer from the pitcher on the table and then pours a glass for Kurt. Kurt introduces everyone to Kirsten and then gestures at Maxine, “Maxine is the latest addition to our elite group of rebels. She was exiled to the Phoenix Project as punishment for the payroll outage, and of course, her vast talents have been completely wasted ever since. That is, until we recruited her to help overthrow the uncaring, ancient, powerful existing order...oh, um...” Kurt suddenly looks embarrassed, realizing Kirsten is part of that order. “Present company excepted, of course,” he finishes.

Kirsten merely raises her glass in response.

Kurt continues, “It turns out that Maxine, in her boredom and search for meaning, began working on creating repeatable Phoenix builds, something that has eluded the Phoenix teams for well over a year. We believe in many great and virtuous things, but one thing we all agree on is that getting builds going again is one of the most urgent and important engineering practices we need right now. Once we get continuous builds going, we enable automated testing. We get automated testing, we can make changes quicker, with more confidence, and without having to depend on hundreds of hours of manual testing. And that, I believe, is the critical first step for how we can deliver better value, safer, faster, and happier.

“Without continuous builds, we are like a car manufacturer without an assembly line, where anyone can do whatever they want, independent of the plant goals,” he continues. “We need to discover problems only when we are in the build or testing process, not during deployment or production.

“I’ve wanted to own this for a year, but my boss, uh, rather, my recently departed ex-boss, didn’t think it mattered. So, I’ve been taking people off my team to work on it in secret and seeking out the best engineers in the company who are willing and able to help. And Maxine has been a tremendous help in an amazingly short amount of time,” he adds.

Kurt pauses. “Uh, let’s all raise a glass to William—he and I had our differences, but he certainly didn’t deserve to take the blame for the entire Phoenix fiasco.”

Maxine raises her glass, as everyone else does the same. She takes the time to clink glasses with everyone around the table.

Looking at Kirsten, she says, “It sounds crazy, Kirsten, but I really think this group can make a big difference. I’ve seen developers wait for

months to get a Dev environment. The lack of environments and centralized builds slow us down in countless ways. In fact, most Dev teams eventually stop waiting for environments or builds and just write code in isolation, without caring whether it actually works with the system as a whole.”

Maxine continues, “Look at what happened last week with the Phoenix release. Better engineering practices would have prevented so much of that. What a waste...”

“We all agree with Maxine,” Cranky Dave says. “But, Kirsten, uh, what in the world are you doing here?”

Kirsten laughs. “I’ve long harbored a suspicion that how we manage technology at this company is not working. And it’s not just the Phoenix release catastrophe. Look at all the things we need from Phoenix that are still years away on the project plan.

“Kurt has been telling me for months about the work the Rebellion is doing. But my aha moment was when Kurt pointed out that we’ve somehow created a system where hundreds of engineers are unable to get simple things done without an incredible amount of communication and coordination,” she explains. “Sure, it’s our job to safeguard the most important projects in the company. But ideally, everyone should be able to get what they need done without any help from us. Somehow, I think Project Management has turned into an army of paper pushers, being dragged into every single task because of all the dependencies.

“We track the work of nearly three hundred people working on the various parts of Phoenix. But, the real effort is even larger,” she continues. “You’d think we have thirty teams of ten people, with each team able to get things done independently. But at times, it’s like we have only one team of three hundred people... Or maybe three hundred teams of one. In either case, something is very wrong...”

She turns to Kurt. “What was that term you used? Watermelon projects? Green on the outside, but red on the inside? That’s what every one of our IT projects is these days,” Kirsten observes wryly.

She continues, “I’ve been here for fifteen years, and we’ve been playing this game of outsourcing and insourcing IT the whole time. The last time around, the CIO proclaimed that Parts Unlimited was ‘no longer in the people business,’ if you can believe that, and outsourced everything.

We eventually brought most of it back in-house, but everything we got back was in worse shape than ever. And we've lost the capability to do some of even the most basic things ourselves. Last year, we had to make a simple schema change for our data warehouse. We put out the request to our normal list of outsourcing partners. It took them about three weeks to get an estimate back to us. They said the work would take about ten thousand hours to complete," she says. "Before we outsourced IT, this was something we could have done in a couple of hours."

Maxine does the math in her head. From her consulting days, she knows one fully loaded engineer works about two thousand hours per year—that's forty hours per week, fifty-two weeks per year, if they don't take any vacation. She bursts out laughing. "That's five engineers working full-time for a year, just to make a database column change?! That's something I could do in fifteen minutes!"

"Yep," Kurt says, with a sad smile. "The data warehouse change requires work from two or three different outsourcers. You'd need to pull together meetings from the account managers from each of those teams. Each account manager would require a change fee and a feasibility study. It takes weeks to get all the technical people to agree upon a change plan, and even then, the tickets bounce back and forth for weeks. It takes a super-heroic effort to actually get the change made."

Dwayne laughs loudly. "You think that's bad? That's nothing! We used to have three networking switches in all of our manufacturing plants. One for internal plant operations, one for employees and guest WiFi, and one for all our equipment vendors that need to phone home to their mothership.

"A couple of years ago, probably during budgeting season, some bean counter looked at those three networking vendors and decided to consolidate them down to one switch. Sort of makes sense, right?" he continues.

"So without asking anyone, they went ahead and did it. And not just in one plant, but in a bunch of the plants. They replaced the three switches with one bigger, beefier switch, and then moved all plant traffic onto it," Dwayne says. "But what they didn't know was that they had three separate outsourcers managing the three different networks. So now all three outsourcers who used to work on their own separate switches had to work on one switch and were suddenly stepping on each other's toes all the time.

“Within a week, one of the manufacturing plants had their entire network knocked offline—absolutely nothing from inside the plant could talk with the outside world. No one could get plant scheduling information, no one could send out replenishment orders, equipment couldn’t get maintenance updates... All interfaces were dead!” Dwayne continues, still clearly in awe of the scale of the outage.

“The only thing that worked was the fax machine. Everyone from every department had to wait in line to send out things like weekly production reports to management, orders for raw materials...” Dwayne says.

Maxine bursts out laughing. “I remember that—it was incredible. We had to buy some USB printers from the local office supply store for a couple of systems that couldn’t connect to the network printers. It was like going back to the 1970s for almost a week.”

Adam mutters from across the table, “Yeah, just like we did to the in-store systems this weekend.”

Dwayne takes another drink of beer and leans back, enjoying having everyone’s attention. “You’re probably wondering why it took a week to restore service. Well, that entire time, no one took responsibility for what happened. All three outsourcers denied that it was them, even when we presented them the log files that clearly showed that one of the them had disabled everyone else’s accounts. Apparently someone got tired of having their changes trampled on by the other two, so they just locked them out.”

Everyone roars in laughter, but Maxine’s jaw drops.

Dwayne continues, “That entire week all three outsourcers kept blaming each other, and the network stayed down for days. It escalated all the way to Steve. Yep. The CEO. Even after he got all the CEOs of all three outsourcers on the phone together, it still took almost twenty-four hours for the network to be restored.”

As everyone jeers, Maxine says slowly, “That’s so interesting. Consolidating network switches isn’t inherently a bad idea. Before, three teams were able to work independently on their own networks. And when they were all put on one network switch, suddenly they were coupled together, unable to work independently, having to communicate and coordinate in order to not interfere with each other, right?”

With awe in her voice, she continues, “You know, after they got put onto one switch, I bet those teams needed to create a master schedule with

all of their work on it. And I'm even betting that they needed to bring in project managers where they probably didn't need them before.

"Holy cow," Maxine continues, on a roll. "They did it to reduce costs, but surely, in the end, it was more expensive for everyone all around. And I bet it took everyone longer to do their work, with everyone having to communicate, coordinate, get approvals, with project managers shuffling and deconflicting all the work.

"Oh, my God. It's just like the Phoenix Project!" she exclaims.

Silence falls upon the table as everyone stares at Maxine in a mix of horror and dawning realization.

"You mean everything that's wrong with the Phoenix Project we did to ourselves?" Shannon asks.

Kirsten looks rattled, brow furrowed, but says nothing. "Yes," says Maxine. "I think we did it to ourselves."

"You are correct, Maxine. You are truly on the cusp of understanding the magnitude and scale of the challenges that await you," a voice says from behind Maxine.

CHAPTER 7

• *Thursday, September 18*

The owner of the familiar voice is, to Maxine's surprise, the bartender the last time she was at the Dockside.

He sets a tray of drinks down next to Maxine and gives Kurt a friendly pat on the back. Then he turns to Kirsten, saying, "Oh, ho—if it isn't Ms. Fingle! Long time no see! Welcome to the Dockside, headquarters of the budding Rebellion."

"Holy cow," Kirsten says, staring.

"Uh, you know each other?" Kurt asks, his usual confident tone missing.

Kirsten laughs. "This is Dr. Erik Reid. You may not know this, but Steve and Dick have been trying to recruit him to serve on the board of Parts Unlimited for months. He's worked with the company for decades. In fact, Erik was part of the initial MRP rollout in the '80s, and then he helped the manufacturing plants adopt Lean principles and practices. We were one of the first companies to have an automated MRP system, and he's a genuine hero among the manufacturing ranks."

"Him?" Kurt says in disbelief, pointing his thumb toward the bartender.

Maxine is surprised too. After all, she took over continuing development and operations of the amazing homegrown MRP system years ago. She's always been impressed at how it codified not only a wonderful way of working that led to fantastic flow but also enabled continual learning, for both line workers and plant managers.

"Don't believe everything you hear," Erik says, snorting.

Maxine quickly sizes him up. He appears to be in his mid- to late-fifties, about the right age to be the progenitor of the MRP system. He has the build of someone large who used to be in great shape. He has shoulder-length, graying hair, reminding her of The Dude from *The Big Lebowski*. But instead of being mellow and cool, Erik is clearly sharp and attentive.

He turns to Maxine with a sly smile. “On behalf of everyone in manufacturing operations, thanks for taking such good care of the MRP system. You’ve helped create and sustain software that is a masterpiece of simplicity and locality. You’re not only magnificently meeting the business objectives, you’ve also created a system where small teams of engineers are able to work productively and independently of each other, with components painstakingly and splendidly isolated from each other, instead of being complected into a giant, ugly, knotty mess.

“A truly magnificent feat of engineering and architecture!” he says, beaming. “The developer productivity you’ve enabled is a beautiful testament to elegant simplicity. And even more impressive is your ruthless eradication of technical debt as a part of your daily work. I’m pleased to finally meet you!”

Maxine stares at Erik. *It’s not every day that a bartender compliments you on the code you’ve painstakingly written and shepherded for years*, she thinks.

“Thank you—I’ll make sure to pass that on to the team,” she says, perplexed, but unable to hide her pride.

“Uh, what does ‘complected’ mean?” Kurt asks.

Erik answers, “It’s an archaic word, resurrected by Sensei Rich Hickey. ‘Complect’ means to turn something simple into something complex.

“In tightly coupled and complected systems, it’s nearly impossible to change anything, because you can’t just change one area of the code, you must change one hundred, or even a thousand, areas of the code. And even the smallest changes can cause wildly unpredictable effects in distant parts of the system, maybe in something you’ve never even heard of.

“Sensei Hickey would say, ‘Think of four strands of yarn that hang independently—that’s a simple system. Now take those same four strands of yarn and braid them together. Now you’ve complected them.’ Both configurations of yarn could fulfill the same engineering goal, but one is dramatically easier to change than the other. In the simple system, you can change one string independently without having to touch the others. Which is very good.”

Erik laughs, “However, in the complected system, when you want to make a change to one strand of yarn, you are forced to change the other three strands too. In fact, for many things you may want to do, you simply cannot, because everything is so knotted together.

“And when that happens,” he continues, “you’ve trapped yourself in a system of work where you can no longer solve real business problems easily anymore—instead, you’re forced to merely solve puzzles all day, trying to figure out how to make your small change, obstructed by your complected system every step of the way. You must schedule meetings with other teams, try to convince them to change something for you, escalate it to their managers, maybe all the way up the chain.

“Everything you do becomes increasingly distant from the real business problem you’re trying to solve,” he says. “And that, Dwayne, is what everyone discovered when they switched out the routers in those manufacturing plants. Before, you had three independent strands, with teams able to work independently but at the cost of having to maintain three networking switches.

“When you put them all on one switch, you complected their value streams, all now having dependencies on each other that didn’t exist before. They must constantly communicate, coordinate, schedule, marshal, sequence, and deconflict their work. They now have an extremely high cost of coordination, which has lengthened lead times, decreased quality, and, in your story, led to a week-long catastrophe that significantly impaired the business, going all the way up to Steve,” Erik says with glee.

“The importance of lead times in software delivery is tantamount, as Senseis Dr. Nicole Forsgren and Jez Humble have discovered in their research,” Erik says. “Code deployment lead time, code deployment frequency, and time to resolve problems are predictive of software delivery, operational performance, and organizational performance, and they correlate with burnout, employee engagement, and so much more.

“Simplicity is important because it enables locality. Locality in our code is what keeps systems loosely coupled, enabling us to deliver features faster. Teams can quickly and independently develop, test, and deploy value to customers. Locality in our organizations allows teams to make decisions without having to communicate and coordinate with people outside the team, potentially having to get approvals from distant authorities or committees so far removed from the work that they have no relevant basis to make good decisions,” he says, clearly disgusted.

“You should be able to create value by changing one file, one module, one service, one component, one API call, one container, one app, or whatever! Which is why putting cross-cutting concerns in one place

is so great, like logging, security, or retry policies. You change it there, and you've changed it everywhere," he says. "Isn't it absurd that when you build a feature, changes sometimes have to be made by the UI team, the front-end team, the back-end team, and the database team?"

"Interesting," Maxine says. "Locality in our code and organization is so desirable, as opposed to what we have now, which is code scattered everywhere!"

"Yes, exactly. Scattered!" Erik says. "And achieving this greatness is never free. It requires focus and elevation of *improvement* of daily work, even over daily work itself. Without this ruthless focus, every simple system degrades over time, increasingly buried under a tundra of technical debt. Just look at the disaster that is the Phoenix build system."

Maxine furrows her brow. "You're saying that Phoenix used to be simple, but now it has become complexed beyond recognition. That Phoenix used to have a great build process, but over the years it has become neglected, taking a backseat to features, and eventually bumped out of the car entirely."

"Precisely," says Erik. "Build responsibility moved from Dev to QA to interns. Tech giants like Facebook, Amazon, Netflix, Google, and Microsoft give Dev productivity responsibilities to only the most senior and experienced engineers. But here at Parts Unlimited, it's the exact opposite."

Dwayne laughs, "At least our builds aren't outsourced anymore. Not too long ago, it cost \$85 each time a build was performed." Everyone, including Maxine, guffaws in disbelief.

Kirsten says, "I hear engineers complain all the time about technical debt? But what exactly is it, besides being something bad?"

Erik laughs. "There are many definitions, but my favorite is how it was originally defined by Ward Cunningham in 2003. He said, 'technical debt is what you feel the next time you want to make a change.' There are many things that people call technical debt, but it usually refers to things we need to clean up, or where we need to create or restore simplicity, so that that we can quickly, confidently, and safely make changes to the system.

"Sometimes it's a build and test system that doesn't give fast feedback to developers, or when it stops working entirely," he continues. "Sometimes it's when simple components become complexed, and you

can no longer reason about it or change it without immense effort or risk of catastrophe. Sometimes it's when decision-making processes or the organizational structure loses locality, forcing even small decisions to be escalated—your infamous 'Square.'

"I've started calling all of these things 'complexity debt,' because they're not just technical issues—they're business issues. And it's always a choice," he says. "You can choose to build new features or you can choose to pay down complexity debt. When a fool spends all their time on features, the inevitable outcome is that even easy tasks become difficult and take longer to execute. And no matter how hard you try or how many people you have, it eventually collapses under its own weight, forcing you to start over from scratch."

He looks at Maxine and says, "Which is why what you've done with the MRP system is so remarkable. Your teams are able to add features at a rate that the entire Phoenix team should envy. And that is only possible because you pay down technical debt as a part of daily work. It's a magnificent example of the First Ideal of Locality and Simplicity in our code and organizations. Well done, Maxine."

Erik stands up. "I'm a little short-staffed tonight. I'll catch you later, and great to see you, Kirsten!"

"Oh, one more thing," he says, turning around. "Think about the engagement scores of the technology employees versus the rest of the business and ponder the differences, especially on the Phoenix Project."

As Maxine watches Erik head back to the bar, she hears everyone burst into conversation.

Maxine says, "I have no idea what just happened." Looking at both Kirsten and Kurt, she asks, "What was that all about? And what did he mean by the First Ideal?"

"I have no idea," Kurt says, shaking his head. "I've known Erik for over a year. I had absolutely no idea he had some connection to the company..."

Dwayne says to Kurt, "I never bothered to tell you because, you know, it didn't seem that important. But one evening he asked me whether I knew anything about configuring Kubernetes clusters. That was pretty strange."

"That's odd," Shannon says. "Now that I think about it, I once had a debate with him about how completely you should or shouldn't isolate the cardholder data environment in order to comply with the PCI Data

Security Standard. He even sent me links to the specific subsections in the standard. He seemed very knowledgeable. An expert, even. I thought it was just because this bar took credit card payments...

"I've heard he's been having many conversations with Bill Palmer, the new VP of IT Operations," Kirsten adds. "Bill told me about how Erik is teaching him something called the Three Ways and the Four Types of Work."

"I've never heard of those," Maxine says. "He only mentioned the First Ideal... I wonder how many other Ideals there are?"

"And what did he mean by engagement scores?" asks Kurt.

"I don't know," Kirsten says. "But I do know that we have some of the highest employee satisfaction scores in our industry... except for the IT department... which I think is negative twenty-seven."

"Is that bad?" Dwayne asks.

Kirsten looks embarrassed. "Very bad."

Maxine is not surprised. And yet, something bothers her. In the Town Hall, Steve talked about how much he cares about employee engagement. What does he think when he sees that the department responsible for the most strategic program in the company is miserable? Shouldn't that worry him?

When Erik walks by with a full beer glass, Maxine gets up and rushes to catch up with him. "Thanks again for the kind words, Erik. You mentioned the First Ideal—How many of them are there and what are they?"

"Ha! That's not the way it works," Erik says, laughing. "In fact, I've got Bill Palmer running hither and yon, trying to find all the Four Types of Work, watch. But... perhaps I can give you all a head start."

Erik and Maxine walk back to the table. "There are Five Ideals," Erik begins. The whole table turns their attention to him. "I've already told you about the First Ideal of Locality and Simplicity. We need to design things so that we have locality in our systems and the organizations that build them. And we need simplicity in everything we do. The last place we want complexity is internally, whether it's in our code, in our organization, or in our processes. The external world is complex enough, so it would be intolerable if we allow it in things we can actually control! We must make it easy to do our work."

Maxine sits back down, opens her laptop (pleased she remembered it this time), and starts taking notes.

“The Second Ideal is Focus, Flow, and Joy. It’s all about how our daily work feels. Is our work marked by boredom and waiting for other people to get things done on our behalf? Do we blindly work on small pieces of the whole, only seeing the outcomes of our work during a deployment when everything blows up, leading to firefighting, punishment, and burn-out? Or do we work in small batches, ideally single-piece flow, getting fast and continual feedback on our work? These are the conditions that allow for focus and flow, challenge, learning, discovery, mastering our domain, and even joy.”

He looks around the table with a smug expression on his face. “And that’s all you get for now. I’ll share with you the other three Ideals when you’re ready.”

“You’re kidding me,” Maxine says. “You’re pulling some sort of Yoda or Mr. Miyagi routine on us? Come on, at least tell us the *names* of the other Ideals!”

“Lucky for you, Young Grasshopper, I don’t have time to argue, as there’s a line at the bar I need to take care of,” he says. “In its briefest form: The Third Ideal is Improvement of Daily Work. Reflect upon what the Toyota Andon cord teaches us about how we must elevate improvement of daily work over daily work itself. The Fourth Ideal is Psychological Safety, where we make it safe to talk about problems, because solving problems requires prevention, which requires honesty, and honesty requires the absence of fear. In manufacturing, psychological safety is just as important as physical safety. And finally, the Fifth Ideal is Customer Focus, where we ruthlessly question whether something actually matters to our customers, as in, are they willing to pay us for it or is it only of value to our functional silo?”

Erik finishes his beer and says with a smile, “Good luck to you all. See you next week.”

“Wait, wait, that’s it?” Maxine says, but Erik is already gone. Maxine looks down at her quickly typed notes:

The First Ideal—Locality and Simplicity

The Second Ideal—Focus, Flow, and Joy

The Third Ideal—Improvement of Daily Work

Maxine stares at the list—all of the Ideals sound nice, but how in the world are they supposed to use them to change the trajectory of the Phoenix Project?

“That was so strange,” Kurt says, saying what everyone is thinking.

Cranky Dave adds, “That bit about the Fourth Ideal hit home. A culture of fear where everyone is afraid to share bad news? That’s us.

“Erik is right,” Adam says. “No one talks about the real problem. Most people aren’t brave enough to say what they think or to do the right thing. They just say ‘yes,’ whether they agree or not. But maybe this creates an opportunity. There are some big, gaping holes in the org chart now,” he says to Kurt. “You should throw your name into the hat for one of them. Maybe even for William’s position?”

Silence descends upon the table as everyone turns to look at Adam and Kurt.

“That’s a pretty good idea, Kurt. You could make a huge difference in the QA organization. You know all of us would be pretty happy about that,” says Shannon, with everyone around the table murmuring assent.

“Maybe,” says Kurt, nodding slowly. “But you know, if we really want to make a difference, there’s another move. I’m thinking about telling Chris that I want Peter’s position.”

Maxine hears some gasps around the table, followed by Cranky Dave’s loud laugh. “You’re right, Kurt. You would definitely make a much bigger difference by taking over a Dev team. We all know we need to change how QA does testing, but the best place to start is by changing how *Dev* does testing. And that requires being a Dev manager...but that brings up a teeny, tiny, little problem...they’ll never give you that position, Kurt,” he says. “You know, because you’re ‘just a QA manager.’”

Maxine winces. Cranky Dave is voicing an all-too-popular prejudice that developers have about QA people, which embarrasses her. QA is often viewed as an underclass, but at least they’re above Ops. *All of which is crap*, Maxine thinks. After all, she started her Ops career in high school, rotating backup tapes, and later, before graduate school, QA—if

it weren't for that background, she wouldn't have become the person she is today. Technology is still too often a caste system.

Adam says to Kurt, "You know I'm a big fan and I love working with you—you're a fantastic leader—but I agree with Dave. There's no way that a bunch of Dev managers are going to let a QA manager take that spot. Maybe you should just settle for William's old role. After all, someone has to lead QA out of the Stone Age and bring automated testing to the rest of the Phoenix Project."

"I have to agree with your friends, Kurt," Kirsten says. "You and I both know that William was never a big fan of yours. He never spoke very highly of you in meetings. They're probably going to bring in someone from the outside."

Kurt grins, seemingly not bothered by Kirsten's observation. In his great William impersonation, Kurt says, "Yes, Kirsten, you are right. Although Kurt shows some potential, it's clear to me that he doesn't understand the testing game. Maybe in a couple of years he'll have the maturity to run the QA organization."

Everyone laughs. Kurt continues in his normal voice, "Folks, here's an opportunity for us to make a difference. But I don't think we can do it from anywhere in the QA organization—QA as we know it is changing. We can't keep being the people who test after the fact. We need to get into the game, and that means finagling our way into the development teams that are actually responsible for shipping features and the quality of their outcomes. Anything else is a waste of our time."

He continues, "In fact, if we can take over Peter's team, my goal will be to show that we can out-perform every other Dev team in the Phoenix Project. Gathered around this table is some of the best technical talent in the company, and we've already created the infrastructure that can bring some great technical practices to the game."

Kurt leans forward. "If I can get Chris to give me that chance, would you all be willing to join the team and show that we can change the trajectory of the Phoenix Project?"

"Hell yes, Kurt. Count me in!" says Cranky Dave. Maxine is surprised that he's the first to volunteer.

Maxine follows. "And me. This is what I want to work on. And I know we can run circles around all the other teams. I've seen the competition up close," she says with a smile.

Everyone around the table chimes in, excited at the potential opportunity. Cranky Dave says, “Okay, we’re all in, Kurt. But frankly, I’m not holding my breath. Adam is right—you getting a Dev team is a long shot.”

Kirsten says, “Kurt, I agree with your instincts. If you want, I’ll write a letter of recommendation to Chris.”

“That would be fantastic, Kirsten,” says Kurt, beaming and obviously genuinely surprised and grateful for Kirsten’s offer. At that moment, Maxine realizes that Kurt has been operating this entire time without any real leadership air cover. *He could get fired for going rogue*, she realizes.

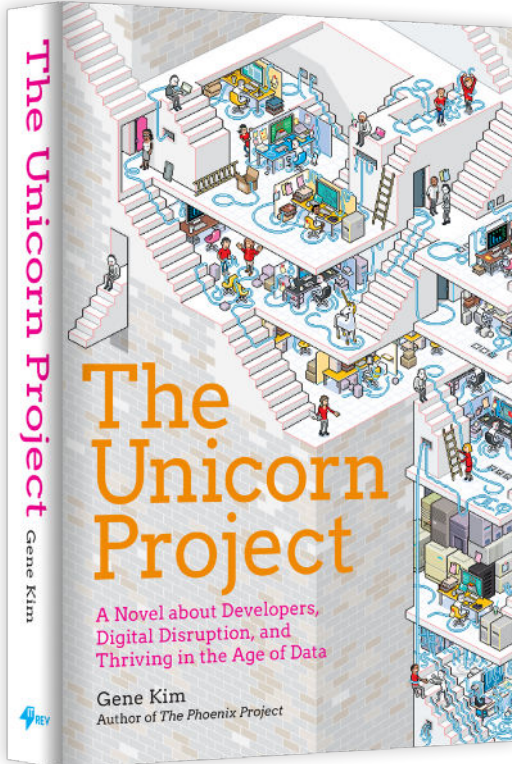
“Happy to help,” says Kirsten. “But let me be clear. I’m willing to write a letter to support Kurt’s ideas, but I really can’t be seen publicly with you all. At least, not yet. People need to see me as impartial.”

“Oh, you’re willing to give us a chance to take a risk and get fired, but you want to stay safely on the sidelines?” says Cranky Dave, halfway joking. Kirsten merely raises her glass to Dave.

Thank you for reading this excerpt of

The Unicorn Project!

On Sale November 26, 2019!



In the meantime, feel free to:

1. Share what you thought of the book (please use #UnicornProject and mention @realgenekim), or share a picture of the cover! (You can find images and so forth [here](#).)
2. Of course, you can find more information about the book and **pre-order it here** or at your favorite local retailer.

The Unicorn Project officially releases on November 26, 2019.

Thank you!

Gene Kim

