

SCRUM

**The Art of
Doing
Twice
the Work
in Half
the
Time**



JEFF SUTHERLAND
Co-creator of Scrum

CONTENTS

About the Book

About the Author

Title Page

Preface

*Chapter One: **The Way the World Works Is Broken***

*Chapter Two: **The Origins of Scrum***

*Chapter Three: **Teams***

*Chapter Four: **Time***

*Chapter Five: **Waste Is a Crime***

*Chapter Six: **Plan Reality, Not Fantasy.***

*Chapter Seven: **Happiness***

*Chapter Eight: **Priorities***

*Chapter Nine: **Change the World***

Acknowledgments

Appendix: Implementing Scrum—How to Begin

Notes

Index

Copyright

About the Book

Scrum is the revolutionary methodology set to completely overhaul the way businesses operate.

Already adopted by world-leading organisations such as **Google, Amazon** and the **FBI**, Scrum transforms the standard approach to project management and team building. In the process, costs and working hours are drastically cut, leading to reported improvements in productivity of a staggering 400-800%.

The reason for Scrum's success is simple: it is based on how people really work, rather than how they think they work. Scrum employs an incremental approach, breaking an overall project down into individual short-term goals to allow for constant assessment of progress. By reducing team sizes and offering an adaptive approach to problem-solving, both speed and quality of work are rapidly improved.

For the first time ever, the man behind Scrum, Jeff Sutherland, shows how it can be applied universally.

The way we work is wrong; this is the solution.

About the Author

Dr Jeff Sutherland is the creator of Scrum and was a signer of the Agile Manifesto, which marked the start of the Agile movement. A graduate of West Point, he began his career as a fighter pilot in the US Air Force, and went on to join the faculty at the University of Colorado Medical School. Dr. Sutherland has served as VP of Engineering or CTO at 11 software companies, managing the last seven entirely using Scrum, and achieving industry-leading, hyper-productive results. He is currently the CEO of Scrum, Inc., the Chairman of the Scrum Foundation, and Senior Advisor to OpenView Venture Partners, where he is Agile coach for venture-funded companies.

SCRUM

*The Art of Doing Twice the Work
in Half the Time*

Jeff Sutherland



Preface

Why Scrum?

I first created Scrum, with Ken Schwaber, twenty years ago, as a faster, more reliable, more effective way to create software in the tech industry. Up to that point—and even as late as 2005—most software development projects were created using the Waterfall method, where a project was completed in distinct stages and moved step by step toward ultimate release to consumers or software users. The process was slow, unpredictable, and often never resulted in a product that people wanted or would pay to buy. Delays of months or even years were endemic to the process. The early step-by-step plans, laid out in comforting detail in Gantt charts, reassured management that we were in control of the development process—but almost without fail, we would fall quickly behind schedule and disastrously over budget.

To overcome those faults, in 1993 I invented a new way of doing things: Scrum. It is a radical change from the prescriptive, top-down project management methodologies of the past. Scrum, instead, is akin to evolutionary, adaptive, and self-correcting systems. Since its inception, the Scrum framework has become *the* way the tech industry creates new software and products. But while Scrum has become famously successful in managing software and hardware projects in Silicon Valley, it remains

relatively unknown in general business practice. And that is why I wrote *Scrum*: to reveal and explain the Scrum management system to businesses outside the world of technology. In the book I talk about the origins of Scrum in the Toyota Production System and the OODA loop of combat aviation. I discuss how we organize projects around small teams—and why that is such an effective way to work. I explain how we prioritize projects, how we set up one-week to one-month “sprints” to gain momentum and hold everyone on the team accountable, how we conduct brief daily stand-ups to keep tabs on what has been done and on the challenges that have inevitably cropped up. And how Scrum incorporates the concepts of continuous improvement and minimum viable products to get immediate feedback from consumers, rather than waiting until a project is finished. As you’ll see in the pages that follow, we’ve used Scrum to build everything from affordable 100-mile-per-gallon cars to bringing the FBI database systems into the twenty-first century.

Read on. I think you’ll see how Scrum can help transform how your company works, creates, plans, and thinks. I firmly believe that Scrum can help to revolutionize how business works in virtually every industry, just as it has revolutionized innovation and speed to market at a dazzling array of new companies and a breathtaking range of new products emerging out of Silicon Valley and the world of technology.

—Jeff Sutherland, PhD

CHAPTER ONE

The Way the World Works Is Broken

Jeff Johnson was pretty sure it wasn't going to be a good day. On March 3, 2010, the Federal Bureau of Investigation killed its biggest and most ambitious modernization project—the one that was supposed to prevent another 9/11 but that had devolved into one of the biggest software debacles of all time. For more than a decade the FBI had been trying to update its computer system, and it looked as if they would fail. *Again*. And now it was his baby.

He'd shown up at the FBI seven months earlier, lured there by the new Chief Information Officer, Chad Fulgham, whom he'd worked with at Lehman Brothers. Jeff was Assistant Director of the IT Engineering Division. He had an office on the top floor of the J. Edgar Hoover Building in downtown Washington, D.C. It was a big office. It even had a view of the Washington Monument. Little did Jeff know he'd end up in a windowless cinder-block office in the basement for much of the next two years, trying to fix something that everyone believed to be unfixable.

"It was not an easy decision," Jeff says. He and his boss had decided to declare defeat and kill a program that had already taken nearly a decade and cost hundreds of millions of dollars. By that point, it made more sense to bring the project in-house and do it themselves. "But it needed to be done and done well."

The project was the long-awaited computer system that would bring the FBI into the modern age. In 2010—the era of Facebook, Twitter, Amazon, and Google—the FBI was still filing most of its reports on paper. The system the Bureau used was called the Automated Case Support system. It ran on gigantic mainframe computers that had been state of the art sometime in the eighties. Many special agents didn't even use it. It was just too cumbersome and too slow in an era of terror attacks and swift-moving criminals.

When an FBI agent wanted to do something—*anything*, really—from paying an informant to pursuing a terrorist to filing a report on a bank robber, the process wasn't that different from what it had been thirty years earlier. Johnson describes it this way: "You would write up a document in a word processor and print out three copies. One would be sent up the approval chain. One would be stored locally in case that one got lost. And with the third you'd take a red pen—I'm not kidding, a red pen—and circle the key words for input into the database. You'd index your own report."

When a request was approved, that paper copy would drift down from upstairs with a number on it. A number written on a piece of paper is how the FBI kept track of all its case files. This method was so antiquated and porous that it was blamed in part for the Bureau's failure to "connect the dots" that showed various Al Qaeda activists entering the country in the weeks and months before 9/11. One office was suspicious of one person. Another wondered why so many suspicious foreigners were getting flight training. Another had someone on a watch list but never told anyone else. No one in the Bureau ever put it all together.

The 9/11 Commission drilled down after the attack and tried to discover the core reason it was allowed to happen. Analysts, said the Commission, couldn't get access to the very information they were supposed to analyze. "The poor

state of the FBI's information systems," reads the report, "meant that such access depended in large part on an analyst's personal relationships with individuals in the operational units or squads where the information resided."

Before 9/11, the FBI had never completed an assessment of the overall terrorism threat to the United States. There were a lot of reasons for this, from focus on career advancement to a lack of information sharing. But the report singled out lack of technological sophistication as perhaps the key reason the Bureau failed so dramatically in the days leading up to 9/11. "The FBI's information systems were woefully inadequate," the Commission's report concludes. "The FBI lacked the ability to know what it knew: there was no effective mechanism for capturing or sharing its institutional knowledge."

When senators started asking the Bureau some uncomfortable questions, the FBI basically said, "Don't worry, we have a modernization plan already in the works." The plan was called the Virtual Case File (VCF) system, and it was supposed to change everything. Not letting any crisis go to waste, officials said they only needed another \$70 million on top of the \$100 million already budgeted for the plan. If you go back and read press reports on VCF at the time, you'll notice that the words *revolutionary* and *transformation* are used liberally.

Three years later, the program was killed. It didn't work. Not even a little bit. The FBI had spent \$170 million in taxpayer money to buy a computer system that would never be used—not a single line of code, or application, or mouse click. The whole thing was an unmitigated disaster. And this wasn't simply IBM or Microsoft making a mistake. People's *lives* were, quite literally, on the line. As Senator Patrick Leahy of Vermont, then the ranking Democrat on the Senate Judiciary Committee, told the *Washington Post* at the time:

We had information that could have stopped 9/11. It was sitting there and was not acted upon.... I haven't seen them correct the problems.... We might be in the 22nd century before we get the 21st-century technology.¹

It is rather telling that many of the people who were at the FBI when the Virtual Case File disaster happened aren't there anymore.

In 2005 the FBI announced a new program, Sentinel. This time it would work. This time they'd put in the right safeguards, the right budget procedures, the right controls. They'd learned their lesson. The price tag? A mere \$451 million. And it would be fully operational by 2009.

What could possibly go wrong? In March of 2010 the answer landed on Jeff Johnson's desk. Lockheed Martin, the contractor hired to make the Sentinel system, had already spent \$405 million. They'd developed only half of the project, and it was already a year late. An independent analysis estimated it would take another six to eight years to finish the project, and the taxpayers would have to throw in at least another \$350 million.

Finding some way around that was Johnson's problem.

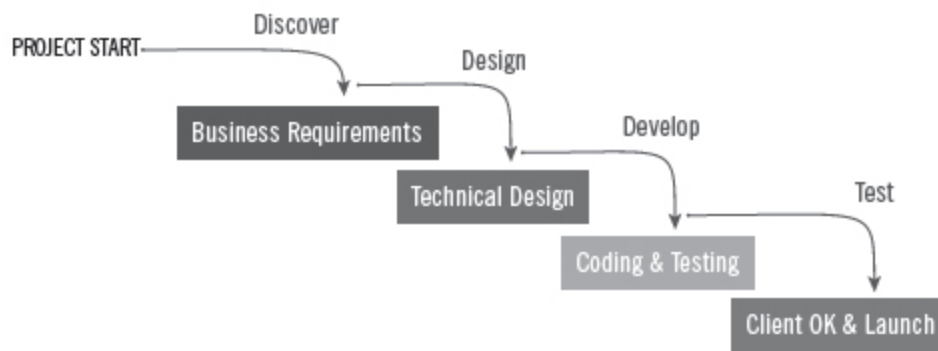
What went wrong and how the situation got fixed are why I'm writing this book. It wasn't that these weren't smart people. It wasn't that the Bureau didn't have the right personnel in place, or even the right technology. It wasn't about a work ethic or the right supply of competitive juices.

It was because of the *way* people were working. The way *most* people work. The way we all think work *has* to be done, because that's the way we were taught to do it.

When you hear what happened, it *sounds* at first as if it makes sense: the people at Lockheed sat down before they bid on the contract, looked at the requirements, and started planning how to build a system that would do all that. They had lots of intelligent people working for

months, figuring out what needed to be done. Then they spent more months planning how to do it. They produced beautiful charts with everything that needed to be accomplished and the time it would take to complete each and every task. Then, with careful color selection, they showed each piece of the project cascading down to the next like a waterfall.

WATERFALL METHOD



These charts are called Gantt charts, after Henry Gantt, who developed them. With the advent of personal computers in the 1980s making it easy to create these intricate charts—and to make them really *complex*—they have become works of art. Every single step in a project is laid out in detail. Every milestone. Every delivery date. These charts truly are impressive to behold. The only problem with them is that they are always, always *wrong*.

Henry Gantt invented his famous charts around 1910. They were first used in World War I by General William Crozier, who was the Chief of Ordnance for the US Army. Anyone who has studied that war knows that efficient organizational capability was not exactly a salient feature. Why a World War I artifact has become the de facto tool used in twenty-first-century project management has never been quite clear to me. We gave up on trench warfare, but somehow the ideas that organized it are still popular.

It's just so tempting: all the work needed to be done on a massive project laid out for everyone to see. I've visited many companies that have people whose *only job* is to update that Gantt chart every day. The trouble is, once that beautifully elegant plan meets reality, it falls apart. But instead of scrapping the plan, or the way they think about the plan, managers instead hire people to make it look as if the plan is working. Essentially, they're paying people to lie to them.

This unfortunate pattern echoes those reports the Soviet politburo was getting in the 1980s just before the total collapse of the USSR. A complete mirage. Now as then, the reports become more important than the reality they're supposed to describe, and if there's a discrepancy, reality is the problem, not the charts.

When I was a West Point cadet, I slept in Dwight Eisenhower's old room. At night, the streetlights would reflect off a gold plate on the mantelpiece and sometimes wake me up. DWIGHT D. EISENHOWER SLEPT HERE, the plate read. And I'd remember that Eisenhower once observed that planning for combat is important, but as soon as the first shot is fired, your plans go up in smoke. At least he had enough sense not to use a Gantt chart.

So Lockheed presented the FBI with all these lovely charts, and the Bureau signed on. Supposedly, the task was now so well planned out that nothing could go wrong. "Look, it's in the color-coded, time-stamped, bar-graphed plan."

Yet when Jeff and his boss, CIO Chief Chad Fulgham, looked at the plan in the spring of 2010, they knew it for what it was, what such charts *all* are, really: a complete fabrication. When the two men started to look at actual development and actual deliverables, they realized the problem was beyond fixing. New defects in the software were being discovered faster than old ones were being fixed.

Chad told the Department of Justice Inspector General that they could complete the Sentinel project by bringing development in-house, cutting the number of developers, and that, by doing so, they'd deliver the most challenging half of the project in less than a fifth of the time with less than a tenth of the amount budgeted. The skepticism in the usually dry IG reports to Congress is palpable. In the October 2010 report, after laying out their nine points of concern with the proposal, the IG watchdogs conclude: "In sum, we have significant concerns and questions about the ability of this new approach to complete the Sentinel project within budget, in a timely fashion, and with similar functionality..."²

A New Way of Thinking

This new approach is called "Scrum." I created it twenty years ago. Now it is the *only* way proven to help projects like these. There are two ways of doing things: the old "Waterfall" method that wastes hundreds of millions of dollars and often doesn't deliver anything, or the new way, which, with fewer people and in less time, can deliver more stuff with higher quality at lower cost. I know it sounds too good to be true, but the proof is in the results. It works.

Two decades ago I was desperate. I needed a new way of thinking about work. And through tons of research and experimentation and looking over past data I realized we all needed a new way of organizing human endeavor. None of it is rocket science; it's all been talked about before. There are studies going back to World War II that lay out some of the better ways that people work. But for some reason people never really put together all the pieces. Over the past two decades I've tried to do just that, and now this methodology has become ubiquitous in the first field I

applied it to, software development. At giants such as Google, Amazon, and Salesforce.com, and at small start-ups you haven't heard of yet, this framework has radically shifted how people get things done.

The reason this framework works is simple. I looked at how people *actually* work, rather than how they *say* they work. I looked at research done over decades and at best practices in companies all over the world, and I looked deeply at the best teams within those companies. What made them superior? What made them different? Why do some teams achieve greatness and others mediocrity?

For reasons I'll get into further in future chapters, I called this framework for team performance "Scrum." The term comes from the game of rugby, and it refers to the way a team works together to move the ball down the field. Careful alignment, unity of purpose, and clarity of goal come together. It's the perfect metaphor for what I want teams to do.

Traditionally, management wants two things on any project: control and predictability. This leads to vast numbers of documents and graphs and charts, just like at Lockheed. Months of effort go into planning every detail, so there will be no mistakes, no cost overruns, and things will be delivered on schedule.

The problem is that the rosy scenario never actually unfolds. All that effort poured into planning, trying to restrict change, trying to know the unknowable is wasted. Every project involves discovery of problems and bursts of inspiration. Trying to restrict a human endeavor of any scope to color-coded charts and graphs is foolish and doomed to failure. It's not how people work, and it's not how projects progress. It's not how ideas reach fruition or how great things are made.

Instead, it leads to frustrated people not getting what they want. Projects are delayed, come in over budget, and, in too many cases, end in abject failure. This is especially

true for teams involved in the creative work of crafting something new. Most of the time, management won't learn of the glide path toward failure until millions of dollars and thousands of hours have been invested for naught.

Scrum asks why it takes so long and so much effort to do stuff, and why we're so bad at figuring out how long and how much effort things will take. The cathedral at Chartres took fifty-seven years to build. It's a safe bet that at the beginning of the project the stonemasons looked at the bishop and said, "Twenty years, max. Probably be done in fifteen."

Scrum embraces uncertainty and creativity. It places a structure around the learning process, enabling teams to assess both what they've created and, just as important, how they created it. The Scrum framework harnesses how teams actually work and gives them the tools to self-organize and rapidly improve both speed and quality of work.

At its root, Scrum is based on a simple idea: whenever you start a project, why not regularly check in, see if what you're doing is heading in the right direction, and if it's actually what people want? And question whether there are any ways to improve how you're doing what you're doing, any ways of doing it better and faster, and what might be keeping you from doing that.

That's what's called an "Inspect and Adapt" cycle. Every little while, stop doing what you're doing, review what you've done, and see if it's still what you should be doing and how you might do it better. It's a simple idea, but executing it requires thought, introspection, honesty, and discipline. I'm writing this book to show you how to do it. And not just in software companies. I've seen Scrum used successfully to build cars, run a laundry, teach students in a classroom, make rocket ships, plan a wedding—even, as my wife has used it, to make sure that the "honey-do" list gets done every weekend.

The end results of Scrum—the design goal, if you will—are teams that dramatically improve their productivity. Over the past twenty years I’ve built these teams over and over and over again. I’ve been the CEO, CTO, or head of engineering of a dozen companies, from small start-ups with a few people in one room to large enterprises with offices spread across the planet. I’ve consulted and coached hundreds more.

The results can be so dramatic that leading research and analysis firms such as Gartner, Forrester Research, and the Standish Group now say that the old style of work is obsolete. Companies that still cling to tried-but-not-true ideas of command and control and that attempt to impose rigid predictability are simply doomed to fail if their competitors use Scrum. The difference is too great. Venture capital firms like OpenView Venture Partners in Boston, where I’m an adviser, say that Scrum offers too big a competitive advantage not to use it. These are not warm and fuzzy people; these are gimlet-eyed money men, and they simply say, “The results are indisputable. Companies have two choices: change or die.”

Fixing the FBI

At the FBI, the first problem the Sentinel team faced was contracts. Every single change ended up being a contract negotiation with Lockheed Martin. So Jeff Johnson and Chad Fulgham spent months unraveling all the contracts, taking the development inside, and cutting the staff from hundreds to under fifty. The core team was even smaller.

The first week they did what a lot of people in these circumstances do: they printed out all the requirements’ documentation. If you’ve never seen what that looks like on a large project, it can be hundreds and hundreds of pages.

I've seen stacks that are several feet high. I've seen this in project after project—people cut and paste and throw in boilerplate, but no one actually reads all those thousands of pages. They can't. That's the point. They've set up a system that forces them to endorse a fantasy.

"There were 1,100 requirements. The stack was a few inches thick," says Johnson. Just thinking about those documents makes me feel for the people who had probably spent weeks of their lives producing those documents that *had no purpose*. The FBI and Lockheed Martin aren't alone in this—I've seen this duplicated at almost every company I have worked with. That tall stack of futility is one of the reasons Scrum can be such a powerful change for people. No one should spend their lives on meaningless work. Not only is it not good business, it kills the soul.

So after they had this stack, they went through and prioritized each requirement. Which is vitally important and trickier than it sounds. Often people simply say that everything is important. But what they need to ask, what the Sentinel teams asked, was, what will bring the most *value* to the project? Do those things first. In software development there is a rule, borne out by decades of research, that 80 percent of the value in any piece of software is in 20 percent of the features. Think about it: when was the last time you used the Visual Basic Editor function in Microsoft Word? You probably don't know what Visual Basic is, let alone why you'd use it. But it's there, and someone spent time implementing it, but I guarantee you, it doesn't increase the value of Word by much.

Making people prioritize by value forces them to produce that 20 percent first. Often by the time they're done, they realize they don't really need the other 80 percent, or that what seemed important at the outset actually isn't.

For the Sentinel team, the question became, "Okay, we're doing this huge project that is vitally important that we've wasted hundreds of millions of dollars on. When will it be

done?” After thinking on it, they promised delivery in the fall of 2011. The Inspector General report from the fall of 2010 is a study in disbelief:

The FBI stated that it will employ an “agile methodology” to complete the development of Sentinel, using fewer employees from the FBI, Lockheed Martin, and the companies that have supplied the major off-the-shelf components of Sentinel. Overall the FBI plans to reduce the number of contract employees working on Sentinel from approximately 220 to 40. The FBI said that, at the same time, the number of FBI employees assigned to the project will also decrease from 30 to 12.... The FBI told us it believes it can complete Sentinel with the approximately \$20 million remaining in the Sentinel budget and within 12 months of beginning this new approach.³

The use of the phrase “agile methodology” shows just how little the IG knew about Scrum. The term “Agile” dates back to a 2001 conclave where I and sixteen other leaders in software development wrote up what has become known as the “Agile Manifesto.” It declared the following values: people over processes; products that actually work over documenting what that product is supposed to do; collaborating with customers over negotiating with them; and responding to change over following a plan. Scrum is the framework I built to put those values into practice. There is no methodology.

Of course Johnson’s twelve-month promise was somewhat misleading. Because, in actuality, they didn’t know; they couldn’t know. The FBI didn’t know how fast their teams could actually work. It’s something I tell executives all the time: “I’ll know what the date will be when I see how much

the teams improve. How fast they'll get. How much they'll *accelerate*."

It was also crucial, of course, that team members figure out what would *stop* them from accelerating. As Jeff Johnson put it, "I handled impediment removal." An "impediment" is an idea that comes from the company that first formed a lot of the ideas Scrum is based on: Toyota. And, more specifically, Taiichi Ohno's Toyota Production System.

I won't go into all the details here, but one of the key concepts that Ohno introduced is the idea of "flow." That is, production should flow swiftly and calmly throughout the process, and, he said, one of management's key tasks is to identify and remove impediments to that flow. Everything that stands in the way is waste. Ohno gives waste a moral, as well as a business, value in his classic book, *The Toyota Production System*:

It is not an exaggeration that in a low-growth period such waste is a crime against society more than a business loss. Eliminating waste must be a business's first objective.⁴

Ohno talks a lot about the different kinds of waste and impediments that can get in the way of production. For Scrum to really take off, someone in senior management needs to understand in his bones that impediments are nearly criminal. I'll tell you *how* to eliminate waste later on in the book. Suffice it to say here that the effect of eliminating waste is dramatic, but people often don't do it, because it requires being honest with themselves and with others.

Jeff Johnson knew that was his job.

It took the Sentinel team about three months to figure out how long completing the project would *really* take. Why? This goes back to that "Inspect and Adapt" cycle I

talked about earlier. Scrum works by setting sequential goals that must be completed in a fixed length of time. In the FBI's case, they decided on two-week cycles, with the understanding that, at the end of each cycle, there would be a *finished* increment of product. That meant they'd have something working, something that could be shown to anyone who cared to look but certainly the stakeholders and, optimally, the people who'd actually be using the thing.

This methodology allows teams to get near real-time feedback on their work. Are they headed in the right direction? Is what they're planning to do next *really* what they should be doing, given what they've discovered during that cycle?

In Scrum we call these cycles "Sprints." At the beginning of each cycle there is a meeting to plan the Sprint. The team decides how much work they think they can accomplish during the next two weeks. They'll take the work items off that prioritized list of things that need to be done and often just write them out on sticky notes and put them on the wall. The team decides how many of those work items they can get done during this Sprint.

At the end of the Sprint, the team comes together and shows what they've accomplished during the time they've collaborated. They look at how many of those sticky notes they actually got done. Did they bring too many into the Sprint and not finish them all? Did they not bring enough? What's important here is that they begin to have a baseline sense of how fast they can go—their velocity.

After they've shown what they've done—and here's where Ohno's ideas come in—they discuss not what they did, but *how* they did it. They ask, "How can we work together better in the next Sprint? What was getting in our way during the last one? What are the impediments that are slowing our velocity?" You can find a more detailed explanation of how Scrum works in the appendix.

And that's why Jeff Johnson needed a few months before he could really tell how long the project would take. He wanted to measure the velocity of each team measured over a few Sprints and then see how much they could improve—how much faster they could go. Once he looked at how many work items each team had finished in each Sprint and then checked how many they had remaining until the end of the project, he could forecast a completion date.

Besides learning how fast the teams were going, he also wanted to know what impediments were slowing them down. What he really wanted to do was *accelerate* those teams so they were producing faster—not by working longer hours (I'll go into why that's a fruitless rat hole that ends up making things take longer later) but by working *better* and *smarter*. Jeff Johnson says his teams increased their productivity by *a factor of three*. They were going three times as fast once they got moving as compared to when they started. Why? They got better at working together, yes, but most important, they figured out the things that were slowing them down, and each cycle, each Sprint, they'd try to get rid of them.

It eventually took the Sentinel project eighteen months of coding to get the database system deployed, and another two months to deploy it to the entire FBI. "Tremendous time pressure," Johnson said when he sat down for an interview. "And you have to understand, the system is used for *everything*. Paying informants. Storing evidence. Case files. Calendars. This *meeting* is in Sentinel."

And the most powerful part of Scrum from his point of view? "Demos. Driving toward a demonstrable product on a frequent basis." Every two weeks the Sentinel team would demonstrate what they'd accomplished. And this show-and-tell wasn't just to themselves. They were taking what they'd achieved and running it by the people who would actually be using the system. Everyone who had a stake in

the project sent someone and that could make for a pretty full house. Records. Intelligence. Special agents. The Office of the Inspector General. Representatives from other government agencies. Often enough, the Director and Deputy Director of the FBI were in the room, as was the acting Inspector General herself. This was *not* an easy crowd.

And that was what made it work, says Johnson. “Scrum is not about the developers. It’s about the customers and stakeholders. Really, it was an organizational change. Showing the *actual product* was the most powerful part.”

Actually showing the product was powerful, because people were, to put it mildly, skeptical of the team’s reported progress. They just couldn’t believe Sentinel’s progress actually kept moving at a faster and faster rate. “I was saying to Congress that with 5 percent of the budget and in twenty months we were going to accomplish what Lockheed couldn’t do with 90 percent of the budget in ten years,” says Johnson. “There was skepticism in the room. We had to provide reports to the Associate Attorney General. We would be transparent with our status, but our audience would assume something devious was going on. Anytime they’d seen those kinds of indicators in the past, the reports were less detailed, and something else *was* going on.”

And that skepticism infected the rest of the FBI. *The guys down in the basement are just going to screw it up again*, was the thinking. This will just be one more temporary system that will fail, and we’ll have to go back to using paper.

Jeff told his team about a passage he had to memorize when he was a Naval cadet at Annapolis. It was from Teddy Roosevelt’s speech “Citizenship in a Republic,” which he gave at the Sorbonne in 1910. It is oft-quoted, and you may be already familiar with it:

It is not the critic who counts; not the man who points out how the strong man stumbles, or where the doer of deeds could have done them better. The credit belongs to the man who is actually in the arena, whose face is marred by dust and sweat and blood; who strives valiantly; who errs, who comes short again and again, because there is no effort without error and shortcoming; but who does actually strive to do the deeds; who knows great enthusiasms, the great devotions; who spends himself in a worthy cause; who at the best knows in the end the triumph of high achievement, and who at the worst, if he fails, at least fails while daring greatly, so that his place shall never be with those cold and timid souls who neither know victory nor defeat.⁵

The team did have some delays as they figured out exactly how fast they could do things, and just how hard things were to do. Finally in July of 2012, they turned Sentinel on. And they had to turn it on all the way, to everybody, all at once. There was no way to stage it.

“It happened from one day to the next. In a criminal case or a counterterrorism case, something in Los Angeles might be related to something in Chicago,” says Jeff Johnson. “We couldn’t allow leads to be lost. *At every point we had to have a clean and known good state.*”

And that state had to be clean and good enough to hold up in court. The data in Sentinel was being used to prosecute people, and its integrity had to be beyond a shadow of doubt.

Jeff was frantic and nervous that first day. He went into his office and turned on Sentinel. It loaded. That was a good thing. And then he tried to approve a document with an electronic signature—a basic everyday task that tens of thousands of FBI employees would have to do all the time. Up came an error message. It didn’t work. He started to