

The Posit Science Story

Scrumban Stories

During a year of cultural change at Corbis, Corey Ladas joined our team as a process coach. I needed help coaching the rollout of Kanban across our portfolio. One day he came to me and said, “I am finding that the projects where they are still using a traditional SDLC (software development lifecycle) process need different coaching from those who have adopted (the Agile software development methodology) Scrum.” What we were seeing was the emergence of what Corey later named Scrumban—the application of Kanban to a starting position where a project team or an organization consisting of many such teams (typically six to eight people per team) had already adopted Scrum as their way of working and coordinating with each other.

We see two common varieties of Scrumban stories in our work: The first is where Scrum has helped an organization initially but improvements have plateaued and stubbornly refused to improve further; the other is where circumstances have changed—the market has moved, customer expectations have changed—and consequently the Scrum approach of two-week timeboxes, known as sprints, of planning, working, and checking in with stakeholders with a demonstration and retrospective is no longer appropriate. In the first category, we often find that it takes a long time to recognize that things aren’t improving. Managers often persevere with Scrum for months or years before they are ready to look for an alternative: two years is typical, whereas up to four years isn’t uncommon. I know this from the experience of receiving emails asking for help and reading the stories. “Things haven’t improved any further for the past two years and we’d like someone to take a fresh look and suggest some new ideas.” This is typical of such a situation.

With the second category, the timeframe is shorter. Quite simply, due to changes in context and circumstances, the two-week sprint approach isn’t working and is causing stress, anxiety, and pain, resulting in dysfunctional behavior and tension among the team, managers, and customers. In both situations, there is motivation to find “an alternative path to agility”—to be able to move quickly, respond to change, and adapt to customer needs appropriately. Regardless of the motivation, given that the organization was already using Scrum, their choice to introduce Kanban in order to take them further means that they, too, have a Scrumban story.

Corey has said, “Scrumban is a journey.” To better understand that, I’d like to relate one such journey, from one of the earliest Scrumban stories, that of Posit Science in San Francisco. Their story is primarily of the second type—their circumstances had changed, and consequently Scrum was no longer serving their needs. Their motivation was to find a new way of working, to meet the expectations of business owners, and to relieve their research and development organization of overburdening. They needed smoother flow, more predictable delivery, and a sustainable pace of work. Kanban proved a good and effective choice as a way forward.

Posit Science Background: The Brain Aerobics Company

Dr. Michael (Mike) M. Merzenich, the founder of Posit Science, had a long history of scientific achievements. In the late 1980s, he was a member of the team that invented the cochlear implant, a device that enables the deaf to hear. In the 1990s his career moved into neuroscience and specifically the field of brain plasticity. He is the author of *Soft Wired: How the New Science of Brain Plasticity Can Change Your Life*.¹

¹ <https://www.amazon.com/Soft-Wired-Science-Brain-Plasticity-Change/dp/0989432823/>

For much of his career, Mike Merzenich had known how important continuous learning was for adults. As a neuroscientist, he knew that the most important role of continuous learning was for the individual and the health of their brain, especially as it aged. Professor emeritus at the University of California, San Francisco, he had many achievements in his field. His devotion to and research of brain plasticity had earned him a spot on the National Academy of Science in 1999. Developing an understanding that brains were plastic, and could be exercised and coached, had disrupted the long-held belief that human brains stopped changing past early adulthood. It had long been assumed that after completing its development in young adults, the brain changed little, beginning its decline, demise, and deterioration. It was assumed that there was nothing medicine or technology could do to prevent it.

Mike and a few like-minded colleagues believed otherwise. They believed that brains could be trained and molded even in adulthood. That belief stemmed from the simple observation of people from different cultural contexts. While everyone would agree that children developed differently based on their nurturing and context, few had thought that to be true for adults as well. But as he had observed, older humans continued diversifying more as time passed. They learned new skills, even late in life. The brain was flexible and changing, its adaptability and plasticity never really disappearing. Could that plasticity trigger change that would negate deterioration, they wondered. Together with his team of researchers, Mike dedicated his career to finding the precise triggers for brain plasticity.

By 2004 Mike was ready to speak publicly on the subject. He gave a TED talk² that February in Monterey, California. People do not forget because the brain has forgotten to remember, he explained. People lose their memory because the brain starts representing the things they are seeing and hearing and feeling less saliently. “When you’re young and you see something surprising, your eyes are attracted to it. You are bright-eyed, literally. Your eyes take a series of snapshots that reveal information about what’s out there.” According to Mike, these snapshots leave a footprint in the brain, which keeps the machinery active. But as the footprint of what is seen or heard becomes less clear and vivid, the machinery behind it starts to falter. As a result, memory loss and neurological decline begin. As time goes on it becomes less and less active and eventually begins to die.

Mike, of course, believed there was an antidote to all this.

Simply keeping your mind active was not enough to fix it, he believed. To truly offset the demise of the machinery in your brain, what was necessary were very specific, challenging activities. Those challenges to keep your brain sharp could be in the form of continuous learning, such as taking up a foreign language or learning to play a musical instrument such as the guitar. Or what he claimed would be the thing of the future, “brain aerobics”—interactive games that take advantage of brain plasticity to kick-start the brain’s natural learning mechanisms.

Games created specifically to engage the otherwise deteriorating parts of the brain could offset declining cognitive abilities. As the scientific community was improving its understanding of specific neurological issues, he felt more confident that training activities could be designed to harness the brain’s plasticity to create and reinforce neural pathways for specific results. And as computer technologies improved, those brain aerobics training methods could become more advanced and more precise. Through the use of complex algorithms, they could monitor responses and adjust in a manner that heightened engagement and became customized to address individual deficiencies.

As with many other enlightening TED talks, this was pure science fiction to the audience. If it were true, if people could sustain their cognitive abilities just a while longer, that would be one of the most phenomenal discoveries of the century. Mike had already initiated the steps to make it a reality. Some months earlier, he and Henry Mahncke had founded Posit Science—the company that would commercialize brain aerobics with a series of interactive computer games.

² http://www.ted.com/talks/michael_merzenich_on_the_elastic_brain

They had chosen the name “Posit” on purpose: It means “to put forward or advance” and it reflected their hope to make a positive change in people’s lives. “Science” also meant something—this wasn’t superstition or belief, nor was it entertainment—this wasn’t a games company; it was a scientific endeavor producing a medical-grade product designed to help people with dysfunction in their cognitive abilities, whether caused through aging or injury.

I first encountered Posit Science in 2009. There is only one other organization I’ve visited that gave me a similar feeling, based on how I experienced the culture, employees, and what they said about why they had joined the firm and what they hoped to achieve there—The Bill & Melinda Gates Foundation. The people at Posit Science had all joined for altruistic reasons. They had joined to do some social good—and to give something back to a wider society. Posit Science paid people well enough, but salaries were perhaps 15 percent under the market norm for San Francisco. This is true also of the Gates Foundation, where salaries are typically below the market rate for Seattle, and the employees all have some altruistic motivation for working there. In both cases, people joining these organizations believe in the mission, and it gives them each a sense of contributing to the greater good.

Posit Science as a tribe consisted of three distinct sub-tribes: the scientists, neuroscientists who had produced the original research; the game developers, who produced the product; and the business people, who took that product to market. Regardless of their background, they all bought into the vision, mission, and purpose and believed in Mike Merzenich. While they all were being adequately remunerated for their work, they all were there for altruistic reasons, too, to give something back to society. If it was only about the money, they’d have been working somewhere else. So although there were three distinct social groups, Posit’s purpose (the vision and mission) was the unifying factor that made them a highly cohesive super-tribe.

Together, the research team and the games developers strived to move technologies out of the laboratories and into the hands of people, where they could do the most good. Those who needed it most urgently, and of primary interest to Posit, were the elderly. Could computer-based training help them “keep their brains”? Could it enable them to be more childlike, grasping everything more vividly? Posit was developing a set of exercises called *The Brain Fitness Program*.

What Dr. Merzenich and Posit Science were aiming to do with their brain aerobics program was to address all three key problems of brain decline: the slowing of the brain’s processing speed, the weakening of sensory brain signals, and the decrease in production of key brain chemicals. They believed that by presenting the right stimuli—in the right order, with the right timing, and through intensive, repetitive, and progressively challenging programs—they could tackle all three. They believed that brain games and training was the solution.

No other species plays with objects and with language as much as humans do. We pursue a wide variety of entertainment for long stretches of time. We like enjoyment. That, of course, is no coincidence, as play is an important evolutionary tool. Physical, cognitive, and social development, as well as general training for emergencies and disasters, have their roots in play or simulation games. As Francis Steen from the University of California, Los Angeles, puts it, “Playing is an evolutionary adaptation for learning, it is a sort of a simulator that allows children and adults alike to imagine and try out different scenarios with little risk.”³ For Posit, those imaginary and playful scenarios could help bring much-needed changes in the brain.

Posit Science’s extraordinary and noble cause attracted the attention of investors. The startup received its first round of venture financing and began operations on October 1, 2003. With the money in place, it was possible to assemble a global consortium of brain scientists to develop, test, refine, and validate exercises that rejuvenated the brain. In the first years, they tested those games in a few chosen retirement homes where they set up learning centers to observe the games’ effects on the residents. They looked for

³ <https://www.newscientist.com/article/mg21428610.300-human-nature-being-playful/>

improvements (or lack thereof) in cognitive function stemming from their exercises. Scientists, determined to have clinically proven technology, needed substantial data that their brain training exercises worked before they commercialized their products.

To summarize: we have a newly formed, venture capital–funded startup with a powerful and unifying vision employing world-class neuroscientists and successful game developers commercializing brand-new science, all in the highly regulated medical industry with a nascent and unproven market located in the high-cost, high–burn rate center of the global technology industry, San Francisco, California. This is what defines them.

By the end of 2005, Posit had positive results. During the Society for Neuroscience annual meeting in Washington, D.C., Mike Merzenich presented results from one of the first studies on brain games. The study⁴ showed that participants using the program had improved in neurocognitive status by ten or more years, on average. The study had been conducted at Rossmore, a retirement community near San Francisco, California, with ninety-five volunteers aged sixty-one to ninety-four. Researchers compared results from memory and cognitive assessment scores in participants enrolled in forty one-hour training sessions with both a control group using a computer and with a no-contact group. Participants using the brain plasticity–based training program had improved. Participants who had completed more difficult levels of exercises showed even greater improvements. Soon after, Posit Science released the first ever *Brain Fitness Program* CD-ROM. In March 2006, they began selling it through a partner network. The price tag was \$395 for a single user. It was expensive, but the value stemmed far beyond the dollars, or so Posit and its team of scientists believed.

This product launch represented the first major milestone, and punctuation point, in the history of Posit Science. Until then, they'd been in scientific mode—researching, experimenting, and slowly burning up their investors' capital. Now, in 2006, they were a commercial entity, seeking to make money from their product and, they hoped, one day become cash-flow positive, no longer in need of investment to continue operations.

Punctuation points in the history of an organization are always good opportunities to introduce change. Punctuation points can take many forms:

- Launch of a first product
- Taking an investment round
- Arrival of a new CEO or leader
- Key employee exit (typically a founder or a creator of intellectual property)
- Merger, acquisition, divestiture, or takeover
- IPO (initial public offering—taking a company public from previous private ownership)
- Regulatory, legal, political, or major economic changes (such as a financial crisis)
- Outsourcing and/or offshoring of work
- Company reorganization
- Retrenchment
- Arrival of a new disruptive competitor or business model (such as budget airlines)
- Arrival of a disruptive innovation into a market (such as commercial jet aircraft into the flying boat and ocean liner transportation business)

Famously, there is the concept of “the first 100 days.” It is used, for example, to refer to the tenure of a new president or head of state or a new leader in an organization. The 100 days starts with the punctuation point and runs for approximately three months. During this period, everyone is adjusting to the turmoil, and a new leader gets to blame his or her predecessor or the conditions that existed before they took control. As a consequence of the punctuation point—or the conditions that existed prior to it, metaphorically a “global warming” condition, which provokes some engineered punctuation point such as a reorganization of a business—the new leader has an opportunity make changes without much resistance.

⁴ <http://www.brainhq.com/news/brain-training-program-enhances-memory-cognition>

Posit had a new product in the market, but it had taken a long time to come to fruition; and the software code was fragile. They took the opportunity to hire a new head of software development, David Hoffman.

Hoffman quickly realized that Posit suffered from a common problem for young software companies: their software code was fragile, and the engineering of their product was such that it could be described as a prototype. This is typical of first-generation software products, where the focus has been on exploring the functionality and features needed to serve a market rather than on the integrity of the code and its underlying architecture. It was first documented by Fred Brooks in his classic work, *The Mythical Man Month*,⁵ in which he said, “Plan to throw one away, because you will anyway.” In other words, the first generation of a product always has code of poor internal quality, and the business discovers that it must scrap the code and start again for a second-generation product.

Until this point in its history, Posit Science wasn't a typical Silicon Valley startup. Since its founding, employees had worked normal, humane hours, which afforded them quality family time and a sustainable life. For people concerned with brain health, they knew all too well that it does not deal well with overburdening. Perhaps more than any other software development organization on the planet, Posit Science developers understood the damage they could do by working too hard while anxious, overburdened, and stressed. However, this comfortable work-life balance that had been a signature of the Posit story so far was now under stress and would soon break. The fragile code behind the Brain Fitness Program was generating a lot of rework as defects were discovered, and maintenance to add minor enhancements was much more problematic than anticipated. Meanwhile, Posit had started to work on a second generation product, a set of games to be called Insight. As things got more complex, the heat started to rise. Their work-life balance was slowly eroding by the urgency of the work and Hoffman's department was, consequently, growing anxious and stressed.

Hoffman decided he had to take action: It was time to start again, and they had to develop a new system architecture and a new, cleaner set of software code that would be much more robust and easier to maintain. I was creating my own, smaller punctuation point as a response to the global warming problem in my department. We scrapped the existing code base and started again for the new product, and I took this opportunity to introduce a new way of working. The department adopted the Agile software development methodology known as Scrum. The product developers were motivated and ready for change. They embraced the opportunity with alacrity. Consultants and trainers arrived, and a popular Agile project management software tool was purchased to help everyone track work and report progress.

The Rise and Fall of Scrum (Part 1)

David Hoffman hired a project manager to lead the changes and help his organization get through all the work that was accumulating. It's at this point that Janice Linden-Reed enters this story and our story, the Kanban story. Janice had had a long career, beginning in the early 1990s, as a games designer, producer, and executive at firms such as Maxis and TEN (Total Entertainment Network). She joined Posit as a senior project manager, attracted to the firm by a close friend, a former games developer who'd joined the executive team at Posit. The work-life balance was the first thing she noticed. The games industry doesn't have the best reputation for maintaining a sustainable pace. She'd spent too many nights at previous jobs sleeping under her desk with a never-ending pile of work on top. Relatively speaking, Posit was much more relaxed.

With the decision to adopt Scrum, David Hoffman believed his people could work smarter rather than harder.

⁵ Brooks, Frederick P. *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2 ed.)*. Reading, MA: Addison-Wesley, 1995.

Adopting Scrum changed many things at Posit, from the layout of the office and the breakdown of work into smaller pieces that could be completed earlier to the variety of new meetings, such as the daily “scrum” and the periodic “sprint planning”; change swept the employees off their feet. It was difficult in the first year. They weren’t used to the overwhelming transparency that came from using an Agile project-management tracking tool whereby everyone could see the state of all of their work at any time. None of the developers had been used to the close examination of their own work on such a regular basis, yet they had desperately needed to change how they were working. The old way wasn’t sustainable. With time, they got used to it and things started to improve. With a better idea of what was going on, they delivered faster. The switch to Scrum was widely regarded as a good thing. Scrum is ideal for low-maturity organizations looking to add process and bring some control to the chaos of their environment. One of the creators of the Scrum approach, Ken Schwaber, named his website *controlchaos.com*. Ken knew exactly what sort of problems he was trying to solve with the design of Scrum, and David Hoffman’s decision to adopt it at Posit was entirely appropriate.

Posit made only one alteration to the canonical Scrum definition: they agreed on a three-week time period for each “sprint” instead of the usual two. Part of Posit’s process was “clinical validation testing.” This wasn’t testing the software for defects; rather, it was testing completed functionality—the actual brain game—to validate that it delivered on the underlying science. Clinical validation testing demonstrated whether the product delivered on the anticipated clinical outcomes and provided the expected medical benefits. This meant testing the product with patients to measure and validate expected improvements to their brain function. This takes time; the game initiates a chemical process in the brain to strengthen neural pathways. To see results takes at least a few days, possibly longer. Consequently, a minimum of one week was needed for clinical testing of new functionality. Given this overhead for every sprint, they decided that two weeks were needed to develop sufficient functionality to be worthwhile testing with patients, and hence, a third week was needed to gather the results of the tests. While Posit’s products, and the general field of brain plasticity, were not yet regulated by the FDA (U.S. Food and Drug Administration), Posit proceeded as if they were. Their training as scientists didn’t allow them to cut corners on effectiveness. Their careers and reputations hinged on these brain aerobics games delivering on their clinical claims. There could be no compromise on clinical testing.

Scrum gave them a regular rhythm. They avoided too much overburdening. They had time and space to architect and code the Insight suite of games with much higher quality than the Brain Fitness Program that preceded it. Insight was launched successfully in 2007. Like so many Scrumban stories, Posit’s story starts with an appropriate and successful adoption of Scrum.

Posit employed almost 100 people in downtown San Francisco; its burn rate in 2007 must have been well over a million dollars per month. There was an expectation that with two products in the market, revenue from sales would start to improve cash flow. The investors’ money wouldn’t last forever. If revenues did not improve, it was foreseeable that the business would run out of money. Executive attention began to switch from science and product development to finance and sales.

The Rise and Fall of Scrum (Part 2)

Janice made an effort to learn everything there was to know about Scrum so she could help the developers make better use of it. She grew to believe greatly in its teachings. She appreciated the predictability, the honesty, and the absence of fear. As much as she liked it, she gradually began to notice that the developers continued to experience many problems. There was still too much work to do. Aside from the creation of the games, the development group had myriad other things to do. Providing support to the users of the Brain Fitness program was one. Participating in the scientific research for the new games was another. Beyond that, they also worked with the learning centers at the retirement homes and needed to be aware of and adhere to FDA and other compliance requirements. They were assisting with the IMPACT study, a joint project of the Mayo Clinic and the University of Southern California, the most sophisticated study to date of the effectiveness of brain training

games. On top of all that, developers also worked closely with the marketing and sales departments. The number of sources of demand and the impact it had on the day-to-day work of the developers was growing as time went by and both the product lines and the customer base grew.

Janice found that no matter how much she tried to help, all of these demands were overwhelming. She witnessed how unbearable it was becoming for the developers. Planning meetings were long and excruciating, sprints were disrupted with pre-emptive, urgent work, and the developers and testers were exhausted. Almost always too optimistic, the development team took on more than they could deal with and missed many deadlines, breaking promises as a result. The trust between the business people and the developers began to deteriorate. Questions about their ability to deliver lurked ominously in the atmosphere around their San Francisco office.

The sprint planning meeting every three weeks was dreaded by everyone. The situation worsened as the product backlog grew larger than ever. By the beginning of 2008, this wish list from business owners, customers, and regulatory stakeholders had grown to more than 800 requests. Sprint planning had become an intense event during which the developers would have to decide what to work on and what to leave until later. The challenge of selecting around forty items from an available set of 800-plus was overwhelming. Any new request for the backlog needed to be analyzed and broken down into so-called stories that were considered small enough to complete within one sprint. Then the stories had to be estimated to determine the anticipated number of hours of work. Next was a triage to pick the work to start immediately versus everything else that had to wait until later. There were seven stakeholder groups in each sprint planning meeting, with two representatives from each group, plus two developers, and Janice as facilitator. In a specialized field such as brain plasticity science, you might expect a highly specialized workforce, and once you add those business functions—including customer care—it is easy to see why so many people needed to attend. Everyone complained about these meetings. They were too long, too stressful, and seemed to add little value, as sprints were constantly being interrupted with new work that was urgent and critical. No one really wanted to be involved with it any longer, they simply wanted to work. Some people stopped attending altogether. As is often the case when people with valuable information fail to attend, participate in debate, and contribute to decisions, the result is poor quality decision making. This leads to further complaints about bad decisions, and a vicious cycle ensues. Janice would try anything she could think of just to make the planning meeting a little bit more bearable. She would bring toys so that people could fiddle with them and relieve some of their frustration. Yet the meetings were so intense that the relief was too little to make a difference. Janice came to fear these meetings. She lost sleep over them. She suffered anxiety and trepidation about how each new sprint planning session would go.

What was happening at Posit was that their circumstances were slowly changing. They were, once again, (metaphorically speaking) in a global warming condition. Things were slowly heating up such that on a day-to-day basis no one would notice the changes, but seen over a longer time span, it was clear that the situation was deteriorating. The business pressure to have successful revenue-generating products was increasing as the investor funding ran down. There was greater complexity and much more urgency in their environment. The business had become reactionary to every revenue-generating or investment opportunity that came along. Planning every three weeks wasn't often enough. The conditions that had enabled a successful Scrum workflow no longer existed.

Too many blocked work items—due to preemption by something even more urgent and critical—meant increased multitasking. Deliverables were taking longer to complete, and releases were increasingly unpredictable. Developers were wearing out and over tired. Others in the company thought they were simply lazy. Relationships were strained. Janice felt that the assumption of laziness and lack of motivation was unfair and untrue. She wanted to help her department of developers. She felt direct responsibility. She initiated conversations to help understand what was wrong and how to make it better. She began researching whether other software development organizations were experiencing similar problems. She sought advice anywhere she could. She found the consultants and coaches hired from their Agile project management software vendor to be unhelpful. They blamed the developers, saying they did not adhere to

all the rules of Scrum, that they lacked discipline. Scrum could not be wrong if it was applied properly, they said. It could not fail; if things were not working, then it could only be the fault of the people involved.

Janice felt this guidance from their external coaches was unsatisfactory, even insulting. This was a team of successful career games developers and Ph.D. neuroscientists. Wasn't it likely that they had needed a lot of discipline to achieve success in their careers? And if they could reverse the effects of brain aging, wasn't it also likely they were capable of reading and following a prescriptive process recipe? Janice knew how smart these people were, how devoted to the product they were, and how motivated they were to use their knowledge and experience to benefit those who might need help boosting their brain power. They were not lazy. They were not rebellious. It was shocking that their paid advisers treated them with such disrespect.

What Janice was hearing from the consultants was rooted in guidance from Ken Schwaber, co-creator of Scrum, who said, "Scrum is designed to work in a context. Your job is to create the context so that Scrum works for you."

This statement really defines Scrum as the antithesis of Kanban's start-with-what-you-do-now approach. Scrum requires that you change your context to facilitate the method of working. It's inward focused and self-serving in nature. From a developer's perspective: "Because I feel overburdened and stressed by the chaos around me, everything in my world must change to facilitate me doing my work without interruption and with high quality."

Kanban embraces the context you have and enables the way of working to evolve, adjust, and optimize to its environment. Scrum requires that you change the environment. For Posit, it seemed that they didn't control their environment, their market, or their impoverished circumstances; they were running out of money and desperate to keep their vision afloat, surviving any way they could.

What had enabled the success of Scrum at Posit a year earlier was that their world wasn't yet sufficiently chaotic. It wasn't yet sufficiently complex. While developing a single product and with plenty of investor capital to burn, the environment was relatively simple. Introducing three-week planning horizons and small batches of work to fit into those three weeks was just fine. As time went by and the scale increased—with more products, more customers and other stakeholders, and an ever-larger backlog of work on their wish list—and with ever-increasing pressure to chase revenue and business opportunities as investor capital was running short, Scrum simply broke down for them. It wasn't anyone's fault. It wasn't a lack of discipline. Nor was it an inability to control the environment and create the context in which Scrum would work successfully. Suggesting that Posit could have modified their environment to solve their problems, was, and would be to this day, wishful thinking. "If only we had deeper-pocketed investors and more patient capital behind us, then Scrum would work for us." "If only new business opportunities didn't arrive so frequently and unpredictably, requiring proofs of concept and demonstrations, scheduled at the client's convenience, then Scrum would work for us." There is no wishful thinking in Kanban, and if you find yourself saying, "If only . . .," then you've already fallen from the path of pragmatism.

Janice kept on looking for explanations and ideas for how to help her developers. For months she used every free moment to watch webinars, read blog posts, and have discussions with some of the best in the field. On her commute to and from work she listened to various podcasts every day.

One day she stumbled upon a blog post that described problems like Posit's. In the post, the author explained how, in an attempt to solve their problems, they had stopped doing one of the essential practices of Scrum—they had dropped the use of time-boxed two-week sprints. Instead of worsening their performance—as they'd been warned by Agile software development consultants—it had helped them. The "permission giver" for this change had come from another blog post, a report of a presentation by Corey Ladas during the 2008 Toronto Agile conference. From Corey's session, the author had taken away the realization that there was another way forward if Scrum did not work for your circumstances: using

Kanban was *an alternative path to agility*. From just these two blog posts, Janice liked how Kanban sounded. Limiting work-in-progress (WIP) seemed a simple yet powerful concept.

Fascinated by the affirmation that the fault was not with the developers but in their method of working, it was Janice's turn to introduce a small punctuation point. She suggested Posit make a shift and introduce Kanban.

Kanban Gets Rejected

Janice was sure her colleagues would embrace any kind of improvement, but to her surprise they rejected any change suggested. Perhaps Scrum had become too much a part of their identity? For two years, they'd had its rules and practices drilled into them by their external coaches. Scrum had become popular in the San Francisco Bay Area, and there was broader social pressure from fellow professionals to be seen as part of the movement. Meanwhile, they'd been criticized, belittled, and ridiculed for their inability to make it work for them. They'd been made to feel guilty for their lack of discipline. They didn't want to be seen as quitters. There was some risk of professional social ostracization for taking a different tack. Scrum had to stay.

It seemed to Janice that while Scrum had worked well in those early days, circumstances had changed such that Scrum's policies and practices were literally hurting the developers. And yet they resisted change. The idea of change seemed to be even more painful than their current situation. Taken aback, she continued to read everything she could find, trying to understand Kanban better.

She began noticing some of the ills described in nascent Kanban literature at the time. During the daily team meeting, it was clear that the developers were working on pretty much everything in the sprint, all at the same time. There was a lot of multitasking, and individuals were clearly overburdened. It had never occurred to her just how problematic this was until she read about limiting the work-in-progress. While she could see the problems—and a solution—her team did not want to deviate from the definition of Scrum they'd been trained to follow.

Janice thought that Scrum did not say anything about limiting WIP. She had never heard that, nor had it been mentioned by any of Posit's professional external coaches. Actually, you have to look deep in Scrum literature, and back to its early days, to find advice from Jeff Sutherland, the other co-creator of Scrum, to find mention of focus. Team members are supposed to focus and not start too much work all at once. However, this guidance never specified a WIP limit or even the concept of a policy to limit WIP. It was just general guidance, loosely worded, suggesting that individuals shouldn't voluntarily overburden themselves. In 2008, it was rare to find a coach who even knew about, much less taught, this Scrum practice of focus.

Janice didn't give up. She kept on planting the seeds of the possible change and improvement. She waited for people to be ready for it. Eventually, David Hoffman intervened. He agreed that something needed to change. He would show the leadership the development department needed. Sometimes people need help to help themselves. They need leadership. He was willing to give it a try, to give Kanban a try.

	BEFORE	AFTER
Iterations	✓	✓
Scrum Master, PO,	✓	✓
Sprint planning	✓	✓
Daily Standup Meeting	✓	✓
Product Owner accepts	✓	✓
Demo	✓	✓
Retrospective	✓	✓
Estimation	✓ By TASK	✓ By User Story (T-shirt sized)
Other		Per Person WIP LIMIT

Figure 1 Posit Science’s Scrum implementation practices, October 2008

However, there was still resistance and fear. The developers pushed back against a full Kanban implementation and a kanban (signal) system to pull work when they had capacity. Janice had to back off and reduce the scope of the changes. In October of 2008, she was able to make just three simple but important changes: she was able to extend their visual board to upstream analysis, introduce personal WIP limits, and drop the Ken Schwaber–style of estimating each request in hours of work, replacing it with a simpler system that simply asked for a “t-shirt size” ranging from extra-small through extra-large (XS, S, M, L, XL). It was agreed by consensus that individuals would work on no more than three things at the same time: their per-person WIP limit would be three. This was visualized on the board by introducing small avatars—photographs of the team members mounted on magnets. Each person had three avatars, and they would place these beside tickets on which they were contributing some effort. Everyone could see who was working on what, who was collaborating together, and which tickets were currently being ignored. The changes to practices are summarized in Figure 1 while the new, extended board is shown in Figure 2.

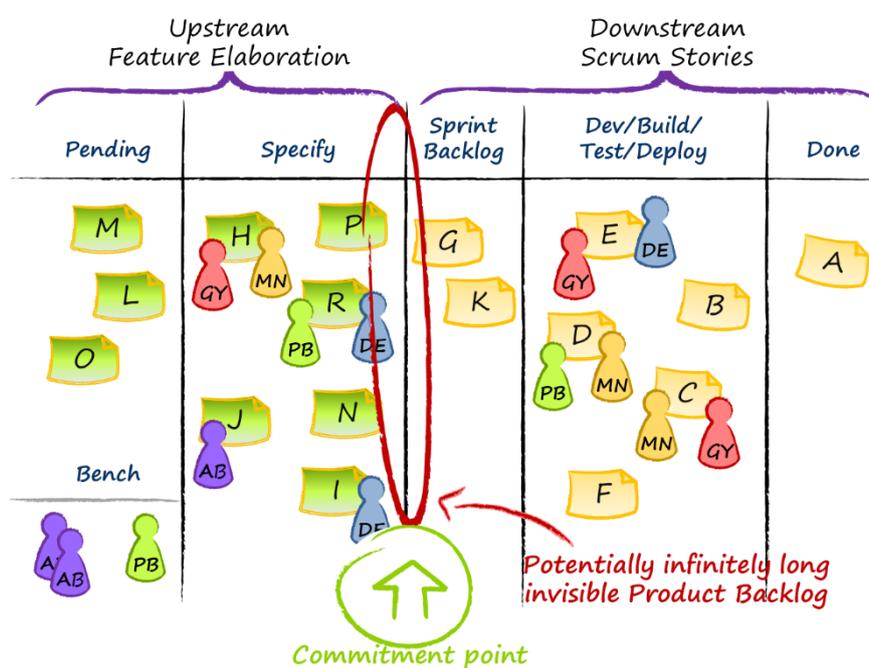


Figure 2 Posit Science’s extended visual board, October 2008

On the extended board, the Scrum process is shown on the right-hand side, or downstream, while an upstream activity to elaborate customer requests—transforming a feature request into a set of user stories—is shown on the left-hand side. Tickets flow from left to right. Every third week, the sprint planning activity provided the commitment point. Stories selected for a sprint are committed, while the product backlog contains a potentially unbounded number of uncommitted stories. The product backlog isn’t visualized on the board, rather, it is stored in the software tool.

This visualization reveals some important details that were perhaps opaque previously: the people doing the elaboration work shown in the Specify column are, in fact, the same people doing development work. Hence, developers from the team multitask among estimated, planned, and committed work and unplanned, upstream, requirements-elaboration work. Requirements elaboration at Posit was unplanned, ungoverned, and disruptive. Part of their inability to deliver on sprint commitments was due to the distraction of working on unplanned requirements elaboration.

As a general rule, we never want to see a situation where workers cross a commitment point and multitask between planned, committed work and unplanned, uncommitted, optional requests. This is especially true when specific commitments to delivery dates have been made.

So, already we can see room for further improvement at Posit. The problem was that the developers were not yet ready for it. Perhaps standing in front of this board every day would help them to see what we could already see? Time would tell. Janice, now in the role of Kanban coach, had to be patient.

The idea behind the change in approach to estimation was to move away from unnecessary precision. It was causing a lot of pain, and the accuracy of the estimate was always questionable. A t-shirt size would provide a broad idea of how big each request was, especially for stakeholders who were not familiar with software development work. Being less precise was certainly easier and faster, and it tended to produce consensus. The hope was that it would also be more accurate, enabling them to deliver on their promises. Janice communicated that the only important metric was, “Did we deliver what we promised?” Delivering on promises affects oxytocin levels in the brain. Oxytocin is the brain chemical associated with trust and some other emotions, such as love. By speaking the language of neuroscience with her neuroscientists she hoped to make them understand and to move them to action. This entire department understood that the hypothalamuses of both the trustees and the trusted would produce oxytocin when deliveries were made as promised. Each sprint completed with a matching promise of functionality would enhance the relationship between the stakeholders and the developers.

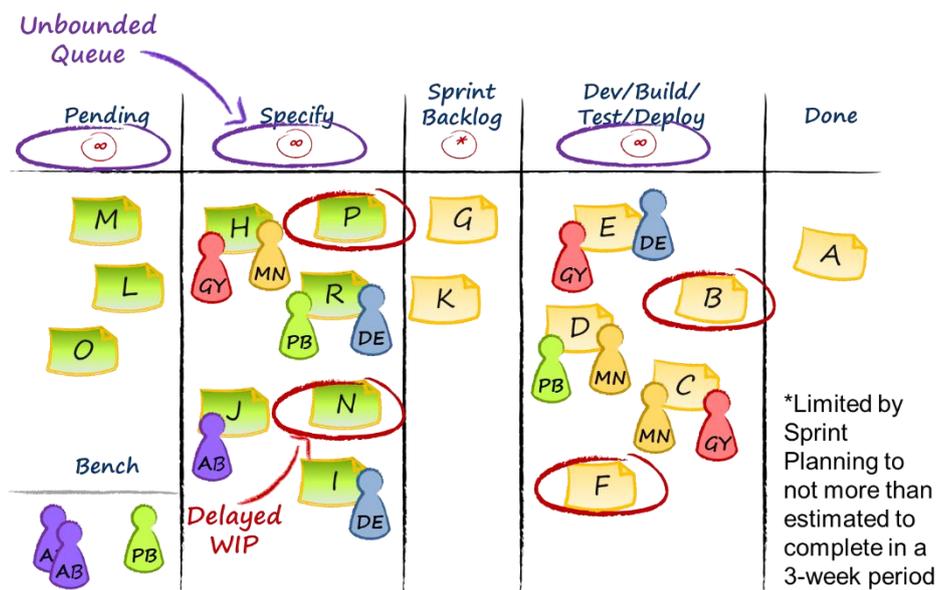


Figure 3 Protecting people from overburdening doesn't protect a workflow from overburdening.

This October 2008 implementation at Posit (Figure 3) isn't a Kanban system. There are no WIP limits assigned to the workflow, and the end-to-end system is not relieved of overburdening. The work in the system can grow in an unbounded fashion, while within the sprint there is no restriction on how much multitasking may happen. Equally, work may be started and then set aside for periods of time. The work in the sprint backlog is bounded only by the planning, and its effectiveness is a function of the accuracy of the estimation process. While there is deferred commitment and pull, it is at the scale of a batch of work that

ought to be completed over three weeks, while a true Kanban system works at the scale of individual requests, one kanban—one ticket—at a time.

This and other variants of partial Kanban implementations have come to be known as proto-kanban, a term coined by the software engineering academic Richard Turner of the Stevens Institute. *Proto* implies an evolutionary predecessor. These implementations are so named because of stories like this one, the Posit Science story. As you will see, this degenerate, partial Kanban system at Posit evolves into a full and proper implementation later. So these degenerate implementations are, in fact, stepping stones, and are an active part of the evolutionary process. Hence, this story isn't only a Scrumban story, it is also a proto-kanban story. It is also one of the earliest recorded examples of using per-person WIP limits with a workflow that involves considerable scale—more than twenty people. At that time, autumn 2008, this approach was associated only with the nascent concept of Personal Kanban, which had not yet been codified or documented.

Be Patient

The business problems described previously hadn't gone away. Under financial pressure, politics became an issue within Posit, and different factions with opinions about how the firm should proceed began to emerge. There was pressure to move into the consumer space; one enabler for that would be a web-based implementation rather than the current products that had to be installed as native applications on a user's computer. Posit was still very much in the business of producing CD-ROMs and shipping physical boxes through a distribution and retail channel.

Both the precision and speed a user attained was important to the games' clinical effectiveness. Data integrity was also vital—losing data, or confusing data from one user to another, would negate the clinical value of playing. These were computer games that were prescribed like drugs: "Play this game fifteen minutes each day and your peripheral vision should improve." They needed to be treated as both medical devices and drugs, the argument went. These were scientists involved in serious play, not entertainment.

Lumosity is now a well-known brand and one that readers may be familiar with. They have invested heavily in consumer marketing, especially television commercials. Lumosity makes brain exercise games, just like Posit does. They were and are peers. During the latter part of the last decade Lumosity was gaining ground while Posit was struggling for market adoption and revenue. The pressure to copy Lumosity's lead was strong. However, if you pause briefly and carefully read Lumosity's advertising or website, you'll realize that they make no medical claims as to the effectiveness of their product. Lumosity does not hold itself to the same clinical, medical-grade standard that Posit does. It's far easier to have a consumer product at an affordable consumer price when you don't hold yourself to the same regulatory regime. Without any doubt, Lumosity employs brain plasticity scientists and their products are based on scientific underpinnings and good intentions, but aspects of non-functional quality on which Posit scientists were unwilling to compromise—such as precise timing of operation and clinical testing prior to launch—may have been areas where Lumosity was able to save cost and accelerate time to market. Posit's identity as scientists and their vision and mission for their company—its purpose—was to make medical-grade products that reversed the effects of aging and repaired damage from trauma such as injuries sustained in automobile accidents or in combat. Lumosity presumably saw themselves differently. Although they were using scientific ideas, they didn't appear to have aspirations of robust clinical results. If their respective products were sold in a pharmacy, Posit's would have been over-the-counter or perhaps "prescription only," while Lumosity's would be off-the-shelf. Lumosity's willingness to compromise and develop a web-based platform enabled them to reach far more people and to mine a wealth of information from those web-based users. Consequently, their finances and valuation were in much better shape.

Things Heat Up and Motivate Further Change

InSight, Posit Science's new suite of games, was set to be just like the earlier Brain Fitness Program, delivered on a CD-ROM and priced similarly. Many people felt that it was expensive for its designated target audience. This issue would gradually become a bigger and bigger concern.

Janice continued to focus on the developers. Implementing a rudimentary proto-kanban system was a small win for her and she saw that things improved a little. The developers were more focused and less anxious about whether they could meet their promises. While they felt relief from overburdening and enjoyed their three-items-per-person WIP limit, the changes did little to relieve the bigger problems of delivering whole projects against an expected schedule. The workflow was unpredictable and there was still too much work, including a portion that was unplanned—a reaction to current circumstances. The team continued to have a hard time coping. Janice realized that the most valuable role she could play was to help the developers realize what was really affecting their performance. If they could see and feel it, perhaps they'd be motivated to implement more changes, little by little. This was why she had liked Kanban in the first place. Its evolutionary nature seemed attuned to the nature of human behavior.

One of the ways she helped was to continually ask them during retrospective meetings how they felt—to let them express their frustrations in a safe environment. She started to change their vocabulary, introducing them to language such as “work-in-progress,” “class of service,” “cost of delay,” and so forth. Armed with a better lexicon to express their troubles, they were able to see new ways to improve.

What continued to be problematic was selecting what to work on and what to leave until later. Asking business owners about priorities didn't help: apparently everything was high priority. When priorities changed because something urgent hadn't yet been selected, the developers simply absorbed the request, taking on more and more work. While individually they were limited to multitasking across just three items, the workflow filled up with committed work. The problem described in relation to Figure 3 was real. After some months, they came to realize that they had to address the overburdening of the whole system if they were to improve their ability to deliver against customer expectations.

One source of demand was generated from tactical decision making by senior management. The pricing model that didn't much concern the development group became their direct problem. More than ever, there was pressure to chase the market. Potential clients and investors were actively pursued. In order to woo customers or investors, Posit leadership came with many requests for one-off demos and feature enhancements. Most of those had to be completed and presented against short deadlines. Unplanned, urgent work pre-empted committed work in-progress. This reactionary, opportunistic business demand could rarely wait the three or more weeks and be planned for a scheduled sprint. Posit needed every bit of business they could get, and no one could say no to these requests. Planning, estimations, and working in strict time boxes became more and more auxiliary and unnecessary in their context. They were breaking the rules of Scrum: urgent and critical items were added to existing sprint commitments. This work was neither planned as part of a sprint nor was it delivered at the end of the sprint; rather, it was expedited to the customer whenever it was needed. Despite the emotional attachment to Scrum that had become so much of their identity over the past three years, there was increasing recognition that its rules were not serving their needs. They were succeeding despite Scrum rather than because of it.

By the beginning of 2009, Posit was ready to embrace further changes. David Hoffman asked Janice if there was more change that could be implemented. Naturally, the answer was yes; so she started to facilitate team collaboration on a better process—one based on, not just inspired by Kanban.

New Insights on Priority, Urgency, and Impact

Posit was now actively working on its third product, DriveSharp, which consisted of three games. Through their research with elderly people, Posit's neuroscientists realized that one of the biggest problems faced by that demographic group is their deteriorating ability to drive, which was mostly due to two factors: failing peripheral vision and the inability to react quickly in more complex driving situations. Driving has always been associated with independence and, for this Baby Boomer generation, owning a car and the freedom to go wherever, whenever is something to which they attach great value. Being unable to drive means the loss of independence and reliance on others—it is a core identity issue. Fiercely independent people give it up sorely. A product that would enable people to continue driving long into their retirement and thus protect their prized independence was bound to do well in the market. Through the three games that were part of DriveSharp, Posit was aiming to extend independence and freedom for the Baby Boomer generation. This time, the channel to market would be in partnership with car insurance companies, with an offer that amounted to “sign up to play this game and receive a discount on your insurance.” Who ever heard of seniors who don't love a discount? This product was a sure-fire winner. Consequently, the development team needed to focus on it immediately and deliver it quickly.

However, they were hindered by the continuing miscommunication between them and their stakeholders. Business owners would give them work items without much thought for how busy they were or what other stakeholders had asked of them. They said yes to everything and missed a majority of their deadlines. They felt miserable. The whole process was ruining relationships between people across the company. There had to be a better way than constantly saying yes and allowing everyone to believe their request was the most important one. The conversation needed to shift to understanding business risks, which would facilitate discussions about urgency and impact, thus enabling a better understanding of when to start new work.

Although the long-term goal was for a more mature organization that could meet customer expectations and business objectives, Janice had to start with small, realistic plans. She focused on issues people were raising at retrospective meetings. She looked through her notes and made a list of sources of dissatisfaction. She would address these one by one.

A recurring source of the team's dissatisfaction was often referred to as “fragmentation.” Constantly changing priorities meant developers were continually pre-empted and pulled in different directions. This prevented them from focusing and completing work with high quality or in a timely manner. There was low satisfaction and little sense of achievement. Janice knew this was a hot button with the team, so she offered to alleviate their complaints by “smoothing the flow” and preventing interruptions and changes of direction.

Janice is looking directly at the emotional pain point—fragmentation—constant interruptions, low sense of achievement, probably lowered self-esteem, frustration, and a lack of pride of workmanship. When she proposed the new kanban system solution, she sells it to the team as a new “flow” system. She doesn't use the word “Kanban” directly. These are Scrum people, so she avoids raising their hackles while offering to alleviate their pain. Good coaching is humane. Good coaching has empathy for the humanity of the individuals involved. Sometimes this is referred to as “the Fight Club school of Kanban” because “the first rule of Fight Club is that you never talk about Fight Club.” If mentioning Kanban risks raising resistance, then don't mention it. Address an emotional pain point directly with your proposal and proceed.

She had to limit the WIP in the whole system, not just for each individual. Her developers were ready to make this change. After several months of watching their board and seeing the cause and effect from limiting WIP, they now understood why it was necessary and the benefits it would bring. With the right WIP limits, Janice could create balance in which both developers and testers were equally busy, yet never overburdened. Work would flow better!

She sat down to discuss all her proposed changes with everyone involved, including business owners. She needed their buy-in and consensus to move forward. The changes felt counterintuitive to many, especially the more experienced ones. She was perhaps fortunate that this entire organization understood brain function, plasticity, and how humans cope with change. They were prepared to follow a path that felt wrong while understanding that it made logical sense: their limbic brains were objecting while their pre-frontal cortexes agreed with the analysis and logic of the proposal.

The next thing to address was the dysfunctional planning and prioritization. She needed to bring some organization and collegial collaboration to the process of selecting, sequencing, and scheduling work.

Contemporaneously to this, I was developing the manuscript for *Kanban* (also known as the blue book) and attempting to codify classes of service, which first emerged in the Corbis implementation in 2007, by identifying the nature of the cost of delay that results in the selection of a class of service. There were four empirically derived classes of service and I'd named them: Expedite, Fixed Date, Standard, and Intangible. Janice had asked me to advise her and help Posit with their transition plans. So Posit had first access to new material and were the first organization to see the association of sketches of delay cost functions mapped to classes of service (Figure 4).

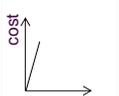
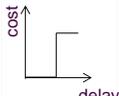
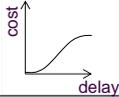
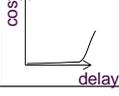
Color	Func	Class of service and its policies
		Expedite – white; critical and immediate cost of delay; can exceed other kanban limit (bumps other work); limit 1
		Fixed date – orange; cost of delay goes up significantly after deadline
		Standard - yellow; increasing urgency, cost of delay is shallow but accelerates before leveling out
		Intangible – blue; cost of delay may be significant but is not incurred until significantly later, if at all

Figure 4 Delay cost function sketches mapped to classes of service

The concept was simple: ask business owners to describe the impact over time for a given function. This would enable a determination of urgency. The discussion would help facilitate scheduling as well as the class of service required after the work was selected and committed. Stakeholders were briefed on the concept and asked to select the delay cost function that best matched the business risks associated with the request. This worked incredibly well. It was perhaps the most easily adopted of any new technique Janice introduced at Posit. It quickly institutionalized and years later was still in use for assessing risk and selecting and scheduling work.

The sketches show the y-axis labeled with the abstract concept of impact. This facilitated various ways of assessing “cost” related to time: opportunity cost of lost revenue, operational expense incurred, subscribers acquired over time, impact on intangibles such as customer satisfaction, brand equity, mindshare, investor confidence, and so forth.

While over the intervening years this set of sketches was extended, it remains the simplest, and possibly most powerful, way of qualitatively tying cost of delay to the class of service used for a ticket in a kanban system. Although quantitative techniques emerged later and are included in Enterprise Services Planning training curricula available from Kanban University, they are much harder to understand, require difficult to acquire input data, and the use of software to run a convolution algorithm in order to establish a “probable cost of delay in starting” function and, from its derivative, a quantitative value for urgency. Consequently,

quantitative assessment of cost of delay remains an intellectual curiosity, while the simple qualitative approach using sketches showing impact over time has proven powerful and easy to adopt. The fact that this technique has prevailed for more than a decade is a strong indication of its effectiveness and robustness. While we may see some refinement over the coming years, I expect this technique to survive and thrive for a long time to come.

In addition to cost of delay, the executive team were also trained in another simple qualitative risk assessment taxonomy that describes the role played in the market by a given feature or function. These roles are:

- Table stakes (commodity features expected by customers; omission is unacceptable)
- Cost reducers (features that save [Posit] cost in development, production, or field service)
- Regulatory (required by a regulator, subject to regulatory changes; omission is unacceptable)
- Spoilers (also known as Catch Up, or Neutralizing features; these copy a competitor's differentiator)
- Differentiator (a new feature, unique to the market)

While this taxonomy made a great deal of sense to the business school-trained executive team at Posit, they rejected it. Their argument was that this taxonomy was clearly for more mature markets with an established set of competitors and well understood customer expectations. Their argument was that Posit was in a nascent and emerging market and, while other brain plasticity firms such as Lumosity existed, they weren't directly competitive. Hence, they would find little to no value in labeling features using this taxonomy. Knowing something was either table stakes or differentiating wouldn't affect their decision making. So I challenged them to come up with something better, something more relevant and attuned to their business and their market. After a short huddle, perhaps fifteen minutes later, they returned with their own taxonomy. It was very simple:

- Existing Market
- New

They recognized that they needed to hedge risk by allocating capacity in their portfolio. They needed to enhance and develop existing products, broadening and deepening their market reach, while they also needed to continue to probe for new markets and market segments by commercializing more of their fundamental scientific research.

I challenged them a little further, and after another short conversation, perhaps only five minutes, there was a consensus that there should be a 60/40 split. Sixty percent of features flowing through their kanban system should be for existing market development, while forty percent should be for commercialization of research and introduction of new products.

So, two very simple methods were introduced to facilitate planning and prioritization: requests would be triaged using cost of delay, and commitment would be deferred until the "last responsible moment" before an item was too urgent, with an appropriate class of service to facilitate flow and delivery; while a capacity allocation would ensure a mix of work intended to manage risk exposure across their product portfolio.

Previously, business owners had anxiety about limiting WIP and deferring commitment. Now, with a better means to assess risk and a new language with which to discuss comparative business risks, they were comfortable with the introduction of a kanban system. Since 2010, methods of qualitative risk assessment have been recognized as essential to facilitate the successful implementation of Kanban.

There was one more obstacle: the developers objected to the naming of the classes of service, specifically the Intangible class of service. It turned out that almost all work likely to be classified as having a deferred cost of delay (and therefore assigned the Intangible class of service) was work proposed by the development organization. This was mostly work on system architecture, code maintenance, and systems infrastructure. They objected to their work being labeled as having "intangible" value and feared it would never be selected.

Responding to their objections Janice engaged in some negotiation: two out of ten slots in the replenishment buffer, to be known as the Top Ten, would be reserved for Intangible class items; additionally, the classes would be renamed.

In 2009, I'd been persuaded by Julian Everett, at the time the super-smart chief architect for the BBC's websites, that cost of delay could be modeled as a linear function. Julian had shown that in discussions with business owners, he could get them to declare a business value for a feature, such as a new set of web pages for the forthcoming season of Dr. Who, and then determine a confidence level to adjust the number. The BBC's website made money from advertising; hence, the number of anticipated ad impressions was the metric for determining business value. If the business owner thought that a new set of pages would generate 1.2 million page views (and hence ad impressions) in a year, but they clearly had only around seventy-five percent confidence in this number, then Julian would adjust 1.2 million to 900,000 and calculate a monthly rate as an average—effectively creating a linear regression for the value. Whether or not the rate of page views was truly flat and aggregated linearly was, in his experience, not important. When making a comparative selection between different opportunities for the same web development team, he found that linear functions were good enough. Given that Julian had real-world experience, my guidance was based on his reports and I initially provided Posit with a sketch of a linear rising line for the Standard class of service.

The Ph.D. neuroscientists were immediately smart enough to push back against this, arguing that typical delay cost functions were “accelerating” and ultimately would tail off as an S-curve. Ironically, my earlier guidance had stated this, but Julian's experience had suggested that linear lines were good enough and a lot simpler—the idea of cost of delay as a constant rate is attractive and alluring. Posit people weren't buying it and, in truth, their gut feeling is correct. Perhaps in Julian's constrained domain of comparative assessment of website features it did make sense, but time has convinced me that it was poor general guidance.

The Standard class of service was renamed Accelerating, and the sketch used was like the S-curve shown in Figure 4. This meant that the word “Standard” was not used. Meanwhile, there was an emotional objection to “Intangible,” and hence the lowest class of service came to be named “Standard” as a unique customization and enabler for Posit Science.

Every change agent—every Kanban coach—should expect some pushback on initial designs. Just like Janice was, they should be prepared to back off in the first instance and implement something shallow, something intended as a proto-kanban, and then be patient—wait for everyone involved to internalize the issues and for motivation to build to enable a full change. Equally, when getting pushback on minor elements such as the naming of a class of service, be prepared to negotiate and make changes. If there is a rock in your way, be prepared to negotiate around it. Texts such as this one are here to advise you and to provide illustrative guidance, they are not prescriptive. With Kanban you have the freedom to tailor and evolve your own unique workflow solutions. Embrace that freedom. Do not feel constrained by the words on these pages.

The Flow System

Janice proceeded with the new flow system. On the day she did that she tweeted, “We just did our last iteration⁶. We have switched to flow.” The relief was tangible. Painful timeboxed sprints were now a thing of the past. Just like the company she had read about, it made no sense to timebox work in their context.

⁶ The Agile software development community often refers to timeboxed team activities as “iterations.” This is a misnomer; as the activity is seldom iterative, in the sense that work will be revisited and improved with greater fidelity, as, say, an oil painting might be by its artist. Instead, Agile software development work is mostly incremental, and each “iteration” contains the completion of a small piece—part of a whole. “Iteration” is used synonymously with Sprint in the Scrum methodology with “iteration” being considered more generic Agile terminology and not specific to Scrum.

Three-week timeboxes weren't helpful to the business owners or to anyone on the service delivery side. Everyone was miserable. Switching to an on-demand flow system met everyone's needs much better.

The Kanban System

The table in Figure 5 summarizes the changes made from the proto-kanban system of 2008 to the full Kanban system introduced in 2009. Sprints (iterations) were replaced with an on-demand flow system with a twenty-one-day SLA. The SLA was chosen to match the previous cadence of three-week sprints. The purpose was to encourage breakdown of work to be small enough to complete within three weeks and to allay fears that work would take longer without the pressure of a sprint boundary or specific delivery promise.

There is a commonly held myth that a lack of time-bound delivery commitment, such as a sprint demonstration, will lead to a lack of focus, laziness, and increasingly long delivery times. There is, in fact, no such evidence from more than a decade of Kanban usage. The fear is one generated by those who sell Scrum training and coaching for a living and wish to dissuade adoption of Kanban or existing customers from switching.

Posit used consultants from a well-known Agile software development vendor as coaches and advisors, who had a long history of actively dissuading the adoption of Kanban at their clients' companies. Ironically, they used a kanban system for their own software development, but as their product was designed for Scrum, they didn't want their clients using it.

There was significant motivation for change at Posit. The consultants had already lost their argument suggesting that it was "a lack of discipline" and the fault of the people at Posit. They had lost their ability to provide leadership and their contract wasn't to be renewed. However, fears they had laid had to be mitigated. Janice did this by including the three-week guarantee of delivery within the service level agreement.

In addition to the twenty-one-day SLA, there was a further change in how estimation was conducted. You will recall that they'd started with the very precise estimation method of speculating how many person hours were needed for each task. This approach was prescribed by Ken Schwaber, one of the two founders of Scrum, in his original book on the subject. At the time, this approach to estimation was the preferred approach of the consulting firm assisting at Posit. The precise estimates actually offered little information value, as they had a very low probability of being accurate. When they introduced the proto-kanban changes, they moved away from precision to a t-shirt sizing approach for user stories. This moves up a level in the hierarchy, as stories usually consist of tasks. Hence, there was less need for analysis and the story level approach was faster. They hoped it would also have greater accuracy and information value.

Now, a year later, they would move away from estimation almost completely, and move up another level of the hierarchy to the feature level. Hence, it was no longer necessary to break features down through analysis into user stories in order to make a commitment and proceed with the work. They would simply request a thumbs-up, thumbs-down vote after the requirements were read out and explained to the team. They took a few minutes to establish a confidence level. If there was strong confidence that the feature could be finished within the SLA, it was marked as ready for selection. If not, its business owner was requested to rethink the requirement and submit the ticket again. These changes are summarized in Figure 5.

	BEFORE	AFTER
Iterations	✓	✗ Flow & SLA
Scrum Master, PO	✓	✓
Sprint planning	✓	✗ Triggered, per feature
Daily Standup Meeting	✓	✓
Product Owner accepts	✓	✓
Demo	✓	✓ Calendar
Retrospective	✓	✓ Calendar
Estimation	✓ By STORY	✓ By FEATURE per SLA
Other		More detailed workflow
Other		Workflow WIP LIMITS

Figure 5 Summary of changes from 2008 proto-kanban to 2009 full Kanban system

The roles of Scrum Master and Product Owner, key elements of Scrum, were untouched. With Kanban, no one gets any new roles, responsibilities, or job titles, at least initially, and certainly not imposed upon them.

Roles and job titles become key elements of an individual's professional identity. Changing a role, a job title, or significantly changing responsibilities tends to meet with resistance—and fear. There is a fear of being, at least initially, incompetent in the new role or with new responsibilities. Such a fear can be allayed through training, mentoring, and a failure-tolerant culture that provides personal safety for experimentation and learning. However, identity runs deeper than just the fear of initial incompetence. Identity provides the means of self-image and for determining self-esteem. Skills, competence, and role played are also key to establishing status in a social group. A new role or job title directly attacks an individual's sense of self and their self-worth. New roles and job titles have both psychological and sociological effect. We might expect seventy to eighty percent of people to have misgivings and trepidation about a new job title or a new role with new responsibilities.

Kanban is the start-with-what-you-do-now method. Kanban also asks you to go around the rock and avoid obstacles to change. If you start by giving someone a new job title, then you are starting by throwing an obstacle in your way. Why do that? Let people keep their existing roles and job titles until they are ready to pull a new identity for themselves.

Each of the three Scrum teams at Posit would work on one feature at a time. A strict one feature per team WIP limit. Replenishment meetings would be triggered on demand when a team needed to pull a new feature to start. As the concept of three-week sprints was dropped, so too the sprint planning every third week was eliminated. These were the dreaded, stressful meetings with seventeen attendees, filled with emotion and anxiety. This brought great relief.

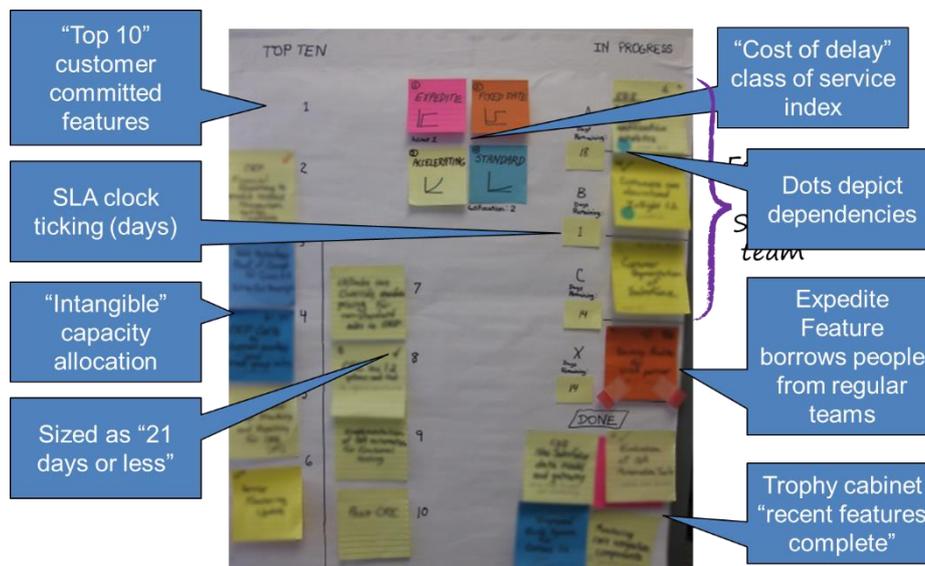


Figure 6 Posit Science replenishment meeting board

Replenishment meetings were facilitated using a small board, as shown in Figure 6. It has four regions:

- The Top Ten
- In-Progress
- Done
- Legend

The Top Ten shows the input queue. In this case, it is a queue numbered 1 through 10. However, work isn't necessarily pulled from the queue in strict prioritized order. Due to the nature of Posit's business, there is a lot of specialization and heterogeneity amongst their workforce and the work submitted. Consequently, their three Scrum teams are not homogenous in skill sets. When a team finishes a feature and is ready to pull another, the first item in the queue may not be a good match for them. They would work their way down the queue until they found the first strong match for their skills. So Posit implemented an almost FIFO (first-in, first-out) input queue rather than the input buffer (sometimes called a "supermarket" in Lean literature) implemented at Microsoft and Corbis.

The Top Ten has a capacity allocation of two slots for Intangible class of service work.

In-Progress shows the features in progress for each of the three Scrum teams, shown only as A, B, and C. They didn't have names. This is an indication that the social cohesion was at the level of the whole department, and people didn't strongly identify with the smaller teams to which they were assigned. The teams were known as Scrum teams despite the Kanban implementation and the decoupling of cadences for planning, lead time, and delivery, together with the removal of timeboxed sprints.

The numbers below the team letter show the number of days elapsed on their twenty-one-day SLA. The picture shows what has triggered the replenishment meeting: the B team has completed a feature and pulled item 1 from the Top Ten. Items 2 through 10 should now shuffle up one place, and the current meeting will select a new item for slot number 10.

The features in progress are also annotated with decorations. The colored circles indicate peer dependencies—items that must be delivered together—same color, same delivery. The small fluorescent tabs indicate blocking issues and flag that the SLA may be in jeopardy.

Done is the trophy cabinet. Done provides a space for features completed in the recent past. Done provides time for reflection and feeling a sense of achievement. Done communicates to business owners, "what we've done for you recently" and the value being delivered regularly.

The Legend shows the classes of service, the colors of the tickets, the delay cost function sketches associated with each, and any capacity allocation or other policies related to classes of service. In this case, Expedite items are limited to one and the Intangibles have a minimum of two.

The board appears to show a fourth team, X. This actually represents the Expedite lane on the board. There wasn't a dedicated team for expedite requests. Instead a feature with Expedite class of service was permitted to break the WIP limit. It didn't, however, completely pre-empt existing work. A cross-functional team from the entire pool of labor would form to complete the request. These individuals would come from any of the other three teams. Assuming that no one team was completely depleted, work would still continue on the current features in progress.

Not every element of Scrum was eliminated. As already mentioned, the Scrum roles of Scrum Master and Product Owner remained, so too did the daily "scrum," though effectively it was a daily Kanban meeting.

Demonstrations, retrospectives, and product owner acceptance would also stay. Demonstrations and retrospectives were scheduled every third week, at exactly the same time and day as they had been with Scrum sprints. No change. Acceptance is fully explained later, in relation to the Kanban board implementation shown in Figure 9. Product owner acceptance continued. However, it became an on-demand activity, with work represented in a column on the board, as shown in Figure 7. The responsibilities of the product owner were unchanged.

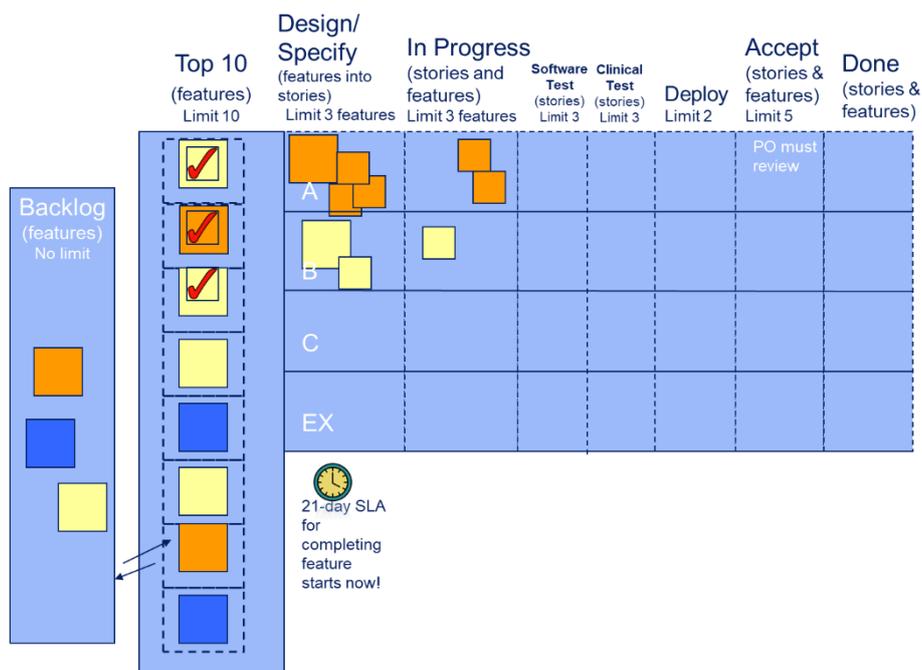


Figure 7 Design for Posit's new "flow system" Kanban board

Figure 7 shows the design for the new Kanban system and board. From a backlog of features, a top ten will be selected for delivery. The Top Ten queue has an interesting innovation. It has a two-phased, asynchronous commitment. In the Microsoft and Corbis examples, commitment is synchronous, in that both the customers and the delivery organization are represented at the meeting and agreement is mutual. The customer says, "I want this done next," and the delivery service says, "In that case, we will do it next for you." The Posit design drops this synchronous commitment. Instead, at replenishment meetings, the business owners get to pick new items for the Top Ten, while the delivery organization doesn't have to commit to any of them. The commitment point occurs when one of the Scrum teams pulls a feature into the Design/Specific column of the board. Only then is there two-sided commitment and the clock starts ticking on the twenty-one-day service level agreement.

An asynchronous commitment has some advantages. It is especially useful when it is difficult to organize meetings involving all of the stakeholders. However, it is typically representative of a lower maturity organization, with lower trust and less collaboration. Some of the “Kanban Magic” is unlikely to happen if you can’t get both the upstream and downstream people together.

The delivery rate at Posit was an average of approximately one feature per week. Consequently, the WIP plus the Top Ten represents around three months of work. Some items may wait ten weeks or more before being selected and pulled onto the Kanban board. Something in the Top Ten is expected to be delivered within the next three months. Hence, the Top Ten can be used as a forward signal for marketing, public relations, or even lead time to prepare delivery. This can happen as soon as an item enters the Top Ten; however, the items are ultimately uncommitted and generate no impact downstream. Consequently, it was permitted for the owner of a Top Ten item to swap it during replenishment meetings for another feature they deemed of greater urgency and importance.

The Top Ten provides a means to forward-signal arrival while avoiding full commitment. It facilitates notice periods and deferred commitment together. The concept of an asynchronous commitment around the input buffer has appeared in other implementations subsequently. Sami Honkonen reported⁷ a version that used a visual calendar to indicate tickets to be started on a specific week up to thirteen weeks in advance.

These asynchronous commitments around the input buffer are predecessors of the dynamic scheduling system now used for Enterprise Services Planning—the approach used for dependency management in large-scale Kanban implementations.

The main Kanban board at Posit features a row for each of the Scrum teams with an additional row for expedite requests. The board is two-tiered. A feature occupies a lane, but a feature is broken down into stories, shown as the smaller tickets on the board. Stories are children of the parent feature. Stories flow across the board. Once completed through Software Test, typically the entire feature would be pulled into clinical validation testing.

There is a danger that features are simply too big to fit within the twenty-one-day SLA. Posit utilized a strategy that I call the “credit card fraud” solution. They would let things that might be too big into the system and hope to catch them quickly, within a day or two. They put a small information request, the thumbs up/down sizing, in front to reduce the likelihood of something too big entering the system.

Notice that some items in the Top Ten have check marks. This indicates that the software developers—the delivery service—believe the feature can be completed within twenty-one days. Naturally, this approach won’t be one hundred percent foolproof, but the percentage that will slip through should be small. Hence, the developers were empowered to flag features that they believed were too big if they discovered this after starting work.

Posit had three options if something proved to be too big after it was started:

1. **Do it anyway.** We want it. We need it. It has some urgency attached to it. We don’t care if it takes more than three weeks.
2. **Trim it down.** Invite the business owners to inspect the analyzed stories and indicate which, if any of them, they deem to be excessive.
3. **Throw it back.** Return it to the backlog and ask the business owners to think again about something simpler.

⁷ Unfortunately, Sami has removed the specific blog post at the time of writing, and there is no reliable alternative reference

Feature Request

Requested by: _____ Date Requested _____
 Feature name _____
 Format: [customer] [action] [purpose]

Description _____

Cost of Delay Classification (required)
 Check the type of Feature per the cost of delay.
 Expedite – critical and immediate cost of delay
 Fixed date – cost of delay goes up significantly after deadline...
 Standard – cost of delay goes up increasingly over time...
 Intangible – cost of delay incurred significantly later

Provide information on one or more of the following (optional)
 Projected Revenue _____
 Opportunity Cost
 • Estimated 6 month **revenue loss** if not implemented _____
 • Estimated 6 month **operating expenses** if not implemented _____
 • Estimated cost of man hours or other resources if not implemented _____
 Qualitative Value (customer experience, quality of service, etc) _____

Suggested stories (optional) _____

New

This portion of the form quickly fell out of use. It is an example of an evolutionary relic

Figure 8 Feature request form at Posit Science, mid 2009

Figure 8 shows the 2009 revision of the feature request form used for business owners to submit new features to the backlog. It introduces a compulsory request for new information regarding cost of delay and required class of service, while relegating the older business case information to merely optional. The business owners are not being asked to drop their existing way of doing things, they are simply asked for some simple, additional information that is useful while not being burdensome to provide. Cost of delay is the gray squirrel—the new species that has entered the ecosystem; the return on investment assessment is the red squirrel—the incumbent species that will be squeezed out by the fitter, stronger alternative.

Within six weeks of introduction, none of the business owners were filling out the optional return on investment section of the form. Cost of delay and the discussions at the approximately weekly replenishment meetings were sufficient to make good quality selection, sequencing, and scheduling decisions. The lower section of the form was truly an evolutionary relic. One year later, when we captured the form from a Microsoft Word document, as you can see, the lower section was still there. Despite not being used for a year, no one had removed it or even discussed removing it.

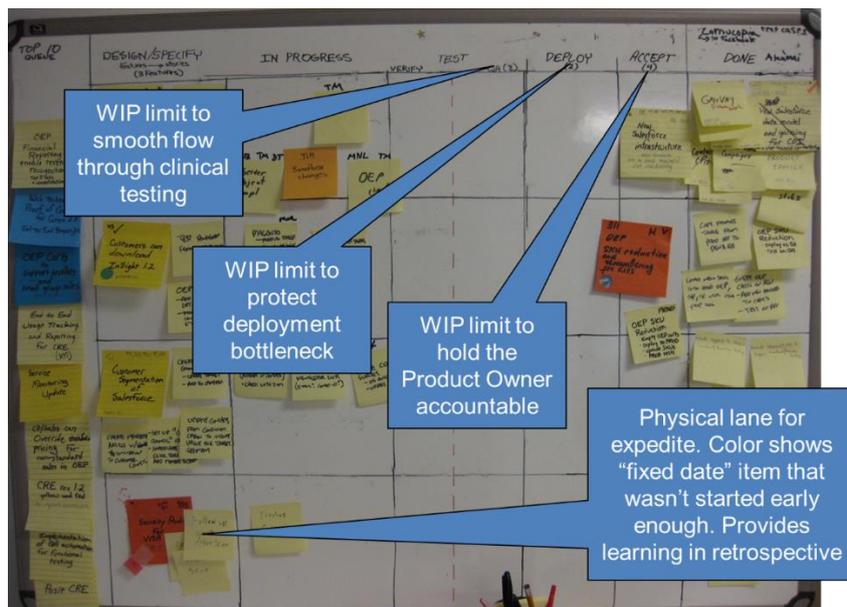


Figure 9 Annotated photograph of Posit Science Kanban board mid 2009

Figure 9 shows a photograph of the actual board taken in 2009. The photograph is annotated to highlight some interesting elements of the design and implementation of the Kanban system and board.

The feature in the Expedite lane is an orange, fixed delivery date class of service ticket. Notably, it wasn't originally an expedite request. As the physical position of the ticket in the expedite lane tells everyone that this feature must be expedited, the color isn't important to communicate that fact for operational purposes. Instead, leaving the original orange ticket and not replacing it with a white⁸ expedite ticket actually communicates that this feature had to be expedited because the team was unable to start it early enough. This is an incredibly powerful educational message and is likely to catalyze improvement discussions. It is just the sort of gentle stressor that the Kanban Method thrives on.

Clinical testing, labeled QA on the board, has a column WIP limit. This tells us that clinical testing is a shared service that serves each of the three Scrum teams and an expedite team if there is one. If the clinical testers were embedded within the Scrum teams in the cross-functional formation prescribed by Scrum, there would be no WIP limit on the column, just as we see for the previous column for software testing.

There is also a column WIP limit on Deployment. The packaging of the code for production was a specialist role and only one person in the company had permission to release code to production—for some younger readers it may be hard to understand the concept of "Gold Code," which is released on a golden rewritable CD-ROM that was in turn passed off to a production facility that manufactured the CDs. The transaction cost of production was high, and it was important that the configuration on the disk be perfect before production. As such, this function at Posit was a potential bottleneck; the WIP limit was there to protect the bottleneck from overburdening.

There is also a column WIP limit on Acceptance. David Hoffman played the role of product owner in the Scrum process. The product owner is supposed to do many things, one of which is accept the delivered product, attending each sprint retrospective and demonstration. However, the global warming problem at Posit—that the money was running out—was consuming executive time. Consequently, David was failing to attend the retrospectives and wasn't accepting the completed work. This was sending a very bad signal to the developers. It was demoralizing. It appeared that leadership didn't care. These people were working hard, often heroic hours, and they put everything they had, often sacrificing their social and family lives, even their health, to keep promises and give the company a chance of success, only to find that the executive leadership didn't appear to care. At the same time, David was aware of this and knew it was poor behavior.

Metaphorically, David was a middle-aged man who comes to the realization that he isn't in the same physical shape he was in while in his twenties. He realizes he is overweight and in poor physical condition, so he buys a gym membership. At first this is great, but gradually his resolve to attend the gym wanes and he finds himself gaining weight again. So to counter this lack of self-discipline, he signs up for a personal trainer. It costs him \$80 per session, two sessions per week. He schedules his gym appointments on his work calendar with a half hour prior and post for the transaction costs of walking to and from the gym and changing, showering, and so forth. He protects that time. He goes for every single session. Why? Because he'd lose \$80 every time he misses!

So, David spoke with Janice and asked her to help him have better discipline to accept finished work. He understood how important it was to send the correct signals to the development team. Janice discussed with him whether or not acceptance was even necessary. Did he need to approve the finished work? This is another use of an important concept in Kanban: acceptance was an explicit policy, and Janice was challenging its validity. However, on reflection, it was agreed that acceptance was still important in a number of ways and that it should stay for governance, risk management, and staff morale reasons.

⁸ Posit actually used bright pink for Expedite, but this was unconventional. When using this case study for training purposes, we've changed it to white, reserving pink for blocking issues. While not historically accurate, this avoids confusion for those new to the method.

The solution was to give Acceptance a WIP limit. Such a simple idea and yet incredibly powerful. There is power in simplicity!

If David failed to show up and accept finished work, then gradually the Kanban board would fill up with work. The WIP limits would prevent the team from pulling new features and gradually the team would become idle. Doomsday is when the entire team is idle and the board is stuffed full of blocked work. We call this a Doomsday Scenario because of the awkward conversation that is likely to happen next and the possibility that the change agent, the Kanban advocate, will find that they are packing their office and looking for new employment. Imposing a WIP limit on an external dependency is a dangerous choice. What if the external party doesn't cooperate and Doomsday arrives? WIP limits on dependencies are not for beginners! However, in this instance, Janice had the mitigating circumstance that David Hoffman, the external dependency, was collaborating. He didn't intend to let anyone down and Doomsday should never happen. The WIP limit and its consequences were there to provide gentle pressure for David to act in a timely manner. What's more, David had to walk past the board from his office on his way to get a cup of coffee. He was going to see it multiple times per day. He wasn't going to be able to forget about pending feature acceptance. Of course, this worked as designed, and a considerable amount of social capital was restored.

Posit delivered new features to their business customers as each was completed, on average one every week. New work would be pulled when an existing feature was delivered, and everything flowed smoothly. It was a true victory for Janice and a major relief from three years of living in chaos and feeling guilt for the inability to make it all better. She always felt that the problems were not with the people—the scientists, developers, and clinical testers—rather the problems were with the system in which they worked; it was not attuned to the nature of the environment in which it lived. She was also immensely relieved that the awful planning meetings every third week had been eliminated. Introducing Kanban had enabled her to achieve a better balance in her life. She was relieved from a lot of the stress and anxiety.

Postscript

In the early summer of 2009, the long-awaited results of the Mayo Clinic study were published.⁹ Encouragingly, the researchers found that the software boosted the brain in ways unrelated to the training. Rather than simply learning to parrot back what they had practiced, participants improved their test scores across a range of brain functions. What Posit had achieved was truly remarkable. Later that summer they released DriveSharp. One of the games within it, Road Choice—later renamed to DoubleDecision¹⁰—was especially beneficial. The accolade for it came at a time when there was a growing debate on whether brain games delivered the benefits their creators claimed.

Janice had great plans for many more improvements, but she ran out of time. Posit had failed to raise their cash flow fast enough, and a major retrenchment was necessary. While the Ph.D. neuroscientist researchers were retained, the entire product development group was let go. Janice moved on to work at another gaming startup in Berkeley, California.

Posit had resisted pursuing the consumer market. Their identity as scientists making medical-grade products ruled their decision making, their strategy, and their investment choices. Meanwhile, their competitor Lumosity had embraced the consumer market and garnered a much bigger marketing budget, essential for selling something few consumers yet understood, and developed a web-based product available on an affordable subscription-basis. They took the market by storm.

⁹ <http://www.brainhq.com/world-class-science/published-research/impact-study>

¹⁰ <http://www.nature.com/news/a-little-brain-training-goes-a-long-way-1.12924>

Lumosity's founders had, like Posit, assumed that much of their market would be aging Baby Boomers and their parents. But when they analyzed their user data, they found the games were appealing much more to twenty- and thirty-somethings, the Millennial generation. Recognizing this early, they began targeting them. This resulted in millions of new users each year.¹¹

Posit showed resilience. The decision to retrench had come early enough that they avoided collapse and remained afloat. They survived with a much-reduced cost base and recovered some years later with the release of their own web-based platform, BrainHQ.¹² Now it offers a broad range of games suitable for a broader audience at much more affordable prices.

While Janice had to leave the brain aerobics scene, she never left the Kanban scene. Kanban changed her life. In 2011, she moved to Seattle and founded the Lean Kanban University licensed training organization and soon afterward took over the Lean Kanban Conference and associated event-planning business. Janice continues to be an important contributor and a driving force in the Kanban movement.

¹¹ <http://www.inc.com/magazine/201312/robin-schatz/from-research-lab-to-market-leader-in-no-time.html>

¹² <http://www.brainhq.com/news/posit-science-launches-brainhq>