

Atlassian

Management processes, such as product planning, are the levers by which large companies are run. To get the picture, imagine a giant Caterpillar loader with a bewildering array of levers. Yank on one lever and the behemoth turns left, pull another lever and a giant shovel swings up, grab another and the whole contraption rotates around its central axis. Clearly, you want to know what you're doing before you start fiddling with the controls. Hence, it would be foolish for a CEO to make an abrupt or dramatic change to one of the management mechanisms that is critical to keeping the company on track, and even more foolish for you to attempt such a feat. No matter how carefully one thinks through the implications of such a change, there will always be unanticipated side effects. When it comes to management innovation, you want to think boldly and act prudently. As important as this is in a big organisation, it's even more vital in a young and vulnerable company.

Make a hasty and ill-advised change to a core management process in a large company, and you may sink your career. Do it in a start-up, and you may sink the company. Too often, people believe that small companies are blissfully unencumbered by the management challenges that confront industry giants. Conventional thinking holds that newly minted start-ups are too young to have to worry much about the long-term, too single-minded to have to grapple with competing priorities, and too nimble to be at risk from bureaucratic inertia. Yet pint-sized organisations are often beset by full-sized management problems. For example, think of all the great rock bands – from the Beatles to the Smashing Pumpkins – that disintegrated because three or four individuals couldn't resolve their differences around artistic direction or the division of credit and compensation.

In addition, small companies are susceptible to a kind of myopia that's uniquely their own: In most start-ups, every ounce of energy gets channeled into scaling up one great idea. After four or five

years of ultra-focused effort, the business takes off. But when that first, brilliant strategy starts to fade, there's no Plan B. All that focus early on often leaves a company with few growth options later on. Again, think of all those one-hit bands that stayed together, but faded into obscurity when they couldn't reinvent themselves. A great product or two can make a company famous and its founders rich, but it can't guarantee enduring success. Small companies struggling to do more with less face a knotty challenge: How to plant the seeds for tomorrow's game-changing products and services, even as they're working from dawn to dusk to harvest the fruits of yesterday's breakthrough thinking.

Mythology and conundrums

Mike Cannon-Brookes, co-CEO of Atlassian, a 30-person, fast-growing software company based in Sydney, Australia, found himself facing just such a conundrum. Taking its name from Atlas, the mythological figure who bore the weight of the earth on his back, Atlassian generates around \$15 million a year in sales. Launched in 2002, the company markets two enterprise-software products: JIRA, a suite of tools that helps project-management teams track complicated workflows, and Confluence, a "wiki" that allows people to create, annotate, and catalogue project-related documents and discussions. JIRA and Confluence, which are licensed to more than 3,200 organisations worldwide, compete against software packages from heavyweights like Microsoft and Silicon Valley startups like Socialtext.

With everyone at Atlassian running flat-out to stay ahead of a rapidly multiplying horde of project-ware competitors, Cannon-Brookes and his co-founder and co-CEO, Scott Farquhar, wondered whether the company's frenetic, dead-line focused work environment might be robbing its engineers of the time they needed to dream up new products or explore out-of-the-box enhancements to JIRA and Confluence.

The 20 per cent rule

Both men have friends who work at Google and both have been intrigued by Google's "20 per cent rule," which allows developers to devote a fifth of their time to any project of their choosing, without first seeking management's approval. At Google, this unmanaged time is the loam in which new service concepts are germinated. Cannon-Brookes hypothesised that what worked for Google's fast-growing engineering corps might also work for Atlassian.

Yet he was mindful of some stark differences between the two companies. Google is a Godzilla-sized company that gobbles up millions of search queries and spits out millions in profits. Google can afford to speculate on the future. Atlassian's twenty-odd software coders, on the other hand, have barely enough time to deliver on their near-term priorities. Adding to this short-term pressure is the fact that Atlassian's financial backers hope to get their money back sooner rather than later—a feat best accomplished, in their view, by avoiding any product development detours or dead ends.

Given this, Cannon-Brookes couldn't see how he could let his developers devote 20 per cent of their time to "out-of-scope" projects, the equivalent of losing the efforts of four full-time coders. Yet he also realised that his engineers, and his company, might benefit from some unscripted innovation. How could he test Google's 20 per cent policy at Atlassian without putting the company's development schedule at risk? Slowly, an idea took shape.

What if Atlassian set aside a single day in which its engineers could work on whatever captured their interest? This would allow Cannon-Brookes to test the 20 per cent notion without having to commit to a big and risky policy change. It might also allow him to answer some crucial questions: If given the chance, would his engineers come forward with worthwhile ideas for new products or features? How

much real development progress could they make in one day? Enough to turn ideas into prototypes? Would the engineers find the experience stimulating or a distraction from more pressing duties? Like many ambitious but impoverished experimenters before him, Cannon-Brookes wanted to test a big idea in a small way.

Delivery room

And so was born Atlassian's inaugural "FedEx Day," named for the overnight courier famous for its promise of next-day delivery. The one-day experiment was designed to test whether a bit of discretionary time for Atlassian's software developers would produce a significant pay-off. As Cannon-Brookes would later write in his blog, FedEx Day was a "heavily bastardised, Atlassian version" of Google's 20 per cent time.

Thus on an April day in 2005, Atlassian's developers wedged themselves into a cramped conference room and started kicking around ideas for dramatically improving JIRA or Confluence. The goal was to get a bunch of ideas out on the table and then spend the rest of the day working to turn the most promising notions into bits of code. With all of their regular tasks put on hold for the day, the developers were free to imagine, argue, and code. There were only two stipulations: First, their efforts had to be "out of the ordinary"; they couldn't work on something that was already in the product development roadmap. And second, by the day's end, they had to deliver a working prototype. The day would conclude with a "show-and-tell" over pizza and beer.

Atlassian's software is highly modular, which makes it easy for engineers to develop extensions, known as "plug-ins," that enhance the core product. Cannon-Brookes hoped that in eight hours, at least a few of his developers would whip up something novel and practical. Having self-organised into two-person teams, the engineers went to work. As the day progressed, the excitement level rose.

Each team seemed intent on producing a mini-breakthrough that would win some peer-group props. By the end of the afternoon, the teams had generated a dozen ‘kickass’ extensions. Ultimately, half of those would ship in subsequent releases of Atlassian’s apps. What’s more, the experiment gave developers the time and space to learn new technologies, suss out new approaches to thorny technical challenges, and jumpstart ideas that hadn’t found their way into the formal product plan. All in all, concluded Cannon-Brookes, the developers did “pretty damn well”.

Evolve and experiment

Having validated the hypotheses behind Google’s 20 per cent rule – that smart folks often harbour ‘off-budget’ ideas that deserve to be explored, that a small reprieve from the daily grind is a tonic for innovative thinking, and that people get most enthused by the tasks they choose for themselves – Cannon-Brookes considered how he might turn the one-day experiment into something more. While Atlassian couldn’t afford to adopt Google’s model outright, it might be possible to make FedEx Day a regular event.

Today, Atlassian holds two to three FedEx Days a year, in which it offers prizes such as dinner vouchers and a gaudy trophy to get staffers’ competitive juices flowing. And it continues to tweak the process. In the second go-round, Cannon-Brookes asked people to post their ideas online in advance of the next FedEx Day, so they could solicit feedback and refine their thinking before they actually started coding. The extra thought and time helped Atlassian’s developers concoct two extensions for Confluence – a dynamic calendar and an ‘auto-save’ feature – that were all but ready for shipment by the day’s end. Says Cannon-Brookes: “FedEx Day spurs innovation by giving people room enough to think.”

Atlassian has begun to integrate FedEx Day into its formal, product-development process. Now, promising enhancements that can’t be completed within the format of the one-day coder fest get rolled into the mainline product development process.

The experiment continues to evolve: the most recent FedEx Day also incorporated a panel of Atlassian’s corporate customers. And in a bid to get customers even more involved, Atlassian recently created a kind of “FedEx Month,” in which developers from the company’s clients were invited to contribute plug-in enhancements to JIRA and Confluence. Over the course of 30 days, outside developers competed for \$30,000 in cash prizes and swag (such as tickets to conferences). In the end, they created a total of 35 plug-ins, all of which were made available for download. Many of the enhancements were “significant pieces of work,” says Cannon-Brookes. Dubbed “Codegeist,” the experiment was a way for Atlassian to capture some of the discretionary time of its customers.

Atlassian is a pre-pubescent start-up. Yet it can teach industry veterans a thing or two about management innovation. The FedEx Day experiment demonstrates that it’s possible to test a bold, potentially disruptive management idea in a low-impact, non-disruptive way. Smart companies don’t do a big product launch without a lot of preliminary experimentation. Typically, this involves building prototypes, collecting feedback from lead users, and garnering data from a regional roll-out before going global. At every stage, the design becomes more refined and the risk of an expensive flop goes down.

So it goes with management innovation. New approaches to compensating people, staffing projects, allocating resources, developing strategies and other management tasks need to be refined through enlightened trial and error. As is so often the case in business, the goal is to maximise the numerator – learning and improvement – while minimising the denominator – time, risk, and cost. Even the most promising management innovations run the risk of being discredited if one starts by making large-scale, systemic changes. So here’s your assignment: Design a one-day experiment to begin testing and refining your bold idea for management innovation.