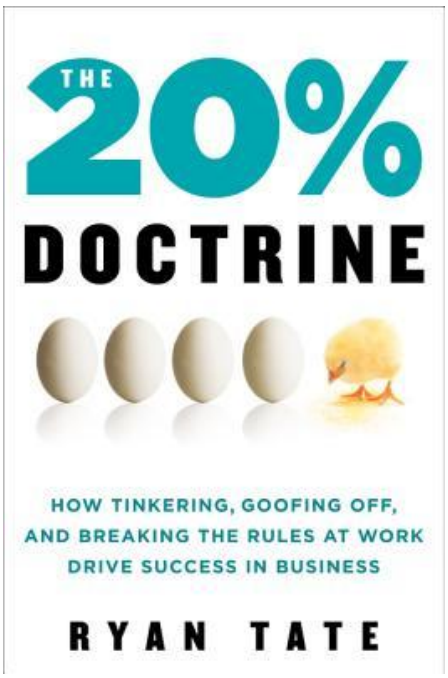

The 20% Doctrine: How Tinkering, Goofing Off, and Breaking the Rules at Work Drive Success in Business

Ryan Tate

The 20% Doctrine: How Tinkering, Goofing Off, and Breaking the Rules at Work Drive Success in Business

Ryan Tate

The 20% Doctrine: How Tinkering, Goofing Off, and Breaking the Rules at Work Drive Success in Business Ryan Tate
Gawker tech-blogger and journalist Ryan Tate reveals how businesses can inspire greater creativity and productivity by allowing their employees to pursue their own passions at work. In *The 20% Doctrine*, Tate examines how companies large and small can incubate valuable innovative advances by making small, specific changes to how work time is approached within their corporate cultures. The concept of “20% Time” originated at Google, but Tate takes examples from all around the business world—from Yahoo! and Condé Nast to the Thomas Keller Restaurant Group, National Public Radio, Flickr and the *Huffington Post*—to demonstrate how flexibility and experimentation can revolutionize any business model, including yours.

The 20% Doctrine: How Tinkering, Goofing Off, and Breaking the Rules at Work Drive Success in Business Details

Date	:	Published April 17th 2012 by HarperBusiness (first published January 18th 2012)
ISBN	:	9780062003232
Author	:	Ryan Tate
Format	:	Hardcover 208 pages
Genre	:	Business, Nonfiction

From Reader Review The 20% Doctrine: How Tinkering, Goofing Off, and Breaking the Rules at Work Drive Success in Business for online ebook

Josh says

interesting read. most of the projects highlighted took place as side projects inside established brands though. of the major projects highlighted, Flickr was the only one that really took place outside of a well established organization such as Google, Yahoo!, NYC Schools or Thomas Keller Restaurant Group (yes, Twitter was mentioned along with its origins - but it was not one of the major, in-depth case studies).

the Huffington Post case study was particularly bothersome to me as an individual who has not had as much success as i would like as a freelance writer - using an overwhelmingly volunteer corps to generate content for a for-profit enterprise. from a business point, i can see how this makes sense, but ethically - using the work of volunteers to generate shareholder value (Huffington is under the AOL brand) is a huge issue for me and a growing number of unemployed individuals who are struggling to find work. yes, it's innovative - but should this be something that we should be applauding as major companies are continuously downsizing? it appears as though it didn't happen, but there was discussion of firing paid individuals and replacing them with this style volunteer-corps throughout the entire organization... i don't think this is kosher. innovation should be rewarded, but labor should still be compensated and paid employees should not be under the threat of losing their jobs to unpaid contributors.

otherwise, i thought the remainder of the cases were intriguing and utilized a number of constraints to their advantage. it highlighted that you do not need massive budgets, superior tech and sprawling org charts to leverage a few good ideas from a handful of creative minds then spread that enthusiasm.

Mary Andrusyk says

There are probably be some good take-aways in this book but not for me. Once I got bored with the case study stories, I lost interest and didn't finish reading the book

---

Deane Barker says

A collection of stories about 20% and "skunkworks" projects and what makes them tick.

The stories were interesting -- the first ones were slanted toward tech, and the later stories branched out -- but I was having trouble drawing any common principles out of it. This was remedied somewhat in the conclusion where the author discussed the commonalities which made the projects all tick.

I found myself getting lost in some of the stories -- they're written with a journalistic bent, and I found myself not really caring about the book topic after a while.

Did I learn anything? Maybe a few things: embrace time constraints to pair your project down to the bare

essentials, make friends in high places when you don't need them so you can call on them when you do, and explicitly ask people to be involved and contribute. These points are all supported by various anecdotes throughout the book.

A solid effort, and I'm glad I read it. But not life-changing.

Valentín Muro says

Not a particularly well written book, it explores with some liberties its main premise but sometimes fails to show how these stories of success are replicable. It is clear that the author goes a great length to show that all of these projects are somehow connected but that doesn't seem to be always the case.

In spite of those remarks, it's a great read for anyone interested in knowing where all the hackathon frenzy comes from and why the entrepreneurial culture is so eager of embracing what they think constitutes the hacker culture in its essence. Paradoxically, what you get from this book is that hackathons have much more to do with companies and capitalism than with the hacker ethic.

Andrew Fallows says

As the kind of guy who loves to explore new things and hold traditions loosely, I love Tate's reflections on dedicating time specifically to these things. Lots of lessons that go beyond Google's original conception of 20% time.

Michael says

I read this book as research for a project at work. It covers some of the principles in common to a few different side projects (Google, Huffington Post, etc) but the case studies, with the exception of Google, are at best loosely tied to the main premise.

Academic Eric says

This supported the concept that innovation has space to develop within time left available for playing around with ideas, new designs and new ways of working. We must be careful not to schedule ourselves too tightly to current processes, projects and points of interest. Future innovations depend on unplanned discoveries.

Mary says

There are valuable lessons in this book, though at times the examples felt like little more than interesting stories and the writing was dry. From 20% time to hack days to volunteerism, there are many ways to explore people's passions and create something awesome. The conclusion does a good job of recapping and

tying together the underlying lessons of the stories to help you with a potential passion project.

I appreciate the examples beyond tech to include education and hospitality, showing that it's applicable across industries. The key point I got from this book is that if you let people scratch that itch and follow their interests and passions, great things can come of it. Ultimately, it's about keeping in touch with our humanity.

Other ideas to remember: having constraints can be good for creativity and innovation. Success stems from those parameters and tapping into the emotional experience of the new idea. Ship, then iterate. Keep standards high.

Scribbler says

Quick and easy read. Nice story-telling style for the examples (e.g. particularly Huffington Post's "Off The Bus" initiative) interwoven with some basic practical advice to chew on. I will definitely be sharing some of these principles with my team as we seek to crank up our innovation.

Martin Van says

About 20% of this book is about 20% time. I found Chapter 3 about hack days to be the most helpful. The book often departs from its thesis and forces in some success stories to fit a "20% time" definition.

Emily Leathers says

Not a bad book, but also nothing terribly exciting. I might come back to it at some point just for the completionism because I didn't dislike it - but on the other hand I have so many other things I'd rather learn.

DeLace Munger says

It was a good read and it was fun to see what goes on in industries other than my own but I found it very difficult to apply any of the suggestions to my career. It's worth it to know that freedom and innovation go hand in hand. I think it'd be a good read for anyone in the software industry or anything internet related.

Todd says

Enjoyed generally but the last two chapters have little to do with an actual 20 percent policy as interesting as the ideas are.

Lucas says

Early Stage - Building Identity

In the early stage of a side project, you're inspired by a new idea, evaluate that idea, refine it, and begin work. Your experiments are frenzied, your potential seems limitless, and your idea is highly vulnerable to dying as a result of fear, boredom, and neglect.

-Scratch your own itch. Thomas Keller (chef) was inspired to build Ad Hoc because he wanted to relive the old days, when, as he put it, "you work five days a week with the same people every day and you take the same days off. Those restaurants are far and few between these days." Jay Rosen launched Off the Bus with Arianna Huffington because he was hungry to improve news coverage after decades of observing how it failed consumers. And Paul Buchheit build Gmail because, as far as he was concerned, email was still broken years after he'd toyed in college with the idea of fixing it.

All these people were building for themselves. Similarly, you should be your own first customer. Make something you really want for yourself. Do it to eliminate something that annoys the hell out of you, or to create something in an area where you feel inspired. The hunger to scratch your own itch is the bedrock on which a passion project is built.

-Be conspicuously different, even defiant. You've got to be visibly different, as both a product and a team. It doesn't hurt to be outright rebellious sometimes. You need a bold message and distinctive identity. It is incredibly hard even for a full-time company like a start-up to break through the wall of noise that is today's marketplace, to get attention, and to lodge in people's minds. It's even harder when you're working in your spare time inside a company with different priorities.

The desire for distinctiveness is why the Huffington Posts Off the Bus shunned news stories that sounded too professional, created a Special Ops team, and launched brands like Grassroots Correspondents and Eyes and Ears that had no equivalent whatsoever in the mainstream media. Yahoo! Hack Day likewise took off because Chad Dickerson pushed the hackathon format into bold, highly visible new territory with the alt-rock concert, oddball hacks, and wafting pot smoke of Open Hack Day, because of his unauthorized press leaks, and because he embraced subversive hacks like "Who's the Boss?" Joan Sullivan put her students in the Bronx in uniforms and took them to museums, cultural monuments, and college campuses that showed them in a very concret way that they were a special cadre of students. And of course Gmail was possible, thanks only to Paul Buchheit's brutally-distinctively!-rough first prototype.

-Connect with people emotionally. You are not just delivering a product, you are creating an experience. Your passion project lights you up inside, and it should do the same for both your customers and your coworkers. You want to create something emotionally resonant, a product that delights people, even the people building it. Even before you know exactly what your product will be, your own high expectations will create an experience for those around you. As Sullivan said, "You do need to believe that [your goal] is possible, because people's intuition...their sense that this is a credible belief...is strong, generally, [on] your team."

If you can learn this lesson and create something that touches people's feelings, you will have at your disposal a powerful lever that can multiply the impact of whatever resources and expertise you invest in your

project. Thomas Keller credits the runaway success of Ad Hoc as both a restaurant and a cookbook not only to the quality of the food but to the fact that the restaurant touched people's "reference points," with family-style dining and dishes like fried chicken, ribs, fish and chips, and roasted sirloin that took people back to their childhoods. Flickr, similarly, took off after its tagging feature turned photo uploading into an emotionally powerful social experience. And of course Off the Bus succeeded where prior journalism efforts had failed because it tried doggedly to create a positive experience for its thousands of contributors and to make human, relatable political coverage for its many readers.

-Build something simple and quick. Make a simple prototype, quickly. If there is one lesson that is ubiquitous among successful 20 percent projects, it's this one. Version 1 of Gmail was build in a few hours and could only read one guy's email. AdSense was prototyped overnight. Version 1 of Flickr was build in just over two months. Hack days succeed by making people build something in twenty-four hours or less.

The rough prototype is also a staple of non-software products. Ad Hoc was built in a four-month sprint for less than a quarter of the cost of Thomas Keller's other restaurants. The proposal for the Bronx Academy of Letters was written in two weeks while Sullivan was teaching full-time. And Off the Bus came together, as Amanda Michel put it, after "Arianna [Huffington] had a conversation with Jay [Rosen], and they made a very fast decision."

-Broke and sweating is your happy place. Constraints are your friends. Embrace tight deadlines and a dearth of resources. Seemingly impossible timelines help you whittle down your idea to its most potent essence and get something out the door, while constrained resources force you to abandon bad ideas and encourage you to seek out novel solutions to problems rather than buying common ones. Flickr was born not just from some spare-time hacking but also from the death throes of a video game company. Ad Hoc came about because Keller, a famous perfectionist, told himself he was going to throw away the restaurant, and so shouldn't spend too much money launching it or take too much time planning it. Google Reader launched because of a one-month ultimatum. Hack days are built on the power of deadlines. And tough conditions in the New York City schools and in the Bronx pushed Sullivan to create a pioneering endowment for her public high school.

Middle Stage - Fighting for Support

In the middle stage of a 20 percent-style project, other people become involved. You are trying to attract customers or users, as well as teammates and allies. You improve the product.

-Iterate fast. You want to improve incrementally and release a steady stream of new versions. The key is to keep each iteration small and manageable. This style of product development has a number of advantages. More improved versions mean more opportunities to get feedback on your product from testers, colleagues, and early customers. This reduces the risk of wasting a lot of time developing in the wrong direction or that you'll be surprised at how people react to your improvements. Keeping your versions small also helps you more quickly determine how much work is involved with each improvement and how long each one will take. In addition, more iterations mean more excuses to talk about your project and more opportunities for other people to spread the word, raising the profile of the project.

Gmail is the poster child for iteration, having incrementally improved over the course of two and a half years from the point where it was described, in Googler Chris Wetherell's words, as "the worst thing ever" to the point where it became the fastest-growing email service on the planet. Flickr, too, evolved via baby steps, gradually transforming from an instant messenger application to a photo website to a full-blown social network. Innovation benefits from iteration outside of tech, too. Sullivan's high school started with just ninth grade and added a grade a year for four years. At that point, it began adding a middle school. "You didn't

have to staff up all grades all at once," former New York City schools chancellor Joel Klein told me. "You could kind of staff up the ninth grade, the tenth and eleventh, and then twelfth, staff up the seventh grade, basically building a culture that was aligned with your thinking."

-Decide what to measure, and set a benchmark goal. To decide how to develop Gmail, Buchheit set a goal of accumulating one hundred happy users inside Google. He later said, basically, that once he got to one hundred users, getting to millions was not much harder. "One hundred doesn't sound like a lot," Buchheit said, "but it turns out that people are pretty similar to each other, so if you can make one hundred people happy, usually you can make more." At Ad Hoc, Keller set a goal at the outset of doing about sixty "covers," or customers, per night. At Flickr, Caterina Fake's main statistic was the bank balance of Ludicorp, which was perpetually in danger of going under. But she had another benchmark that she watched, total user sign-ups per day; she knew Flickr was in danger when it briefly dropped below ten.

It's great that you are passionate enough about something to launch a side project in your spare time. But you cannot be guided by emotions alone. You need an objective yardstick with which you will measure your project's progress, and you should set a goal that can be measured with this yardstick. Try not to get lost in or obsessed with statistics, but do have some numbers to keep you grounded.

-Be needy. The best way to enlist people as allies is to show them concrete things they can personally do to improve your project. Wetherell showed the source code for a prototype version of Google Reader to a smarter programmer, knowing the programmer would wince at the low quality of Wetherell's code and be tempted to rewrite it himself. It worked. Also at Google, Buchheit's AdSense prototype was crude, just as his Gmail prototype had been. Google assigned a whole new team to AdSense, which completely remade the product and launched it in time to fund Gmail. "An extremely talented team was formed to build the project, and within maybe six months a live beta was launched," Buchheit later wrote. At Off the Bus, meanwhile, Michel successfully recruited citizen journalists by showing them how their professional expertise, their geographic location, or their political background made them especially valuable to the Huffington Post's political coverage.

-Be confident. WHile you want to make clear to people how you need help, at the same time you also need to show zealous confidence in your future success. This might sound obvious, but it's important. The people you need to help your project, and the people you need to buy or use your project, will be able to sense your level of enthusiasm and dedication. In this sense, being a great builder of side projects is like being a great entrepreneur. You need a deep-seated conviction that you are doing the right thing. One of the reasons Wetherell had the chutzpah to bypass the usual Google channels and convince an engineer with root access to deploy Google Reader on a server is that he was truly confident the product could take off. Buchheit was so confident in AdSense that he was willing to defy an order from Gmail's product manager, Marissa Mayer. "I don't remember Marissa ever liking [AdSense]," Buchheit told me, "but I wouldn't have taken them down even if she did ask."

-Go outside. A great side project walks the line between the host company's core competencies and outside technologies and ways of thinking that the company hasn't embraced. It brings a piece of the outside wold inside the corporate walls. As such, it is wise to reach out to the outside world repeatedly for feedback and inspiration. This could include a software, music, or art Hack Day; it could also include a loosely structured forum like BarCamp, which has been used as the basis for creative workshops in tech, politics, health care, city planning, and other fields.

You can get useful perspective on your project at any event or meeting that puts you in touch with people beyond your usual coworkers. Keller and his team didn't realize Ad Hoc could be a viable permanent

restaurant until Yountville, California, locals started approaching them on the street begging them not to shut it down. The demoralized employees behind Flickr got a much-needed boost by demonstrating a very early version of their photo-sharing software at O'Reilly's Emerging Technology Conference. The Bronx Academy of Letters brought professional authors to its fund-raising board and into its classrooms to upgrade its students' educational experiences. Likewise, you can learn a lot by embracing outsiders.

-Keep going, even if you have to turn. Creating a successful product or organization is often a long and tough road, even if you've already got an entire company built up around you. It took Buchheit two and a half years to launch Gmail. Off the Bus had nearly a year and a half to run before the election, and there were lessons all along the way for its leaders. Even relatively short periods of time can feel long when you're putting in intense hours, as Flickr's founders did with their fourteen to eighteen hour days.

But it's not enough to persist. You also have to be on the lookout for indications that you should change course. It took Michel a few months to admit to herself that she needed a journalist like Marc Cooper working by her side. It took financial desperation for Fake and Butterfield to realize that they should try to "pivot" their video game company around a new photo-sharing idea. It's possible that your own most promising future might require you to turn on a dime. So be ready to do just that.

Last Stage - Growth and Resolution

When your project reaches its later stages, you push up against the challenges of life inside a company built around some other idea. You need to grow to survive. You need resources to grow. You need smarts to make the most of your resources. And eventually you need to resolve your idea's future within the company if all of that energy and bother are to last.

-Find patrons and allied projects. When your project is in the advanced stages of development, it begins to bump up against problems that require not just the human capital of time and energy but the old-fashioned kind of capital. Gmail, for example, needed more hard drives and servers; Hack Day wanted to invite hundreds of geeks to camp overnight on its corporate campus; the Huffington Post needed to develop a kind of Web page that didn't exist in its publishing system. This is when you need friends in high places. I'm telling you this now, so you can cultivate these patrons before your project desperately needs their help. As you put together your side project, seek advice, counsel, and encouragement from trusted mentors and superiors within the organization, people you may be able to turn to later.

You may also be able to get help from another project, particularly an officially favored project, by convincing the higher-ups that your project is crucial to the other's success. Recall how Wetherell was able to obtain some measure of official sanction when his politically weak Google Reader project was deemed to be a "dependency," or required subcomponent, of the "iGoogle" customizable home page project. This help from the iGoogle team made Google Reader possible.

-Highlight investments that will benefit everyone. One of the best ways to get your company to bet on your project is to convince your company it is not betting on your product. There are invariably techniques and technologies developed for a side project that end up having benefits for the entire organization, even if the side project is folded up and forgotten about. So show your bosses features, improvements, and technologies that have strong potential to be spun out to benefit the entire organization. They'll be much more inclined to provide resources to that sort of product development than to development tightly coupled to your personal vision.

The Huffington Post's tech staff strongly resisted Michel and Cooper's insistent requests for a "big news

page" where they could showcase their Iowa caucus coverage, a page that could be laid out like the HuffPos front page or like its section pages but that was actually neither. After HuffPo tech relented, the big news page turned out to be a much bigger deal, being used hundreds of times by HuffPo proper versus just a handful by Off the Bus.

-Bend over backward to make do with existing systems. You are more likely to get resources if you're clearly pushing to the limit all the resources you've already got. Off the Bus never did get most of the tech it wanted, not even a simple Web form where writers could type in their articles directly. Instead, they ran almost everything off email and a giant Excel spreadsheet. Buchheit begged spare hard drives and servers off other teams at Google. Flickr made do with tags rather than splurging on fancy image-recognition software. Even Keller, owner of two of the fanciest restaurants in the world, bought second-hand and pulled things from storage when it came time to furnish Ad Hoc.

-Communicate constantly with the mother ship. Side projects tend to be disruptive in nature, so giving your supervisors frequent updates, particularly in your project's later stages, will help protect it from controversy. It is likewise a good idea to give colleagues in other groups frequent communiques about anything you are doing that might impact them. Keeping up a constant flow of communication will help not only to smooth over bruised egos but to build political support for your project, laying the essential groundwork for when you want to graduate a 20 percent project into something bigger.
