BY SWIZEC TELLER

Why programmers work at night

A book about programmers, by a programmer.

Swizec Teller

This book is for sale at http://leanpub.com/nightowls

This version was published on 2013-10-21

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Tweet This Book!

Please help Swizec Teller by spreading the word about this book on Twitter!

The suggested hashtag for this book is #nightowls.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#nightowls

Contents

From disgruntled developer, to founder, to burnout

I was 21 when I quit my first real job.

What started as a great opportunity for a high school senior to sharpen his coding chops, turned into daily frustration over management, stress and terrible working conditions.

The small advertising agency was a creative's dream. Offices showered in natural light. Large windows overlooking a lush green park, playful decor, walls covered in old design projects, sofas in the lounge, shelves full of books on design and marketing. A relaxed culture.

It was amazing. But also terrible.

Nobody had any real idea how programmers function. How we think, how we work, what we need to feel productive. For a long time I was the only programmer on the team.

On day one I was given an old laptop, a comfy chair and a spot on the large desk occupying most of an open-floor office. Across from me was a chatty copywriter. On the right was the main designer. In the opposite corner sat another designer.

The boss who talked on the phone a lot was hiding behind a tall book shelf. When somebody went to ask him about something, the whole office could hear. Eventually a project manager joined our big desk … she wasn't even half as quiet as the boss. I still know that printers often need to be talked to angrily to get a move on.

At first I was happy. The work was fun, it was new, and I even got paid to do what I loved. Jackpot!

But as months passed the job became more and more stressful. The projects became a slog, always the same. Waves upon waves of client work made it completely impossible to work on that one cool internal project. By the time I was finishing up coding for the previous project, a new one was already specified and all conversations with the clients done.

I could never get a word in on the design part. They thought I didn't care, but I just didn't have the time to attend meetings.

At one point there were so many things on my plate I couldn't even get more than a half hour of focused

work until everyone went home already.

Word of Google being a super awesome place to work for reached my radar at about the same time. A company by engineers for engineers. A place you can *work* at.

I dreamt of founding the perfect company to work for. I didn't know what that meant, but I knew it was going to be cool and I would love working there.

After a few months of working on a side-project late into the night I quit my job and started working from home. Finally I could work only on what I wanted, there was nobody to boss me around and I could work distraction-free as much as I wanted!

Nope.

Mum didn't get "working from home" at all.

"You're home arsing about all morning. What do you mean you couldn't wash the dishes and vacuum the floor and cook lunch for everyone and your room is still a mess?"

"But muuuuum, I was *working*. No I don't have time to sit down with you for an hour when you come home every day. I'm *busy*!"

"Are you even making any money? Did you pass any exams yet this month?"

"Sheesh mum it's not all about money …"

"Then you're not working!"

Not that this has been much different now that I *do* make plenty of money by working from home. Mums just don't understand. I guess nobody who doesn't work from home understands …

But eventually the side-project started taking off. Two guys agreed to work with me for free. I couldn't really pay them anyway. We convinced a local hackerspace to give us an office - also for free - and we set off to build the next big thing in news consumption. We were going to take over the world!

We didn't.

The startup never left my mind. At any given moment I was either architecting a cool piece of tech, pouring my misguided ideas into code, creating business strategies, thinking about what the others were doing, or learning to be a better entrepreneur.

It got so bad I was listening to entrepreneurship podcasts whenever I had more than 10 minutes of "free" time, just so I could squeeze that extra bit of productivity into the day. My standard was 14 to 15 hours.

Everyone spent at least eleven hours a day in that dark hackerspace basement. We occupied a sofa, two tables,

and three whiteboards.

There was no time for deep thought. We wanted to work work work. One by one we crashed and burned.

The other programmer was the first to go. He simply stopped coming to the office. He didn't answer any of my calls or emails. For a few weeks even his friends were asking me if I'd seen him.

I don't know what happened, but he was a cool guy and I still feel bad for doing that to him. Last we met he was working for an established startup and finishing his studies. He didn't seem to hold a grudge.

He was the first to give up on us, but I was the first to start burning out. I started taking long car rides to nothing. Just sit in the car and drive aimlessly around the city listening to bad music on the radio and looking at the stars.

I needed to be alone. At the office everyone expected something from me. Everyone expected me to be awesome, to inspire everyone to work hard, to woo investors, to keep my spirits up, to be perfect.

They actually didn't, but it sure felt like it. In my eyes I *was* the startup. Without me there was nothing.

The car didn't judge.

I became a husk. I spent whole days aimlessly clicking around the internet. "Work" had become writing two

lines of code followed by an hour of lolcats. My nights were spent doing everything I could possibly think of that didn't involve the startup.

In March 2011 our investors and the other cofounder kicked me out of the startup I had poured my life in for a year and five months. I was devastated, but I deserved it. I was a terrible boss, I drove my guys like slaves, I didn't listen to just about anyone.

On my way out an advisor slash investor said: *"Dude, I know this hurts. But you need to learn a lesson."*

"Screw you, no I don't!"

But he was right. What I was doing was completely unreasonable, practically psychopathic. That is no way to treat programmers. The job I tried so hard to escape was nowhere half as bad.

Ever since I have been actively searching for *the* best way to be a programmer. *The* best way to treat programmers. *The* best environment to keep programmers happy, productive, and sane.

This book is about what I've learned.

Introduction - version 0.4

Hello, I am Book.

You already know the guy who created me from that foreword, you might think he's silly, but he's alright.

I am Book! You will like me.

It's going to take about an hour, fifty-three minutes and nine seconds to read me through, but take it easy. Or fast. Either way, I suggest a nice cup of tea and a comfy chair.

I am here to tell you I love programmers. Even though they sometimes prefer to be called developers, or software engineers, or computer scientists. Whatever *your* favorite version likes to call themselves, go give them a hug. They're awesome folk!

They might seem a bit strange at first, but I'm going to help you be *the best* person for programmers to be around. If you happen to be a programmer yourself, I'm going to tell you how to be the best programmer there is.

Keep in touch

Swizec is starting a newsletter about the things from Book. He's going to share interesting things about productivity, programmers, office culture and stuff like that about twice a month.

Sign up at: http://swiz.ec/nightowls-list[1]

You can also poke Swizec on Twitter as @swizec[2] or send him an email at swizec@swizec.com. He loves hearing from people.

Whom Book is for

You should read me if you are a programmer or have to deal with them a lot. If you've ever waited until 11am for one of them to show up at work, or couldn't get them in bed until 3am no matter how sexy you dressed. This book is for you!

If you're a mum or a dad and have a pup programmer in the house, don't worry. I'll tell you all about why they stay up all night and sleep through school. It's because they're doing awesome things in their spare time ;)

[1] http://swiz.ec/nightowls-list
[2] https://twitter.com/swizec

If one of your parents is a programmer and they keep making corny jokes like *"The two hard things in computer science are cache invalidation, naming things, and off-by-one errors. Ha ha."*. I'll help you understand that too.

What Book contains

I have roughly four sections.

First I'm going to talk about whether programmers do in fact work at night or do they just like to think they do. For some reason they really like bragging about how little sleep they got last night. Swizec is just like that!

Then I'm going to tell you *why* they feel night-time is best time. Some things in the day-time just don't gel well with those guys, understanding will help you keep them at a minimum.

Lastly, there's going to be practical stuff. Both for people around programmers as for programmers themselves. The very last section might be called *"Lifestyle tips for top productivity"*.

Everything you read here has been tried by somebody. Either Swizec or some of his close friends. None of that writing things that sound good on paper, but nobody can do. Only things that work are allowed.

What is not in Book

I am not hard science. Take everything with a grain of salt, if something seems strange, tell Swizec. If something doesn't work for you, tell Swizec. If something sounds downright dangerous, tell Swizec.

You should consider me a collection of anecdotes, personal experiences and interpretations of scientific literature on the topic.

How finished is Book

Swizec seems to have a life outside this book so it's taking him a while to finish. You might still find some notes to himself, usually inside square brackets, or sections that clearly haven't seen an editor yet. He's very sorry about that and promises to get to it as soon as possible.

If you are reading via the Github repository[3], that's wonderful! Make sure to tell Swizec what's wrong! Github Issues exist for a reason.

If you're reading via purchased pdf, that's *awesome*! You are a gentleman and a scholar and a great person

[3] https://github.com/Swizec/nightowls

to boot. Swizec asked me to say thanks for supporting the project.

Right now I'm version 0.4.

The most interesting sections are **About flow** and **Working with programmers**. There's some interesting science about creativity and sleeping habits in **Why programmers work at night**.

The section of tips *for* programmers was mostly written during long drives on a road trip around Europe. It's all very interesting and is probably the most well researched part of the book. But Swizec still needs to mark up the references and clean up the writing.

If you're into data, you should check out the preliminary analysis in **Do programmers work at night?**. Some 500,000 Github repositories were considered, but there's still much more data hiding in there.

License

Book cover courtesy of @ponywithhiccups[4]

[5]

[4] https://twitter.com/ponywithhiccups

[5] http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_GB

[6] http://swizec.com

[7] http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_GB

Do programmers work at night?

Before I can go on, we should first agree whether programmers do in fact work at night or is this just some myth perpetuated by male bravado and our romantic notions of a motivated person doing magic all night. Perhaps we just value exceptionally hard work and working all night is a way of proving you are exceptionally motivated and awesome.

To research this book I talked to some ten programmers and read hundreds of comments on the original essay, Twitter and various Reddit threads about the topic. Essentially, yes, programmers do work at night. They are the loud ones, the romantic young lads and lasses who feel night is the only place to work.

But a large portion of the programmer population is also completely baffled by this. Claiming that those working at night simply cannot schedule their lives, have no familial obligations or don't understand how spectacular starting your day in the morning can be.

A large part of this divide, I think, is simply that

it depends. The core reasons for *why* we work at night, or very early in the morning, or lucked out on a conductive day-time environment appear the same and mostly have to do with deep thought, flow and focus on one's work.

Another venue worth exploring is whether our perceptions might not match reality at all. For instance, I was certain that 1am was when I reached peak productivity and could get the most done. But looking at timestamps of my Github commits on various projects shows that I am much less a night coder than I am a late afternoon and evening kind of guy.

Then again, most adults working day jobs would find it incredibly odd that I should get the majority of my work done from 10pm to midnight. My habit of "doing email" at midnight also confuses many.

It depends

Generally speaking, programmers do seem to work at night - just take a moment to look around the internet. Plenty of advice on stimulants, romanticizing late nights and infinite bravado about how little sleep everyone needs to function.

Talking to individual programmers reveals a different picture. Many will admit to working mostly dur-

ing the day, at worst late in the afternoon. Infinite amounts of people admitting they hate working at night and mostly do it when they have to chase a deadline or two.

The one dividing factor between the two groups seems to be age. Not so much because age has anything to do with it (although it does dictate our sleep-wake cycle to an extent), but because age is a good indicator of what stage in life people are at. Their lifestyle if you will.

Somebody whose kid needs to show up at kindergarten every morning at 7am can't well afford to sleep until noon. Just as a student is forced to code in the evening because school comes first.

The main lifestyle factors appear to be:

- freelancer or staffer?
- student of some sort?
- have pet projects?
- spouse and/or children?

Types of programmers

You can divide programmers into roughly two groups - freelancers, founders, indies who set their own schedules, and those whose schedule is dictated by the organization they're in.

Those running a larger business fall into a grey area of sorts, because even though their schedule is their own to pick, they must still conform to the organization lest they hinder their employees' work. A common pattern here is apparently to work on management during the day and code during the night.

Different work patterns show up in each group.

Staffers are more likely to work during the day, mostly brought on by the needs of collaboration with others. Even working from home, they have to be available for Skype calls and must be relatively prompt in answering email and comments on various issue trackers. Despite living in the 21st century, physical presence at a common place of work is often still required.

After all, it's very difficult to collaborate when somebody is stuck on a bug at 11am and their colleague only answers email at 3am twice in a blue moon.

On the other hand, freelancers mostly dictate their own schedule. For many this is the reason they became freelancers in the first place.

It seems more likely for freelancers to work evenings and nights. Often this isn't out of a preference for when they feel most productive, but out of guilt over how much they got done during the day. Perhaps there's just an over-representation of workaholics in

this population, but talk to any freelancer for long enough and they will likely complain about how little they get done.

They get just as much done during the day as their employed counterparts, but they personally *feel* all the time wasted on overhead. Mostly because they can't (or won't) charge for it.

Because much of their income is directly correlated with how much work they can get done, freelancers tend to push themselves into situations where the deadline or budget is just a smidge too tight. As soon as something goes wrong - and it always does - they are forced to chase deadlines, which usually involves a lot of late nights.

Age of programmers

Sleep researchers have discovered a link between circadian rhythms - our body's internal clock - and age. In a BBC documentary The Secret Life of Your Body Clock[8] they performed an experiment on teenagers that showed their brain activity is much better in the afternoon than it is in the morning. Yet society forces most of their learning to happen in the morning.

[8] http://www.youtube.com/watch?v=uL97Ms6JUfQ

As it turns out, everybody starts life as a morning lark. Any number of parental anecdotes can prove that people habitually get woken up at 6am by their toddlers.

But as we age, our cycle starts shifting more and more towards the night, hitting its nocturnal peak around the age of 21. It is therefore not surprising at all that teenagers and college students are known for "being lazy bums who sleep in all day".

At 21 our cycle starts shifting back into its "natural" state. By the time we reach our 65th birthday we once more find the same affinity to early mornings as toddlers. Our preferred wake up time is once more around 6am.

When was the last time your grandfather didn't wake up with the sun?

Lifestyle of programmers

Even more important than circadian rhythms, age is a good indicator of the relative lifestyles of programmers.

Few people have a family of their own at 22, while many or even most do when they are 40. This has a big effect on when people can code because it means

juggling different priorities and things that get in the way of solving hard problems.

A lot of younger programmers work predominantly at night because they simply can't work during the day - there's classes to attend, schoolwork to finish, if they're very young simply doing the chores their parents give them is more important than any coding they might have going on. It's more of a hobby after all.

So most of the coding happens at night - lovingly supported by their young person circadian rhythms.

Older programmers usually have a day job and are generally expected to put their programming on a higher priority. Since programming is suddenly the most important thing in their lives (usually after family and similar considerations), they devote more time to it.

Not only does a lot of folk wisdom talk about getting work done during the day, days being longer than nights gives a great statistical advantage to work happening during the day.

After all, it's called a "day job" for a reason.

Pet projects

Which brings us to pet projects.

Programming is the sort of job you can only do if you really love it. It really is kind of addicting and most people I know can't let it go even outside their work hours.

Yes, there are many programmers who love programming as a day job and want to have other hobbies they can do in their off time and don't even want to look at code.

But even those have a pet project now and then. That little thing they do when nobody is looking, a tiny bunch of code that's done just for them. For the love of their work.

For play, if you will.

Play is very important in creative professions. Feynman once said[9] he's become a bit disgusted with physics, too much was expected of him. So he started playing, completely without consequence. This led to work on the spin of electrons for which he won a Nobel prize.

Having no obligations brings better work, solves more interesting challenges and refreshes the brain.

Not everybody is a famous researcher at a university, or a googler with 20% time, so they have to play when

[9] http://pythonwise.blogspot.com/2011/09/on-importance-of-playing.html

nobody is looking. The only time nobody expects anything of them. At night. During weekends.

Fun projects happen *around* work work.

Pet projects are also something to show off, a cool hack that makes everyone's ears perk up. As respect is gained in the community, so fuel is added to the myth of programmers who convert caffeine into code.

What about other creatives?

The title of this book is *Do programmers work at night?*. Your natural question might be "Do *only* programmers work at night?"

I don't know the answer to that question, but it seems like a lot of them do. Writers come to mind, as do designers and artists. Michelangelo was known for painting in 40 hour stretches until he fainted of exhaustion.

If you are one of those people, you might well find yourself in this book as well, and I wouldn't mind hearing from you.

But I can't write a book about everyone, my experience is in the programming world. Programmers are the people I have an inside look at, I don't know the first thing about artists and while I might know a

thing or two about writers, I personally prefer mornings for writing and nights for coding.

All I know about other creatives is that many of them complained they are not explicitly included in this book.

Statistics

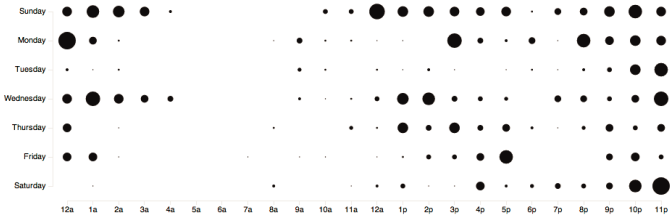
[this needs to be expanded and pulled into a full blown section]

I created a small script that crawled Github search pages for about a week and collected a list of just over 500,000 Github repositories. Another small script then spent roughly fifteen hours going through this list and fetching each repository's punchcard.

As you might imagine, a punchcard tells us what time of day commits were made to a specific repository.

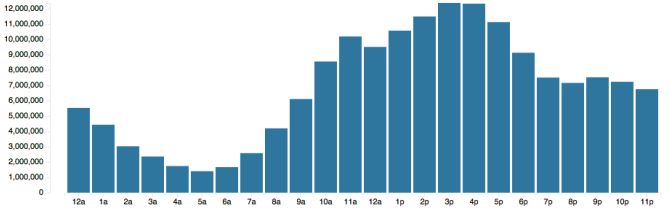
For instance, here's a punchcard for all the repositories that I own; you can get yours at http://nightowls.swizec.com[10].

[10] http://nightowls.swizec.com



Swizec's punchcard

To find out when programmers *actually* work most, I collected all of this data into simple histogram - all 164,509,270 commits of it, or roughly 6 gigabytes. The results are interesting, although perhaps not surprising.



Commit histogram by hour

As you can see, most commits happen between 3pm and 5pm, just when a normal workday is concluding and people have to make sure all of their work is committed so they can go home for the day. This would imply that most programming still happens during normal working hours and that a lot of people use Github for work projects.

More interestingly, even though the volume of commits drops off dramatically - almost exponentially - after 3pm spike, it does not reach the very low volumes of early mornings. In fact, it remains at about 60% of peak volume for the remainder of the evening!

This is interesting on two fronts.

Firstly, it suggests that a lot of programming activity happens in the late afternoon and early night - for an industry that only works during working hours, commit volume at 9pm should be around the same as 5am. Practically zero that is.

Secondly, it's interesting that the volume remains so level until midnight and then start dropping dramatically until reaching the 5am low point, before it again starts rising in a bell curve towards the peak.

Unfortunately, my data does not allow me to tell which projects are things people work on in their spare time at home and which are purely work projects. A lot of interesting things could be gleaned from the data if I could figure that out. Right now I would say that our histogram hides two bell curves.

One bell curve has a peak at 3pm and the other at about 9pm, combining them together would produce the data we are looking at, I think. This warrants further study.

Keep in mind, even though peak commit volume is at the end of a normal working day, more than 5,000,000 commits in my dataset were made between midnight and 1am. That's a lot of programmers producing a lot of work.