



By Saeed Gatson
April 6, 2016

How Taking More Breaks Will Boost Your Productivity

Software Developers live in an interesting paradox. On one hand we're lazy. That's why we like to automate boring repetitive tasks like running unit tests or deploying new environments. At the same time Developers are often celebrated for logging numerous work hours. The stereotypical caffeine-infused all-nighters are all too common.

Working without breaks is a bad habit that can jeopardize business, health and the life you're working toward. 10, 12, or 14+ hour days are only sustainable until the reality of mental and physical limits strike.

It's clear that we should **protect our productive time**, but **what's not clear is that we should protect our break time.**

It's very easy for Software Developers to become overworked. It makes sense from a management perspective. If you have a highly productive coder assigning them, more work will make the overall project more likely to succeed. The problem is that this is just short-term thinking, which normally leads to burnout.

There are two types of breaks you can take. I like to think of them as macro breaks and micro breaks. Macro breaks are the long periods of rest that normally happen over a couple of days. Think the weekend or a couple weeks vacation. Micro breaks are much shorter. Like a water cooler chat, or a short lunch break. In this post, we'll cover why Software Developers should take more micro breaks.



Why you should take breaks?

Few people realize this, but you tend to get more done when you step back and recharge the brain and body. Studies show that performance increases after breaks of all durations: from extended vacations down to microbreaks of 30 seconds.

John Trougakos, Associate Professor of Management at the University of Toronto, says, job-related stress can kill your productivity. **He suggests that, “disconnecting from work can do wonders for people’s energy and mindset.”** To prove his point, he references a study wherein 59 percent of people surveyed said more breaks would improve their work happiness.

John goes on to say that mental concentration is similar to a muscle. It becomes fatigued after sustained use and needs a rest period before it can recover. Just like a bodybuilder needs to rest between set at the gym.

Why you think you don't have time to take breaks?

If you don't think you have enough time to take very short breaks you're probably spending a lot of time doing things that aren't productive. It's important to take a step back and ask "Am I doing work that really matters?" **A lot of Software Developers, myself included, end up doing unnecessary busy work.** For most there is a lot of time that is wasted without even knowing it. Finding the time to take breaks, which will help you in the long run all comes down to managing priorities. For more details and thoughts about this subject check out this post on [Why You Don't Have Enough Time \(And How to Have More\)](#).

How to make sure you're taking the time to recharge. The Pomodoro Technique

One of the easiest ways to make sure that you take time to recharge is to add breaks into your working system. This can be done by using the Pomodoro Technique.

The Pomodoro Technique is a simple yet effective way to organize how you work. The main idea is that you set a timer for a set amount of time. During that time you're supposed to focus on completing one goal. After that timer goes off you then set another timer which will represent a short break. That's it. One time period for working followed by another timer for breaks.

6 high level tips for Pomodoro mastery

There's a lot of information out about the Pomodoro technique, and you can really deep dive into this subject. To see one of example of, check out John Sonmez book **Soft Skills: The software developer's life manual**. He has a full chapter where he covers the Pomodoro Technique. Here are a few high level tips that I feel will give you the most bang for your buck while using the Pomodoro Technique.

1. Monitor how many Pomodoro's (blocks of work time) it normally takes to complete a task.
2. Learn to protect your work time from external interruptions.
3. Make estimates on how many Pomodoros a task will take.
4. Take a few minutes during Pomodoros for recap and review of the current task. This could be in the first few minutes.
5. Organize your working time around how many Pomodoros you'll need to finish your goal.
6. Every few Pomodoros make one of them personal by taking some time for yourself.

Everyone likes to work differently so, in the end, you'll want to keep testing until you find something that works best for you, but keeping these six things in mind will definitely help.

Working with different time splits

The common variables in the Pomodoro technique that is often changed is the time periods set for working and breaks. These can change throughout the day or by more regimented. It's another thing that's up to personal preference. However there are a few common time splits.

- **25 on, 5 off.** The most common type of split is the 25 minutes of work followed by 5 minute breaks. This is done four times in a row and then the break is extended to 15 or 20 minutes. This is classic Pomodoro time split.
- **50 on, 10 off.** An extended Pomodoro Technique is the 50 minutes on 10 minutes off strategy. Very similar to the other techniques but it good for Software Development because there can be a long ramp up time before you start writing lines of code. After investing those few minutes you'll want to make the most of it and avoid context switching.
- **52 on, 17 off.** One of the more unconventional yet successful strategies calls for 52 minutes of intense work followed by 17 minutes of being completely removed from work. The key here is to work 100 percent during the 52 minutes and break 100 percent during the 17 minutes. By fully focusing on work or relaxation, employees are able to make the most of their time. According to this report, the top 10 percent of productive workers take this route.
- **20+ on, 10 off.** Don't break the flow strategy. This is something that has personally worked for me. I have trouble getting started but having a timer helps me to get going. For example, I can just say I only



have to do this difficult task for 20 minutes then I can stop. However if I'm on a roll and the timer goes out I'll just keep working until I hit a wall or drop out of flow. So personally small times help because I use them to build momentum. I've found that two thirds of the time I'll just keep working.

Not every time split will work for everyone all the time. Try experimenting with different periods until you find something that works best for you, because in the end that's what matters most.

How to track on and off periods

I find that having an app is the best way to track breaks and stops. As software developers we are already at our computers most of the day so using a web application is pretty accessible. I'm currently using an app called [Moosti](#), but there are hundreds of other options that are easy to find. You could also just use an egg timer, stopwatch app on your phone or whatever. **Don't over complicate things. Just pick something and go.**

How to handle the unexpected when working

Not everything we do as Software Developers will fit nicely into Pomodoro-like time splits. Things will come up like doing software updates, long build times, or system outages. A common question that comes up is what should you do during the unexpected. John Sonmez answers a few of these questions in a video where he goes into [some questions about the Pomodoro Technique](#). The main takeaway is that you have to be pragmatic.

When things outside of your control happen you have to come up with a new plan that makes the most sense for that situation. You don't have to blindly follow the rules. Come up with something that works for you. There's no wrong answer here.

“You can only set work for you that you’re actually going to be able to accomplish so don’t try and control the world, you can only control you.” – John Sonmez

What to do during your break

Another common question that comes up is what should I do during my breaks. It might seem silly but taking breaks can be hard. Especially for Software Developers with looming deadlines. So coming up with things to do during your breaks (other than keep working) can be tough. Here a few ideas to help get your gears turning.

- **Listen to music.** Playing or listening to music is a great way to boost creativity. It can also be a productivity enhancement and touches everyone in a different way.
- **Do some journaling.** By journaling you can expose your thoughts and unwind. You don’t have to share what you write with anyone. Just the activity alone can up your creativity.
- **Do some doodling.** Much like writing, drawing can be therapeutic and can help you take your mind off work for a few minutes. Another good thing is that it can be done at your desk without most people noticing.
- **Try guided meditation.** This is trending idea, but meditation can help improve your productivity by relaxing your mind. Meditation can help lower your blood pressure and reduce stress, making it a great thing to do during a break.
- **Play a quick game or solve a puzzle.** Just because it’s a break doesn’t mean you can’t think. Putting critical thinking problems and brainteasers in the break room is a great way to encourage workers to think in new ways.
- **Move away from your desk.** You could try walking, reading a book in another room or taking the all-important lunch break, just to name a few options.



Bringing it all together

The idea that you need to take breaks seems counter intuitive. Hopefully, this article has explained not only how taking breaks throughout the day can help, but also gave you strategies on how you can work them into your day.

Remember to guard your break time just as much as your work, because doing so can help your productivity skyrocket.

Select a goal, choose a time split, set your timer, and go. We can talk again during your break.

Are you currently taking strategic breaks during your day or planning to give it a try in the future? Let me know in comment section below.

ABOUT THE AUTHOR



Saeed Gatson

Saeed currently works as a Software Developer In Test remotely from Eugene, Oregon. You can find him thinking out loud at SaeedGatson.com or on Twitter [@SaeedGatson](https://twitter.com/SaeedGatson).

Related Posts



REDUCE CLOUD COSTS: TIPS FROM DEVOPS

How to Reduce Cloud Costs: Tips from DevOps Engineers

AUG 01, 2022 / BY HARDIK SHAH



CREATE A DATING APP LIKE

How To Create a Dating App Like Tinder

JUL 29, 2022 / BY PRATEEK SHARMA



CERTIFICATIONS TO GET YOUR PATH INTO CLOUD

Best Cloud Certifications to Get Your Path into Cloud Development

JUL 27, 2022 / BY VALENTINA SYNENKA



5 Ways Programmers Can Increase Their Overall Productivity

JUL 22, 2022 / BY CHARLIE FLETCHER



How To Evolve SaaS Product Offerings Based on Consumer Needs

JUL 18, 2022 / BY TERESA HOFFMAN



UX Trends That Will Shape 2022

JUL 15, 2022 / BY MATTHEW COOPER



Top 5 PHP Frameworks for Web Development

JUL 20, 2022 / BY SHAHEER ZEB KHAN





QUESTIONS

Top 15 Angular Interview Questions and Answers

JUL 13, 2022 / BY VIKASH KUMAR

[←Previous post](#)[Next post→](#)[report this ad](#)