The M/Movement_

## THIS IS FOR THE DEVELOPERS

A reason to ***commit***. A reason to ***write***. A reason to ***speak***.

Developers are humans. We have preferences, tendencies, likes, dislikes. Much like anyone who works for a company in any industry, we would like to believe that the entity for whom we perform hours of work, day in, day out, believes in us much the same.

But what about when we don't feel that's the case? You see ***resentment***. You see ***burnout***. You see ***attrition***. That's not fun, and both hurts and hinders the company and employee all in one fell swoop.

Fixing this is indescribably complicated, and efforts to do so should be given the deserved respect. That said, there are a handful key principles that are designed to see developers start to feel that their contributions are worthwhile, and that their opinions hold water.

## SO WHAT?

These principles were conjured and designed by developers who feel that the aforementioned work environment isn't unchangeable or gone to the dogs; that things need to, and can, change in a company with such an atmosphere.

In other words, your ***reluctant job*** should be (or become) your ***passionate career***.

So, without further ado, here are the ***M/Movement Principles***.

# 1. HAVE FUN_

Need it be said? As children, we grow up wanting to have fun without being told to do so. Contrary to popular belief, our need to do this does not ***disappear*** as we grow older and wiser, even though the ways in which we do it might morph and transform in more refined and perhaps expensive ways.

## MYTH

> ***I'm not paid to have fun, so I've got no business wasting the company's time sheet by playing around.***

From a LinkedIn post written in 2013, here's a direct counter-argument:

> ***[...] can you tell your CEO that you spent the morning tinkering around with an idea? If the answer is yes, you are in good shape. If no, start looking for another job.***

That was the better part of eight years ago, and one would dare suggest that individuals still struggle with this in ways that they may not ***monitor***. Furthermore, some are actively criticised for this by their less open-minded peers, which makes for an often unbearably ***toxic*** work environment.

## SO WHAT?

Think about what you ***enjoy***. It doesn't necessarily have to directly do with your job, because you may well disregard legitimate elements that could be a part of your job if you ***confine*** yourself to the current situation.

> ***I enjoy customising my terminal theme, wallpaper, and shell environment to my preferences, which is all a part of developing on macOS. I also enjoy Python development, which my work does very little of.***

# 2. STRUT YOUR STUFF_

In a company full of highly qualified software engineers/developers and other similar positions, you're amidst a vast and shining pool of bright minds who are primarily paid to come up with ***creative solutions***.

To recap, these people are ***intelligent***; Their minds are idea machines. To suggest that they can't (or don't) have innovative ideas in the software realm because they're devoted to their job is akin to saying that accountants don't know how to budget personally because they're only good at doing so for their company. Balderdash.

## MYTH

> ***I only know how to do work that the company needs, because I never work with anything else.***

That's simply untrue for the majority of people. Granted, the amount of time you spend on something does generally dictate your proficiency in that area, but it has little to do with the speed or style at which you learn; more importantly, the ***value*** you place on learning is ***independent***. If you want to learn about a new technology, be wary of thoughts along the lines of "I'm so far behind the times that it's too late for me to catch up". That's ***self-doubt***, not skill.

## SO WHAT?

Be ***honest*** with yourself and ask if you have anything you'd be excited about sharing, because people want to learn more and have conversations. Oh, and be ***excited***! Sure, developers can often be the least social of the tea party, but that doesn't mean to say that we forego all human qualities.

If you have something you'd like to share, the first thought you should have is not "It might not be good enough", or "That's already been done." As a counter to the latter, if you demonstrate something that's already been done, your demo provides a learning opportunity for you, so ***someone always profits***.

# 3. STAND UP FOR YOUR WORTH_

## MYTH

> ***Your effectiveness, passion, energy, and efforts should be bound by the limits of other people's work philosophy and approach.***

You're told that you can't do something because of security policy. You have to jump through endless hoops because of red tape. You feel that there's no point in innovation or creativity because you'll be shot on site.

If that sounds at all familiar, this philosophy rejects the notion that it is right or morally acceptable for companies to apply blanket policies to your work which fixate a ***systematic*** tendency of ***failure, defeat, and insignificance*** upon their employees. That's hard to swallow, but not impossible to fix.

If you are stopped from doing something in the tech world because the company isn't comfortable with the risk that it exposes them to, the individuals involved in that decision should then ***work with you*** to find out what it was you were trying to achieve and help you do that. They should not (capital ***NOT***) send you away capitulated and scolded to ne'er return to these lands. Your idea was most likely a ***valuable contribution***, and that didn't change.

## SO WHAT?

If you have a project, initiative, or direction that you truly believe in, remember that this concept came from the mind of someone who is paid for their education, skills, and ability to be flexible and creative. This hypothetical idea that you have which you would like to see become a reality is, in a significant and meaningful part, ***what they pay you for***.

Help them see that. ***Follow it through***.

# 4. BE THE CHANGE YOU WANT TO SEE_

Simply put, if you feel something is lacking in your place of work, ***do not idly*** sit around waiting for an otherwise absent initiative to materialise out of thin air. You are the single most effective proponent of your own idea, which is really the converse of saying that there is no such thing as a mind-reader. If you'd like to see something happen in your place of work, ***light the fire*** to your idea tank.

## MYTH

> ***I need to wait for management to come up with an initiative before I can start coming up with proposals or submitting any proof of concept.***

Sure, if your company had filled its job postings for mind-readers this quarter, you might be able to rely on the above philosophy, but if not, you simply do not know what other people are thinking. An approach to a common goal through your execution may ***tackle a challenge*** that had not been previously considered by the other party; by contrast, an approach to a unique goal will almost certainly tackle such a challenge, making the reward all the more worthwhile.

## SO WHAT?

Be ***brave***, and truly consider what it is you'd like to implement or suggest within your workplace that would benefit the company and therefore its staff.

As a developer, you occupy a unique position that wields both vast perspectives and great power, and there is little stopping you from seeing that come to life aside from ***perceived*** social dynamics.

# 5. "COMPLIANCE" IS NOT "EXISTENCE"_

If your company is in an industry that is prone to a close watch by auditors and enforcers of policy, this philosophy warns of a dreary tale, whereby the fierce knight (okay, developer) is skirmished by wanderers unknown.

Simply put, if you are a developer whose job description does not explicitly state a dedication to interpreting and living their life by compliance, carefully consider what such ***offenders*** you deal with that are ***wasteful*** in the timeline and skill set that often defines your career. Oh, and remember that the timeline is your ***life***, since you most likely spend most of your waking hours at your job.

## MYTH

> *We're a {insert industry} company, so of course we deal with compliance. It's just part of the job.*

The fact that the above statement comes under the "myth" heading is slightly misleading. Take careful note of the word ***part***. Yes, compliance work may be a part of the job as per implicit expectations, but how often will a ***quantifiable*** cap be placed on that? You can be certain that a company will keep a careful watch over how many hours they pay their contractors to get work done, but how often will they ensure that you, as a developer, don't get ***overrun*** by compliance, remembering that it doesn't form the ***core*** of your job description?

## SO WHAT?

You need to be your own best friend here and keep this work in check, ensuring it doesn't grow cobwebs throughout your career. Of course, if you have no problem with an unlimited amount of compliance work, or if your company doesn't have a strong focus on compliance, then this page doesn't apply to you. If it does, carefully consider your ***career goals***. Does this compliance work add to your skill set? Sharpen your skills? Give you creative and technical prowess? Contribute to the greater good of the company and the developer community? There's a strong likelihood that my answers agree with yours here.

# 6. SHOW APPRECIATION AS EMPATHY_

There's very little doubt that if you, as a developer, worked hard on something that you knew would benefit other parties, you'd like to know in some manner or form that your efforts were for something and ***appreciated***; often, such work comes in the form of an ***improved user experience*** or less frustration.

While the fruits of such efforts likely haven't changed over the years, the social model around these changes and therefore the way that customers communicate their resulting reactions have. Often we have move to Slack or Microsoft Teams, and the majority of work is done ***remotely***. As a result, being thanked for work in an area that does in fact benefit others is often either ***overlooked*** completely, or graced with token appreciation, which is in turn overlooked.

## MYTH

> ***I shouldn't expect thanks for the work that I do because it's just my job. I'm sure people appreciate it, but they just don't have to say it.***

The above works if everyone spontaneously and unanimously decides that compliments, gratitude, and appreciation are redundant human traits, and communication should be reduced to ***grunts and nods***. One may feel the smallest of hunches that this won't happen for the next millennium. Referring to the title page of this document, ***developers are humans***. We exhibit human traits, which means we feel good when we receive ***praise*** or positive recognition for our efforts.

## SO WHAT?

If you, as a developer, enjoy receiving such praise from others for your work, then you can easily deduce that the same applies to others in the same way. Psychology teaches us that ***oxytocin*** and ***dopamine*** are released by consistent and repetitive acts of ***kindness***, so why not do that by showing thanks to those that have either made your lives, or your co-workers' lives easier?

# 7. ONE DOTH NOT LIVE IN A BUBBLE_

Have you ever had an email ***ignored*** by a colleague, only to re-send the email again about 12 times (albeit kindly worded) and still hear crickets? Perhaps a more pressing question is this: Have you ever done that ***to someone else***? If so, then you already know what's coming.

## MYTH

> ***I'm sure they'll find the answer somewhere else. They've stopped emailing me, so I'm sure they forgot about it. Someone else will handle it.***

If you ever see the movie Pay It Forward (spoiler alert), you'll learn that it features a young boy who does good things for other people; the movie is written and directed in such a way that the viewer is to be taken aback and ***inspired*** by how seemingly ***small acts of kindness*** and attention, where it is usually neglected, can have ***unimaginably positive effects*** on others.

While this is becoming a comparison between a box office feature film and a company that produces software, the parallels really are no different. If you reply quickly to someone's email and help them as much as possible, you oil the gears of their day, and therefore their job. To put it another way, if you can tell me that you've either never had one of those days where everything "***just works out***", or such days unmistakably have absolutely no correlation with people who put effort into the small things because they want to ***see others succeed***, you may stop reading.

## SO WHAT?

Be that person! When you reply to your 51st email of the day, stop thinking about how you would kill to be in the Bahamas sipping at a ludicrously expensive cocktail, or how your co-workers always get on your nerves. ***Visualise*** the impact on the person you're helping by responding to that email in good time and with above-and-beyond intentions. ***It always helps.***