

Software Diaries - 2



Irum Godil

Copyright © 2024
Independently Published
ISBN: 9798343906974
Printed in USA

Conclusion

Constructive feedback can be hard both for the receiver and the provider. However, it is important that if we are giving feedback to someone, we try to give them as much context and data and give it to them in a sincere faith of helping them.

Alternatively, when we receive feedback and are not happy about it, it is best to step back and identify how much of it is personal versus meant for sincere growth.

Once we understand the perspective of the person who has given the feedback, it will help us to build an unbiased narrative around it.

Also, certainly if receiving that feedback appears detrimental to us in the moment it is very very important to detach ourselves from it, and step away for even a little while and revisit it with a clearer mind.

Preface

October 2024

Recently, over the last couple of months, I have been asked in at least two different contexts, by two different people, as to what have I been writing these days?

Someone told me that I should write my next book on *Copilots*.

My response has been that I have been on a break, as my books really do not go anywhere, and I do not have the bandwidth to invest in marketing.

However, here I am starting to write the Preface of my next book, 'Software Diaries 2.'

So, what has changed?

I recently started working on a different project, and as I have been learning and ramping up, it has hit me that writing is my outlet, to let my thoughts out.

It actually helps me to be productive and creative.

But most importantly, it helps me when I convey my thoughts to the world, as I feel the urge to share my learnings, with others.

Hence, I have written this book in a blog- format, with mini-pieces of learnings and advice, as I have continued to grow in my own learnings.

then it automatically shifts our reactions.

- At one point, when I was given a some feedback, it took me two weeks to just sit with it. Taking a vacation and getting away certainly helped, but after I returned, I had a candid conversation with the person and my other co-workers. I tried to understand the intent behind those points, and also wanted to learn from my colleagues if they had experienced something similar.

Once I cleared the air, it actually helped to learn that the person was only relaying information in *good-faith*, so that if I wanted to grow, I would be cognizant about it.

In retrospect some of those points were actually true and they helped me see how I could change.

Constructive Feedback

Constructive feedback can be one of the hardest things to absorb in this industry, but here are some tips on taking things in your stride.

At many times in my career, receiving constructive feedback has completely torn me apart for weeks and months at times. However, as I learnt to deal with what the person had actually meant and how it helped me in the grand scheme of things, it has been worthy of documenting my learnings.

- Recognizing feedback as 'constructive' is in and of itself a win.

When someone tells us something that we do not like to hear about ourselves, it is only natural to be sensitive and defensive.

However, when we start framing those points as 'points of growth,' rather than accusations,

Acknowledgements

- A special thanks to Apo Song, my colleague, for encouraging me to write my next book and acknowledging that my views and writing mattered. He was the one who asked me to write my next book on Copilots, maybe that would be the next one!
- Also, a very special thanks to my nephew, Asif Moosani, for calling out that my views mattered and at the minimum, the world would get to read my thoughts, as he had enjoyed reading my other books.
- When I had written my first book 'The Path', one of my colleagues, Shubhra Tiwari, had exclaimed that it was great that I had done something outside of work and followed by passion.

My response to her was that it was kind of hard for me to be so stuck up on the computer and type so much outside of work as well. At that point she had suggested to use different modalities for voice to text conversion. As part of working through this book, I have tried to use voice-text in multiple places. I really wanted to give her my special thanks for calling that out.

- Last but of course never the least, all goodness in all my work is only and only attributable to the help and support of The-God and that He has blessed me with the best of parents and family that has been on my side through my peaks and valleys.


- Any other milestone, like a promotion, bonus, job raise, should be **double-rewarded** and thanked. We are living in an economy and market, where the cost of living is on the rise every day. However, sometimes we lose perspective that how blessed we are compared to still the rest of the world. Instead of constantly looking down on what rewards we did not receive, it is imperative to acknowledge what we did get.

Again, this mindset will really help us be our best selves, instead of looking down on things that others got and we did not.

continue to move to more challenging steps
with a positive mindset.

- If someone has been helpful to us, then in addition to ensuring that they are acknowledged, we need to really count it as a blessing.

In this day and age, where human interaction is becoming very rare, it is hard to find 'empathy.' I always try my best to acknowledge the efforts of all my colleagues and folks who have gone above and beyond in helping me. Here's a snapshot of sending a 'Thanks' message to one of my co-workers.



Awsome

Thanks to [redacted] for always being there and
making a difference in my career.

From Irum Godil

Productivity Strategies

Morning Dopamine

The key to kicking off morning dopamine is to **get some dopamine going**.

I am not a morning person and hence starting anything in the morning whether it is exercise, getting ready for work or preparing breakfast, everything is very hard for me in the morning. However, what I have learnt along the way is that one strategy to get over this disconnect is *to just keep going*.

- The key is to accomplish one, two or three things, without judgement or questions.
- It is just to act as an automaton and keep executing, without thinking.

The eventual result is that once we do one, two or three things, the dopamine gets released, as a result of getting something accomplished.

Once this dopamine starts kicking in, it continues to kick in and self-feeds itself to help us execute more complex tasks.

Even if our environment may not make us feel that those are worthy accomplishments, practicing gratitude for those things will help us maintain an optimistic mindset. It will help us to continue moving through the woods and uplift our spirits. That will eventually be a ‘game-changer’ for our own mental selves and others, in this cut-throat culture.

Here are some things for which I try to continuously express gratitude.

- If the smallest part of a system works, it is imperative that we say thanks and pat ourselves on the back.

Distributed, large-scale systems, with multiple authentications, and security layers, can be very challenging to navigate. By exercising gratitude for the smallest things, we would

Be Grateful

We are living in such a competitive world that every accomplishment and a step forward is something to be grateful for.

Most of us keep hearing about exercising gratitude for our personal happiness. However, practicing gratitude is not just a tip for our personal lives, but will also pave the path for our professional success. We are conditioned into believing that only substantial successes are worthy of acknowledgement. As a result, when we are stuck, or when things are not always straightforward, we are always stuck on 'how hard or bad things are.' Similarly, comparing ourselves with others, is another drainer, because mostly if we start looking around, we feel that things are always working for others. However, practicing gratitude is a mindset that needs to be cultivated even for the smallest of the accomplishments.

Try-To

This strategy is about telling yourself that, 'I will **try to do** something, rather than do something.'

For the most part when we want to do something we are scared that we will fail. The idea of wanting to succeed at all costs, sometimes, just stops us from doing anything or prevents us from even taking the first step.

The narrative that I have tried to use, is to tell myself that, 'I will try-to succeed' or 'I will try-to do,' instead of saying that 'I will do.'

When we commit to wanting to get somewhere, all sorts of complexes and thoughts sometimes take over to demotivate us. For instance, if I wanted to understand a code-base, and I committed to getting it all done, then any blockers in the way, create a sense of failure, which in effect creates more de-motivation.

However, if instead of committing to getting something done, we commit to 'trying-to' get

something done, then any progress in the path is progress. Similarly, any road-blocks are not permanent failures, as we are still on track to 'trying-to' get there.

Personally, I have found this strategy very effective, in my pursuit of 'trying to be successful or trying to grow,' instead of absolute statements such of wanting to be successful or getting-there.

Being the Orchestrator

The Power of Hello

Making Progress

This strategy is about how to make progress when **you are not able to see any results**. Or how to keep going, when despite all the hard work, nothing seems to come through.

Some tips that help in the process of making progress are as follows:

- **Focus on learning:**

The goal really is that instead of pushing oneself to get results, one should reframe the situation to, 'What can be learnt from this technology or situation?'

This would involve putting the impact and actual deliverable as the side-result, and focus on building the mental muscles. The value-add of doing this would be that when one would come back to the same or a similar problem, next time, they would be better equipped to deal with it.

- **Reframe the Problem:** Try to re-define the problem, that you are stuck on, in some way.
For instance, I once spent a few hours looking at a code-piece trying to figure out, how would it render as React-Markdown. All along, my search was around how the current code was working.
But later it worked, once I figured that the input was supposed to be bullet points directly and not html tags.
Reframing my search to how to render html text into react markdown, vs trying to readjust the existing code to adhere to what I wanted it to do, did the trick in my case.
Sometimes, when we are blocked, it is because we are looking at the problem in a different way, than it really is. Asking the right question, about what are we really trying to solve, is sometimes the key to move forward.

Looks do matter.

I am personally sometimes really conscious of wrinkled tops, or a bad hair day, even when I wear a cap.

But why?

I think the Software Industry is as much human as it is IT-based.

The human instincts are still about looking normal and decent. And yet, if one is super-smart technically, they could get by with the same pair of tops and jeans for many many days.

But to say that, looks do not matter, would be incorrect.

As part of our regular meetings, conversations, presentations and especially when one moves up the ladder(s), how one comes across certainly has a value-add.

The way I see it is that the money and time that we spend in looking good, is actually an investment.

- **Exercise Gratitude:** Although, I have a full chapter on this below, but when you do see light at the end of the tunnel, if you do, then really practice gratitude no matter how small the win is. Celebrating small wins and progresses are as crucial, as solving the end-game. As mentioned earlier, when we get something going, it actually releases dopamine to get more going.

Break Things Down

Do Appearances Matter In Tech?

It's an interesting rhetoric that the software industry introduced the concept of no-shirts and ties, and flattened everything to be purely about business.

However, one still wonders if and why appearances matter in the technical world?

So, do they?

The answer really is, 'Yes and No.'

Appearances do matter in this industry and yet one can totally get by working in pjs.

Especially, since Covid, many times I dial into calls, either the first thing in the morning, or sometimes at late nights, and keep my camera off.

It certainly is a huge benefit of working in the IT world, or with computers that one really does not have to worry about how they are looking.

And yet, at many times, I do not want to step out of my house with the fear that my looks are not up to par.

So, why and how do looks matter?

Abstracts

The Happy-Path

This strategy suggests that one should focus on the 'happy-path' of execution, i.e. the path of MVP success, while hypothetically assuming that there would be no errors in the system.

The term 'Happy-Path' means writing a feature in a way that assumes no errors, in order to get the working scenario done as a proof of concept.

A few months back I was really stuck on understanding all the parameters of the system and how would they come together. However, as I started talking to one of my managers he called out in a completely different context to consider the happy path. Somewhere that created a spark in my brain that probably I really don't need to be a perfectionist here and don't have to worry about every single scenario from day one but rather just working on a single working scenario and to end

would be a goal to begin with. This led me out of my perfectionist paralysis mode which really helped me carry through for a feature POC.

Be Candid

This strategy is about being honest and candid to your true-core, when faced with something contrary in your career.

documenting my learnings/practices for my fellow-colleagues in good-faith.

- Another application of this strategy is that when things may not be completely working out is that instead of focusing on how hard or difficult, they are, or how people around us may be viewing us, it is to still 'Do our best, despite ourselves.' In this event, we can step in to help someone on things we know, or questions we can answer.

Collaboration vs Competition

This strategy is about **nurturing collaboration** over competition amongst your colleagues.

I come from a culture, where hard-coded competition is ingrained in children from day one, which includes, at times competing with one's own siblings. I once remember I once remember when a teacher called out to me something along the lines of whether I competed with my sisters or not and I said that ohh I don't compete with anyone on which occasion she pointed out that ohh you probably don't compete but I'm sure your sister does the period the point of the story was that my sister was definitely smarter and better because she was more competitive. This analogy of encouraging competition amongst each other and colleagues actually eventually works to our own disadvantage. It is sad that some organizations continue to harbor competition amongst

colleagues in an effort to call out individual efforts but at the end of the day the business as a whole actually loses when people are just trying to outdo one another instead of building on top of one another.

Work in Good-Faith

Always **continue working in good-faith**, even when you feel that you are not valued or needed.

There are many times in our careers when we may feel despaired. For instance, switching teams or jobs etc. or not being as smart as everyone on the team, are some instances.

One strategy to overcome these times is to continue to work in 'Good-Faith.' What this means is that we keep working, despite of where we will get, whether we will be appreciated or not, whether we will be promoted or liked or not.

Some practical tips in this scenario are as follows:

- If one is onboarding to a new project, then instead of only focusing on what one has left behind, or what the future would be, it is to focus on 'What can I contribute to this cause.'
- For this specific case, I have started

The Power of Sharing

Learning Strategies

Learn By Writing

One of the key methods to learn is to **employ the muscle of writing.**

In the current age of technology, although it has been a great innovation to use phones and keyboards but it has also been disadvantageous in many ways.

Technology has stripped us off our writing capabilities.

I was describing the other day to someone, that how as part of learning a new product technology, I spread papers all over and use my pen and pencil to write. Although I do not have any scientific evidence of this skill but I am sure that it makes a difference. It is sort of the 'organic' way of learning.

When one is writing they are actually creating a direct communication pathway between their words and their neurons. Writing-by-hand, is one of the most effective ways of learning.

acquire will only be *part of the story* and not the full story.

This perspective helps one to not be a perfectionist and not to undermine one's efforts, even if the results may not be visible in the overall short-term.

The idea is to continue to focus on learning, making progress and continuing to move on. When we only focus on succeeding in the current scenario, then any setbacks, challenges and issues totally tear us apart.

However, if we focus on only working on a piece of the bigger puzzle, then we are in a constant state of transformation, growth and improvement, where lessons from one story help us weave the next one.

The Game is Not Over

This strategy is about believing that, you are always working on something bigger than what you are working on in the moment.

One key strategy that I use to continue to make progress is that 'I am always working on a bigger problem.'

Suppose one is working on a problem or technical issue, such as learning the architecture of a new system. They may feel discouraged that they will never be able to know or conquer it all. In such an event, the 'The Game is Not Over' strategy entails the following:

- It is to re-iterate that one is always working on a bigger problem.
- The current challenges and setbacks are only part of the bigger-picture. As a result whichever learnings and experiences that one would

Getting Your Feet Wet

The analogy that I like to use here is when I started to enter the pool the first time in my late 20's. The idea was that swimming has never come intuitively to me, but what I kept pursuing was, how to get 'my feet wet in the water, by just dipping my face, or trying to stretch while holding a rod.'

When it comes to onboarding to new technology, I like to apply the same parable.

Here are some steps that I have been employing to get accustomed to a thick set of documents or learning something new:

- Get your feet wet. So, first and foremost, just peek through the deck of notes. Just browse through without any real goals. This is similar to window shopping, when one is just browsing and seeing, what is out there.

What this does to our brains is that it exposes the breadth of all the information that is

available, without intimidation of having to conquer it all.

- Make a list of Top-3 questions that you want to get answered, while learning this new stack. For instance, here's my list from one of my last learnings:

1. What is Technology X?
2. What were the last 3 major deliverables here?
3. How does it work with Technology Y?

incremental progress. It is about not being taken away by inertia, but rather continuing to move despite the roadblocks.

This is my mantra for today, 'Succeed by Resolving to Fail.'

This is my mantra for anytime I am blocked by the paralysis of not knowing anything about anything. It is to be ready to fail and failing through it all. It is doing things believing that yes we will fail.

As we are continuing to fail, we are still treading steps and carving a path.

That path is the eventual light to get to the end of the tunnel.

Resolve to Fail

This is akin and opposite to the idea of 'trying-to be successful'. Instead of trying to be successful at all, this tip involves asserting and wanting to fail.

When we start our day by saying that we want to succeed, that takes us to a state of being a perfectionist, which in some cases, eventually paralyzes any progress that we could make. However, when we resolve to start our day by failing, then every step will be progress. Whether we stumble, whether our computers blank out, whether we are stuck on 100 min customer service calls, whether we do not get unblocked, either way we will continue moving through.

The key to success is not perfection. The secret lies in being able to 'Continue Through Regardless, without obsessing on whether all of it is getting done or not.'

For that, it is important to have a perspective of

Black Box vs White Box Learning

When faced with a technical ramp up of any kind, one questions if they need to know everything all at once?

One strategy that I like to apply is that of separating ideas into Black and White boxes.

Essentially, this entails listing down every single piece of information that one needs to learn, e.g. documents, terms, code, etc. One would then classify them into 'White or Black Boxes.'

To begin with everything is a Black-box. What black-box means is that one does not really need to deep-dive, unless they have to.

My strategy for all black boxes is to

Extend the Runway

Success Tips

way, I started with a theme of documenting:

- Purpose of the Blog
- A North Star Vision for System Setup
- What questions did I encounter in the process and how did I get through?
- What legend could I leave behind for those coming next?

Setting up Systems

Every time I am asked to set up a software system, some part of my hyper-subconscious brain gets activated with a lot of pessimism. It is as if I am determined to believe that I will only face road-blocks along the way.

Although, this is not all schizophrenic and there is a truth to how difficult our world is when it comes to setting up systems, I was thinking about what has led to this paranoia?

The earliest system that I recall setting, which never worked out for me, after spending months of efforts was the Derby database code, as part of my Graduate degree work. I kept spending days and days but kept running into one issue after another. The question really is that, 'Why is setting up Software Systems so complex, still?' Why is it that we never have a seamless end-end way of just getting it there? Or will AI-agents eventually fix it all?

Some things that have helped in this process are as follows:

- When asked to set up a system, do not commit to doing it overnight. Ask at least for 3-4 days, if you cannot ask for a full week. Yes, we all know that the devil is in the details, so make sure that you have enough runway
- Review all the steps in a Breadth-First way first, before jumping on getting started. It helps to know all the things that are needed and calling out perhaps which ones seem more involved than others
- Befriend at least 2 people who are in your shoes, i.e. tasked with the same or who have done this previously. Keep sharing progress and updates.
- Self-Gratify. As you accomplish even the smallest of the steps, give yourself a pat on the

back, get a cup of coffee or move around for a stretch.

- Always do more than one thing in a day. So, whether it is about setting up a system or something else, have something on your plate to do, when you get stuck. Instead of being stuck for too long, change your attention for a little bit on something else. Distract your brain with a different problem. This will hopefully clear some brain fog and may help you when you come back to the problem.
- Make a story or a movie out of your setup problem. We have been trained to work in a very linear way, where we follow steps, double-click on problems, reach to our networks and continue the cycle till success is achieved.
 - o In one of my system setup incidents, I decided to convert the experience as a blog. So, instead of going the traditional