Software Diaries - 2

Author: Irum Godil

Introduction

Preface

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 suggested that I should write my next book on *Copilots.*

My response has been that I have been on a break.

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 helps me to be productive and creative when I get to convey my thoughts and learnings to the world.

Hence, as I am continuing to ramp up new dimensions, I am documenting my learnings for myself and others. This book is documented in a blog format.

As of July 2025, I have decided to publish this book with the work done so far. However, as I continue to find more tips and tricks to share, I will either update it in the same book here or start a new edition of the same.

Thanks for your interest in this work!

-IG.

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 also the one who asked me to write my next book on Copilots, which God-Willing, may be my next endeavor.

- Also, a very special thanks to my nephew, **Asif Moosani**, for calling out that I should continue to write, as he called 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, then, that it was great that I had done something outside of work and followed

my 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 using different modalities for voice-to-text conversion. As part of working through this book, I have tried to use voice-text in multiple places, although have not been totally successful yet. 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* who have been on my side through my peaks and valleys, always.

Outline

This book is structured into different sections as follows.

Success Tips:
This section talks about the overall mindset of being successful in this and other industries. It touches on the things to keep in mind when we feel that we are stuck, or if we are not sure as to why things are not working in our favor. It is more about the soft side of our brains and how to nurture it so that we can be successful in our journeys.

Productivity Strategies:
This section provides tips to be productive, especially, in the case of heavy-duty ambiguity and complexity.

Learning Strategies:

This section talks about learning skills that have been helpful for me and the pointers on the same.

Abstracts:

This section outlines some random and abstract thoughts about the culture of the software industry.

Conclusion:

We are living in very uncertain times. As much as AI is supposed to empower us, many folks in the industry are struggling with questions such as, *what is their future and value-add in this environment.*

I have put my thoughts on the same in the concluding thoughts.

Success Tips

Why is Love Everything

According to Steve Jobs, "The only way to do great work is to love what you do.[1]"

The reason passion is everything in our personal and professional lives is that when we care to do something, we tend to persist better, we tend to go the extra-mile and work is not work at that point.

Once someone asked someone during an all-hands about how to be successful and the person described many mechanical things. At that point, I called out that we should also pursue our passions and interests.

The main struggle we have today is that do we

---

[1] https://news.stanford.edu/stories/2005/06/youve-got-find-love-jobs-says

always get to do things that we love to do, both in our personal and professional lives?

Practicality, everyday struggles and having to do things that we have to do generally always trump our passions.

So, how do we really get to do what we love to do, while still succeeding in our material lives and staying on top of what we want to achieve practically and financially?

For instance, I certainly love to write but would that mean that I would switch careers to be a full-time author? Most certainly not, till I get to the point of being a bestseller, hopefully one day.

So here are some ways to find passion in our everyday life, in the midst of practicality and cut-throat competition:

- First and foremost, it is important to identify what we like. Sometimes this is a revelation to us and sometimes we just know from day-one as to what we like to do vs. have to do.

- For instance, I did not know that I had a passion for writing till much later in my life. But from a technical perspective, I just know that I love being in the UX world, or now in the AI space.

- Once we know or have the revelation of what we like, we need to find opportunities to make time and space to grow in that space, even if our current circumstances, work, and life-situations do not permit for the same.

- So, for instance, with writing this book, I have at least found something to lean my passion against the industry that I am in. Similarly, at work I am always on the lookout for learning and growing in areas that naturally call to me,

whenever possible.

- Lastly, remember that everything that you will
  do in the pursuit of finding and living your
  passion would be a journey to that end. So, for
  instance, once I was put into an end-end
  project. Mid-way into the project, I naturally
  gravitated towards the client-side of the project
  as and when the opportunity came. Later, I was
  once again seen as the client-champ, which
  really was a win-win.

Being true to ourselves and our passions is really at
the heart of success in both our personal and
professional lives.

Whether we must run technical errands or perform
our personal responsibilities, there are challenges
in all directions.

But passion for what we are doing is the *key* that

16

paves the path to help us get through.

It helps us go the extra-mile, softens the hardness of the challenges in front of us, and provides the torch to stand in the dark night.

Love really is everything, whether it is for our worldly careers or metaphysical ambitions. Finding that love and trying to nurture our paths with it, on a regular, consistent and everyday basis in both our professional and personal life, is really at the heart of all success.

So, if you have not yet found or are not able to pursue your passion, continue to keep looking, as Steve Jobs would say.

Resolve to Fail

*This is akin and opposite to the idea of trying-to be successful, as mentioned in one of the productivity strategies below. 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 is progress. Every blocker that we experience in the process is a mover towards our goal.

*The key to success is **not perfection**. But rather, the secret lies in **being able to continue through***

regardless, without obsessing about whether it is all getting done or not.

For that, it is important to have a perspective of incremental progress.

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

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 fail through it all.

The secret is that as we continue 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.*

The Game is Not Over

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

One key strategy that I have used to make progress is to maintain the mindset that I am always working on a bigger problem.

Suppose one is working on a problem or a 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 *Game-Is-Not-Over* strategy entails the following:

- Re-iterate to oneself that one is always working on a bigger problem.

- The current challenges and setbacks are only part of the bigger picture. As a result, all learnings and experiences that one would acquire will only be *a segment of the bigger picture.*

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 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 picture, then we are in a constant state of transformation, growth and improvement, where lessons from one experience weave into the next one.

The Power of Sharing

We are living in a very competitive world. From pretty much early childhood, whether it is at home or school, in even innocent things like playing with each other, we are taught to compete and win. However, what we forget is the basic principle of nature.

Giving only increases and not diminishes; in whatever form it is done.

The tech world today is known for its cut-throat culture, where people are always waiting to outdo others and/or finding a way to do it all. However, one lesson I recently learnt is the importance of sharing in technical landscape. The principle is that **sharing never reduces opportunities and only creates more.**

Even at times when we may feel that there is a dearth of opportunities or layoffs are common, *sharing builds a culture of trust.*

Sharing does not have to be only about sharing money or items, but it includes everything. It includes being open about a discovery we may have made, knowledge about how to onboard a new system or calling out roadblocks that we may have faced, so that others could avoid them.

Whichever field of profession you may be in, and however hard the climb may be, do not hold back information. Yes, sometimes that is what our environment teaches us, but just remember *that your soul is always bigger than the culture around you.*

Even if you may not win in the current game, just the fact that you have shared will create a new opportunity for you to conquer. Always remember

that giving never reduces and never deprives the giver from goodness but only creates more of it.

Being Transparent

*This strategy entails being **honest** to your core and **transparent** to who you really are.*

I was once asked in an interview if I could compete with team XYZ, and my answer to them was 'No.'

Yes, it does not sound right in this time, where we are taught to compete at all costs and be on top of everything, no matter what. But I just knew deep inside that I could not.

If I had signed up for that project, because I had not been honest to myself, I certainly would have dragged myself into much deeper claustrophobia and eventually failed even more miserably.

The question in the industry of course remains, i.e. should you be honest about who you are, or should

you put up with the expectations of the world at all costs?

Of course, survival trumps everything and it all depends on what choices one has.

But even in my case, partly, I did not say no, because I was clear on my alternative choices. I said no, because I just knew that I had to protect myself. And this was not about a lack of confidence, this was driven by data, that no matter how hard I had tried to stay on top of that game, I knew my counterparts were just better.

The key learning was that at the end of the day, our *own personal existence matters*. If we cannot breathe, we cannot execute.

So even though it is important to put ourselves up for challenges and expose ourselves to unknown terrains, *but one has to be true to who they really*

are in their core. And when they are not, it really does not serve anyone, including themselves.

Productivity Strategies

The Happy-Path

(This chapter is dedicated to my manager Anand Vaidyanthan).

The term *Happy-Path* is about writing a feature in a way that assumes no errors, to get the working scenario done as a Proof-of-Concept (POC).

This strategy suggests that one should focus on the basic MVP success, while hypothetically assuming that there would be no errors in the system.

A few months back I was really stuck on understanding all the parameters of a system and how they would come together. As I started talking to my manager, Anand V, he emphasized in a completely different context to consider the *happy path* only.

Somewhere that triggered a spark that I really did

not have to be a perfectionist and did not have to worry about every single scenario from day one, but rather just needed to focus on a single working scenario.

This tip was so effective that I felt it must be documented in gold, for all our endeavors when we feel stuck. Instead of thinking of all the variations and wanting to get it all done, we need to just focus on the *Happy-Path*, before we start thinking of all the things that could potentially go wrong.

Morning Dopamine

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

I have generally not been a morning person, so doing anything in the mornings takes a real toll. 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 to just act as an automaton and keep executing, without judgment and self-criticism.

For instance, when brewing coffee, the first thing in the morning, instead of thinking of all the things to clean in the kitchen, just focus on having a simple

breakfast. Similarly, at work, just boot a system and do a basic task initially, without judging if that is the most important part of the day.

The eventual result is that once we do one, two or three things, some dopamine gets released. Once this dopamine starts kicking in, it continues to keep kicking in and self-feeds itself to help us execute bigger and more complex tasks.

Try-To

This strategy is about telling yourself that you will
try to do *something, rather than doing 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 codebase, and I committed to getting
it all done, then any blockers along the way would

create a sense of failure, which in effect feed into more de-motivation. However, if instead of committing to getting something done, if we commit to 'trying-to' get something done, then any progress in the path is still progress.

Similarly, any roadblocks are not permanent failures, as we are still on the track of *trying to* get there.

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

Start Where-You-Are

This strategy is about aligning what we know and what we have to our disposals, over all the things that we need to learn and do not have to our disposals.

Akin to the idea of the happy path, it is about recognizing what we have, instead of looking at the mountain ahead. So, say if one had to start hiking on a mountain then this strategy would entail focusing on the muscle-power that one has, instead of the peak ahead.

The human mind naturally gravitates towards all the things that could go wrong and all the complexities that one would encounter in a situation. This strategy is about putting those in our blind spots, while we gather the tools that we already have to our disposal.

Once we recognize what we do have, we can then slowly start to include the things that we do not have.

A case study here would be that of building an executive strategy around a best practice in the industry. Both how much to do and what to do would give us anxiety by default.

This strategy would entail putting down 1-2 technologies that we know of in the area and then identifying anything else that we know deeply in that area. In addition, listing out 2-3 folks whom we could count on in this operation, would also be part of the starting point. Once we have this list, then we would continue to expand in each direction, in the same fashion.

This strategy is about getting over that initial hump of the load that we must conquer. It is to start with the 1-2 dominos that we have control over to

determine and build the course of the next set of dominos, that we do not know how to conquer.

Making Progress

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

Some tips for this strategy 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 the actual deliverables on the backburner and to focus on building the mental muscles.

The value-add of doing this would be that when we would come back to the same or a similar problem, next time, we would be better equipped to deal

with it, and hence would have a better chance of getting to the actual deliverable.

Reframe the Problem:

Try to re-define the problem, in some way.

For instance, I once spent a few hours looking at a code-piece trying to figure out how it would render as React-Markdown. I was stuck all along, while trying to understand the existing rendering code. But eventually reframing my search to how to render HTML text into React Markdown, vs trying to readjust the existing code did the trick.

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 or a different question, about what we are trying to solve, is sometimes the *key* to move forward.

Exercise Gratitude:

Although, I have a full chapter on this below, when we do see light at the end of the tunnel, if we do, then we really need to practice gratitude, no matter how small the win is.

Celebrating small wins and progress is as crucial as solving the end game. As mentioned earlier, when we get something going, it releases dopamine to get more of it going.

Break Things Down

This is a no-brainer strategy of always breaking things down.

The main question is that when one does not know where and how to start, how does one know how and where to break things?

One advice that I was once given was to *break my day into multiple parts.*

Somehow, I have learnt that by Wednesday my anxiety is generally half of what it is on a Monday morning. Although nothing has really changed from the perspective of the problems at hand, but somehow there is a brain shift when we have moved through half of the week.

So, in essence, even breaking the day into multiple parts is similar. The mornings can generally be the most uptight time, as one starts from the peak of ambiguity. But as one continues to break out every 3 hours or so, then just knowing that one has covered something helps.

In the same vein, breaking the problem down would involve classifying the pieces of a big unknown puzzle into small sections.

As they say that 'Rome was not built in a day.' Continuing to work on those small segments, even if we do not always know what those actual segments are – can help us to keep resolving the bigger problem.

This is also at the very heart of dynamic programming, i.e. to solve a big problem by solving its pieces and then using the solution of the sub-

problems to arrive at the solution of the more complex task.

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 recall a time in Grade-9 when one of my teachers had asked me if I was better than my sister. I told her that I do not compete and compare, in general. Her response was that well your sister definitely does.

The point was to assert that somehow people who are competing are better than those who are not.

Now, again, we are living in a world of cut-throat competition, but how do we know that competition is always the end-winner.

And even if it is, there is absolutely no reason for us to compete with our own teams and colleagues. Collaboration is always more effective than competition.

The key here is to identify traits of each team player and stack them against how they complement each other. One person's weaknesses can be another person's strengths and aligning those *with* each other, rather than *against*, is the key here.

Dealing With Ambiguity

The key to getting over ambiguity is to put
structure *to the ambiguous domain.*

In my last publication, I had called out the project
of system-wide-flighting that had led me to my
Principal-level promotion.[2]

The key strategy that I had learnt at that time, of
heavy-duty ambiguity, was to document every
single point of ambiguity as a task or a work-item
and then to work on each one of them, as I would
do for a defined task, from someone else.

Essentially the biggest blocker of ambiguity is not
knowing what to do and how much to do. But once
we put some boundaries and start digging through

---

[2] https://www.amazon.com/Software-Diaries-Irum-Godil-ebook/dp/B0CQDT8MW5

something, then we get better at doing it and the structure continues to grow.

Work in Good-Faith

*This strategy suggests **working in good-faith**, even when one feels that their efforts are not valued or needed.*

There are many times in our careers when we feel despaired. For instance, when switching teams or jobs etc., or at times when we may feel that we are not as smart as others.

One strategy to overcome this mindset is *to keep working in good faith.*

What this means is to keep working, despite of where we will get and whether we will be appreciated 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, one could focus on: *what can I contribute to this cause.* For instance, this book is a classic example of this scenario. As I started working in a new domain, where the results were not clear, documenting my learnings and journey, for others, in good faith, has certainly helped me to contribute and grow in my actual work.

- Another application of this strategy would be when things are not working out. Instead of focusing on how hard or difficult everything is, and how people around us may be viewing us, the strategy entails *doing our best, despite ourselves.* In this scenario, one would step in to help someone on things they know of or questions they could answer.

Working in good faith is important when we are feeling stuck and not sure of our paths. Instead of completely giving up or succumbing to what the eventual result would be, this strategy entails doing our best, with the hope that our spirit will lead us to the actual results.

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, it has also been disadvantageous in many ways.

Technology has stripped us off our writing capabilities.

As part of learning a new product technology, I spread papers all over and use my physical pen and ideally pencils or color-pencils to write.

Although I do not have any scientific data for this methodology but when one is writing with their hands, they are actually creating a direct

communication pathway between their words and their neurons. It is actually an *organic* way to learn and to build the neural networks.

Get Your Feet Wet

*This strategy involves **exposing oneself superficially** to a technology or domain, without setting any real expectations.*

The analogy that I like to use here is when I started to enter the pool for the first time in my late 20's. Swimming to this day does not come to me intuitively and even the other day someone was trying to give me tips on how to back-float.

However, one can get their feet wet in the water, without having to swim through the deep end of the ocean, or without swimming at all.

When it comes to onboarding to new technology, I like to apply the same analogy. Here are some steps that I have been employing to get accustomed to a thick set of documents or learning

something new:

- Peek through the deck of notes and browse through without any real goals. *This is like window shopping to just browse.* But it actually exposes our brains to the information out there and opens our neural segments.

- While browsing, make a list of top 3 questions for each category. For instance, here's my list from one of my last ramp-ups:
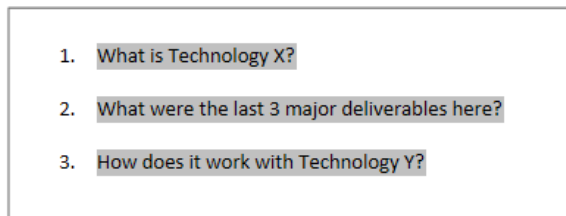
1. What is Technology X?
2. What were the last 3 major deliverables here?
3. How does it work with Technology Y?

Figure 1: Top-3 Questions

- Breadth over depth and depth over breadth. Identifying where to go deep is as important as

going deep everywhere. As our scale of tasks increases, we certainly cannot get everywhere. But breadth shows us all the paths we must go to, and depth shows us how to traverse those paths. Technically, from a graph perspective, getting one's feet wet would be doing a breadth-first traversal of everything that needs to be done, before diving deep into anything.

This technique can also be documented as 'Black Box vs White-Box Learning'. Black-Box means just touching the surface, without knowing any details. As one is ramping up, it is important to call out as many areas as black-boxes and then only dig deeper into the relevant areas.

Extend the Runway

This one is my favorite strategy. The biggest barrier that I have found in my career is when I have resolved to do things overnight, just to recognize that nothing is mostly possible overnight.

Extending the runway means giving oneself enough bandwidth to get the results.

So instead of expecting results overnight, one would expect to finish the tasks over a longer period of time, say a week or a month.

This learning is extremely important as, somehow, we are ingrained to finish everything right away.

But my main learning through this industry has been that 'Nothing works!' or definitely 'Nothing works right away.'

So, the fair and wise way to make things work is to not expect them to happen instantly, even if they appear to be seamless.

But rather it is to give ourselves a bigger runway.

Case Study: Setting up a System

The above strategies can be applied to a case study of setting up a system. The case in question is about setting up software systems, which are still mostly pretty complex.

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

- When asked to set up a system, do not commit to doing it overnight. Ask for at least 3-4 days if you cannot ask for a full week. We all know that the devil is in the details, so make sure that you have *enough runway.*

- *Get your feet wet* only initially, i.e. review all the steps in a breadth-first way first, before jumping onto details. Identify all the steps that

are needed and perhaps call out which ones are more complex than others.

- Befriend at least 2 people who are in your shoes, i.e. tasked with the same work or who have done this previously. Keep sharing progress and keep thanking folks for their support.

- 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, distract your brain with a different problem. This will clear the brain fog and will help you when you come back to the

problem.

In conclusion, be creative with your problem set e.g. make a story or a movie out of your setup problem. Writing this book has been the same for me, i.e. I have taken the experiences of a hard-core distributed system set up and converted it into a novel-story of learnings, that others can also benefit from.

Abstracts

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 would dial into calls, either the first thing in the morning, or sometimes at late night, 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, many times, I do not want to step out of my place with the fear that my looks are not up to par.

So, why and how do looks matter? Looks do matter!

I am personally sometimes really conscious of wrinkled tops, or a bad hair day, even when my hair is covered. But why?

I think the Software Industry is as much human as it is IT-based. The human instincts are still all 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 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 on looking good is an actual investment.

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 also paves 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.

Practicing gratitude is a mindset that needs to be cultivated even for the smallest of the accomplishments.
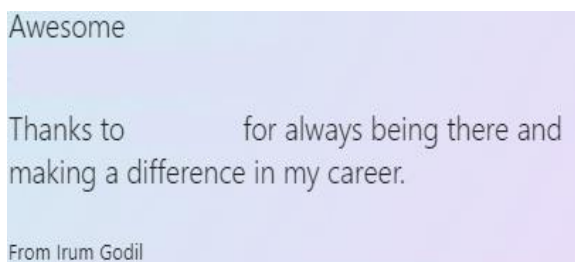
Even if our environment may not make us feel that those are worthy accomplishments, practicing gratitude for them helps us to maintain an optimistic mindset. It helps us to continue moving through the woods and uplift our spirits.

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

- If the smallest part of the 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 continue to move towards

the 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
  real blessing. Currently, 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.

Awesome

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

From Irum Godil

Figure 2: Thanking your Network

- 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 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 to be our best selves, instead of looking down on things that others got and we did not.

Constructive Feedback

Constructive feedback can be one of the hardest things to absorb in this industry.

At many times in my career, receiving constructive feedback has completely torn me apart for weeks and months at times. However, as I have learnt to deal with what the person had actually meant and how it has 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, then that automatically shifts our reactions.

- At one point, when I was given 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 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 have to be cognizant of it.

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

Constructive feedback can be hard both for the receiver and the provider. However, it is important that if we give feedback to someone then we try to give them as much context and data and always give it to them in a sincere way 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 only meant for growth.

Once we understand the perspective of the person who has given us the feedback, it would 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 important to detach ourselves from it, and step away to revisit it with a clearer mind.

Conclusions

Navigating In the Future of Uncertainty

We are living in a paradoxical world. Industrialization is climbing a new peak with AI, and yet we are scared that using AI will leave us with no jobs.

So how do we find the motivation to carry on?

I will conclude this book with the ending notes on shared optimism, for myself and my audience on this topic:

- We need to approach AI with optimism of the *NextGen of Learnings.* It is not about what will be obsolete or replaced, but what new discoveries we can make.

- If the world was all automated, our problems would not end, they would only change. As engineers we will then be focused on those new set of problems.

- The light in the current moment and at the end of the tunnel, is not that we will have nothing to mine, but that we are currently standing on a *Gold Mine. AI is that Gold-Mine.*

We need to think of AI as our *third hand* and unleash it to let us move faster and better, but also towards the *Impossible.*

So, start your day by questioning *what can AI do that I have always wanted to do?*

Can it solve cancer for starters? Can it solve time-travelling? Or can it set the stage for solving any/all of these?

The world before and after AI were both challenging, but with a different set of opportunities at the same time.

Not using AI because we do not want to empower machines is not the answer, because machines are already empowered.

We just need to utilize them to our advantage.

Hiding away in a corner, that everything is over, is not the way, because **now is our time.**

It's a new time, the age of AI, but the age of us as well – i.e. *Us at the heart of starting and learning to use AI.*

No one knows what the future will be, but no one knew what the future, that is today, would have been 5-10-15 years ago either.

Let's move forward with the hope of creating, shaping, reshaping and writing a different, as well as our individualized future, instead of shying away with the fear of failure and the unknown.

Let's move forward with leveraging AI for both our personal and professional successes, while resolving to make this world a better place, in our names, while we can!

Thanks for reading this book!