Chapter 10 | How to Hold an Axe

I heard a story about a famous film producer who built and kept a team around him for decades. Sure, they make great movies, won awards, and got rich. When asked about the team, the producer said: The team speaks in shortcuts. We know what we mean when we say something.

Having been member of elite athletic teams, emergency response teams, and a civilian member of a military team, I can recall the jargon, the slang, and the verbal shorthand phrases we used. Some phrases slip into everyday use and sometimes become clichés.

In building our team, we too craft verbal shortcuts into our language and communication. They intent to convey critical thinking points and critical steps.

In the last chapter, number 9, called "The First Minute", I introduced the question: "Are you walking the dog? Or is the Dog Walking You?" This question helps our team explore who or what is in control of a process. The other day, I accidently touched our new washing machine and it make funny noises, a click, then started to spin. I wondered if I caused that. I wonder if I turned it on; if my accidental touch hit the control panel. I tripped and the washing machine steadied me. The washing machine was already in a cycle. My touch didn't cause a damn thing.

"Are you walking the dog? Or is the dog walking you?" reminds a programmer (or anyone) that sometimes things happen and sometimes we make things happen. We need to know the difference. Academically and formally, one is questioning whether a relationship between two events is causal or correlated. I suppose our team could ask: "Are you confident you have complete control over the causal aspects of the behavior?"

We're humans working together. We prefer cute. We like slang. Like that famous movie guy, the linguistic shortcuts reminds us what matters to us and reminds us of traps we have all fallen into.

Today, I'll explore the phrase "Trust the tools". The phrases include:

- PST baby - Primary, Secondary, Tiertiary
- Begin with the end in mind (H0)
- You're a tool maker, make tools.
- The long pole
- It is just work.
- Baby Steps
- Where are you going? What do you have? What do you need to get there?
- Train the way you fight; fight the way you train.
- Maintain parallel structure.
- Let the data be your guide, follow the data.
- Story Telling with Data; let the data tell you a story.

- If you don't look past the horizon, there is no tomorrow.
- It is time to paint with a little brush.
- Charlie Mike - up/over, down under, through or around
- Box unknown with known.
- When skiing trees, look at the gap between the trees and not the trees.
- When troubleshoot, be suspicious of the last thing you did!
- Know the difference between busy and working.
- Don't let the enemy of better be best.
- When swimming towards the shore, make sure it is getting closer.

I could not call this episode: “Trust the tools” because the word “tool” has several alternate definitions in American slang, some are rather derogatory. Instead, I named the chapter: “How to Hold an Axe.” Software developers, as I have mentioned before in this podcast, are the modern-day tool smiths. While we may not stand at a forge and bang on metal, the thought process, the design process, our operational understandings reflect our heritage linking us with earlier tool smiths – whether they stood by fire and bang metal, nor napped at stone with other stone. We recognize each other as smiths.

Of the fundamental tools is the humble axe. The axe recognizable with a stone fastened to a stick is about 8,000 years old according Wikipedia. The axe is about 8,000 years older than software. The hand axe (sharpened stone without an attached wooden handle) is about 1.5 million years old. That hand axe is still recognizable in our kitchens and workshops as a knife.

Axes do not come with instruction manuals. Most, at least those sold in the United States, do not come with warning signs or instructions. Yet both ends of an axe are dangerous. It is a sharp chunk of metal attached to a stout stick of wood. The humble axe is one of our oldest tools. Ancient human tied sharpened stone to sticks then whacked stuff.

As a software developer I use some of the most modern tools on the planet. Software, as a tool, is an invention of the 20th Century. I acknowledge pre-cursors to software found on weaving machines and tabulating machines of the later part of the 19th Century. Some of these machines relied on instructions punched into paper. For practical definitions, software as we know it dates from within the recent 100 years – the 20th Century.

Why is a thoroughly modern tool smith doing discussing sticks and stones (I mean axes and knives)? We, software developers, make tools. “Trust your tools” is a pole-star phrase, a guiding phrase used on our team. Cliché is not the right term, but a touchstone or a unifying concept to our team. Like asking a teammate: “Are you walking the dog? Or is that dog walkin’ you?”, it serves as a fundamental principal for our teamwork.

Our common ancestors used tools to make other tools. They used tools to make the stone axe. That’s less obvious to us all, isn’t it? Archeologists recognize stone tools because these tools were fabricated by humans. Someone sat to shape and sharpen a stone – normally flint or chert

– with another tool. The artisan flaked away bits of stone to create cutting edge to meet the needs of the day such as rendering tree into firewood or cut meat following a successful hunt.

I sure would love to take the moment of this chapter and this topic "Trust your tools" to detail the right way to do all things including using the suite of tools that my teammates and I use to make software applications. The very moment, I try, I fail. Right and wrong have little meaning to the artisan. I acknowledge that both right and wrong are my opinions.

I have used an axe as a walking stick, a hammer, an anvil, a step, a weapon, oh, and as an axe to help maintain hiking trails on our land here in southern Vermont. I have even tossed axes at targets. I cannot sincerely tell you the right way to use an axe. I'll not tell you the right way to write software either.

We, software developers, differ in every possible way. Each of us practices our craft with a specialization and sufficient pride that our opinions resonate like doctrine. Oh, I have doctrine and I whack teammates with my doctrine striving for uniformity. That's called team building and it also relates to the training of apprentice programmers.

The apprentice is taught that there exists the RIGHT WAY to do something. The apprentice learns the rules. The journeyman (and I can't find a gender-neutral phrase for this term) supervises apprentices and enforces the rules. Along the way, the journeyman gradually discovers that the very strict rules and doctrine have flexibility. The doctrine is often best-effort, evolving, and improving.

The master artisan interprets that public norms, public expectations, writes the rules, creates the doctrine.

The apprentice learns the rules. The journeyman enforces the rules, and the master writes the rules.

There is so many ways of exploring this relationship. The young music student learns finger placement and scales. There is a right way and a wrong way – listen to nearly any teacher to hear variations on this phrase.

In our shop, I will emphatically tell a young program that embedding procedural code inside of an Oracle APEX page is wrong. Put your code into a package that is compiled in the database. It will run faster; it is stored compiled; it is easier to find; it is consistent with the team's practice. And yet, we deliberately break this rule when we need to. It takes time, effort, and research to know when to break a rule.

One might think that knowing how to hold an axe is simple. Grabbing the wooden stick whilst holding the sharp metal head away from your body is a good start – but it is it?

Where should one start a lesson on holding an axe?

Or writing software?

Or building a tool?

Beyond the obvious statement that the sharp metal head is dangerous, nothing else is entirely obvious to a rookie or a newbie. I could start the lesson about how to hold an axe with out the axe.

This is how you stand – creating a firm athletic stance with feet below the hips, knees slightly bent, hips open with the foot below the dominant hand back behind the other foot.

Where does the power of an axe swing come from? I may just explore the strongest and largest muscles employed during the swing of an axe.

And sometimes the twinned statements of identifying the sharpest and strongest part of an axe needs to strike the target precisely. The first statement requires knowing the anatomy of an axe and which bits are the strongest and sharpest. The second statement requires understanding the anatomy and vulnerability of the target.

A tennis lesson, a squash lesson, baseball batting lesson all start with the same fundamentals. Even a skiing lesson starts with the basics of body mechanics and the importance of deliberate movement. Your first piano lesson, or violin lesson, began with someone sculpting your body to optimize strength, movement, and agility.

I sit at a desk with three monitors. The body mechanics of thriving and enjoying a lifetime of software development followed the same basic lessons. Learning to master an axe, a kitchen knife, writing prose, and writing software share common roots in the phrase: "trust your tools." The first of these lessons is trust your tools.

Lesson One: Know your tools, their strengths, their weaknesses, and the off-book uses.

Lesson Two: Take care of your tools

Lesson Three: Develop skills, practices, and strengthen your abilities regularly.

Lesson Four: Know the difference between the right tool, a good-enough tool, and the wrong tool.

Lesson One: Know your tools

I took a hiatus during the winter holidays at the end of 2020. We celebrated many of the holidays. We did the menorah, latkes, and cider doughnuts for Hannukah. My niece got the prayer right many nights (she was a bit rusty). We lit the candles again, adding one per day starting with the solstice expressing hope for the return of light and hope in 2021. We honored a few of the Christian traditions too at Christmas. In short, the family pod unit cooked often and a lot. Each holiday required its own feast.

Feasting in rural Vermont means knife skills, butchering, food prep, and related. We buy meats from neighbors. We buy a half-lamb each year from Andy. We buy beef from Sunrise Farm. We buy chicken and eggs from Cooper's Coop. We have three freezers here and in the mudroom, which is as cold as a refrigerator, we have a plastic tub filled with our potatoes and one with our carrots.

My niece moved in with us during the duration of this pandemic. I observed that I got super grumpy when she grabbed my "Chef's knife". I discovered that I regarded that one tool with propriety and strict ownership. My brain screamed out: "That is my knife." This emotional response surprised me. I do not feel that way about any other knife or tool in the kitchen. And one should never feel proprietary when getting help in the kitchen.

In short, I bought Rachel the identical knife. And that mean teaching the 36-year-old woman how to handle a knife and how to get to know her knife.

Simultaneously, we both pushed our skills as we explored and enjoyed preparing for our feasts. We studied her body mechanics: body position, weight distribution, foot position in the very same analysis one goes through when handling an axe or a tennis racket.

Some mornings start with sharpening and care of the knife: don't touch metal stuff like drying racks with your knife; never drag your blade across a surface to scoop food – flip your blade over and use the dull side; learn the strengths of the knife. Sharpening carries me to a Zen and peaceful place. We hone and polish the blades through various grits then finish on a leather strop. You can see sharp (No my fellow geeks, I didn't say "C#" which is a programming language. And to my music teacher friends, I still didn't say "C#"). In the right light, you can see the sharpness of your blade.

Caring for the blade means also caring for the sharpening stones and the honing leather. It bring pride.

We sharpen our chef's knives. We sharpen the lovely boning knife and the paring knife. As we worked, we studied the various grips we took on the knife or knives. We'd roll a blade shifting the blade from the wrist-side to the finger-side (seemingly backwards). We'd hold a knife short with a hand up on the bolster and even a finger down the back of the blade. Each movement of the knife and our grip reflected the intended cut.

For New Year's Eve, we prepared a crown roast of lamb. This dish is rather posh and fussy to prepare. So fussy, most would order a crown roast from a professional butcher. We had four racks of lamb. Two from the 2020 lamb and two from this year's lamb. I save them for special dinner parties. That's not every-day food. With the pandemic, there were no parties and no honored guests. The fancy lamb remained frozen.

We pulled two racks from the freezer, each rack contained seven ribs. One possible presentation of racks of lamb ribs involve roasting both then tenting the rack with the bone tips interlaced. Looks lovely. The "oohs and aahs" go for the quality of the roast.

We opted for the crown roast. You made a small "S" shaped cut at the base of each rib bone, near where the backbone was. This "S" shaped cut permits the rack of ribs to form semi-circle. Then to enhance the look, one then "Frenches" the tips of each rib bone. You remove the meat, connective tissue, and fat on the top 3 centimeters of the bone. I observed that we used three knifes as we worked. Chef's knife, the boning knife, and a dull kitchen table knife – like a butter

knife. The dullest knife scraped each bone tip baring it and cleaning it. Our hands rotated the knives in our hands. When done with the knife-work, we tied the ends together forming a circular and crown-like presentation of the 14 ribs.

We cleaned and cared for the knifes – well except for that kitchen table knife. That poor thing got tossed into the dishwasher.

Each section of each knife had a purpose. Each grip anticipated the next movement.

During the holiday, we cut through potatoes for latke. We served up a duck and a turkey. We prepared fish for the seafood chowder.

None of this is a surprise to the musician, the cook, the plumber, the wood worker, the stone mason, the lawyer, the surgeon, or anyone who has mastered a craft. Don't hand a surgeon a number 11 blade when a number 10 is requested.

What is the strength of my chef's knife? It is multi-purpose utility knife. It is pretty good at a lot of jobs. It has the right size and weight for cutting and chopping veg. It can work around a bird or a larger cut of meat. The primary strength of my chef's knife is the edge. It is the sharpest knife.

What is its weakness? Oh, lots of things. It doesn't maneuver around bone like my boning knife. I can just pick and flick at connective tissue like can with my boning knife. Carrots and potatoes stick to the side of the blade. And after this season of holiday feasts, I observe that I have sharpened my German steal to an angle that is too acute or too fine. The factory grind provides Rachel's knife with a more stable edge. My edge has microscopic chips.

What about off-book uses? I love smashing garlic under the flat of the knife. I grind garlic to a paste using the dull side of the blade. I smash a garlic head with the pommel or butt of the knife.

What about off-book uses for my favorite axe? Too often with these bad knees I use the damn thing as a walking stick. It is a terrible walking stick and a not very good hammer, but who want to carry a lot of tools.

What about off-book uses for our Oracle database, our programming language PL/SQL, or our rapid development framework called APEX? I am stumped. I have used SQL for doing working-out basic math. Where others may open a spreadsheet, or calculator, I am happy pushing formula through Oracle. I have used it as calendar to inform me how many days exist between now and, say, my birthday. Heck, we used an Oracle database to upload and manage 400,000 PDF documents to AWS S3 (cloud-based storage). I don't think that was ever in the vision of the original designers.

Lesson Two: Take Care of Your Tools

The axe I use for trail maintenance is stored in leather guard. The edge is shiny and reasonably share. Not like the sharp I have on the knives. The care of the knives include dozens of tricks to

protect the sharp edge. Never scrape with the sharp edge, never whack at bone with the sharp edge. When done, I wash with care then dry.

In November and December of 2020, we learned that major software vendors suffered significant breaches from bad actors. Solar Winds and Microsoft both got hit. For those of us that watch and track these behavior, the hacks bring us concern. Solar Winds is a suite of tools used by IT professionals to monitor their own tools. Using Solar Winds constituted good practice for decades. You'd get graphs of performance and notification of aberrant events. It's the tool for the people you depend on to keep your technological infrastructure safe and functional. My partner and I debated bringing it in to our infrastructure, but we found AWS met our needs without added costs. It is really the first time in decades, I haven't had Solar Winds (or its competitor) deeply ingrained in our operations.

But we do use Microsoft in all of the same ways that Microsoft customers got impacted. That's close. Frankly it is all close. Software and IT people constantly worry and mitigate against these threats. It is very real work and part of every day's thoughts, part of every week's effort, and part of every annual plan.

Taking care of your tools includes affirmatively making the steps to improve the security posture of our systems. Like washing my knife, then drying with a soft cloth, we patch our system, monitor our servers, maintain a constant vigil for the bad-schtuff that goes on.

There was a series of days in 2018 when our graphs showed more and more red. Red is bad. Somebody, something started impacting our customers and our performance. It started the day that the Government of Puerto Rico published the link to our software on their website. We got hit and hit hard from eastern Europe and south Asia. We worked our way through it step by step, minute by minute. Had we not been prepared and taken care of our tools; we would have been swamped.

The adage holds true: take care of your tools and your tools will take care of you. The process involves that careful, disciplined, and affirmative effort – nearly daily, within each action really.

Lesson Three: Develop skills, practices, and strengthen your abilities regularly

Stephen Covey wrote a book about some pretty-good habits, amongst them was "sharpening the saw". This pole-star phrase reminds us to seek continuous improvement and renewal professionally and personally. I have nothing to add to Covey's statement.

Some portion of every day I find a way to learn something new and explore – lending to my love of on-line instructional videos. The community of Oracle APEX developer, people scattered around the globe found a way to rally professionally on Twitter, on Slack, and on a website called APEX.WORLD

We share with each other. And we observe each other as peers. We learn from each other. And as such, we appreciate each other professionally. I am thankful for this community.

Lesson Four: Know the difference between the right tool, a good-enough tool, and the wrong tool

In any moment, the wrong tool serves as the right tool and often it doesn’t matter. My favorite axe helps trim limbs, make pointy tips and then helps me drive them into the ground as fence stakes.

When programming, we often face similar problems. My chosen programming languages have weaknesses. A few years ago, when working with JSON data in an Oracle database, the process of converting and rendering the data useful required boat loads of code. People informed me I was using the wrong tools – it was inefficient. In fact they were telling me that I was wrong. Not so. Oracle’s PL/SQL represented a good-enough tool during those days. And I stood at the leading edge of their technology. A wave of continuous development and innovation let me feel like I was riding a wave. In time, my good-enough tool became the right tool. Thank you, Oracle.

Regardless, it was the right tool for me. Oracle PL/SQL is a language I have used since first learning Turbo Pascal in the mid-1980s. I am pretty good at it. Learning new languages when having absolute mastery of one is not hard.

That’s not really the point of the statement within the framework of a team. We know our strengths. We know our missions. We know the objectives. Collaborating with a teammate this week, I asked this human write a narrative of a complex process. A word processor such as Microsoft Word is ideal at this task. I am writing this episode in Word. It is the right tool.

Throughout the work, the teammate tried to communicate programming instructions in this same tool. Microsoft Word behaves like an idiot. It lays squiggly lines informing us of typos. It wants to enforce proper formatting. It hates our capitalization. Word makes pretty code look entirely ugly. Most non-programmers think code looks ugly anyway and here is old Word reinforcing the wrong message.

Frankly, we don’t have a good tool for blending programming code and narrative text. They are incompatible. So, Word become the good-enough, hate-it-the-least option.

We all get to ask, what is the right tool? What will it take to learn it, find it, use it? As a team, we evaluate these questions together and rather constantly.

I entitled this chapter: “How to Hold an Axe” letting axe stand in for 8,000 years of tools, the phrase we tend to use is: “Lean into PL/SQL” or “Lean into APEX”. The phase means to remind the team to learn, to actively study the tools we use. Find the optimal and most power ways of using these tools made by other tool smiths. Know the weakness and the strengths.

Asking a team mate to trust their tools serves as a challenge. First, to know if you make an optimal and strong decision. Or, if not, why not.

Closing

Recognizing a team that functions well as a team can be a joy. Watching athletes coordinate efforts while pushing a ball down the pitch closer to a goal fills us with joy. The non-verbal communication follows drills and discussions and the development of a team-focused verbal shorthand.

We all know how to encode complex messages in quick phrases with co-workers. You never really want to start each message from scratch. When asking a tool smith if they understand the power of their tools, we also see new opportunities. Maybe we need to create new tools? Maybe we should improve the tools we have? Maybe it doesn’t matter at all.

The goal of any team is to score points and move closer to objectives. Whether filming a movie, writing software, pushing a ball across a field, or trimming tree limbs along a mountain trail in Vermont. Streamlining communications and developing verbal shorthand stands as a common element with strong teams.