An Open Letter to Knowledge Base Writers: Change Your Practice to Experience Base Writing

A recurring issue in working in technical and customer support roles is the over-reliance on Knowledge Based articles to guide new technicians and brand ambassadors through the process of operating a user-interface such as a software repair or an e-commerce order tracking request. While Knowledge Base articles are written by seasoned support, knowledge cannot be imparted without experience.

Experiential-based learning has a history of utility and is not new. However, with the advent of self-service website access for both customers and support roles, Knowledge Base information is increasingly less relevant as an initial resource for trainees. Practically, a Knowledge Base contains a mix of tutorials as well as an overview of a given procedure and or policy. Much of the knowledge gained in training, depending on the level of hands-on-learning, is not quickly absorbed as a more in-depth walk-through. I also believe a job-requirement to writing a Knowledge Article is to have a love for the English language. I'm kidding. Sort of.

Being detailed as possible in a procedure is not making anything too easy for a trainee. Being wordy isn't what is being requested. Rather, detailing options in a procedure, and how to rectify an error is what I mean by being detailed. Ambiguity in instruction exists because Knowledge Base steps are terse. Online Forums and their threads, by analogy, can be like a good knowledge article. A forum may be a place where some chat idly, but it is also a place where many specific actions or problems can be elucidated and elaborated. Not everyone needs an explanation why some action must or must not be taken in a procedure. But there is no shortage of disk space for elaborating on something on the off-chance that something does need to fixed. The format of the forum, however may not be applicable to an Experience Base. What is applicable, is the concept of a thread- that is, the ability to branch off specific pathways in a procedure like a flow chart and to interactively access and find the specific error's solution or action needed.

There is much ado about standardization. So much can be standardized, it is uncanny at times. I thought how weird it is that I am writing a post advocating for the standardizing of obscure knowledge bases into standard-capable instructions, but it's really about ensuring uniformity where uniformity is needed.

An Experience Base may come in the form of an animated timeline or sequence of events in a process. A Step-by-Step instruction set can also be very effective at communicating each necessary step in a procedure. What is lost between the lines in a Knowledge Base instruction, is when a common error or mistake is encountered, and there is no mention of those common mistakes. This happens when a trainee is attempting to replicate a procedure for the first time. This is ameliorated with the granularity of a more frame by frame walk-through of each step in an instruction.

Auditory learning and Visual learning are both important aspects of a training experience. A knowledge base may be entirely textual or entirely auditory, but they can both be insufficiently instructive if they do not address potential errors that may be encountered. While making mistakes can often be a learning and positive experience, a support department may not have the staff to assist with troubleshooting every possible error that may be encountered. Nonetheless, an Experience Base allows for a readily interactive play-by-play resource for those who are just learning the basics.

Knowledge can be imparted only to those who already have the same knowledge. In a way, Knowledge Bases are more of a reference for those who already know a procedure but are merely accessing

information, rather than a procedure. In a way, it really is only useful for those who are in the know, who know-how, but may have forgotten a minor detail. A more accurate term for Knowledge Bases would be Information Bases or Directories.

A few Google searches resulted in Experience Bases which I think are good resources:

https://serc.carleton.edu/introgeo/enviroprojects/what.html

"Experiential learning is a well-known model in education. Kolb's Experiential Learning Theory (Kolb, 1984) defines experiential learning as "the process whereby knowledge is created through the transformation of experience."

https://www.igi-global.com/chapter/contextualization-decision-making-decision-support/11245

"Experience Base: It is a collection of practices that are expressed as chunks of knowledge, reasoning, and context. A contextual graph is an element of an experience base for a given problem-solving situation. This is the (future) knowledge base of intelligent systems, which are of higher granularity than previous knowledge bases in expert systems."

A related but important aspect to Knowledge Bases is User Experience (UX). The same principles that are important to a customer are also important to the support role. A support agent is also a user of information systems (IT), and must navigate through a known procedure or an unknown procedure. The latter may be known to others, and thus can be replicated best through an Experience Base. User Experience is relevant to changing how Knowledge Bases are written and accessed as a resource.

A very well known fruit company is known to employ the following practice. Some details have been changed to protect their anonymity and avoid promotions. Let's say a product such as a new cell phone, is packaged in a box in a very standardized way- so that each consumer opens the box and sees a quick start guide, before they open a smaller package under it that contains a plug and a phone. The goal of user experience is to try to find the most intuitive actions that a common consumer would do when they open a box. If there was a need to charge a phone for at least 20 minutes before powering it on, the quick start guide may instruct the user to plug phone in for 20 minutes.

I have not bought a new phone in a while, and I admit I have not read the quick start guide, but my point is, if a manufacturer wanted a consumer to read a quick start guide or follow the instruction manual to ensure they did not miss an important step, it would be placed in an easy to spot place, such as the first item inside the box after opening it. Unless, of course, you open the box from the bottom by accident, like me. A company wants the act of opening a box This Side Up to be part of the User Experience because they are counting on you to see the same thing as someone else opening the box This Side Up, in hopes that you don't miss the same thing. A good User Experience makes all the difference in ease of use. Advocates of User Experience are aware that two people might not see or think the same thing when they open the box of the same pristine condition and This Side Up, but it at least attempts to address the issues of user experience by giving it a name, a thing, a belief, a way of life.

This concept can be applied to both advanced procedures and beginner tutorials. All advanced users begin somewhere, therefore the assumption that everything is already known at a higher level is not true the more advanced one is. There are always new software updates and browser formats that change

the way users (i.e. you and the customer) see an instruction, and interpreting a step verbatim may ironically have more than one meaning.

One could analogize the user experience to computer science and science in general. The scientific method is something I have some experience with. After I graduated from college, I had a Bachelors of Science in Molecular & Cell Biology. I worked in a lab for a couple years with microbes. A single experiment requires a control group and a test group. The scientific method may involve one or many variables. To accurately test a hypothesis, one has to test one variable at a time. If more than one variable changes from a control group and a test group, the result of the experiment cannot be considered affected by a single variable until each variable is tested separately in a controlled environment.

So why am I bringing this up? Well, I admit I was not a very good biologist. I could not keep track of all the variables! So I thought computers might be a tad bit simpler, because two new computers, with the same operating system, browser version, same amount of RAM, hard disk space, computer speed, and video memory would perform the same, right? Well, it turns out, sometimes the answer is no. And as a computer technician spending years troubleshooting computers that appear to be the same, I know many assumptions can be made of how something will perform, but I will not make those assumptions.

So, it does bother me when I can't figure out why two computers of similar specifications operate at different performance levels. The key to making equal products, whether it is software, or hardware, or a knowledge resource, is a clearly articulated Experience Base that can be replicated by a user.

There are no wrong questions when troubleshooting. A customer might see "123 Main St" when they are trying to type in their address on a form. They call customer support, and tell the support representative, "My address is wrong. It says "123 Main St". The support agent thinks, "where did I see that? I don't know how to fix this though." The support agent thinks back, and remembers and says, "Oh! That is just an example, you can click in the box and that will disappear." The customer says, "Oh, thanks!" I see it disappeared now."

You may know what I'm talking about here- the gray suggestion text before you begin typing in an online form. To you, you think it's already common knowledge that those letters are not written in stone. It is our intuition only as experts that we click inside the box and that text disappears. The customer, being overly cautious that they are making a purchase for hundreds or thousands of dollars, wants to make sure they are not buying an expensive product that will arrive at 123 Main St. And that means they don't want to click anywhere, even on the box that says 123 Main St. An expert cannot assume what a customer doesn't know.

These are just some simple examples, maybe frustrating to others, but this is where I believe that Experience Learning is important at every stage of support. It is a humbling experience to be a trainee. I do not want to suggest that I know more about what a Knowledge Article should contain, because I don't. What I do know is *how* it should or could be written, and it should or could be written as if the person was detailing every common issue they had when they first experienced it. I also understand they might not have encountered the same issues as I. Detailing the same errors they had or heard about from co-workers will help those reading it because there is never enough supervisors to help every one with a simple little procedural action that can be quickly accessed in a resource. That helps the bottom

line of any company, because most people like me do not want to frequently rely on others when there is already a resource that doesn't cost another human salary to host.

There is another aspect to user experience- the organization's situational awareness. Not all bugs or trending issues are known to upper management and vice versa, which is why workforce management is designed to help identify user issues before they become a trend. An essential component to a website's functionality is its system administrators, and network administrators. Then there is the website developer and host, who may be the same person. Sometimes a software or infrastructure as a service may be licensed and the developers do not work for the organization. Reporting website issues need not be a complex chain of command. I have worked in organizations where there was a close interaction with website developers and the call center agents who use the CMS applications everyday and know the good practices and the bad. The good practices are when the floor is able to directly forward their website or tool issues directly to the maintenance or developers, rather than it needing to be "approved" or filtered through multiple levels of management. While many organizations rely on outsourced software services for their IT support as well as web development, it should never be assumed that the floor level call center agents are unable to understand all the technical reasons for a website being designed as such or for suggesting a website cannot have any flaws.

A unidirectional, vertical organization may rely on software monitoring tools more than positive feedback channels that are bi-directional or multi-directional. This is not necessarily a bad thing, as it can improve efficiency in operations that have already been optimized with a well-trained crew. Having the ability to report errors and issues in an organization that have various elements of untrained or under-optimized components and resources requires more feedback mechanisms for improving situational awareness. Much like artificial intelligence, situational awareness is an organizational capability for implementing dynamic changes based on any single agent. An important bug can be identified by any particular user, and therefore a channel for reporting and identifying known and unknown issues is crucial to adapting to a change of operations.

As call center agents most frequently experience the day-to-day issues with a website, it is odd that not every call center has a resident "bug-fixer" team working on the floor with the agents, asking them if they are experiencing any technical issues. Sometimes I wonder how many times a website is designed by someone on contract, leaving no documentation as to how everything was written, making it indecipherable to the next contractor who is unable to use a standardized programming language that has a fixed meaning and is able to translate any jargon back to several levels of management down to the floor where the agents need a quick fix, or need to wait for a server update.

Many organizations do have and actively pursue their bugs fixes but the information overload with knowledge bases can make software support appear invisible or in a far off place. Having an open-door policy with a department contact directory that is actively encouraged for bi-directional communication for technical and instructional issues is a better way to for an organization to maintain internal bonds and bypass much of the red-tape that exists in a large, multi-faceted organization.

Quixote, LLC