

Tacit knowledge

Tacit knowledge is knowledge based on experience and observation. There wasn't some law or procedure handed down from on high. It's just the way that things are done. It's something that is immensely powerful because it's relevant. It can be directly applied to the activities that will need to be done in the future.

One example of tacit knowledge comes from a buddy of mine who does handyman work for a living. He and I like to work on projects together because we have great conversations. Recently we were building drum raisers for the church in my garage. He was constantly sprinkling all sorts of bits of knowledge. We built the frame before attaching it to the decking rather than attaching the frame to the deck one board at a time. Why? The answer was that getting it all to line up works better that way. How does he know? He's done this general sort of thing (building cabinets and platforms) enough that he just knows.

He knew all sorts of things, like making sure that the crown of the board, the slight curvature that's normal for all wood, are all going up. Why? Because when load is applied it will tend to bend it down to a flat or neutral position.

Another example was something I learned in college several years ago. It was about loop structuring. An instructor of mine suggested a loop structure that did an initial read of a record before the top of a loop. The structure I was using at the time performed the read just after the top of the loop. The result was a loop that had a big if block inside of it. When she suggested the new structure of putting the read statement before the loop, I asked why? Her answer was "It just works out better that way." She knew from experience that the structure worked out better if you performed the read before the start of the loop (and at the end of the loop.)

Tacit knowledge is the basis of apprenticeship programs. You learn how to do something through gaining experience with someone else who already has more experience than you. This approach illuminates small subtle ways of making a better product or working more quickly which would be hard to capture into a formal textbook on the subject.

Some folks just know how to make things work from experience. You may be hard pressed to get them to explain their knowledge to someone else or even to write it down but the knowledge of how things should be done is there and it's built upon experience.

Explicit knowledge

Explicit knowledge is, on the other hand, knowledge that can be quantified. It can be written down and clearly communicated to another human being. It's tangible. There's no need to gain experience. It's something that has been converted to a rule.

For instance, one may observe that apples fall from trees and know how to shake the tree to get them – tacit knowledge – and they may also know that gravity causes all objects of mass to come together. The observations in the first case, tacit knowledge, were converted to a law in the second case, explicit knowledge.

Explicit knowledge is the type of knowledge conveyed through articles, books, seminars, and video presentations. There is no need to have direct experience with something to have explicit knowledge about it. This is one of the criticisms of college students who are just graduating. They have a lot of "book knowledge" (explicit knowledge) but lack real world experience (tacit knowledge). Clearly then we know that no matter how great the explicit knowledge it can not always replace tacit knowledge. On the other hand since the Gutenberg printing press explicit knowledge is transferred at a much more rapid rate. You are reading these words and learning more about the different kinds of knowledge because I've been able to convert tacit knowledge of how things work into a set of rules that you can use.

Understanding the conversion process

Thomas Kuhn, a philosopher of science, argues that science comes in cycles of "regular science", "crises", and "revolutionary paradigm shifts". This model demonstrates the relationship between explicit knowledge and tacit knowledge as it relates to the development of new science. We start with tacit knowledge which is the core of the scientific method.

The first step in the scientific method is to observe – in other words to develop experience. It is from there that we hypothesize how the thing we're observing works. The next step is to test the hypothesis and finally if incorrect we test again. The output of the scientific method is a nugget of explicit knowledge – a tested hypothesis or theory about how the world works. This explicit knowledge can be easily and quickly communicated between humans without the need for extensive experience. Kuhn, however, realized that this is not the end. Eventually flaws, gaps, or errors in the hypothesis will arise. This will create a crisis where the explicit knowledge comes into question. This tacit knowledge, refined by the focus brought by the explicit knowledge, begins to challenge the explicit knowledge under which the world seems to operate. Eventually, the pressure brought to bear by new tacit knowledge creates a revolutionary paradigm shift that creates a new explicit knowledge that becomes the basis for all new tacit knowledge to become known. This process happens over and over again in small ways and in big ways. Simple things like the discovery of quantum physics radically changed our view Newtonian physics. There was a crisis because we were trying to apply a hypothesis – a thoroughly tested one – to an area where the hypothesis didn't hold true. The net effect is that both kinds of knowledge are necessary; however, it is explicit knowledge that is easier to convey.

Converting knowledge

Converting tacit knowledge into explicit knowledge is a real trick – a real leap of faith. It takes someone with vision to see the line that fits between all of the dots of observations and experience. It's a bit of magic not unlike the process of converting requirements into design or a caterpillar converting to a butterfly.

The task of making the conversion isn't just as simple as making the connections. There are some issues with the way that we process information and the makeup of our brains which hinders our ability to make these conversions.

In [Blink](#), Malcolm Gladwell raises some interesting points. In Chapter 4 "*Paul Van Riper's Big Victory*", Gladwell points to the work from Jonathan Schooler who

researches verbal overshadowing. The point of his study is to understand why we're able to recognize faces quickly until we are asked to describe them. His work explores how the verbal part of your brain interacts with the mental parts of the brain such that it's both difficult to describe pictures as words and how the process of even attempting this seems to reduce the fidelity of the original image.

In Chapter 5 "*Kenna's Dilemma*", Gladwell revisits the topic by sharing how we adjust our preferences to plausible-sounding reasons. Jonathan Schooler did an experiment with Timothy Wilson. In this experiment experts and college students were asked to subjectively describe which strawberry jam they liked best. The experts and the college students by-and-large agreed. That is unless the students were asked to quantify their rankings at which point all correlation between the experts and the previous students disappeared. This illustrates the problem with converting knowledge from tacit to explicit. When asked to create an explicit perspective on the jam they created plausible sounding reasons for liking one jam over another and then adjusted their perceptions to match the plausible-sounding reasons. In other words, they changed the way they remembered the jam to match what they described.

Gladwell continues and points out that experts have somehow overcome the limitations that seem to plague the less expert in part by creating a more precise understanding of how to rate foods. The structures that the students came up with didn't have the fidelity necessary to accurately portray their experience.

In the software development world, a world of creativity, it's sometimes difficult to get to the true best practices, the rules, the guidelines and techniques which lead to good software development. Undoubtedly there are issues such as different languages which subtly shift the best practice and make it more difficult to identify. However, at its core the issues remain the same and we must learn to convert our knowledge in requirements specification into an actual requirement.

Getting the conversion done

So if expert food testers can quantify the differences in jams with no real difference in the outcome, so too can we convert tacit knowledge of how software can and should be

developed into the explicit knowledge that we need to be able to communicate with the entire development team – or the entire organization.

The food testers had very precise scales and very precise attributes they were rating. So too can we clearly define the characteristic behaviors that must be present for good software development. We can also clearly define the metrics where by those behaviors are measured.

Instead of just saying that you "keep in touch" with developers it's possible to define an "informal communications" attribute which can be measured. Once the attribute is defined it's possible to define points on the continuum of that attribute. From this a workable understanding can be created. For instance, a 0 on a 0 to 100 scale might mean that you never talk to the other team member." A 100 on the scale might be that you share office space with them.

The actual value and the specific attributes that you select for your organization are not necessarily important, but the ability to quantify what you have is important. You may learn, for instance, that in order for your team to work well together they need to have a certain level of informal, non-structured interactions.

The explicit knowledge becomes a periodic set of events that support the underlying observations. The explicit knowledge is that every quarter there should be four hours of time set aside so that team members can communicate without structure. This clearly misses the underlying knowledge of how teamwork is developed and the level of team work that is required for the group to operate effectively – however, this tacit knowledge isn't necessary for most cases. All people need to know is what to do – to create informal communications – not necessarily why it works.

Breaking the rules

One of the interesting observations that we take from Kuhn is that sometimes it's necessary to break the rules – to go outside of the explicit knowledge. No where is that more true than running sound for a live event. There are a set of rules – or guidelines – for how to setup things so that they sound good. These rules are extremely effective for

those who don't have a great deal of experience. They simplify the process and make it manageable.

However, an experienced sound person often breaks (or severely bends) the rules to suit their needs. While you may "never" want to use a large diaphragm condenser microphone on a noisy stage, this may be precisely what an experienced sound engineer does. He does this because he has the tacit knowledge that contains a more precise understanding of how to make this work than the explicit knowledge in the rule. So go make some explicit knowledge – and then go break the rules.

Source: http://articles.techrepublic.com.com/5100-10878_11-6117372.html