Most seasoned professional software engineering teams probably understand the immense value of [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) in their jobs, but it seems to me that the concepts of [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) isn't used much outside of software engineering, even when [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) has existed for way more than a decade already, which is quite a pity for me.

So how [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) can be used outside of software engineering? Let's show it using the following example:

1. You've a front-line customer service job(sitting on a booth with the customer on the other side while you're using a computer to do the work) which demands you to strictly follow a [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) covering hundreds of cases(each of your cases will be checked by a different supervisor but no one knows who that supervisor will be beforehand), and the most severe [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) breach can cause you to end up going to jail(because of unintentionally violating serious legal regulations)
2. You've to know what cases should be handled by yourselves and what have to be escalated to your supervisors(but no one knows which supervisor will handle your escalation beforehand), because escalating too many cases that could've been handled by yourselves will be treated as incompetent and get yourselves fired, while handling cases yourselves that should've been escalated is like asking to be fired immediately
3. As the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) is constantly revised by the upper management, it'll change quite a bit every several weeks on average, so the daily verbal briefing at the start of the working day is always exercised, to ensure all of you will have the updated [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), as well as reminding what mistakes are made recently(but not mentioning who of course)
4. Clearly, a [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) of this scale with this frequency and amount of changes won't be fully written in a black and white manner(it'd cost hundreds of A4 papers per copy), otherwise the company would've to hire staffs that are dedicated to keep the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) up to date, in which the company will of course treat this as ineffective and inefficient(and wasting tons of papers), so the company expects **EVERYONE**(including the supervisors themselves) to **ALWAYS** have **ABSOLUTELY** accurate memory when working according to the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)
5. As newcomer joins, they've about 2 months to master the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), and senior staff of the same ranks will accompany these newcomers during this period, meaning that the seniors will verbally teach the newcomers the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), using the memory of the former and assuming that the latter will remember correctly

Needless to say, the whole workflow is just asking for trouble, because:

1. Obviously, **no one can have absolutely accurate memory**, especially when it's a [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) covering hundreds of cases, so **it's just incredibly insane to assume that EVERYONE ALWAYS have ABSOLUTELY accurate memory** on that, but that's what the whole workflow's based on
2. **As time passes, one's memory will start to become more and more inaccurate gradually**(since human's memory isn't lossless), so eventually someone will make a mistake, and the briefing on the upcoming several days will try to correct that, meaning that the whole briefing thing is just an ad-hoc, rather than systematic, way to correct the staff's memories
3. Similarly, as newcomers are taught by the seniors using the latter's memory, and **human communications aren't lossless** either, it's actually unreasonable to expect the newcomers to completely capture the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) this way(**because of the memory loss of the seniors, the information loss in the communication, and the memory loss of the newcomers, which is essentially the phenomenon revealed by** [**Chinese whipsers**](https://en.wikipedia.org/wiki/Chinese_whispers)), even when they've about 2 months to do so
4. As each of your cases will be checked by a different supervisor and no one knows who that supervisor will be beforehand, and supervisors will also have memory losses(even though they'll usually deny that), eventually **you'll have to face memory conflicts among supervisors, without those supervisors themselves even realizing that such conflicts among them do exist**(the same problem will eventually manifest when you escalate cases to them, and this includes whether the cases should actually be escalated)
5. Therefore, **overtime, the memories on the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **among the staff will become more and more different from each other gradually, eventually to the point that you won't know what to do as the memory conflicts among the supervisors become mutually exclusive at some parts of the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure), meaning that you'll effectively have to gamble on which supervisor will handle your escalation and/or check your case, because there's no way you can know which supervisor will be beforehand

Traditionally, the solution would be either enforcing the **ridiculously wrong assumption** that **EVERYONE** must **ALWAYS** have **ABSOLUTELY** accurate memory on a [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) worth hundreds of A4 papers even harder and more ruthlessly, or hiring staff dedicated to keep the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) up to date, but even the written version will still have problems(albeit much smaller ones), because:

1. As mentioned, while it does eliminate the issue of gradually increasing memory conflicts among staff overtime, having a written version per staff member would be **far too ineffective and inefficient**(not to mention that it's a serious waste of resources)
2. When a written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) has hundreds of A4 papers and just a small parts of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) change, those staff dedicated to keep the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) up to date will have to reprint the involved pages per copy and rearrange those copies before giving them back to the other staff, and possibly highlight the changed parts(and when they're changed) so the others won't have to reread the whole abomination again, and **this will constantly put a very heavy burden on the former**
3. Because now the staff will rely on their own copies of the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), **if there are difference among those written versions, the conflicts among the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **implementations will still occur**, even though now it'd be obvious that those staff dedicated to keep the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) up to date will take the blame instead(but that'd mean **they'll ALWAYS have to keep every copy up to date IMMEDIATELY**, which is indeed an extremely harsh requirement for them)
4. As it'd only be natural and beneficial for the staff to add their own notes onto their own copies of the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), when those written versions get updated, **some of their notes there can be gone because those involved pages will be replaced, so now those staff might have to rewrite those notes**, regardless of whether they've taken photos on those pages with their notes beforehand(but taking such photos would risk leaking the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)), which still adds excessive burden on those staff
5. As you're supposed to face customers at the other side of the booth while you're using a computer to do the work, it'd be detrimental on the customer service quality(and sometimes this can lead to the customer filing formal complaints, which are very major troubles) if you've to take out the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) in front of the customer when you're not sure what to do in this case, even though it's still way, way better than screwing up the cases

Combining all the above, that's where [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) for the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) can come into play:

1. Because now the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) is a soft copy instead(although it still works for soft copies without [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control)), this can be placed inside the system and the staff can just view it on the computer without much trouble, since the computer screen isn't facing the customer(and **this largely mitigates the risk of having the staff leak out the written version of the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure))
2. Because the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)'s now in a [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control), each staff will have its own branch or fork of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure), which can be used to drop their own private notes there as file changes(this assumes that the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) is broken down into several or even dozens of files but this should be a given), and **their notes can be easily added back to the updated versions of the files having those notes previously added**, by simply viewing the diff of those files(or better yet, those notes can also be completely separate files, although it'd mean the staff have to know which note files corresponds to which [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) files, which can be solved by carefully naming all those files and/or using well-named folders)
3. Because the written version of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)'s now centralized in the system(the master branch), **every staff should've the same latest version, thus virtually eliminating the problems caused by conflicts among different written versions from different staff members**, and the need of the dedicated manual work to ensure they'll remain consistent
4. Clearly, the extra cost induced from this [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) application is its initial system setup and the introduction to newcomers of using [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) at work, which are all **one time costs instead of long-term ones**, and **compared to the troubles caused by other workflows, these one time costs are really trivial**
5. Leveraging the issues and pull requests features(but using blames as well might be just too much) in any decent [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control), **any staff can raise concerns on the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure)**, and they'll either be solved, or at least the problems will become clear for everyone involved**, so this should be more effective and efficient than just verbal reflections towards any particular colleagues and/or supervisors on difficulties faced(if called for, **anonymous issues and pull requests can even be used**, although it'd seem to be gone overboard)

So the detailed implementation of the new workflow can be something like this:

1. The briefing before starting the work of the day should still take place, as **it can be used to emphasize the most important** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **changes and/or the recent mistakes made by colleagues**(as blames not pointing to anyone specific) in the [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control), so the staff don't have to check all the recent diffs themselves
2. Whenever you're free, **you can make use of the time to check the parts in the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **of your concern from the computer in your booth**, including parts being unclear to you, recent changes, and even submit an anonymous issue for difficulties you faced on trying to follow those parts of the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)(or you can try to answer some issues in the [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) made by the others as a means of helping them without having to leave your booth or explicitly voice out to avoid disturbing the others)
3. When you're facing a customer right in front of you and you're unsure what to do next, **you can simply ask the customer to wait for a while and check the involved parts of the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **without the customer even noticing**(you can even use issues to ask for help and hope there are colleagues that are free and will help you quickly), thus minimizing the damages caused to the customer service quality
4. To prevent the [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) from being abused by some staff members as a poor man's chat room at work, **the supervisors can periodically check a small portions of the issues, blames and pull requests there as samples to see if they're just essentially conversations unrelated to work, and the feature of anonymity can be suspended for a while if those abusers abuse this as well**(if they don't use anonymity when making those conversations, then the supervisors can apply disciplinary actions towards them directly), but don't always check all of them or those supervisors would be exhausted to death due to the potentially sheer number of such things
5. Of course, **you still have to try to master the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **yourselves, as the presence of this** [**DVCS**](https://en.wikipedia.org/wiki/Distributed_version_control)**, which is just meant to be an AUXILIARY of your memory, doesn't mean you don't have to remember anything**, otherwise you'd end up constantly asking the customer to have unnecessary waits(to check the [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)) and asking colleagues redundant questions(even with minimal disruptions), causing you to become so ineffective and inefficient all the time that you'll still end up being fired in no time

Of course, it's easier said than be done in the real world, because while setting up a [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) and training new comers to use it are both easy, simple and small tasks, **the real key that makes things complicated and convoluted is the willingness for the majority to adopt this totally new way of doing things**, because it's such a grand paradigm shift that's wholeheartedly alien to most of those not being software engineers(when even quite some software engineers still reject [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) in situations clearly needing it, just think about the resistance imposed by the outsiders).

Also, there are places where [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) just isn't suitable at all, like emergency units having to strictly follow [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure)s, because **the situations would be too urgent for them to check the** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure) **in** [**DVCS**](https://en.wikipedia.org/wiki/Distributed_version_control) **even if they could use their mobile phones under such circumstances, and these are some cases where they do have to ALWAYS have ABSOLUTELY ACCURATE memories**, as it's already the least evil we've known so far(bear in mind that they'd already have received extensive rigorous training for months or even years before being put into actions)

Nevertheless, I still believe that, **if some big companies having nothing to do with software engineering are brave enough to use some short-term projects as pilot schemes on using** [**DVCS**](https://en.wikipedia.org/wiki/Distributed_version_control) **to manage their** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure)**s of their staffs, eventually more and more companies will realize the true value of this new ways of doing things**, thus causing more and more companies to follow, eventually to the point that this becomes the norm across multiple industries, just like a clerk using MS Office in their daily works.

To conclude, I think that [**DVCS**](https://en.wikipedia.org/wiki/Distributed_version_control) **can at least be applied to manage some** [**SOP**](https://en.wikipedia.org/wiki/Standard_operating_procedure)**s of some businesses outside of software engineering, and maybe it can be used for many other aspects of those industries as well**, it's just that [SOP](https://en.wikipedia.org/wiki/Standard_operating_procedure) management is the one that I've personally felt the enormous pain of lacking [DVCS](https://en.wikipedia.org/wiki/Distributed_version_control) when it's obviously needed the most.