How We Went Remote

OSCON 2014

@vmbrasseur

Before I really get into today’s presentation, please take a moment to check all your devices and make sure they’re set to silent.

ALSO: Please save your questions for the end. I’ve left plenty of time for those.

https://www.zotero.org/groups/oscon_2014-remote/items

Also, most everything I'm about to tell you comes from my own experience. I've also collected a ton of telecommuting-related articles and resources in this Zotero group. Don't worry if you don't get the URL now. I'll show it to you again at the end.

Introductions

VM (Vicky) Brasseur

@vmbrasseur

I'm your presenter VM Brasseur, but because we're all friends here you can call me Vicky. You can find me on Twitter @vmbrasseur.

I'm a freelancer, doing technical business and management consulting and open source strategy and tactics through my company shoeless consulting. On a more 1:1 level, I also do career coaching for tech professionals.

Poll time!

Let’s start with some quick audience participation.

Currently telecommuting?

By a show of hands, how many of you out there already telecommute part- or full-time?

Want to telecommute but not allowed/possible?

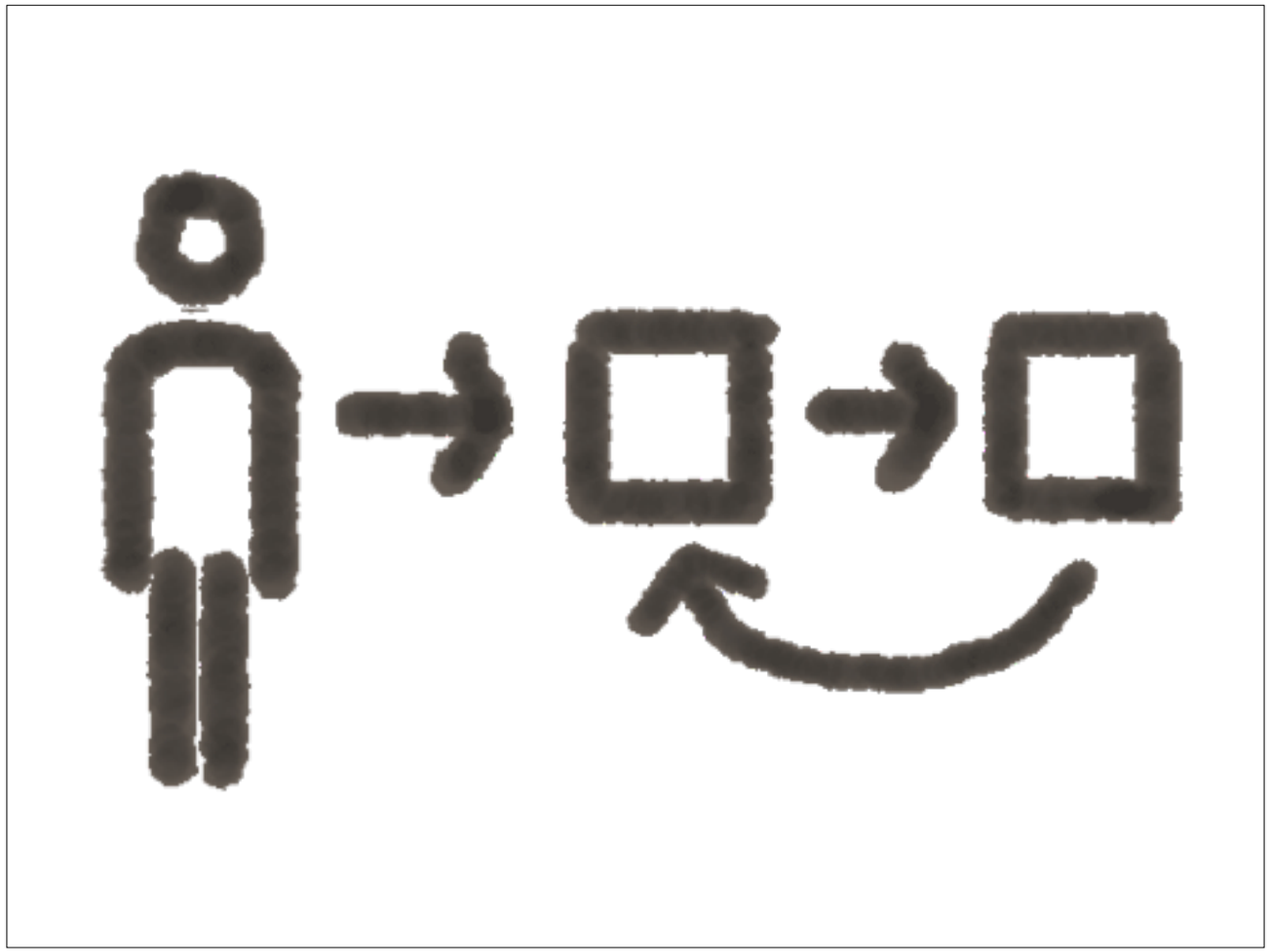
Hands down. Next question: How many of you would like to telecommute but your company can’t or won’t do it right now?

Trying to transition to telecommuting?

OK. Hands down. Next question: How many of you are at an organization which is thinking of starting up a telecommuting program?

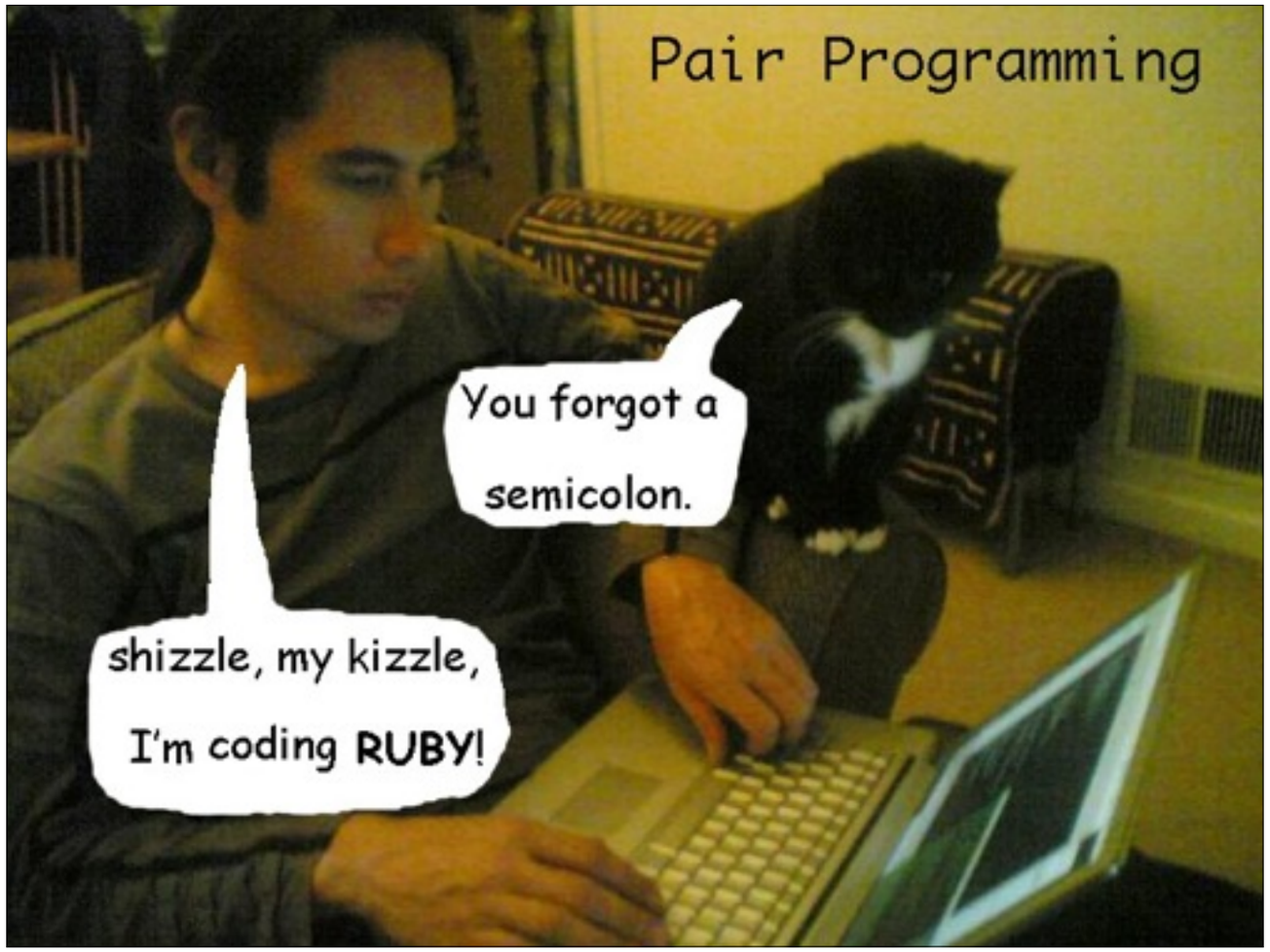
So now that I have an idea of where y'all stand, let me tell you a bit more about what I'll cover today. I'll be drawing on my own experiences, both from taking my own (ex-) engineering department remote and from working with clients to help them do the same, to give you some guidelines on how to do so successfully.

Now here are few notices about what I will and won't cover.



I’m going to spend a lot of time talking about processes, both some you’ll need to consider and also how to define them.

Also, this is going to be a fairly high-level presentation. There is no one solution which applies to all organizations, but there are some guidelines which everyone should consider. Still: Your Milage May Vary.



Also, this talk focuses on software engineering & related processes. Much of it applies equally well–at least conceptually–to many other roles such as customer service, accounting, etc.

“Remote” doesn’t have to mean across the country; it could also mean across the street. Most of the things I’ll discuss apply not only to telecommuters but also equally well to those working onsite. Good processes apply to everyone.

The plans and processes you put in place in order to support a successful distributed team are, conveniently, plans and processes which also help support successful onsite teams. There are no losers here and no wasted effort.

One thing I will NOT cover in this talk is how to hire telecommuters. Good hiring practices—for telecommuters or not—is something which is far too large to cover in a few slides. Also, there's free beer downstairs right after this and I don't want to risk running over. ;-) If you want to know more about this, come and speak to me after the talk.

Why Go Remote?

So, those caveats out of the way, let's cover some of the reasons why you might want to take your team remote.

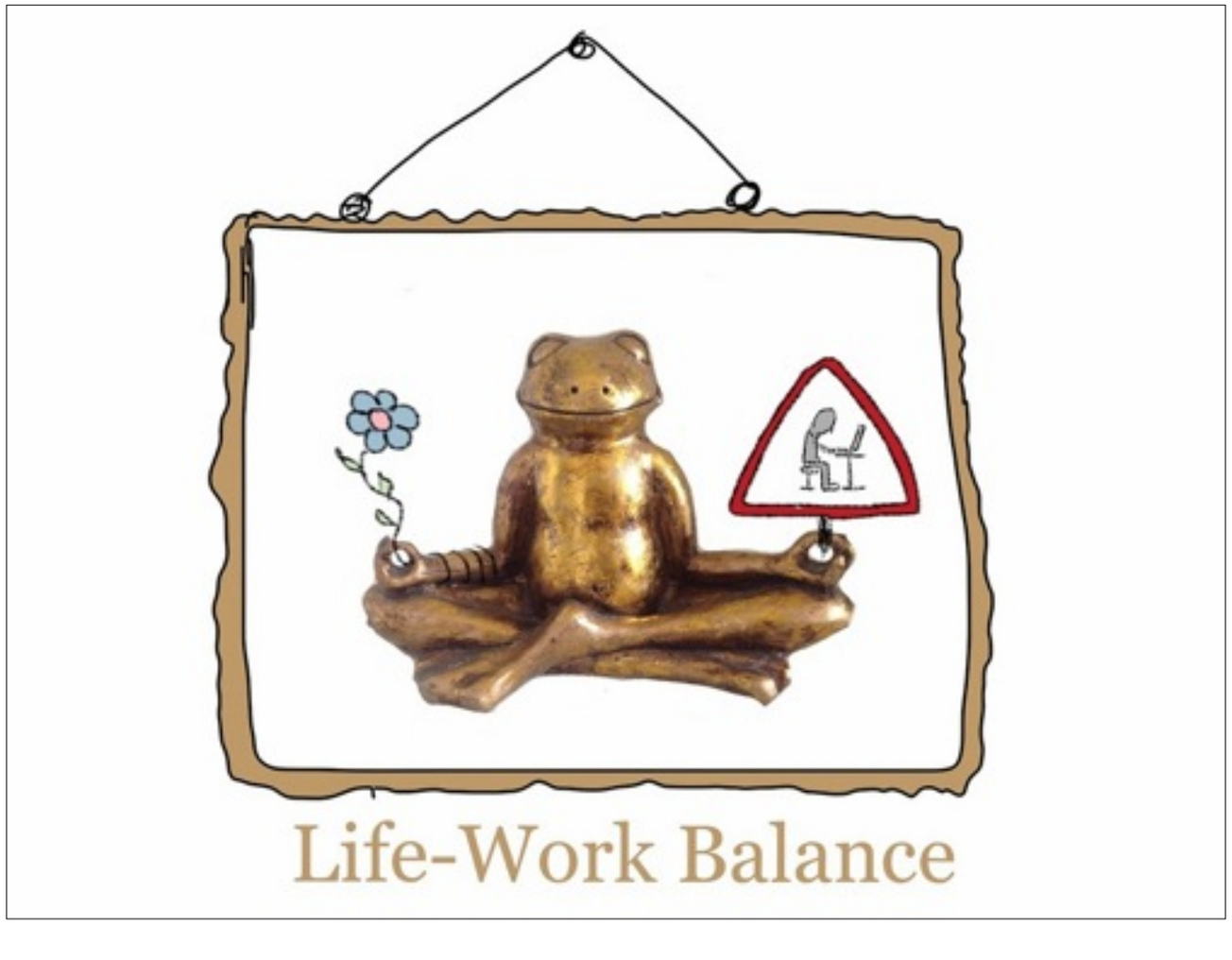
First and foremost: HIRING. This is usually in the forefront of my mind, since I work with a lot of startups in…

…San Francisco. Despite what we may think there in the Bay Area, we do not have a monopoly on the best talent in tech. In fact, right now we in the Bay Area have far, far more open positions than there are people to fill them. Right now we have hundreds of companies fighting to fill positions, but they're unable to find enough people to hire locally. Yet they keep forgetting that great people can be found everywhere.

Not limiting your search to your immediate geographic area means not only do you increase your candidate pool, you also drastically reduce the competition you'll be facing to hire those great people.

When you hire remote, not only is your candidate pool larger but it’s also stronger. Can’t find that C++ programmer with aerospace domain knowledge in Brooklyn? You may find her in New Mexico.

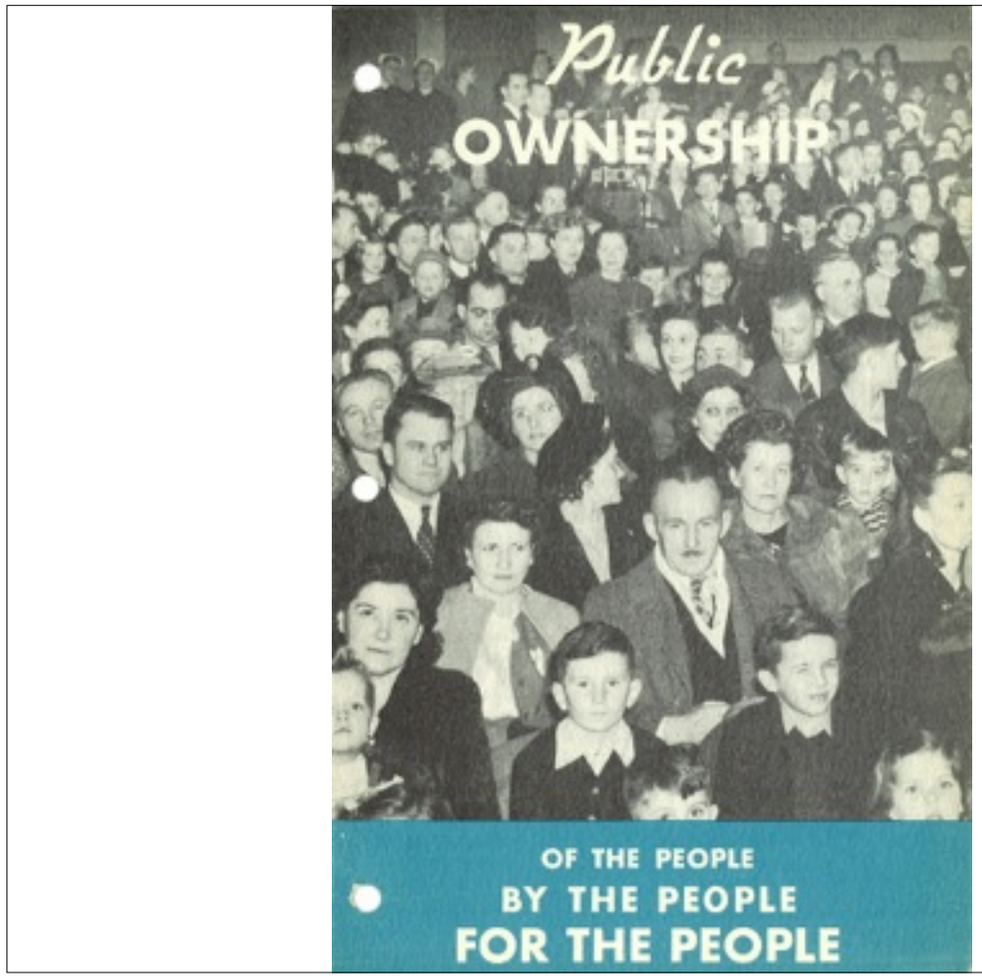
Another large advantage of telecommuting is increased staff retention.



Bottom line is: Flexible jobs == happier people. Happier people stay.

Enabling people to telecommute creates drastically improves a person's job satisfaction. There are several reasons for this…

People who telecommute feel (and are) more trusted and valued by their organization

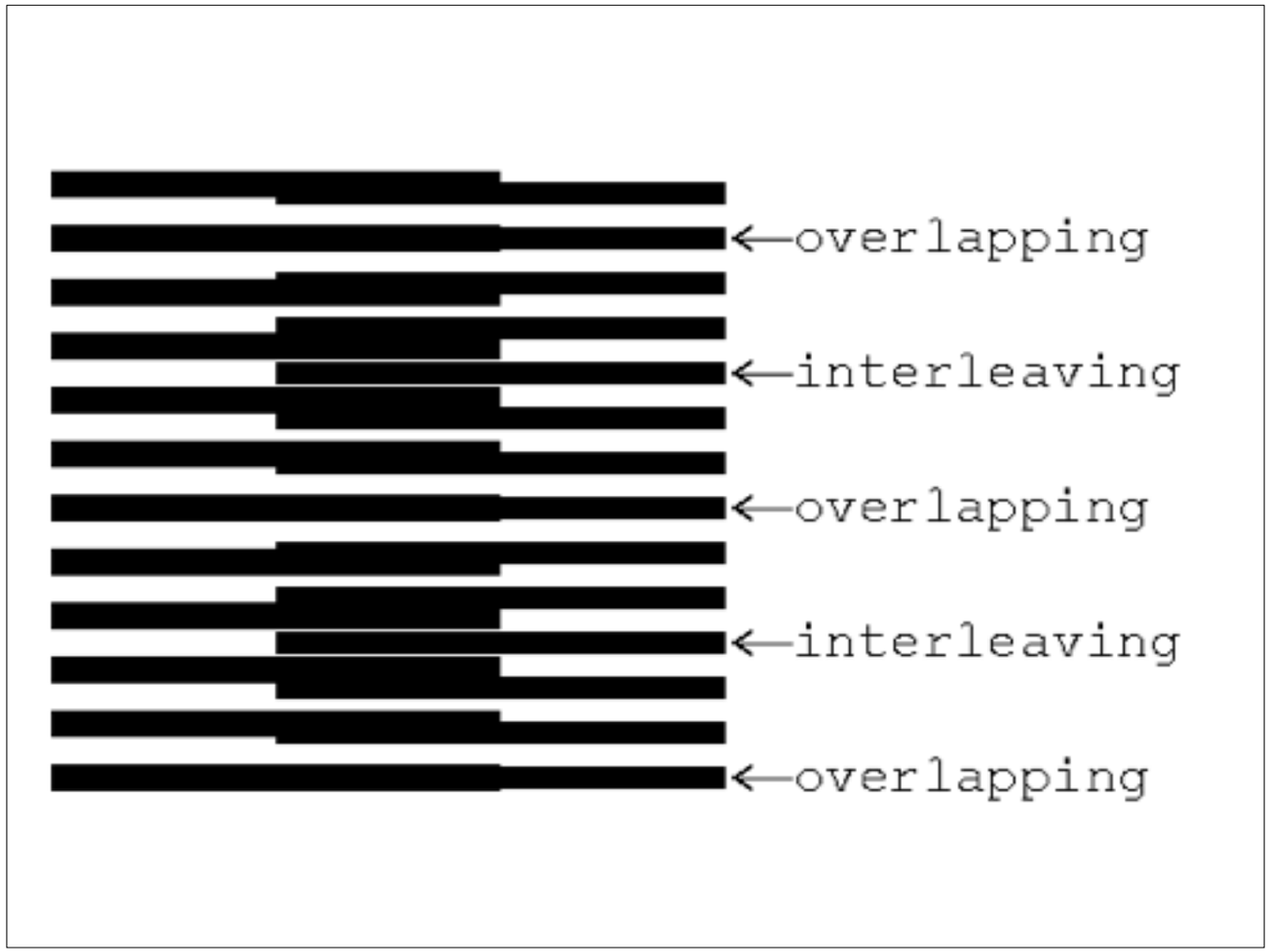


Thanks to that trust, they have an increased feeling of ownership in their product and tasks.

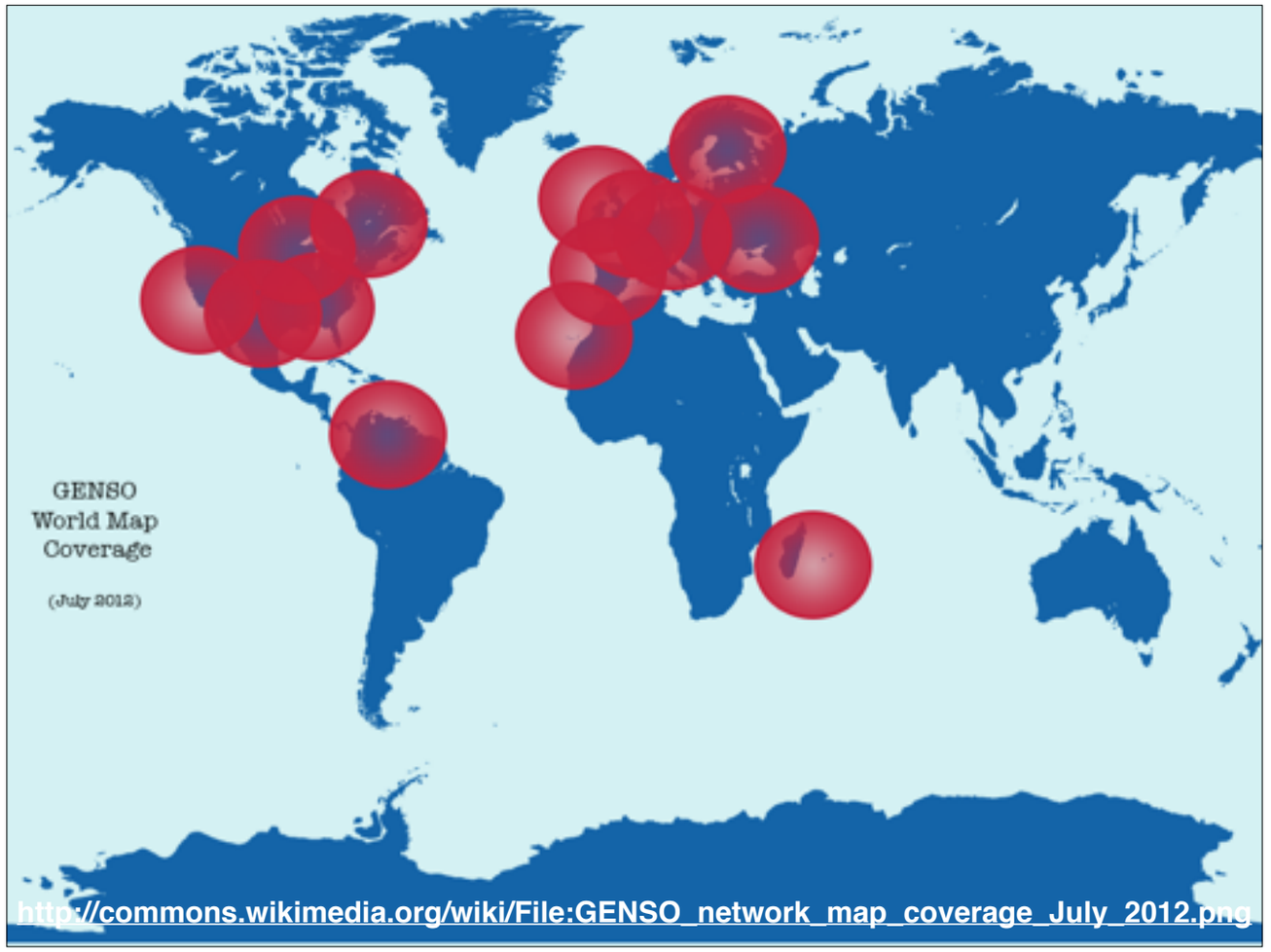
They have almost no commute time or the associated expenses.

They have more adaptability and flexibility, allowing for stronger family relationships and more time for hobbies and other pursuits.

All of these things together add up to much higher per-person productivity. They just get more done and the work they do is of higher quality.



Another reason to have distributed teams is for increased coverage.



http://commons.wikimedia.org/wiki/File:GENSO_network_map_coverage_July_2012.png

Having people distributed across timezones gives you more coverage over more hours of the day (in case, ya know, anything goes wrong).

Flickr user jm3. CC BY-SA
https://www.flickr.com/photos/jm3/98180697/

More coverage == more resiliency. More coverage -> faster response time -> less downtime

In Case of Disaster: people who telecommute can still keep the business running even if the main office is under water.

The final and most popular reason for telecommuting is that it can save your organization a boatload of cash.

Thanks to improved staff retention, you save on hiring. Studies show that replacing someone will cost the organization a minimum 150% of the salary of the position you're filling. Then you tack on training costs/time… Suddenly you see how keeping people on board and happy affects your bottom line.

Flickr user legozilla. CC BY-NC-SA
https://www.flickr.com/photos/legozilla/3552677287

Facilities expenses. Fewer people == smaller office space. Lower catering bills, lower everything.

There are those who think that they can save money because people in other areas are willing to work for less for the same tasks. SLIPPERY SLOPE. Don't Do This. Don't be a cheap bastard who low-balls people just because you want to save a few bucks. If it's worth it to you to pay $130,000 for a great software engineer living in San Francisco, where your company is located, it's worth it to you to pay $130,000 for a great software engineer living in Montreal. Pay for the value you receive. Don't pay what you can get away with.

Objections

Now that we've summarized some of the reasons why you might want to implement telecommuting, allow me to list out and shoot down some common objections to remote working. Some of these have some merit, but most are just FUD. And ALL of them have solutions.

Trust is a two sided coin. On the first side, there are a lot of managers & execs with trust & control issues. They don't believe people will work if they're not currently under an ever-vigilant gaze.

On the other side of the coin, there truly are some people who can't be trusted (for one reason or another) to be productive without constant hands-on supervision. However, these are also the same people who wouldn't do much AT the office either, so…

There's also the related matter of accountability. How will you be able to tell whether people are actually getting their work done? How will you hold them accountable? Well…how do you do it now? If you need to have someone in the office in order to see that they're fixing bugs or rev'ing designs, then I propose that you're doing it wrong. It's about as smart as having someone stare at the server all day to make sure it's still running. There are better ways to do that.

There is the very real concern of security. If you're dealing with sensitive information or secret intellectual property, it's vital that you set up a robust system to secure this information. Encrypted disks on laptops, VPNs, SSL everywhere, etc. This is a legitimate concern but it's a solved problem.

There's a belief that it's necessary for people to be collocated in order to form social and cultural bonds. As anyone here who's ever spent time online can attest, this is not actually the case. Very rich cultures are possible without collocation. Collocation isn't the necessary part for building a great culture. Communication is. I'll talk more about that in a bit.

Sometimes (more often than should be), remote team members end up out of sight and out of mind, neglected by their team even if they're great contributors. For telecommuting processes which aren't well-established or are poorly implemented, there's the very real danger that telecommuters may become 2nd class citizens or even forgotten. Note: this isn't a problem with telecommuting. It's a problem with a poor telecommuting process.

One very real complication with telecommuting is the problem with benefits & taxes. While it's not quite as complicated if all of your telecommuters are within a single country, due diligence is still needed. Does having an employee in a different state mean that you must pay taxes there? Are insurance laws different? These are all considerations you must take into account, but don't do it alone. Seek professional assistance immediately.

Method

OK, now that I've summarized some of the advantages of and objections to telecommuting, let's talk about how to make it happen and be successful.

First: please note that this is not the sort of thing which you can implement overnight. Many of the steps can be established in parallel, but all of them will take some time to do correctly. As with most worthy efforts, this is a marathon, not a sprint. Do it right, don't do it right now.

Culture

Organizational Culture is an important part of implementing telecommuting. It is also the element which will take the most care and time to do correctly. If you're an all new company or team, this can be easier. You can set up the culture to embrace telecommuting right up front. But if you're trying to change an existing culture, then be prepared to travel a long road.

You can walk the halls of OSCON & hear any number of sessions focused on shifting culture, so I won't go into this in depth too much here. Instead, I'll just highlight a few things you'll need to keep in mind.

Ask yourself: What does your organization actually value and need? The product your employees create or having the butt in the seat? Be honest. There is no wrong answer here. The needs of your organization are the needs of your organization. But if the answer turns out to be “butt in seat” then be aware that no amount of effort will make your telecommuting effort successful. Save yourself a lot of time and headache and bail on it now.

Another important culture change which is necessary is to stop speaking of “onsite team members” and “remote team members.” Instead, just speak of “team members.” If the only difference is that some are in the office and some are not, then there’s really not much difference at all so please stop focusing on it.

Flickr user jasoneppink. CC BY
https://www.flickr.com/photos/jasoneppink/77348594

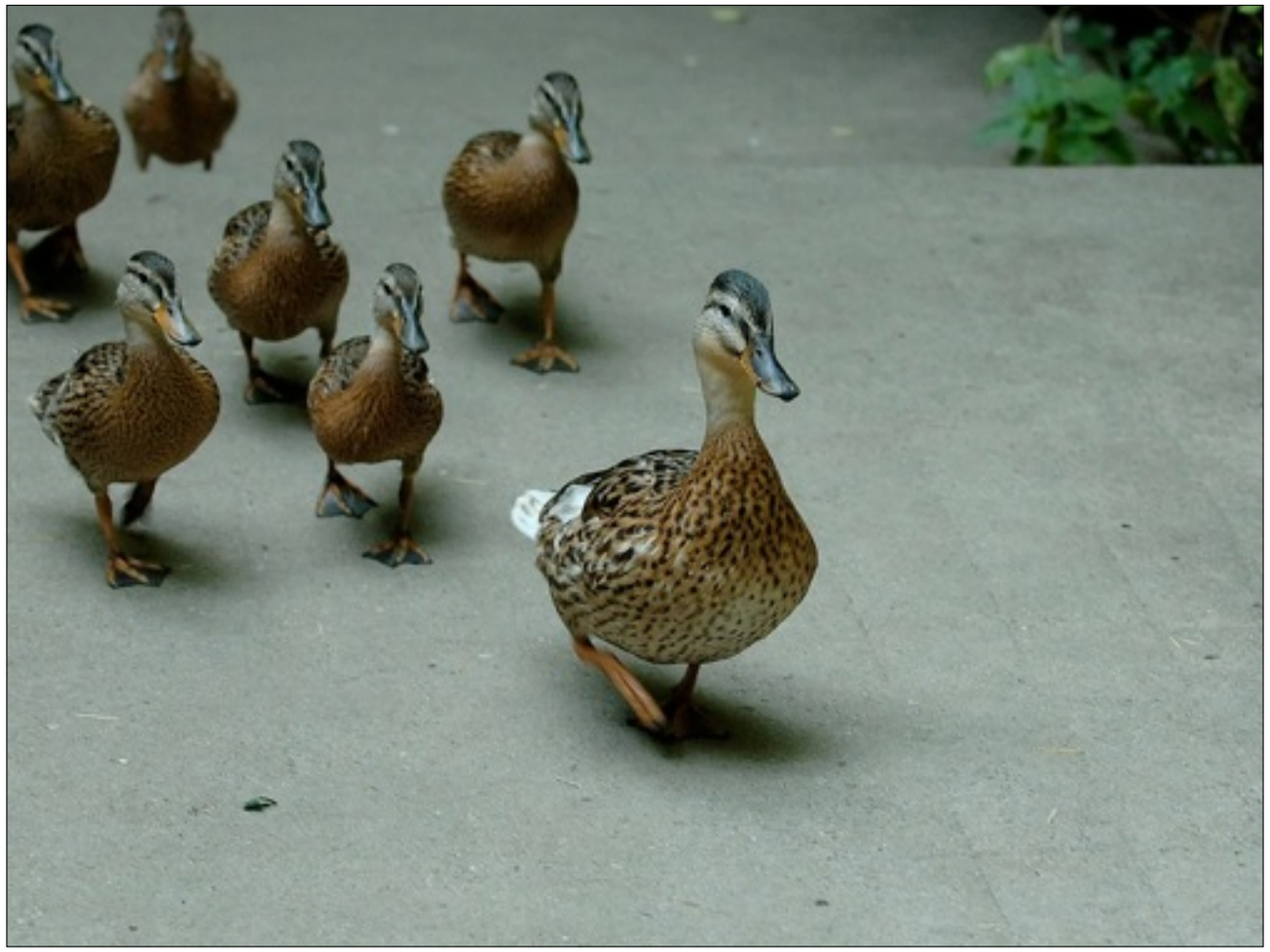
The processes & tools which are established for telecommuting employees work just as well for those who are onsite.

Flickr user ontario_wanderer. CC BY-NC
https://www.flickr.com/photos/ontario_wanderer/3496185271

Therefore everyone can and should use the same processes and tools. Optimize the processes for telecommuting then onsite will “just work.” Everyone will benefit from the improved workflows and tools and the organization will enjoy maximum flexibility.

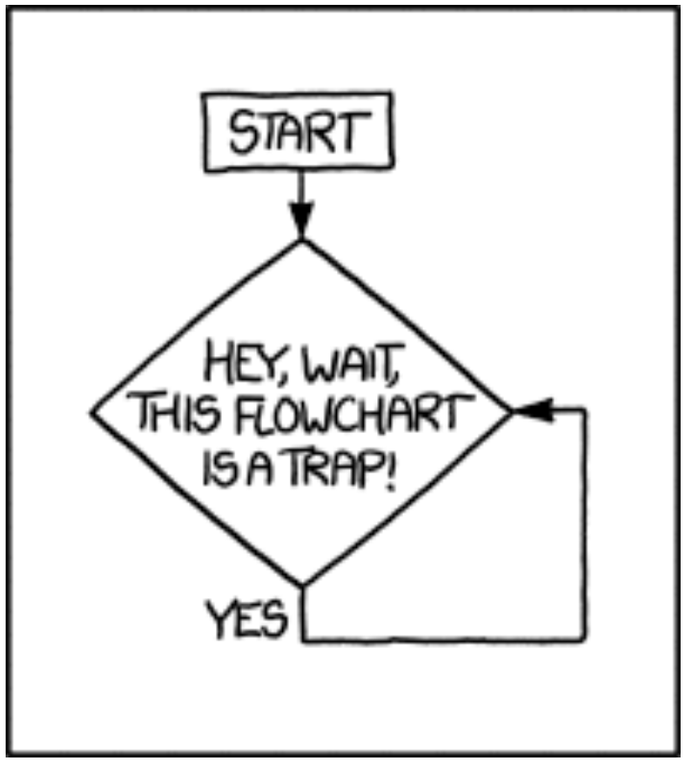
Establishing this mindset and a commonality of processes and tools helps to eliminate the Us vs. Them camps which can spring up between onsite staff and telecommuting staff. Everyone is on the same footing. Everyone is operating the same way. Everyone is pulling in the same direction toward the same goal.

Another way to eliminate the “us vs. them” tension is to make sure that telecommuting is seen as normal. It’s not a perk, it’s just the way those roles operate. If it starts being seen as a perk then those whose roles don’t allow for telecommuting start to resent it and those who can do it.



As with all big changes, these cultural changes will fail if you don’t have complete support from the top. Back your play. Lead by example. Provide moral and resource support.

Processes and Policies



There's often blowback in our industry against processes. People think TPS reports and Pointy Haired Bosses. And, yes, that can happen if it's handled poorly. But codifying processes and policies is the second most crucial part of mounting a successful telecommuting effort.

As with all processes and policies, the key is to make things EASY and USABLE for everyone while still being effective. This doesn't only include the people who're telecommuting, but also those with which they're working onsite. Consider everyone whom the process might affect.

As well, all good processes and policies should include both expectations as well as rationales. Everything should be clear and explicit. Telecommuting team members don't have as much of a chance to verify before following an obscure process. They'll just assume then move forward. This can cause friction. Therefore: Don't force them to assume. Explain everything.

Having these processes well defined also allows everyone—telecommuter or not—to be on the same page. When everyone is following the same processes and understands the reasons behind them, it's harder for an us vs. them mindset to sink in.

However, just because there’s one true set of processes which everyone shares, that doesn't mean that they’re carved in stone. Everyone should know to be on the lookout for ways to improve the processes or even for processes which are no longer effective but are only being followed out of cargo culting. These processes must change or be eliminated.

I've mentioned a few times now that your policies & processes need to include the "why" of the matter. It might be tempting to save time by skipping this, but DO NOT. It is vital that people understand why they're being asked to do things certain ways. This empowers them to make the right decisions in cases where the processes aren't clear or when a situation arises which isn't covered by policies. You're going to hire great people. Give them the information they need in order to make the right decisions.

So, a few examples of some of the policies you may need to define in order to make your team successful:

Care and feeding of a bug

Coding or other style guides

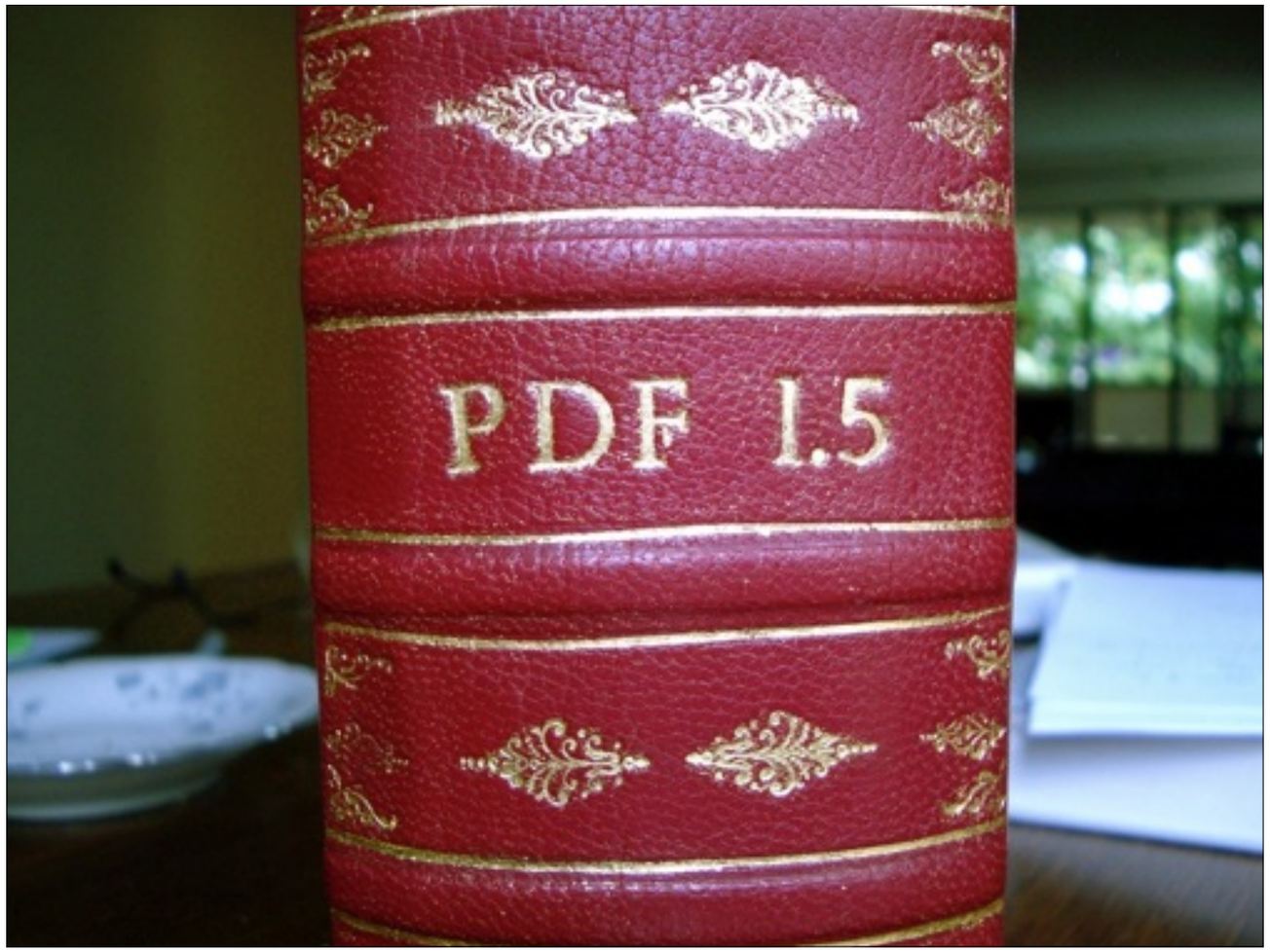
And other related dev stuff such as version control and the like. But don't forget that your team is only one part of a larger organization and that it must integrate well with that organization. Therefore don't forget to define things like:

Expense reports

Time off requests

And there are matters of security.

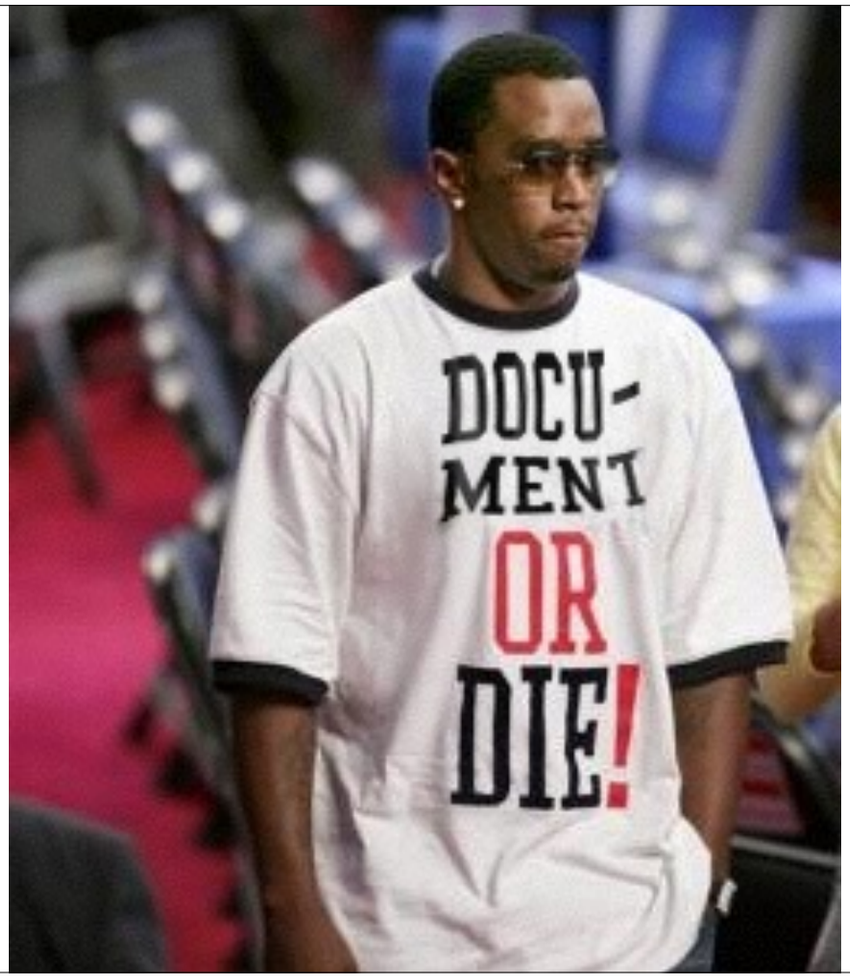
These are examples of processes with which you'll need to collaborate with other departments. Odds are you'll find that they're very open to the idea of codifying process and that the process y'all come up with will make everyone's jobs easier.



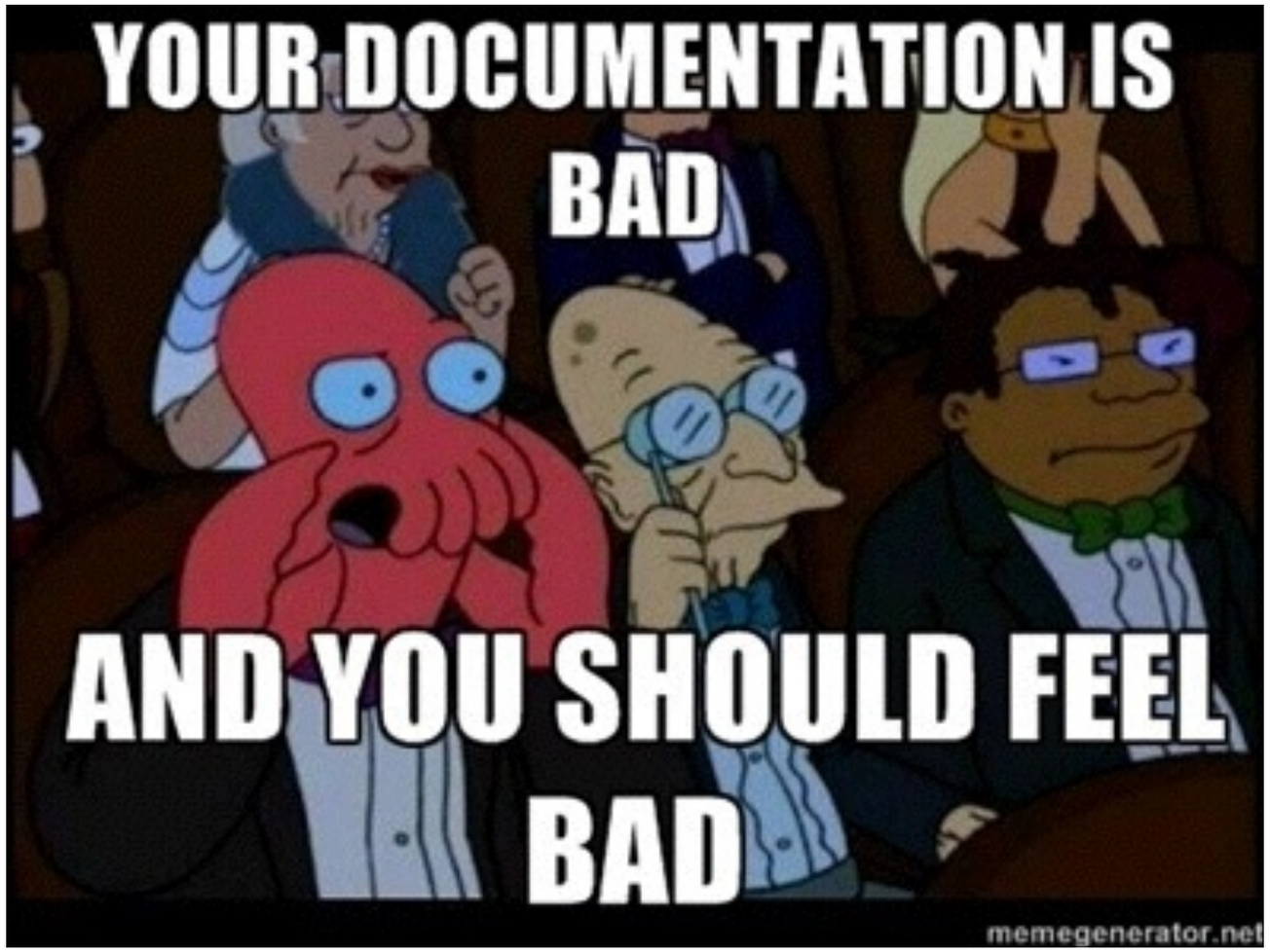
For example, when I took my first team remote I cooperated with Finance to come up with a new way to submit expense reports using only PDFs and email. It not only enabled my team to do this easily from wherever they were, it also ended up saving Finance a boatload of time and effort compared with the previous process. They loved it. We loved it. Everyone was a winner. So we rolled out the process across the entire company.

Documentation

So now that you have all these lovely processes and policies, what're you going to do with them? You need to WRITE THEM DOWN.

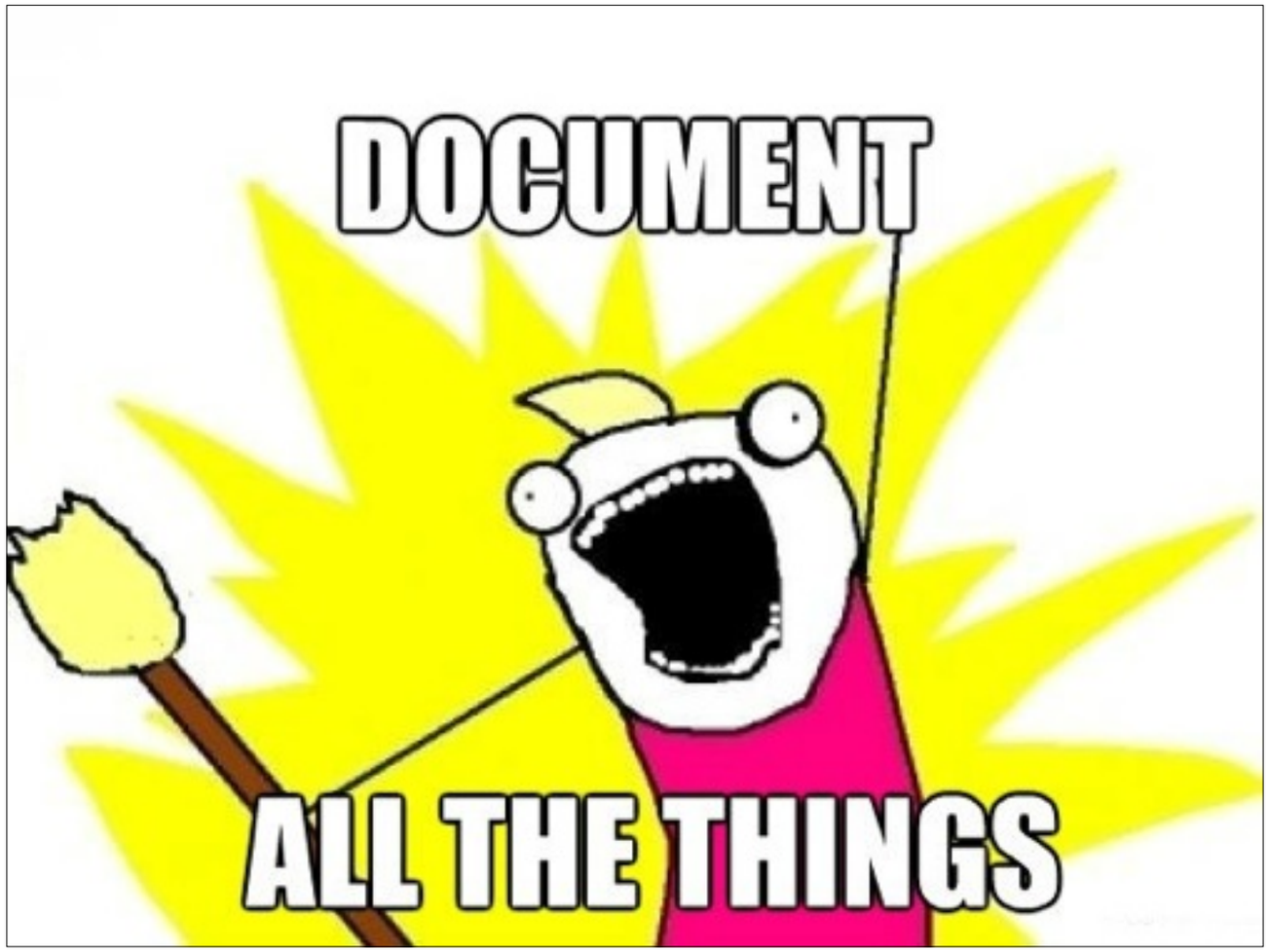


While documentation is important in every organization, it is absolutely vital in one which is distributed. This may feel foreign or even dirty to those of you who are all “Agile,” but it’s absolutely necessary with a distributed team. Skip it at your own peril.



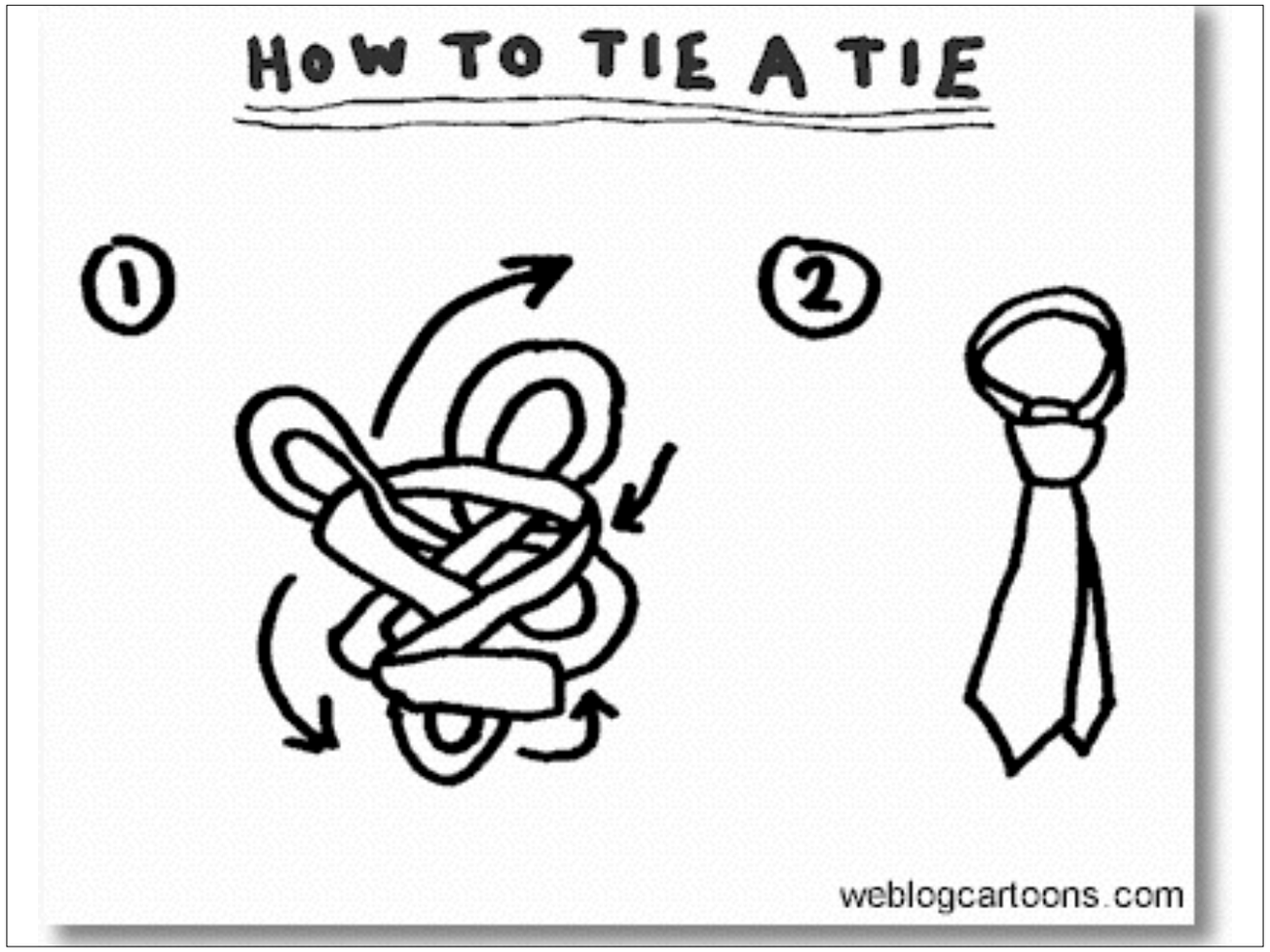
You must foster a culture of documentation. It must get to the point where a person committing a change without also committing associated documentation will feel just a little dirty.

For the record, I don't actually support doc shaming, but I couldn't resist this slide. Because, after all, why not Zoidberg?

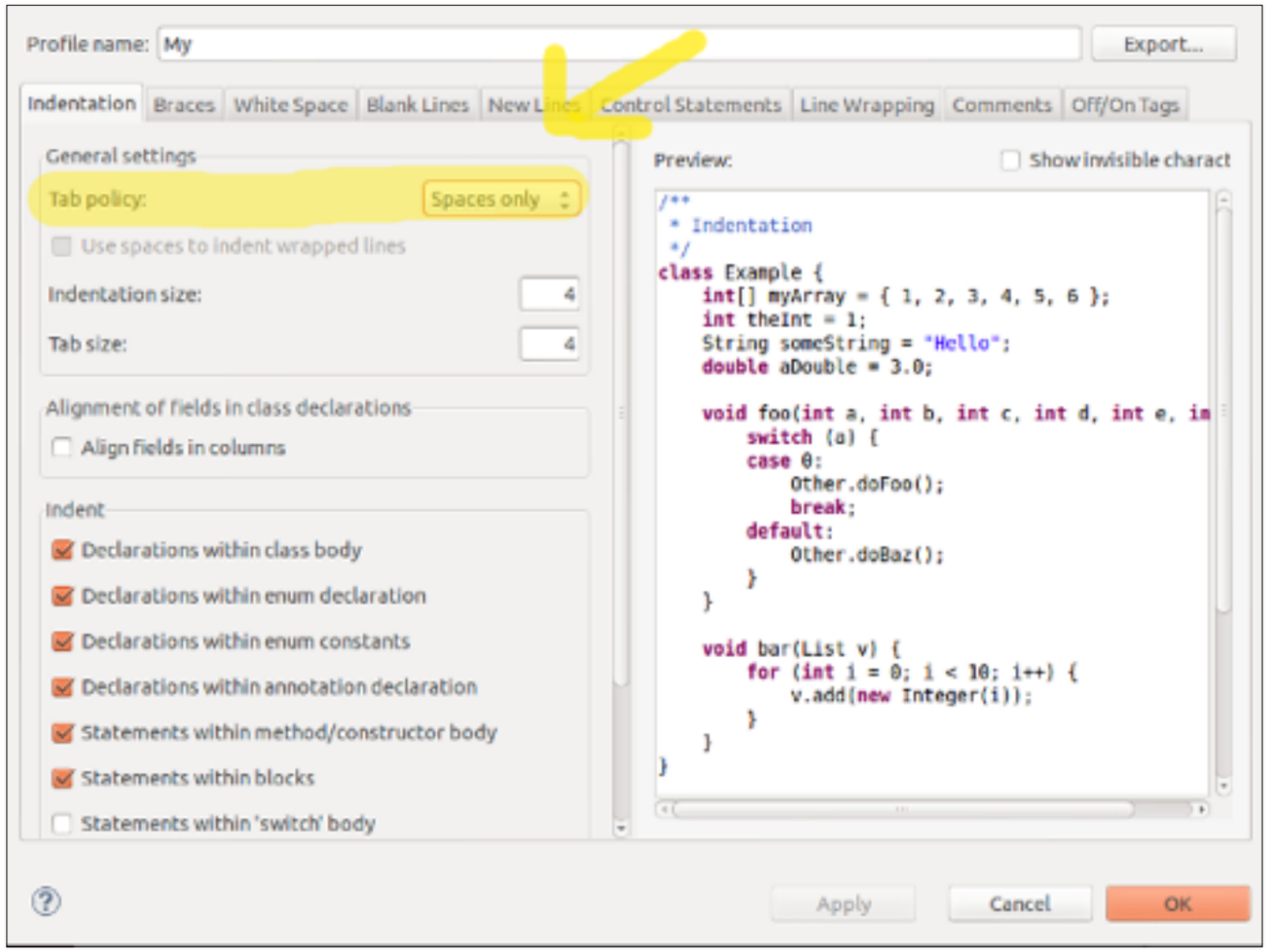


Ya just knew Allie would be making an appearance, and here she is. :-)

DOCUMENT ALL THE THINGS! (such as…)



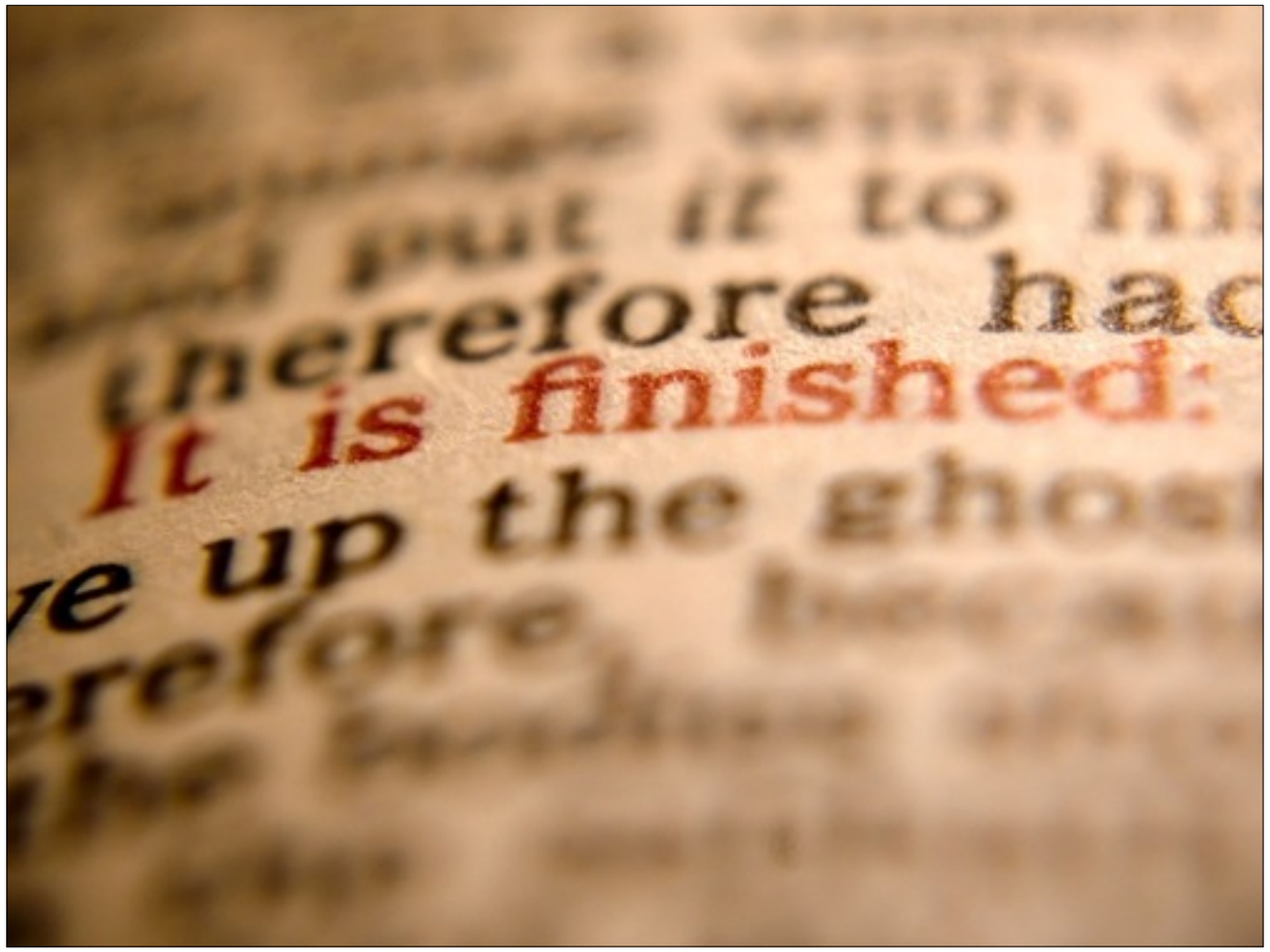
Processes & policies



Style guides

APIs

And other things of cultural significance such as inside jokes and jargon



Imbue the entire team and organization with this idea: if it ain't doc'd, it ain't done. You can even add commit hooks to your version control system to check for docs & reject the commit if they're not there. ;-)

It does no good to have these docs if no one can find or access them. Therefore they should be kept in a single, easy to find, easy to access and use repository. Code docs can be excepted here, but if you can find a way to integrate them then all the better.

The best tool for this job is THE ONE YOU'LL USE. Period. Dot. End. The most obvious idea will be wiki, but if you find your team won't update a wiki then try Markdown and a Github repo. Have an org which kicks it old school? Then use troff and man pages. I don't bloody well care what you use, just as long as you use something and it works for your organization. Don't be dogmatic. Be pragmatic.

Tooling

And lookit that: a brilliant segue into a discussion about tooling. Using the wrong tools is like walking around town wearing shoes that don't fit. Sure, they may look cute but you're setting yourself up for a lot of pain.

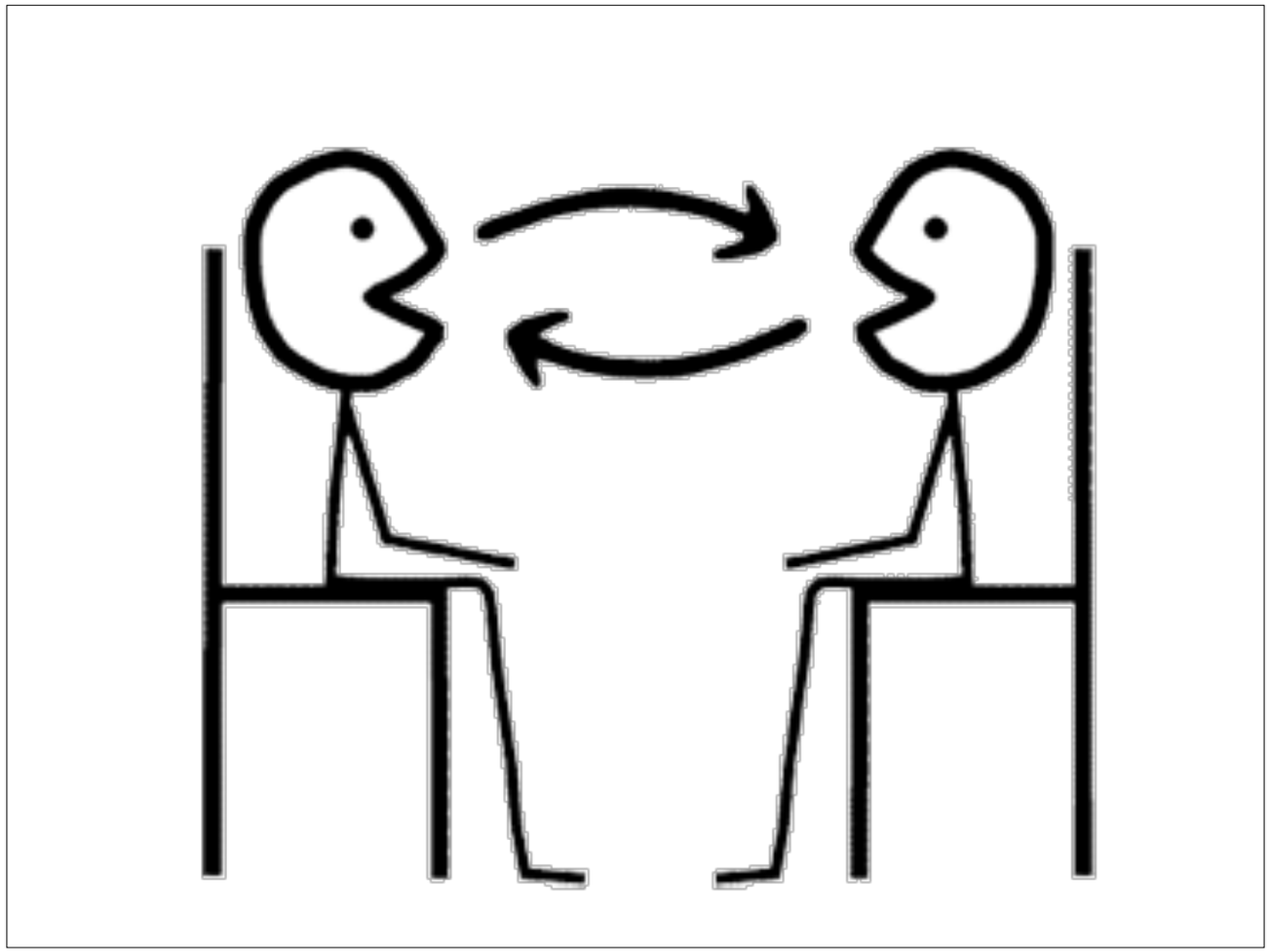
Every team will have a raft of tools they'll need in order to be successful while distributed. Conveniently, as with everything else I've covered so far, the tools chosen apply equally well to onsite people. Everyone, remote or not, should use the same tools and processes.

- Documentation
- Version control
- Issue tracking
- Project management
- Whiteboard
- Pair programming
- Screen sharing
- Video & audio conferencing
- Chat/IM/IRC
- Etc.

The tools needed will vary by team. But some you may need to consider are: Documentation, version control, issue tracking, project management, whiteboard, pair programming, screen sharing, video & audio conferencing, chat/IM/IRC, etc.

How to choose tools

Because tool requirements are vary per organization, I'm not going to make specific recommendations for "Must Use!" tools. But I will tell you how to approach selecting the right tools for your organization. This isn't rocket science but it's still shocking how few teams do it correctly.



During this process, please make sure to include others not only from the team but also from elsewhere in the organization. You don't have to (and shouldn't) take it to full committee, but you should make sure you get feedback from a good selection of people.

Step 1: Gather Requirements

What are the categories of tools that the group will need? What functions must each of those tools provide? What integration points do they need? Prioritize these. Try to keep “nice to haves” off these lists and limit yourself to pure requirements to try to keep things simple. Gather these requirements up front. Use them to guide and inform the entire process. It will take time to do but it will save both time and goodwill later.

Step 2: Collect Candidates

Everyone jumps to this step first, but that’s a mistake. Don’t start looking for candidate tools until you have a good idea of your requirements. Otherwise you’ll fall in love with something which is unlikely to be suitable. Start your collection by looking @ what’s already in use at the organization, then add recommendations & other tools you've found in research.

Step 3: Evaluate Suitability

You have your reqs, you have your candidates. Time to put ‘em together. Chocolate & peanut butter. Any on which you can’t decide? Pilot them.

Step 4: Roll ‘em out

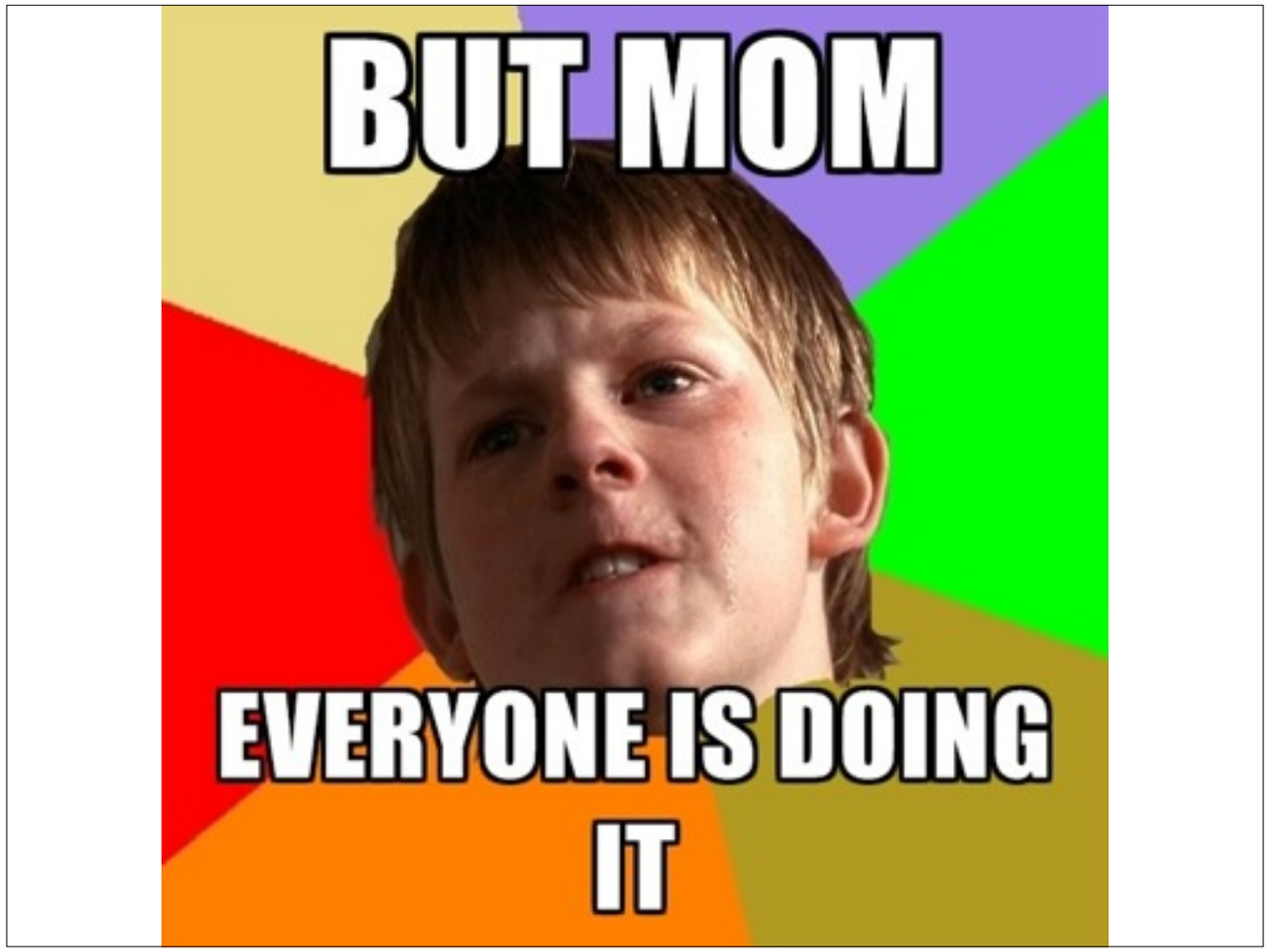
Once you’ve selected the tools you need, do a gradual roll out, providing training to each group as you bring them on board. While you do so, use it as an opportunity to document the process, policies, use cases, and expectations for the tool. Also, use this as an opportunity to finesse (and document!) the training for the tool. By the end of this process, everyone should know not only how to use the tool but also why it was selected and what it’s expected to do for the team.

Step 5: Review and Reevaluate

Now that everyone knows why the tool was selected and what it's supposed to do, please make sure that they need to speak up should the tool stop doing what it's supposed to. Maybe a change in process is needed. Maybe more or different training. Maybe the tool just isn't the right one for the job. The sooner you learn that, the less damage the tool will cause. Don't keep wearing those shoes just because they're cute. Take them off before you get blisters.

How NOT to choose tools

See? Not rocket science. Yet I find most organizations won't follow even those simple best practices. Instead, most places will use one of the following strategies for tool selection:



Trendy

Quickest to implement

Cheapest

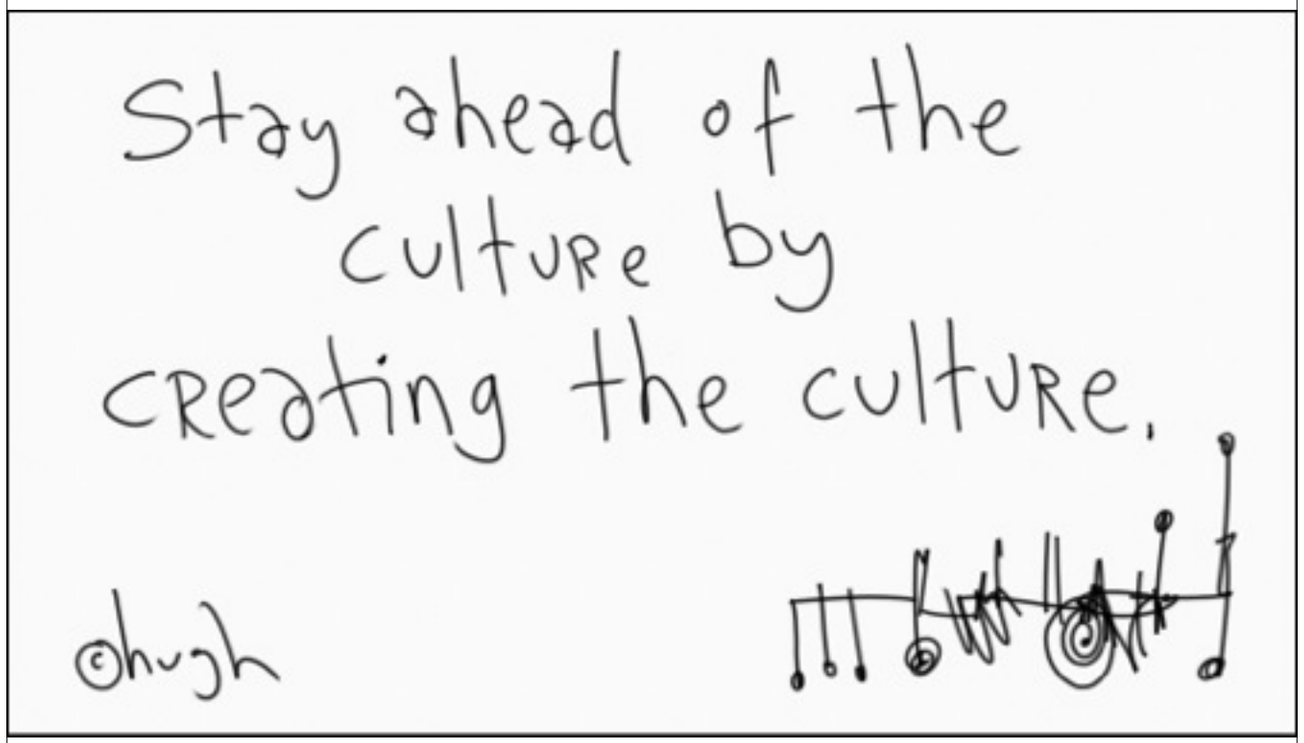
Newest

Politically driven

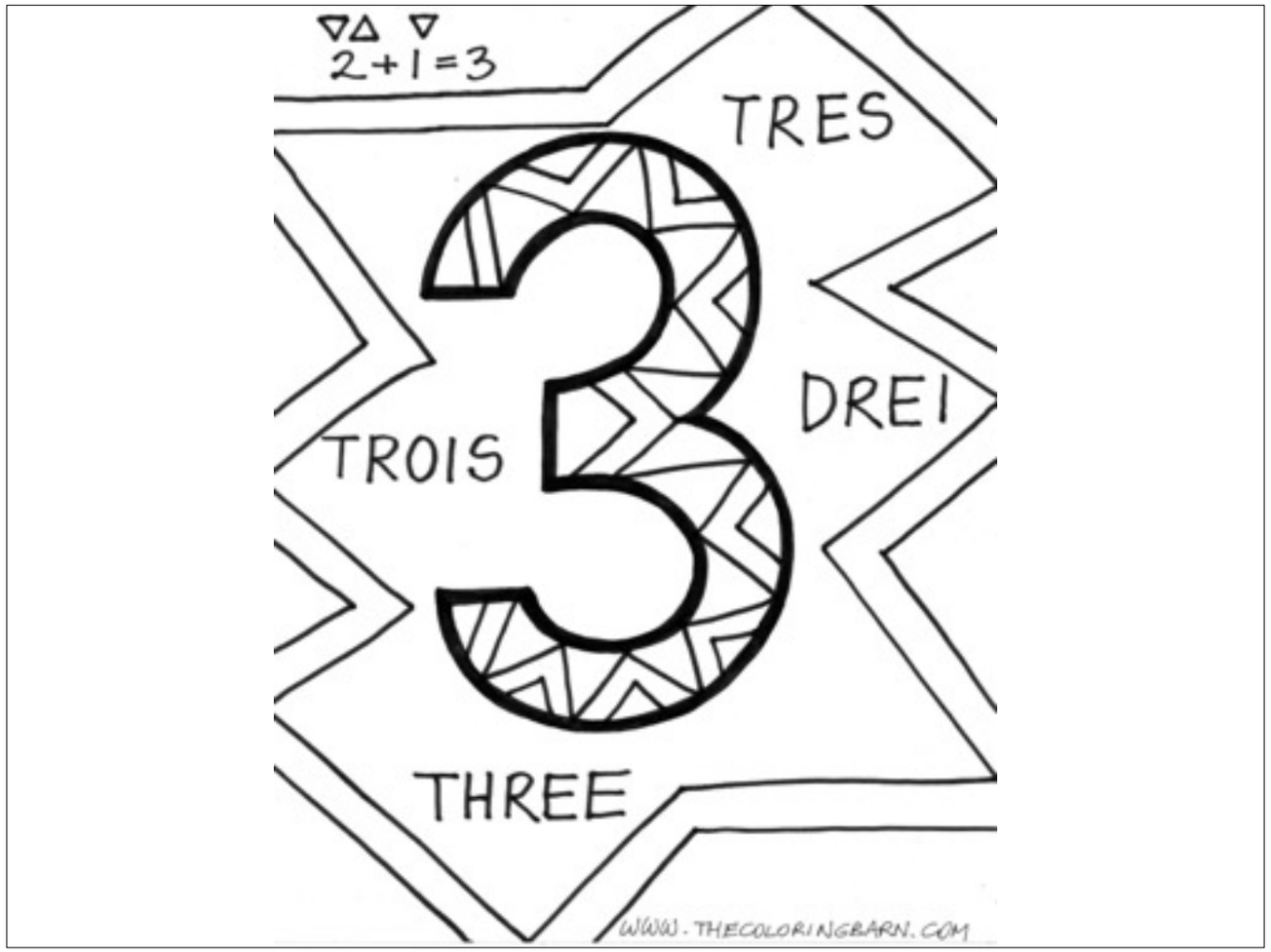
The bottom line is: The only good reason to choose a tool is if it does what the team needs and does it well. All other reasons are either lazy, incompetent, or utter bullshit. Or, I guess, all of the above.

Communication

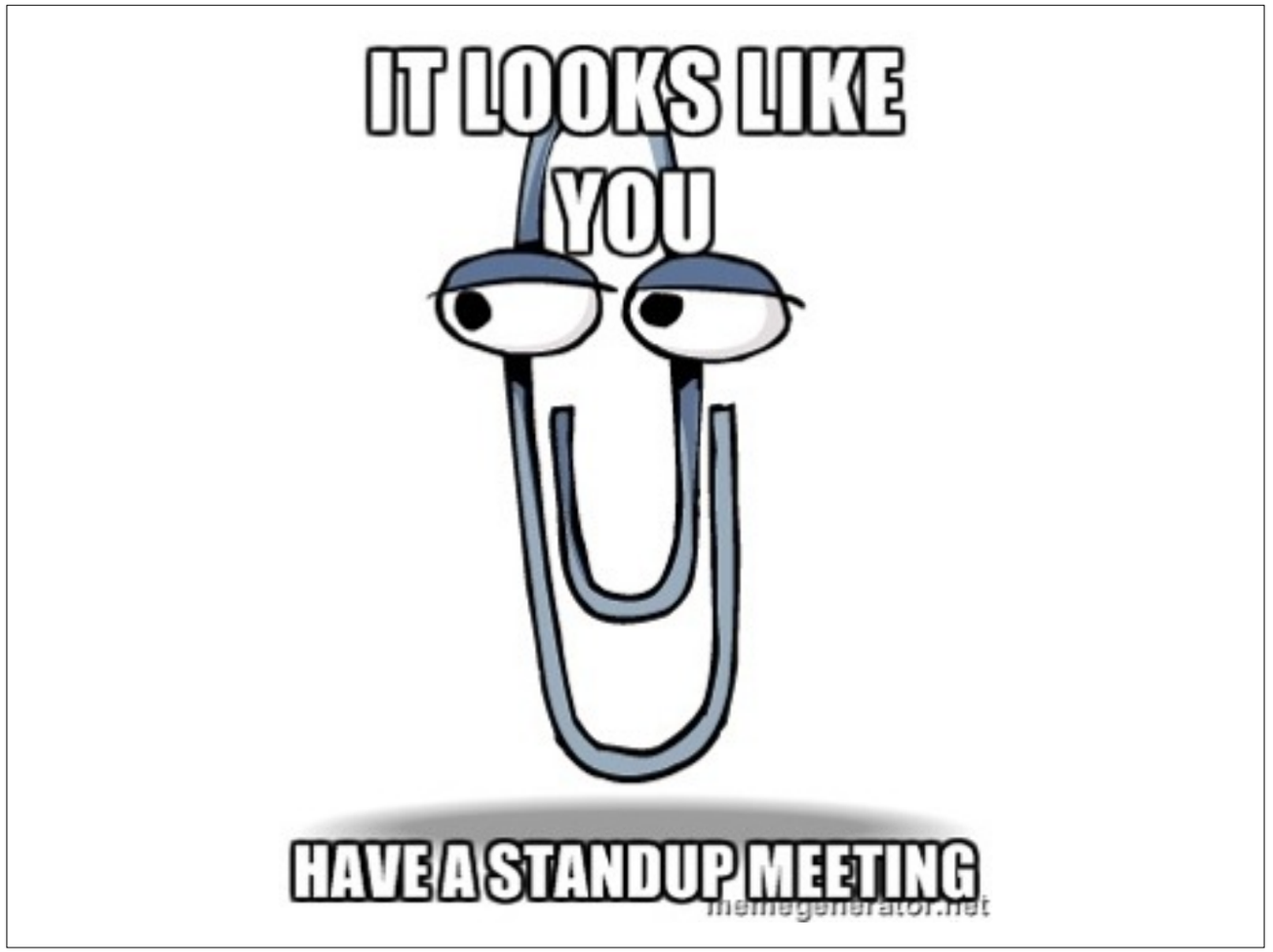
Many of the processes and categories of tools we've discussed so far are aimed at supporting one thing: Communication. This is the **most important** element of any good team, but because so much can & is communicated by facial expressions & body language, it's important you have open & high traffic channels for communication for your remote staff and that everyone use them more or less constantly while on the clock.



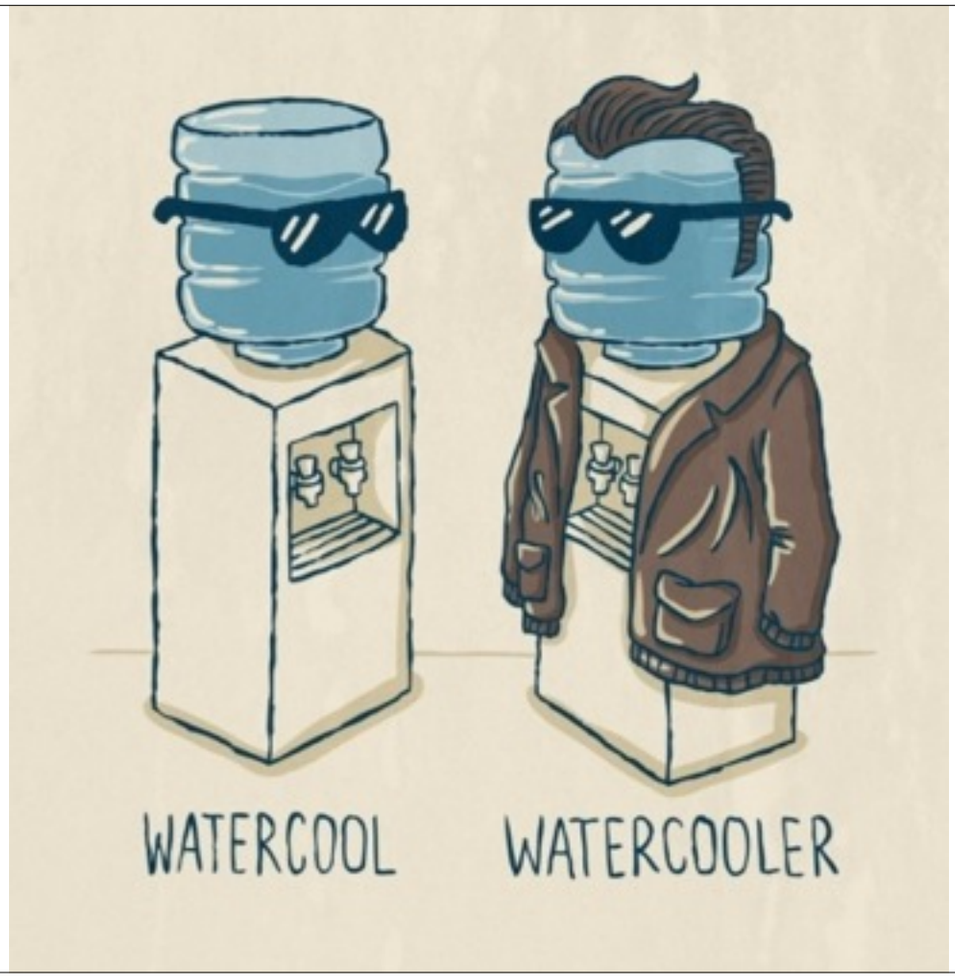
Maintaining robust communication between everyone leads to the elimination of those “but what about the culture” objections. There’s still a great culture. It’s just all online. We are the children of the internet. We understand this stuff.



Therefore, to create and maintain that robust communication, I suggest three complimentary approaches. And remember, these suggestions apply equally to onsite and remote employees.



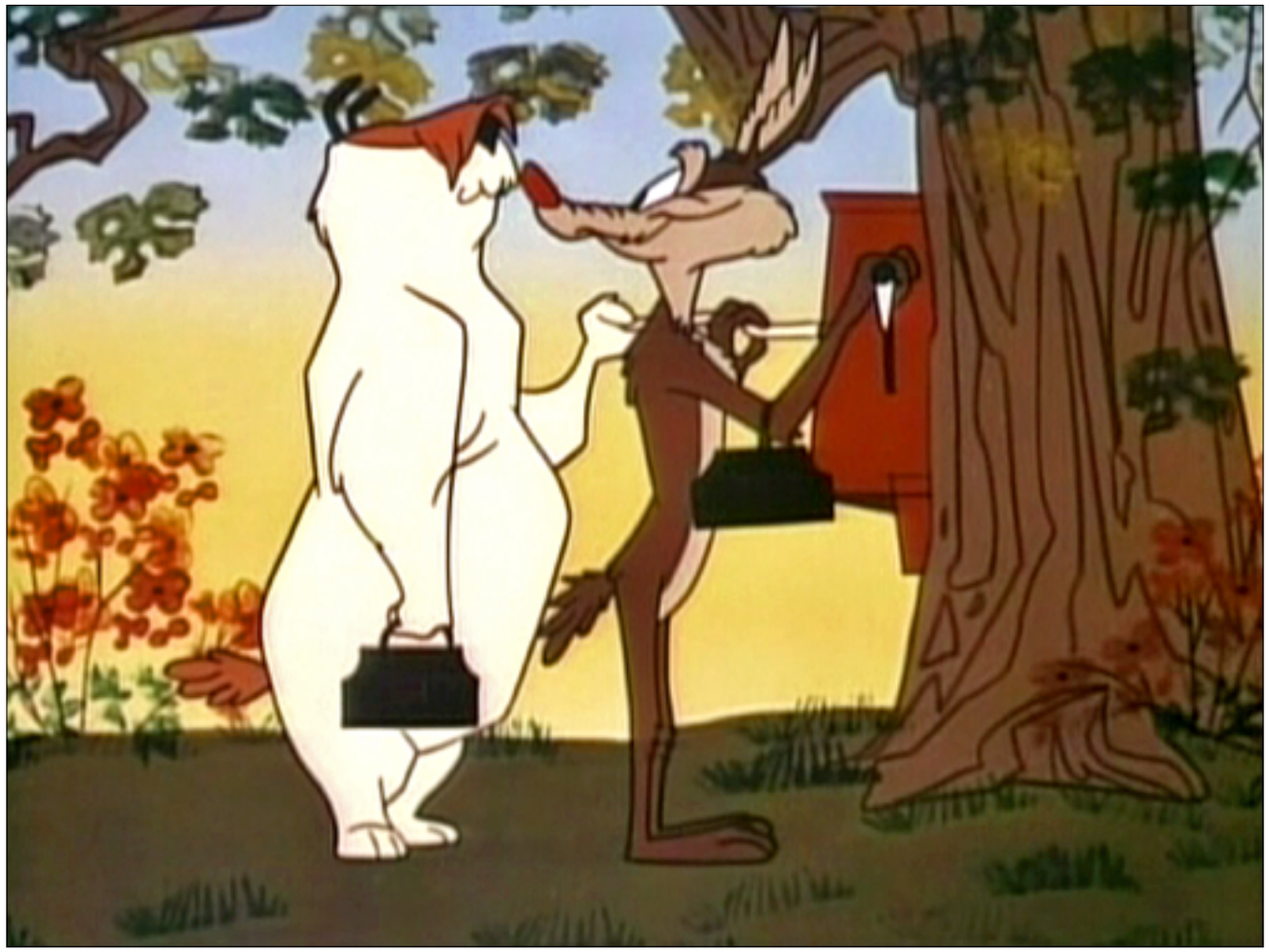
1. Have some sort of daily or frequent standup-type meeting. These should be over Skype or Google Hangouts or what have you so everyone can see/hear each other. They shouldn't last more than 10 minutes. Ever. This suggestion enforces frequent contact between all members of the team, allowing them to develop deepening knowledge of each other in small chunks over time.



2. Make sure there’s an online water cooler in your chat system. This channel is mostly social in nature. It’s where people can be themselves, blowing off steam, sharing links, building camaraderie, and developing inside jokes and jargon. This is a great way for people to get to know each other and bond as people and as friends.

3. Several times a year (or at least two), bring everyone together at the mothership. Fly them in and put them up for a week, having them come into the office or perhaps have everyone gather at a single off-site location. Online chatting and collaborating is great, but there's nothing like sharing a conversation over a meal to bring people closer together.

These approaches serve other purposes as well. For instance, the daily standup enforces that there is a period every day when everyone's schedule overlaps. While, sure, flexible schedules are great it's still necessary for people to be able to know when and how to contact their team members. This schedule coordination can be accomplished in any number of ways, but the standup is a good starting point.



And the water cooler, beyond just being a nexus for cat GIFs, also functions as the online “office.” Are you on the clock? Then come into the “office,” aka the chatroom. This helps with those accountability and trust issues mentioned earlier. If you’re working: you’re in the chatroom. If you’re not in the chatroom, your manager may virtually come looking for you to see why you’re not on the clock.

Example

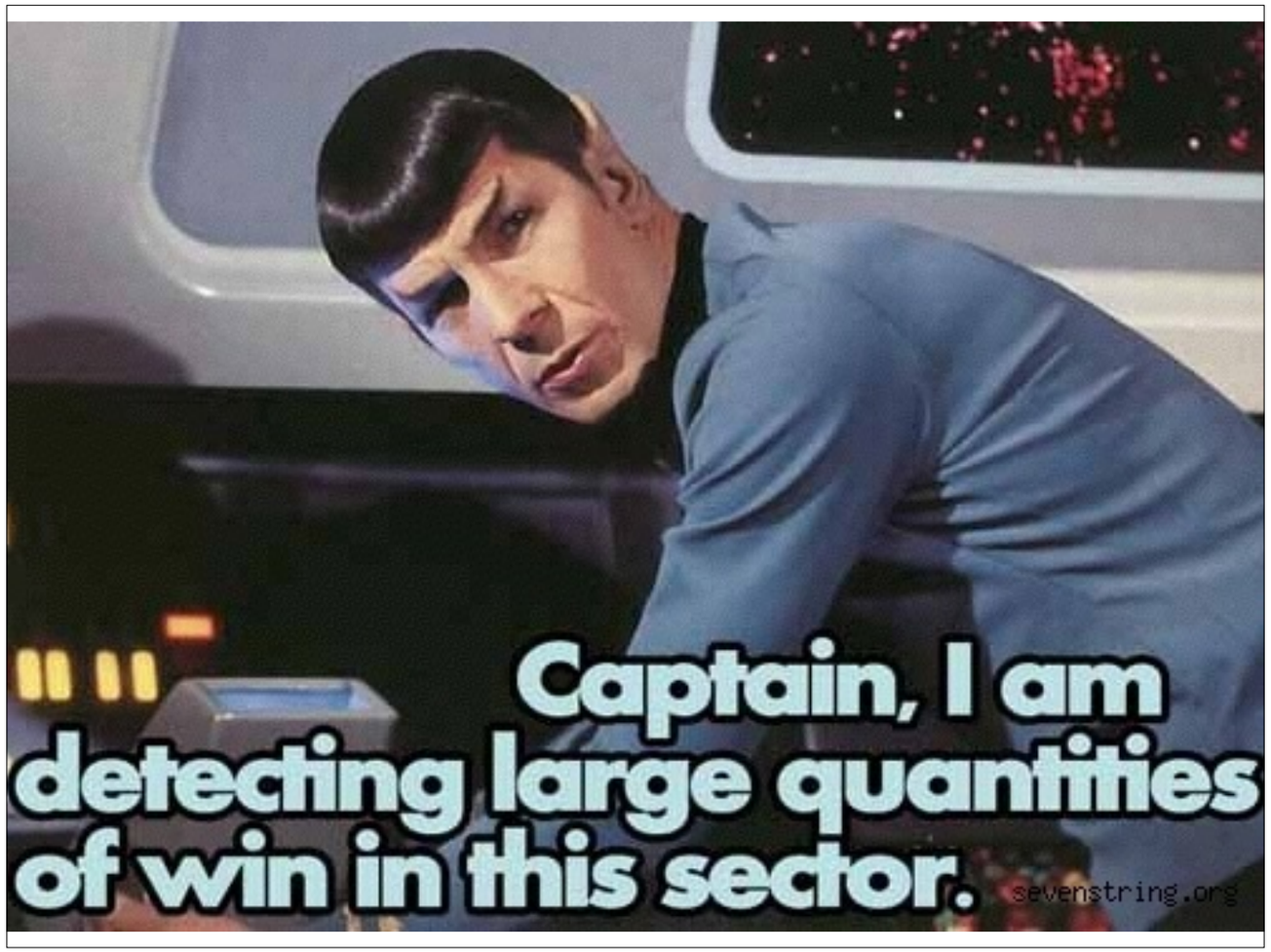
For various reasons, my entire department ended up having to either go remote or be hired to work remotely. I'd kinda seen this coming & had been setting up the processes, just in case. By the end, we had people in three different states, most of whom were in CA but none of whom were in the office.

Our primary requirement for tools was that they be simple yet effective. We didn’t have time/bandwidth to monkey w/complicated and feature-burdened applications. Therefore, we chose:

- Pre-existing in-house Jabber for IM and chatroom
- Pre-existing wiki for documentation
- Pre-existing CVS for version control
- Bugzilla for issue tracking
- Skype for calls and meetings
- In-house built Javascript Kanban board
- PDFs for time-off or expense requests

So, what results did we see out of this? Naturally, YMMV, but we saw:

- Easier and shorter hiring cycles
- Higher productivity
- Higher quality
- Happier team members
- Led the way for the rest of the company to do the same



In summary, in that particular case going remote was Full of Win.

https://www.zotero.org/groups/oscon_2014-remote/items

So that’s about all I have for you today. Just to remind you, I’ve collected a ton of articles and other telecommuting-related resources in this Zotero collection. Check it out.

So now, as we enter question time, I leave you with a photo of my two favorite coworkers. Because, yes, I practice what I preach and am a telecommuter.

Now: Questions?