


How to run an engineering focused privacy review program

giles.douglas@gmail.com

PEPR 2019

Introduce. Goals of presentation

A dark, blurry photograph of a city street at night. The image features a series of out-of-focus lights in the background, creating a bokeh effect. The lights are primarily white and blue, with some red and orange lights visible in the lower portion of the frame. The overall scene is dark and atmospheric, suggesting a nighttime urban environment.

Why engineering focused?

Why do we want this to be engineering focussed? There are lots of methods for improving privacy for endusers. Engineers and product managers are the people designing and implementing products, and we need to communicate nuance effectively to them. Privacy isn't easy to get right - we need to collaborate.



Collaborate with people designing and writing products

Influence early

By collaborating with people who are working on the end product, we can embed in their product lifecycle, and guide teams early in the product process. Ideally it is much cheaper to do this in the beginning than at the end of the lifecycle.

This worked better than pure compliance or program focus



We tried other mechanisms to achieve this - more compliance-y with checklists, more program management where we work with people to make sure they follow a process. Most products tend to be highly iterative, and only by collaborating are we able to get things right. Working directly with other engineers we found it easier to reach mutual understandings and to find out what the teams want to do - and can we change assumptions or implementations to reach a goal?

Need inclusive definition of engineer.

This is NOT “Software Engineer”

Lots of people are good at this!



This means we need to stretch our traditional engineer definition. This is not just software engineers - although there are lots of people who come from that background that are good at this - but we need to be more inclusive about people we need to work on this.

Diversity is a strength

Convince people they are technical



Having a diverse team working on this makes it successful. We need different opinions. Spend some time convincing people on the team that they really are technical. Don't fall into the silicon valley trap of designing everything for a hypothetical poweruser who lives in Palo Alto. There's a whole world out there and we need different people to do this. Backgrounds for people include journalists, lawyers, mathematicians, PhDs, crypto - runs the entire gamut. We need people who can challenge the assumptions and bring their experiences to bear



Need criteria to make people successful

What work product is expected?

It isn't necessarily code

This does present some interesting career problems - we can't just evaluate everyone by the same yardstick of "how much code did you write". We need to establish criteria for people to be successful and document what we expect them to produce. This /could/ be lines of code, but often it is designs, documents, policies, UI mockups, white papers and the like.

Where do I find these people?

They're everywhere!

Enjoy taking things apart

Want to talk to people and understand



You might think it is hard to find people that fit in this role (after all, there are only $O(100)$ people in this room). Fortunately they are everywhere, they just don't necessarily know that we are hiring for this - it isn't a common thing one would search for after all. Typical traits for success can be inquisitiveness - taking things apart is fun! - and people who want to talk and understand someone else's point of view. Empathy and Diplomacy are key. You can't just be entirely against online advertising and beligerant in viewpoints - privacy takes negotiation skills.

Vertical Team alignment



Right, so we found some people. How do we want to organize them? Well, one way to approach it is to vertically align your teams with product functions. Eg at google we could have a “geo” team and a “search” team. This allows some specialization in the different technology stacks that may be used, but moreover there is often different terminology that needs to be understood to be effective. It takes time to learn this. Having people vertically aligned makes it clear to partner organizations who they should talk to.

But also horizontal!



But we couldn't just solve this way. There are a lot of cross-cutting horizontal problems - for example anonymization, cookie handling, centralized logs, location services, payments - these are things that have implications on many products. So we also setup teams to become experts in these areas. This results in a bit of a matrix but with great program managers who can shepherd the process it does work.



This isn't just approvals

Guiding teams to
understand privacy

What are the roles of these teams? If we set them up to “just” be approvals, it isn't going to work. Their goals are to guide people to understand privacy and provide a resource to answer questions for the difficult issues. Ultimately this requires pushing down of expertise into the product teams - often we found that people would go join them as their privacy expert because they really were engaged and committed to a particular product. This feels bad as the owner of team - losing your best resources! - but ultimately it is actually better as it spreads information around the company. We need to empower people to do the right thing

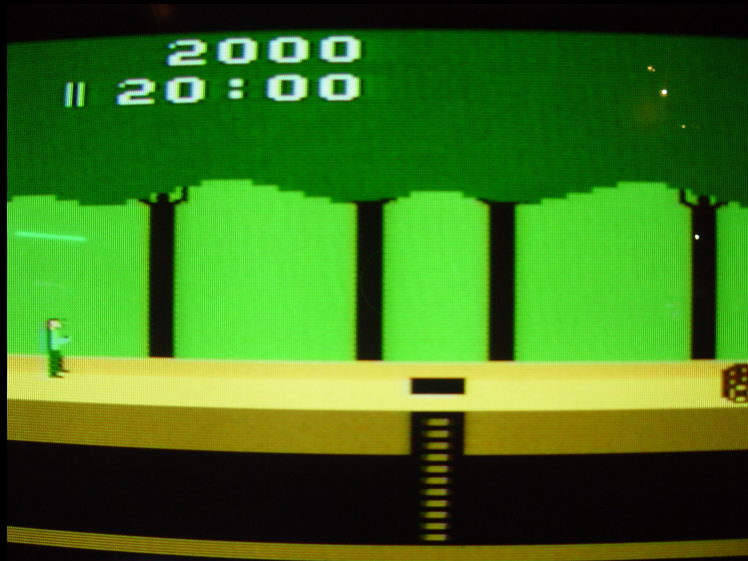
Want process to be repeatable

Want comprehensive review



With apologies for the headache inducing pattern - we need this process of review and advice to be repeatable. If teams are going to come to you as an expert you'd better be able to give them similar answers (I'm not going to pretend that we have to give identical responses because that isn't realistic)

Be careful of pitfalls



So pitfalls of this approach...? This isn't a pancea, and there are some things we observed that we needed to quickly correct for.

Explicitly State Assumptions

Don't give false sense of security



You have to state your assumptions to teams. Try to create principles or axioms. Much as Sha pointed out yesterday, the basis from which you are working needs to be clear - and it should also be clear to the team you are working with what should be expected, when it should happen, and what you are committing to do. It almost does turn into a legal function.

We don't want to give a team a false sense of security either (much like the photo!) - they need to know what you have guaranteed. Having someone say "well, this must be ok, the privacy team gave me their approval" is a dangerous path.

Reviews by nature are
subjective

But objectivity is key

Establish norms and how
to communicate



Reviews are always going to be subjective, and often you can end up in contentious situations. We need to try to be objective, and we can only do that and focus if we establish norms. It is great to standardize how to collect information, but only do it in ways that are low cost to implement. Getting in the way of someone's development process by imposing form t574 that you need to fill in isn't going to make you any more productive. Don't celebrate catching people outside of these norms - help them! There's nothing that says you can't go in and collaborate to fix something.

Work around contention by establishing and leveraging relationships with the other privacy stakeholders. Legal should be your bff.

Extract patterns

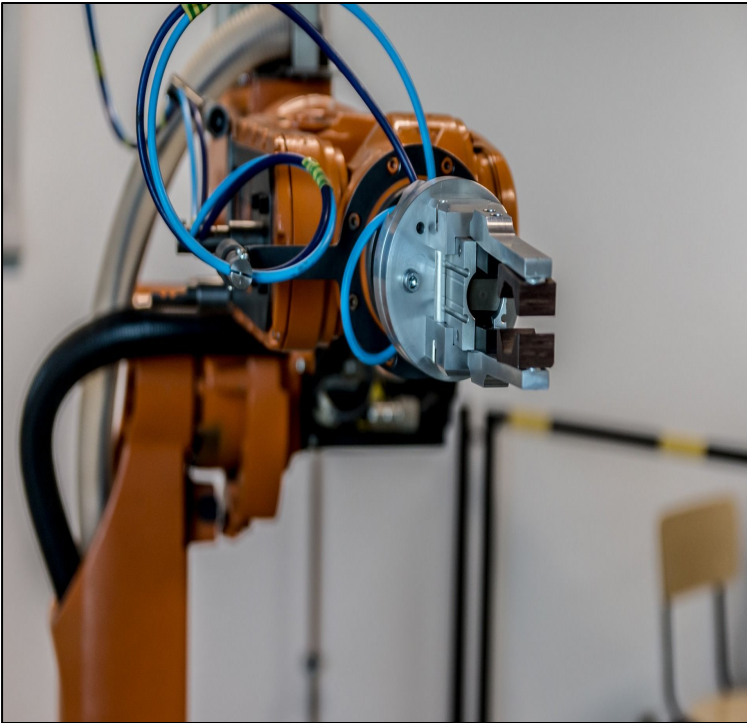
Find infrastructure opportunities

Clearly explain escalation



Engineering types are good at this because they are good at spotting patterns and finding infrastructure opportunities. No one really wants to do the same thing over and over again. This requires time to make these changes, so need to be careful to prevent overload/burnout of team by everything is a crisis.

And moreover, sometimes you are going to have to agree to disagree with someone and escalate a decision. This needs to be clearly defined.



Automate where possible

Automation is your friend. Do you really want to check that Dave needs to have access to the account system every time he asks? Can you codify this? Can you verify this? Whilst it is great to check things at the time you create them, you really do need to have mechanisms to ensure that the truly risky/scary parts of the system are not going to randomly fail in the future. This doesn't have to be immediate notification - but plumb in ways to find when it happens and remediate.

Make it measurable

Volume of engagements?

What do people think?



And finally, we really do need to measure. But what do we measure? We ended up looking at volumes of engagements, and simple surveys to say what worked and what didn't. Did we change your product? Do you feel more comfortable? This is a metric for measuring the process, not the individuals on the team.

Questions?