

How to not get overwhelmed

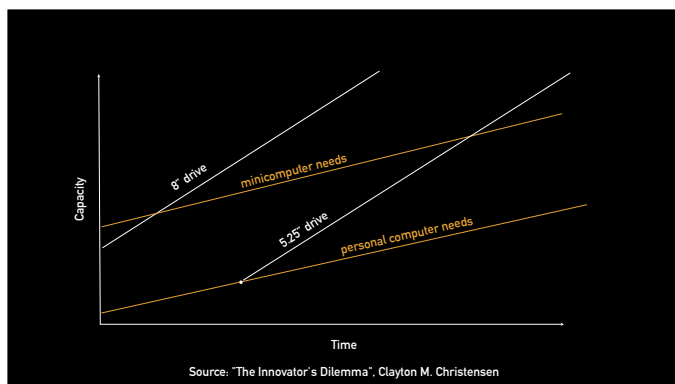
Maria Farrell's talk, on rewilding the internet, was the perfect opening for this meeting. Everyone's been talking about it since.

Here's the thing: over the course of the meeting, as we've been talking about things we can do to help, I've started hearing a lot of reasons why we *can't* do things. Company A is too big; all my family and friends use app B; we'll never be able to prise control out of the grip of Platform C.

And it reminded me of a talk at a RIPE a long time ago, by someone who looked a bit like me, but had a different first name and her gender wasn't quite finished yet, which touched on Disruption Theory.

Not "move fast and break things" disruption. This is about how huge, unstoppable incumbents can find themselves entirely obsoleted by competitors that are small, strange and *worse*.

The lesson is: there are things large companies can't do, even if they want to.



Here's what disruption theory is. Let's talk about hard drives. Time on the X axis, drive capacity on the Y axis.

[BUILD]

Clayton Christensen saw that minicomputers typically ran using 8 inch hard drives.

[BUILD]

Over time those drives got bigger, and they got bigger faster than minicomputer customers needed.

[BUILD]

Then the manufacturers of those drives started to develop smaller, cheaper 5 1/4 inch hard drives, and tried to sell them to their existing minicomputer customers. The minicomputer industry wasn't interested. Why would they use something that was slower and crappier than what they have? Yeah it's cheaper, but it's not cheap enough to make a difference to the cost of a minicomputer. Their existing customers wouldn't stand for it.

[BUILD]

So that left an opening for a bunch of small companies to make personal computers. Compared to minicomputers? These things sucked. You couldn't use them for serious work. They were practically toys. And they bought the 5 1/4 inch drives.

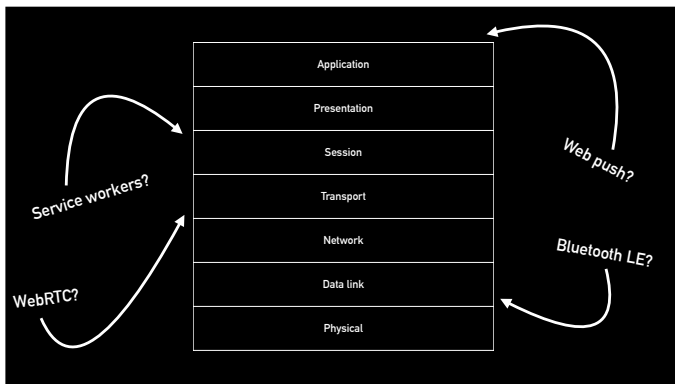
But what happened is that personal computers got better, *faster* than minicomputers got cheaper. And that wasn't a technology issue - that was a business choice. The minicomputer makers were trapped into chasing higher profits per unit, and so kept trying to improve their own product, beyond what was useful to their customers -

[BUILD]

until the PC got good enough for those customers and they all switched.

Disruption is this: when a company gets big and profitable, there are *things it becomes incapable of doing*.

But we know all this, because we've done this too.



Remember that we are networkers. We work in layers of abstraction.

Geoff Huston said that the reason the internet took off wasn't because it was the *first* way to send messages or pictures or even video over the phone. It was because, once you connected your modem, you no longer needed permission from the phone company to run any application you liked. We built a layer that was independent of the layers below, and which others could use to build the layers above.

But there are more layers out there than in this model.

The last couple of years, if you haven't seen me around, it's because I was spending some travel budget on going to software development events. You learn an enormous amount about our networks from seeing how developers use

them. The end-to-end principle, *at the IP layer*, isn't that important to them. They have their networking model, and it looks like networks of servers or containers, fronted by load balancers and DDoS protection, serving client networks covered by NATs and NAT-PT devices.

And yet... because everything is layers of abstraction, developers have still found ways to pull off global end-to-end communication. Innovation is still happening. A great example is web push. This is a regular web standard.

[BUILD]

A web page can, with your permission, wake up your device and send you a notification. Then the web app in your browser reestablishes communication and you have an end-to-end link again. That's way more straightforward and power efficient than trying to maintain an always-on IP connection with every device on the planet.

We don't need to fix every layer in order to fix one layer. That's literally the concept of TCP/IP.

[BUILD]

And outside of this room, there are way more "layers" than we think of in the OSI model. Web standards alone move way more quickly than anything we do. In fact, if you're not already following web technologies: start. Not because you'll be implementing them yourself, but so you can understand what modern applications look like and how their network needs keep changing. Last Monday, the same day Maria gave her talk, a new version of Safari came out, with support for declarative web push, which is an enhancement of that web standard.

I got interested in crazy things like web technologies, and aviation safety and improv theatre, because I think that the very best results happen when different disciplines crash into each other. Which is a thing that large companies just can't do. They become a really amazing machine at *their specific discipline* and are incentivised against following up on things that might lower revenue.

But I need to talk about the dirty secret of public speaking...



It doesn't really DO anything

Public speaking doesn't really *do* anything. It feels like you're doing a lot by standing up and giving a talk that gets huge applause (thank you in advance) and people are talking about it - but talking isn't doing. And it's very rare to give a presentation that actually gets people to stand up, en masse, and start doing things differently.

Maria's is the closest to this that I've seen in a very long time. I heard the reaction in the room. But if we have spent this week persuading ourselves that we - the people in this room - can't find any way to help, then that's an enormous lost opportunity.

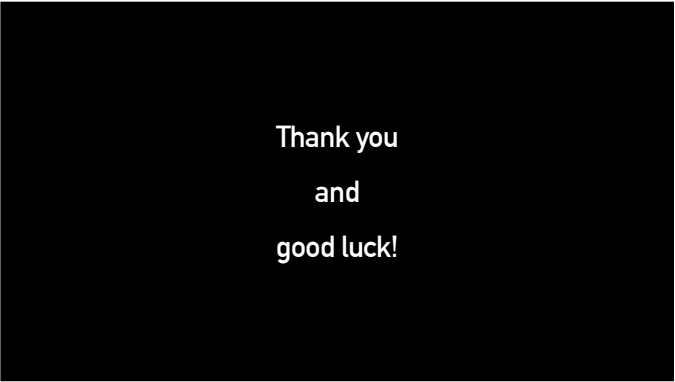
I shared a stage with Maria nine years ago and I said this: for the specific things that we do, it's not just that we CAN help - it's that we are the ONLY ones who can provide THAT help.

And any one thing we do will not be enough. But remember what else Maria said. Building an ecosystem is not just about planting seeds. It's not just about growing your patch. You build ecosystems by *connecting them to other ecosystems*. And there are so many interesting ways to connect ecosystems these days.

If you have something in mind that you want to do, and if it feels hopeless because the biggest companies in the world have hired so many smart people to just dominate that whole sector -- there are things they are incapable of doing, *even if they want to*.

Start by planting your garden, and then -- whatever you do -- connect your garden to others.

We're counting on this. Good luck.



Thank you
and
good luck!