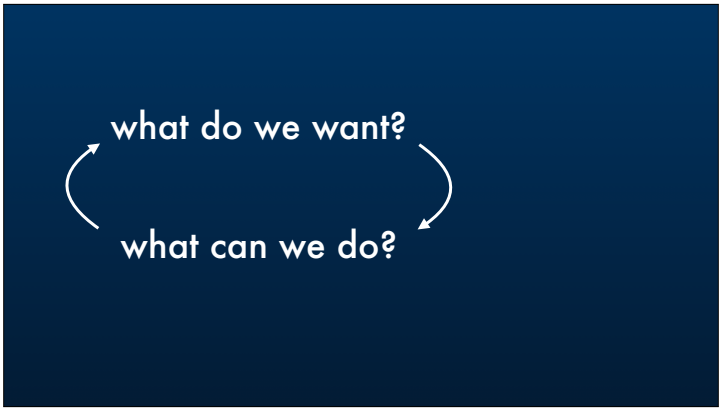
I'm Anna Wilson and I used to be a networker. Now I'm a junior developer.

CLAW is a workshop about crisis management. It's not just the exercise, but the big exercise at the end of the event is the centrepiece. How on earth do you pull that off when people can't travel?

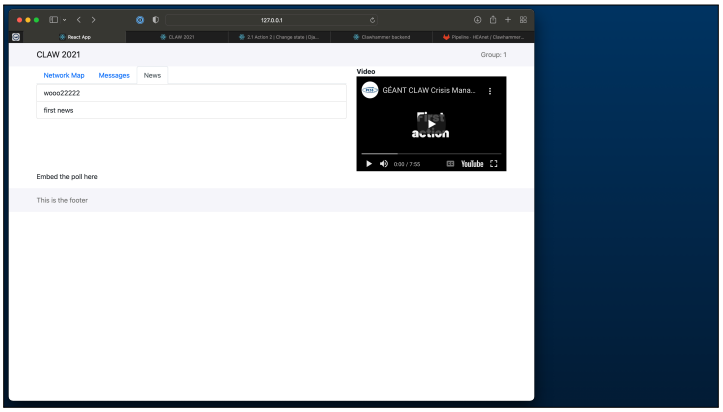
Last year, we ran a "choose your own adventure" thing, simplified so that it was straightforward to run. This year, we wanted to do something a bit more sophisticated, less linear, and definitely something more powerful than a Zoom call. For that, you need some sort of tooling. For THAT, you probably need to write some software.

But we're a team based around running an event. Not that much dev experience. Hard deadline. Risk is HIGH.



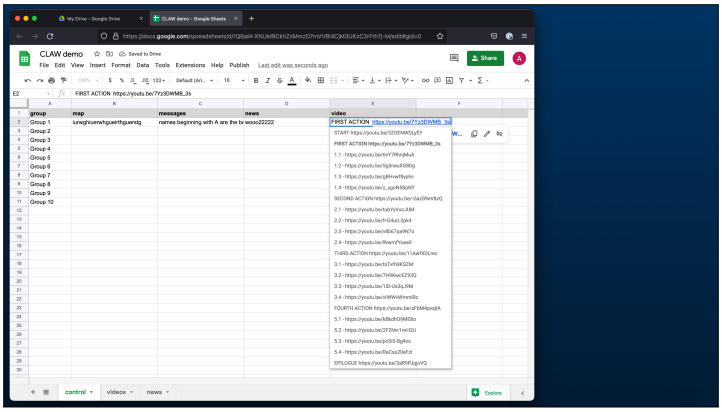
So here's the problem we're trying to solve. I'm willing to write some code, but I need to know what we want. Charlie and the team are ready to design the exercise, but need to know what we can actually pull off.
To break out of this, I make my initial proposal:
What we need is a web app, that we can push documents to the page. So a team makes a choice, and then we reveal documentation or a news report, or a video or something.
Charlie goes great, but what does that

look like? And I go — give me half a day to work on this.



Web dev has changed a lot in the last decade or so. There's enough abstraction now that you can get started REALLY quickly and get the shape of something in literally a few hours. Don't need to worry about wiring it up, or getting functionality right. A few hours with React and Bootstrap was enough to get a sort of interactive screenshot. Where I wanted to load in content, I could stick it in an iframe. Literally just embed a page from somewhere else.

What I didn't need right now was a complex backend. Everything might get thrown away. So at this point, the backend…



…was a goddamn spreadsheet.

Yep. A spreadsheet with a bit of code to turn it into JSON format. That's all we needed at this point.

This really paid off. If a picture tells a thousand words, this was worth a thousand meetings. I could not only show it to Charlie and the team, but they could play with it and already get a feel for what the end user would experience. Which meant we got straight to…

---

we don't want this:
➡ only lost 0.5 days work

we can use this:
➡ very quickly iterate

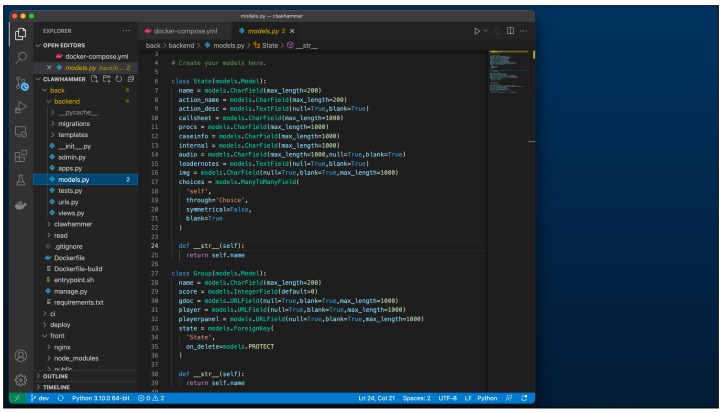
…if this was the wrong direction, we'd know really quickly.
If we wanted to keep going, it meant that Charlie could easily see and evaluate the risk.

---

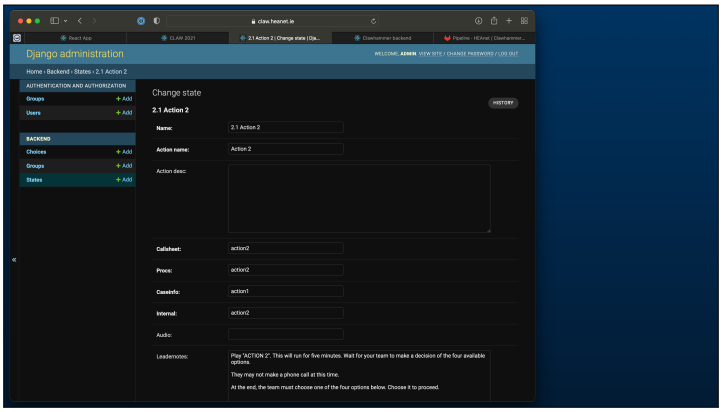
now we do this lots of times

Now we do this lots of times.

I'm still working on this like I'm in a hurry. What's important right now isn't that this is sustainable. We're not a dev team, we're an event team. What's important is that we discover what we want. Like, we probably do need a backend.

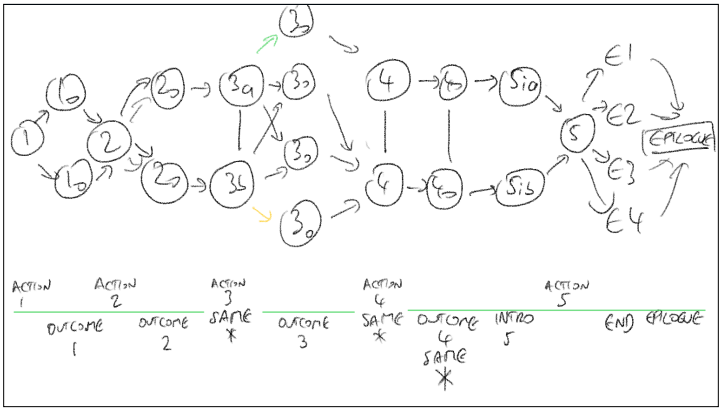


Fine. Let's bust out Django. Why Django? Because step 1 is install the software,
step 2 is define a couple of models for the data I want to store…



…and before I even get to step 3, I already have an admin interface where I can enter data.

And now, the logic of the whole thing is starting to become really clear.
I can go back to the team and say, here's a design constraint.



I'm gonna have these things that I call states. Every group has a state, and I'm gonna need you to design the exercise around moving them from one state to another. So now we can talk about what a state is, and how it impacts the design. If someone makes a choice early in the exercise, and we want that to have an impact later on, what does that do to the stuff in between?

So now I never do more than about a day's programming without going

back and having a quick meeting. And the meetings are GOOD.

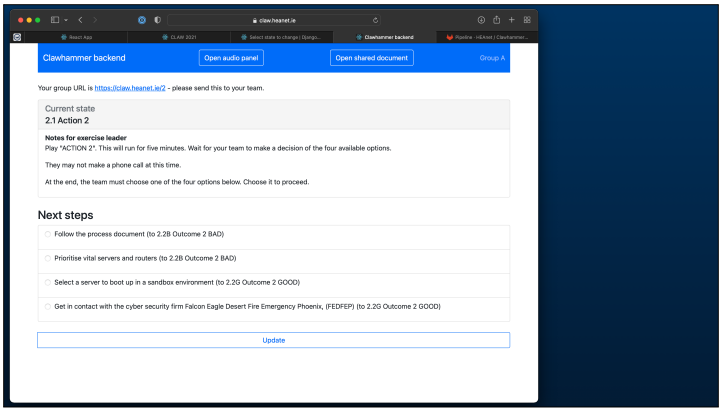
I know how to progress,
in detail

Team knows what capabilities
we have that are useful

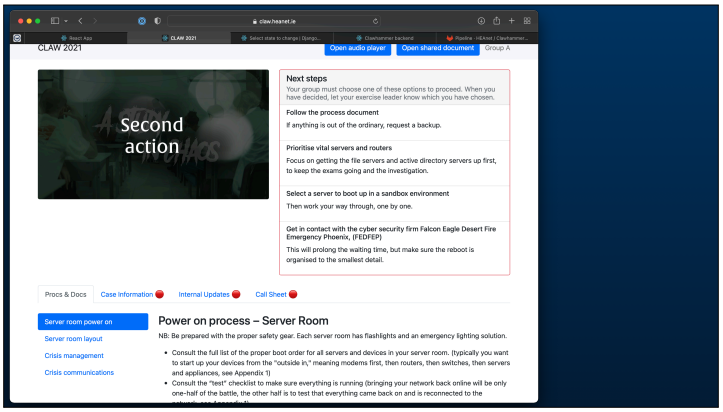
I come out knowing how to progress - so I need the team to put themselves in my shoes a little bit, and learn a little bit about the technical code

They come out knowing what capabilities they can use - so I need to understand the work they're doing, and put myself in their shoes.

THIS IS THE SECRET. THIS IS HOW YOU PULL IT OFF.
The coding doesn't really matter. It's a coin flip on whether it'll ever work. But if it's ever going to work with the resources you have, this is how you make it work.



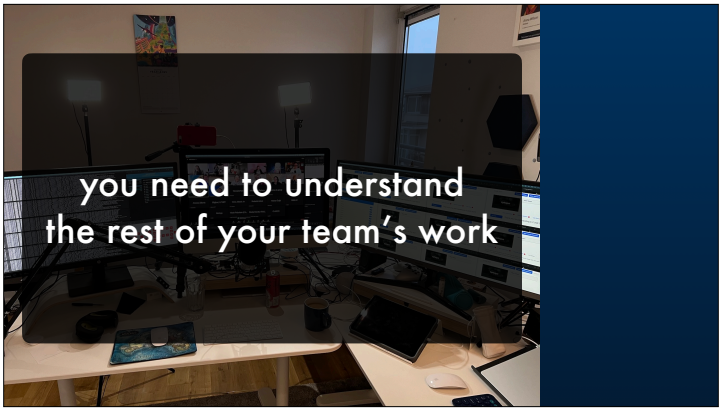
And we kept this going right up until the days before the event. Our first run through, I made a really bare bones backend for the exercise leader, and it was a shitshow. He was just way overloaded. So “we learned a lot”, right? But in this case, yes. I knew exactly what I needed to do to support them.



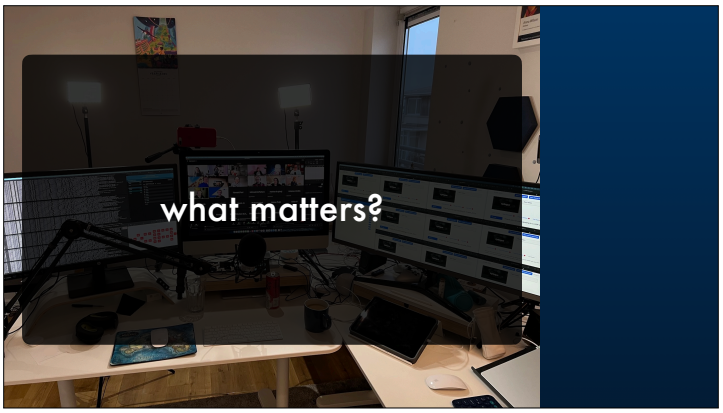
I was making a bunch of changes really late in the day, but at this point I had a lot of confidence in them.

And on the day itself, it worked! I was on call to hop into rooms to help troubleshoot, I had basically every screen I owned pressed into service for this, but the beefed up backend empowered our exercise leaders to handle most of what came up, and they had a ton of fun.

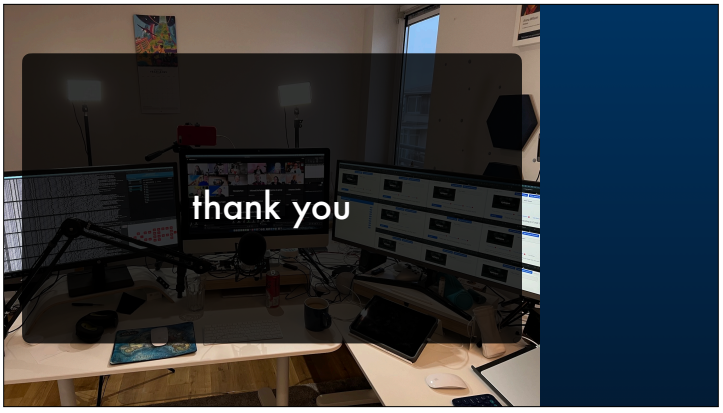
So, what did we learn from this?



Development is not a separate job. You need to have at least a rough idea how to do you colleague's job - and you need to be able to teach your colleague the very top of level of how to do yours. If you're a developer, this is about learning what's really important to getting the job done, and what's not. If you're not a techie, this is about being patient, not being scared, and learning to explain what's important to you in a level of detail that doesn't come naturally.

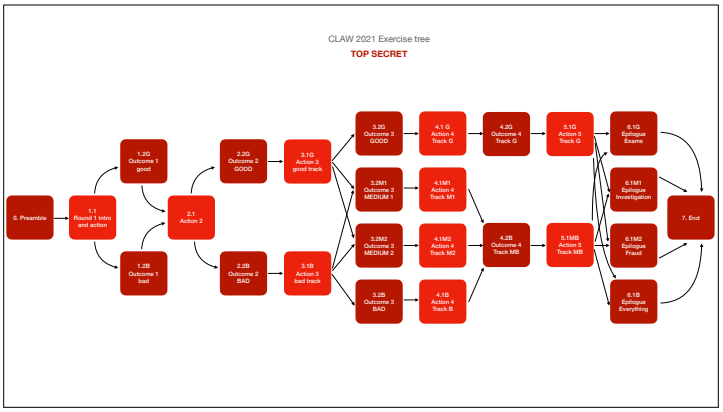


Usually people go “oh we spent time developing something! can we reuse it?”
in this case: no!
that wasn’t a priority! we’re an event team. it’s not unusual for us to spend ten workign days on something, use it at the event, and then throw it away.
BUT if this stuff is useful, now we have loads of information to sit down and work out the resources it would take to make somethign reusable.
which, in my experience, is the kind of information that managers LOVE to have.



Usually people go “oh we spent time developing something! can we reuse it?”
in this case: no!
that wasn’t a priority! we’re an event team. it’s not unusual for us to spend ten workign days on something, use it at the event, and then throw it away.
BUT if this stuff is useful, now we have loads of information to sit down and work out the resources it would take to make somethign reusable.
which, in my experience, is the kind of information that managers LOVE to

have.


CLAW 2021 Exercise tree
TOP SECRET
0. Preamble
1.1 Round 1 Intro and action
1.2G Outcome 1 good
1.2B Outcome 1 bad
2.1 Action 2
2.2G Outcome 2 GOOD
2.2B Outcome 2 BAD
3.1G Action 3 good track
3.1B Action 3 bad track
3.2G Outcome 3 GOOD
3.2M1 Outcome 3 MEDIUM 1
3.2M2 Outcome 3 MEDIUM 2
3.2B Outcome 3 BAD
4.1 G Action 4 Track G
4.1M1 Action 4 Track M1
4.1M2 Action 4 Track M2
4.1B Action 4 Track B
4.2G Outcome 4 Track G
4.2B Outcome 4 Track MB
5.1G Action 5 Track G
5.1MB Action 5 Track MB
6.1G Epilogue Exams
6.1M1 Epilogue Investigation
6.1M2 Epilogue Fraud
6.1B Epilogue Everything
7. End

rrrrr
