MapCamp 2021 - Challenging Assumptions

Okay. Hello and welcome to the firms of our sessions today. Quite a long day, but hopefully a really good one I've been into. Mind expanding for provoking sessions this morning, we are the resilience track, um, were just about to hear some talks around the, the theme of challenging assumptions. Um, prior to today I've done adaptation.

I've been in one group around adaptation. Um, and then it's really just been. Energetic today. Um, we've had a real conversational feel, um, to the actual talks themselves and lots of audience participation. And I'm hoping that we can create the same experience for you today. So we've got four speakers, three talks, and we're having a bit of a tag team effort for one of, uh, for one of them, uh, which is why you've got four panelists in front of me.

Um, I'm going to leave everybody to introduce themselves and give us a bit of context as to the reasoning behind their choice of topic. Um, perhaps a little bit about themselves, what they do, um, to give a view on what they do in the world that makes them care about this with a broad theme is challenging assumptions, and we're going to start with Lisa keel.

Nia, um, describes herself as a lean and agile consultant and sinner can effin, um, never get that right every year, connect an addict. And the hoping with. Kind of jacked up format. We're going to have some time to talk a little bit about that as well. There's can you introduce yourself, tell us a little bit about your talk.

Sure. So, um, I want to talk today about where assumptions come from. Um, as you heard, I'm a Canavin addict, I've got more experience with Connecticut and the Waze mapping that I do have more experience with Wardley mapping than I did last year. I've been at every map camp. I'm a huge fan. Um, I said I'm leading the national consultants.

I'm actually going back to being a developer for the first time in a very long time this year. So I'm super excited to be getting into the code. That's probably all you need to know about. Um, and the rest of you'll get from my talk. Uh, so I sent out a tweet, um, early in the summer and somebody I had done coaching work with, reached out to, uh, brought me into their organization.

So I've been, I've been doing it for a week. Uh, so far I haven't written any code, but I have run things from the command line and that's making me really happy.

Brilliant. Looking forward to it. Thank you. Okay. So Sarah wants to talk about where assumptions come from. Um, I love it. Biology where words come from. So, um, it means to take something upon yourself. Okay. Um, you can also think of assuming something fictitious, C2 based, assume a costume assume a role. Um, and there's something about when assumptions aren't necessarily true.

That's really interesting. Well, I wanted to use Chris Argyris, his ladder of inference to talk about how these assumptions come to be. And the first thing that happens of course, is it's we act, and we act as we act, as we look around us, we understand our environment and that's our behavior. Um, hopefully as adults, our behavior is based on the whole letter and there'll be some stuff that's, you know, your system to seeing data accurately because we observed the data.

It's probable that like most of all, most of our stuff is actually based on the things further down the letter. So the first thing we tend to do is we'd surf the data. Um, we look around us, there's way too much information in the world for us to take care of everything in it's some favorable to us, but we filter the data through our senses and through the bias that our senses gets.

And I'll talk a little bit more about that as well. From what we see, we make assumptions. We generate theories hypotheses about what might be going on. Um, from that we see they conclusions. We start seeing patterns and what's around us. Um, from that we, we create the conclusions and from that comes our beliefs.

Okay. So our beliefs are our understanding of the context in which we live. And the really interesting thing is our assumptions don't need challenging when they fit the context. They're only really a problem when the context is wrong. So, um, they've, uh, they snowed was saying they, they started talking about these things as cognitive heuristics rather than cognitive bias, because most of the time, these heuristics are actually really helpful to them.

The problem is that we filter data based on the beliefs that we've generated. And the confirmation bias can lead us to continuing to hold those assumptions, even when the context is inappropriate, even when it's changing. Okay. So this is why we need to challenge at least some of our assumptions all the time, because the way in which we filter data Israel.

Now, of course, some of our behaviors are very visible. Some of them, we probably wouldn't behave that way in front of other people, some of our beliefs. So espouse, we heard people say, you know, I believe Jesus Christ is the son of God. I believe I'm a good programmer. Um, and some of the things, a lot of the things we believe we don't really think about, and some of those beliefs will be wrong for anyone who thinks that, you know, oh, I ran rallied somebody once actually, who said, you know, I don't have.

Any superstitions. I don't have any wrong beliefs. I said, um, are you married? He said, yes. I said, would you sell me your wedding ring for an identical wedding ring plus a hundred pounds? Because a rational person, especially when married to a rational wife would be very happy to do that if he would no, because his wedding, we have meaning to him.

It's just a piece of jewelry. It's just piece of battle, but we all have these polices. So hopefully that helps you to examine where your beliefs come from. We all, should. We have invisible beliefs, we have visible beliefs. Um, there's some lovely things. Organizations do this too. So organizations will have espoused values that they say they hold and then they'll have how people actually act.

Um, I've was reading about a values-based hiring and firing where if somebody doesn't actually holds your values, um, you will let them go. And if you don't fire them based on your values, you don't actually hold those values. Cause they're got to. So we have festivals values, invisible values, physical beliefs, invisible beliefs.

And of course everything being about these sort of rights. So unless we felt a bit house, unless we unlearned things and those filters have declined, um, you can think of those beliefs as being enabling technology, right? For the rest of what we do with our lives. So, because we've got those cognitive biases, this is true of habits as well.

By the way, habits are basically actions that we no longer need to think about. We've done them so often. Um, everything everything's there allows us to do other things with minimal energy expenditure. So they're enabling technology to allow the new things to come in. Okay. So there you go. I thought that no, you know what?

I've had the latter on its side. So I'm going to show you a little chart and this is, this is not based on any science. This is just in my head. Somebody will come along and tell me I'm wrong. That's okay. This is how I think of how much energy I spend looking at different situations. So I, we spent quite a lot of time thinking about what will probably happen.

And we gradually shift all of that stuff over to the right and it forms up at least. So we'd already done the thinking. Eventually it dies away, but we've spent a lot of time getting that stuff into our head. We don't tend to think about the stuff that might happen like global pandemics for instance.

Okay. And of course, when those plausible things happen to us, because we're not prepared for them, it has a really huge impact. So this is your black Swan. The really interesting thing is we kind of know this happens. Things go wrong, but we didn't expect bang. Now we're in trouble, but we very rarely think about the things which are plausible that might go, right.

I hear so many times people looking at what they might do in a situation and going, oh, that won't work, but you haven't tried it. You don't know whether it'll work or not. And this is one of the big reasons why we need to challenge our assumptions, not just because of the things which might go wrong, but because of the things which might go right, that we're missing out on, and which also might have an extraordinary impact.

Okay. So I want to show you some tricks for looking at what might plausibly work, not just what, what might probably work. And of course, you've heard earlier, I've got a cadet in addiction. It's a problem. I need to deal with it, but I can't get away from it. So there you go. Now, you, you, if you have a cup of cross connected, be careful, um, you will get hooked.

Um, go look it up. I'm not going to introduce the whole thing today. Cause I haven't got very long. Um, what I will say is there's predictable things clear and complicated, predictable things have, you know, outcomes you can see coming. And the context in which you understand those outcomes is well understood.

It's not changing, um, chaotic stuff. It's actually this in emergency your house, burning down. It resolves itself, not necessarily in your favor and it sells itself really quickly. Um, I will talk a bit about how to use chaos in a positive way, but the really interesting thing is the complexity. So human systems are complex.

Human beings are complex and that means stuff emerges. Okay. So we can see with hindsight how something came to be, but we couldn't possibly have predicted it. Um, and th th within the map, if you, if you think if your walls that it's kind of obvious that the custom built in Genesis stuff, Genesis actually often starts in chaos, chaotic, you know, urgent ideas.

Um, but the custom built stuff's definitely probably more complex. Um, and then gradually moves to something stable and predictable as technology evolves and becomes. So, um, you can kind of see that there's a really easy mapping between more thematic connection, but the movement in the maps is also complex and particularly getting through the inertia is complex, guessing through a nurture is a wicked problem.

That's also cultural. So the movement in the map, the development of these beliefs is complex too. Okay. So I'm going to show you how we handle complex stuff. And this is hat tip today. So who taught me all this stuff? Um, he says we have to have a portfolio of probes. So a probe is something where you try something out that is safe to fail because we're not in chaos.

We're in complexity. We have time to try something else for case. So that's the idea. If you're already chaos, you have to act and act really quickly. It might be safe to fail, but you've got nothing to do. But if you're in complexity, you can try things out and you can see whether it's working or not.

You've got some space to do that. So what you want to do is you want to have multiple concurrent probes going on. That's the idea. Okay. So we're going to have several different things we're trying out at the same time. Um, it's not like an experiment where you need to isolate the probe because you're going to repeat that experiment, gave him the game and get the same results in complexity, trying to experiment, changes the context.

Um, if I put somebody in there, poke somebody, they'll poke somebody again. Eventually they turn around and go, well, you stopped doing that at me. Okay. So context has changed me, poking them twice changed how they felt about the doing that. So, um, the context changes we can tell in retrospect with hindsight, which probe is having which effect so we can try multiple pros concurrently.

Okay. So pro pass to have five criteria. You have to know how it's know whether it's succeeding. How are you going to tell whether this is working? Uh, you have to know if it's fairly might not be the same thing. There'll be the opposite of the thing that you look at for success. Don't know. Um, if it's succeeding, you have to have a way of amplifying it.

That means making sure it carries on succeeding or make it succeed at a bigger scale, maybe, but you have to be able to amplify it. Otherwise it goes away. Um, if it's fairly, you have to be able to dampen it. So stop the effects from happening, um, which is why we recommend, you know, try something else with a small scale, try it with fewer people, try it with less time, less efforts, et cetera.

That's a general heuristic for safe to tell probes. Um, Uh, so we're succeeding with fairly waive and flying. It's a way of dampening it. The fifth thing you have to have is coherence, which is a realistic reason for thinking this probe is a good idea. It has to be plausible instead of predictability in the complex space, we have disposition, which is how likely is it that this idea is going to work.

So the probable stuff is really well disposed to work. The plausible stuff is less disposed to work, but still safe to give it a try. So you have to make sure really rigorously that that stuff really is safe to fail. Cause it might well fail. Okay. But if you've made it safe to fail, then it's okay to try it.

Right. So it has had this realistic reason for thinking. It's a good idea. Dave says a sufficiency of evidence, the progress I always say, can you think of a scenario where this works? Because if you can, it's possible. If you can't, you probably want to try something. Okay. So that's your probe portfolio? Um, there's some really nice ways of coming up with this.

One of which is this shallow dive into chaos. So can Nevin has a number of dynamics in it. And one of them is when you're, if you're stuck, can you can't come up with these good ideas, try separating people. So in chaos, um, you've probably heard that saying always put your own mask on before you help somebody else.

Right. We're separated. We have to act for ourselves. We can't connect to each other. We've got nothing to lose, but it's also means we're not biased. So really easy ways to do shallow David's Kelsey publicly done it, right? Brainstorming on post-it notes. Um, so that you're, you're not affecting each other.

We've done things where we taken three developers. I was one of them and we independently came up with a new user interface. As an idea, we actually ended up keeping turf those in that situation. Cause two would go to them. We refined them. We brought them back into complexity, refined them, refined them, worked on them until they worked and were really good.

Um, I learned a lot about pioneers and the quality of that coat. Uh, I am a settler. I like seeing codes. I reset to for fun, but not that guys. It was, it was dusty coat. I ended up rewriting bits. So lesson, if you're going to get the hockey pioneers to do this stuff, free factor it for refactors. If you've got somebody who likes to refactor that stuff or rewrite it, make it stable because it's got to end up in that stable place.

Okay. So we're going to take these ideas, refine them if they're not well disposed, you want to really rigorously, um, check. Are they good probes? Do they meet those five criteria? Um, de Snowden has this lovely thing called ritual dissent, which he uses. You present your probe to a group and they're really mean and attack it.

And they're not allowed to say anything positive because the idea is that it hardens it up. And also the ritual of only being allowed to say negative things means that people will more likely to say negative things. Whereas if they were allowed to say positive things as well, they might be nice too.

And that's not the idea. Um, so we're going to rigorously check that our probes are good solid pros. Okay. Right. There's a couple of other things that they suggest you might want to have in your pro portfolio. One of them is a bleak. That means you're not just addressing the proper director. You're looking at the ecosystem of constraints, the context around that problem and putting yourself in a better place where that context is concerned.

Okay. So. For instance, if I want to introduce, uh, uh, methods like behavior driven development is one of the things I teach. Um, it gets steps and tests this talking to each other and the deaths and testers are at loggerheads and they refuse to talk to each other. I will back off trying to introduce that and I might go, okay, let's look at the quality of how, how easy is it to get this thing to production?

Oh, it's really hard. Cause it's buggy. Okay, well, let's try fixing the bugs and eventually it might lead back to the desk and testers getting on a little bit better. All right. So I've not addressed that problem directly. And as a coach, I tend to back off anything where people go at me because it's just not useful to push that.

Okay. That's not what I was exposed to. You can tell right there, I'll go at something obliquely instead. And I'll usually have two or three of these going on. The other one is what's called NY. And that means to bring in an expert from somewhere else. And I'm going to use a map to show you what, what happens when you do this?

Okay. So we started a map where the customer needs this very abstract map. We start with the customer need and the customer wants to be able to do something with the thing you're producing for them. Okay. There are a bunch of other people who usually also want to be able to do things so you'll have some security people, or they want to stop people, being able to do things.

That's usually what the security people want. Um, maybe I want to be able to maintain this code in five years time. Um, the auditor wants to be able to trace whatever we sold to the customer. We, we need to be able to provide proof to the regulator. There's a lot of things that we need to be able to do.

And there's lots of different components. When we think, look, start mapping these components, we can think about what capabilities they enable. And I found this is a really great way of actually getting started with maps, with people who aren't used to it. What are you able to do with these things? Okay.

So we can start thinking about who else could you set? So rather than looking just at the customer needs, we can look at what other customers we could potentially service. So, as an example, um, I am an online book seller. Um, I sell online books. Who else can we use this book, the selling bit of this thing for well, where I'm a sense.

So we're going to use it for whoever wants to buy everything else. And we've got this bit, which allows us to scale our computing, who might want that. Who, what about that? That's made KWS. Okay. So this is called acceptation the reuse of things for purposes, for which they would not decide. So you can do something that Dave Snowden calls managing for serendipity, which means introducing a bunch of experts from other domains.

And looking at what they see when they look at your capabilities, when they look at your map, where they look at what you're doing. So you might introduce an anthropologist or as Sue keeper, a veterinarian. Um, they've always used to seem to use animal people, examples where they do this. You might have introduced somebody who builds aircraft until it codebases see what they pick up from it.

And this is managing for serendipity. Okay. So, um, Mel said, come my set, a lovely to go earlier. We need to get people who aren't talking to each other to talk to each other. And I thought that was just so. So we need to do this with ourselves as well. We need to introduce experts in different belief systems, diverse perspectives.

If we do this and it doesn't necessarily have to be people we like, but we can get plausible ideas about when we meet might be going wrong. Um, and I just wanted to call out. Simon has been talking a lot about learning from China, right? And I know China's got some human rights record problems, haven't we all, but there's a huge amount that we can learn, right?

So it doesn't necessarily have to be people we agree with, particularly people, not people whose belief systems we necessarily share. Otherwise we don't get that diverse. So I just wanted to finish with, um, some observation of data for people who like data in changing contexts. You won't like this one. We are now at three, 413.7, eight parts per million CO2, which has 0.2 up on my last talk.

Thank you very much.

Thank you, Liz. Start again. Sorry. That was brilliant. That was really, really good. Thank you. Um, so first to the panel, before I say anything up to the floor thoughts, um, there's so much in there, there's a theme around black swans. I want you to comment on, but first from the panel, what did. Absolutely loved it.

Um, uh, I've always been a fan of lists and I'm still learning stuff. Um, so yeah, I, I, I'm taking away that props and experiments are very much different things and I've struggled with that. Uh, could you, I don't know, more elaborate on it more, I'm like eager to learn more on the difference. Uh, sure. So experiments are repeatable.

Um, in experiments we seek to understand what we can do in the context. Usually a physical reality physics doesn't change. Um, however much we want my wants it to, um, so we should be able to take it, take an experiment, do it, do it again, do it again. Get same with. When you do that with people, for instance, in a complex system, you will not get the same result.

There's a lovely thing called the Hawthorne effect, where basically they went to a factory and they watched people. They said, do these people work better with the lights turned up and they turned the lights up and everybody worked better. What faster? And they said, okay, let's do the opposite because obviously we're scientists and we should turn the lights down.

And they turned the lights down into another factory and everybody worked faster because they were being watched. Right? So you, you get, you get things you don't expect. You often often get the opposite of what you expect in a complex system, and it might still be a good thing. So, um, you have to be really super curious about that correlation with hindsight as to what it is, you're seeing have ideas of what failure looks like, ideas of what success looks like, but don't get hung up on them because the real outcomes will emerge and they might not be what you actually.

Um, interestingly just does actually happen a lot insides. There are lots and lots of, uh, products that have come from experiments where instead of yes, that works or no, that doesn't work. The, the result was for that's funny. Um, you know, I got a whole talk on acceptation, um, of things that came about space invaders is my favorite.

Um, as the aliens cross the screen and there's fewer and fewer of them, it takes us processing power. So they get faster and faster that became a feature of the game. So there you go.

Anybody else before I jump into my own question and then, uh, quite a few that we've got in from the panel, um, Fred, um, I'm a bit curious on your view because there's, this mentioned that. There's a big benefit in being able to experiment without the pressure and the chaos and all of that, because it gives you that preparedness.

Uh, but whenever we head towards black Swan, we are usually in that chaos at it cannot always be prevented. And, um, I'm curious about how do you transfer that, uh, pleasant, pace, uh, simulation or experiment or probing into one where you no longer have that control, but the response has to still happen, uh, with a lot more uncertainty going around, right?

And the more uncertainty, uh, the more your assumptions and everything don't have to context in which they are valid. Sure. So, um, Dave will probably be able to answer this way better than I can follow all of his stuff. It's really good. Um, throw constraints around the problem or get yourself out of the problem.

That's literally what you have to do. Um, so chaos happens because of a lack of constraint. Right. Fire burns until it beats the constraints of no oxygen, no fuel, um, nothing else to burn, uh, Huber beings keep growing their ecosystem until nature stops us. Um, and it will stop us. Something will stop us, right?

We have, we've got to shrink ourselves back down. Otherwise we're going to hit those constraints really hard. The fire will go out and that's our civilization. So I think putting constraints around things, um, is one way getting yourself out. The problem is the other. You've got to buy yourself some space to do things.

And one of the really interesting things I found is that you can create, you can create safe spaces. Uh, psychological safety is really important. One of the ways that I see get psychological safety is by repeatedly empathy, all the stuff that's working, all the good things. People are doing, all the good things you see, just calling them out, calling them out, calling them out.

We're not really prying to see stuff going, right, because we don't think about it. It just filters into the belief system. And then it's gone. We see things going wrong and it makes us feel bad when we see things going wrong and then we pay attention. So you have to be very, very consciously hanging on to all the, and flying all the stuff all the time.

Right. And it's, it's a habits to get into. Um, I highly recommend it and it helps to build that psychological safety. And then people have a base of working stuff of which they can try things out. And they know that, you know, if it fails, occasionally it's fine. Cause the last five things were. And they know that they can test it, make sure it's safe to fail.

They know that people who've got their backs. They know that when somebody says, Hey, have you thought about this going wrong? It's not personal. Um, the other thing is cost, but talking about, because as we heard earlier from a few people, um, I have first had it from Jake bloom. Um, Simon mentioned it today.

If it's on the map that arguing about what's on the map, this arguing with each other. So again, it's not personal in that state. And that's just a couple of examples. Uh, I'm new to mapping. So it's so interesting. And I think I'm gonna just going to ask to ask your opinion here. You've got quite a few questions, um, because I know that there's a particular kind of flavor around the diocese talk, which is, um, around not mapping, right?

So having a problem that perhaps either using a different sense making framework versus, Hey, how could everybody help me with this? So I'm wondering, I want enough time to launch some of those questions together, but I wonder if you want to, let's just run through some of the questions. I think we've got four here.

If you're good at doing sort of one or two sentence answers, that would be great. And that just means we can kind of keep the question and announce contextual. Yeah. I can say a couple of rounds, water maps, identifying positive or negative black swans writing a sort of map as a component is it's a black Swan.

It's when the context has changed and that thing is no longer applicable. So what you want to do is be able to spot the weak signals, the plausible stuff. Um, I'm not going to go into detail, but again, um, listen to people's stories, listen to the unusual perspectives. Go look at Dave's stuff on that.

That's what I'll say. Um, the flex, your city stuff is really exciting. No, that's really good. Well, I'll, I'll, I'll jump ahead and ask one question because it kind of related to that, how frequently and how do you iterate this process of going back? So erroneous assumptions are not necessarily useless.

There were once, correct. Analyzing them gives you an understanding of what's changed how frequently and what's in practically, how regularly do you do that? Do you want to just be pulling up maps from six months ago and saying let's review every component? Um, how would you advise people to do that? Um, in terms of both cadence frequency, and then what, you know, how do you prioritize what you're reviewing?

Um, I think the only thing that I do regularly on a C for myself is try and amplify the things which I know I'm doing. Okay. Repeatedly. Um, we're so cruel to ourselves and it means we're always scared of doing things wrong. So by, by saying, you know, here are the things I'm doing right here are the things I'm doing well.

And thank you to everybody who sent me really positive stuff, because that really helps as well. Um, it gives me the ability to try things out and if I do fail once or twice, it's okay. Um, that's almost the only thing that I do on a regular basis. Um, as far as, uh, challenging the assumptions goes, it's an ongoing thing, depending on what kind of assumptions in what kind of place you're looking for.

And it's, it's not a discreet. I do it today. I do it at nine o'clock every morning. It's just an ongoing complex, fuzzy things. So for instance, um, I did a Harvard bias test. I found out I had a. Um, I thought old people were all they sick and useless. So I countered that with diverse perspectives by going and deliberately finding older role models, people who are still healthy and working and fit.

Um, and it's made me feel a lot better about my aging as well, which is really good. Um, I grew up with white privilege. I have various forms of racism, again, diverse perspectives going and finding those role models, following those people, uh, getting that, uh, different belief into my head. So, well, we have more questions I have now more questions, but we'll leave those to the end.

Um, I'm looking at actually our running order and I think we changed it so much. So you're going to last one. I'm going next. Surprised, sorry. My, my, uh, my bad for missing that, I'm really excited about this specific. So Messias was kind enough to step in after a change in order life. Um, you know, just life basically meant we had to rejig things a little bit, um, member of the community, not very active Wardley mapper.

So I thought this was a really interesting talk purely because it was almost, um, by direction in terms of the help we can give. And I want the, the aspire advanced, experienced mappers, um, to be thinking as we're listening on, how would you tackle some of those challenges or how would you perhaps add value to the process through Wardley mapping?

My feeling is that, um, map camp is a learning experience and it's also a great experience to be exposed to new problems, new mechanisms. So I thought this would be fun, retires, um, introduce yourself and a little bit about what inspired you. Okay. Alright. Um, I'd like to share my screen, but yes. Now I should be able to, um, if I can find it again.

Yes. Is that working? We're seeing your full desktop right now? Yes. Oh, my full desktop. Oh, well then you see my foot doesn't matter. Uh, well, is it the problem? Is it readable? Maybe I should try again. Let me just do that so we don't have to worry about it. How about now? That's perfect. Excellent. Uh, the metaphor is an alien crashes on earth.

Um, they have no ship, they don't know what to do, they need to navigate. So of course they draw a map. They don't know what to put on the map at this point. Right. If you go in and you know what you're looking for, you can, you know what you're mapping. Um, but this alien doesn't know. So they start. Whatever they see, right.

It could be three buildings, clouds, uh, roads, anything really, uh, cars. And then, um, as they are moving and using this nap map and changing it and trying to find a, a may, maybe going through the same place twice accidentally, and then they start realizing that some of the stuff is not useful on the map, in their context, right.

The clouds move, the cars move, uh, unless you're a meteorologist or, or a traffic, uh, person mapper or something, then those have no value to you. This, this is sort of, um, the metaphor I'm trying to use here to break you out of the idea that this is not worldly mapping, what I'm doing. It's something I started doing.

Uh, I think 2015, it's somewhat inspired in recent years by worldly mapping, but just to sort of break you out of that, uh, that context for, uh, for the next 15 minutes. So my name is , I'm a principal consultant at Atlin. We do software strategy, design and modeling, and, and that's sort of where, um, where this comes from, right?

I I've been doing domain modeling at first, uh, at, at the clients and, um, and, and design. And then you always bump into, you know, this age, old problem, you have these good ideas, but you can't make them happen for whatever reason. So I started doing sort of, um, gorilla. Trying to not knowing what I was doing was somewhat inspired by context mapping and domain driven design, trying to capture stuff, making things explicit and, and gradually that sort of grew into, well, I still wouldn't call it a proper method, but, but something in any case where I'm just mapping everything and these, these maps are huge and chaotic and only make sense to people who were actually involved in the mapping.

So they're not really useful artifacts for, for, for other, uh, So, and how I did while you work with whoever's in the room, right? That's a, that's a starting heuristic. Uh, you, you don't always have the right people, but you have some people who know something you'll never get a perfect map. I think that message has been around today already, but you tried to bring in sort of a very diverse group, uh, of, of, of people and roles and, and opinions, which, you know, Liz was already talking about as well.

Um, and you start with what they know, right? This is the most comfortable thing for people to map. If they can start, if they're a software architect, you know, have them map architectural components, they're product people, maybe it's, you know, features or capabilities, you know, organization, markets, time, money, all these sort of very, uh, easily understandable, uh, concepts and easily mappable, uh, things.

That's where you start. So you have sort of a frame to start putting all our things in, and then you start looking for, uh, relationships between these things, right? You connect things, you overlay them. Um, No, I'm talking about this in the very abstract, uh, for lack of time, but you try to build a sort of big layer map of things that, that are related to tech is related to the product and the product is related to the, to the customers and, and these kinds of things.

You start looking for the cow parts, right? How people really communicate and not, not the hierarchy, but the value creation network, uh, dependencies, how knowledge flows in the organization, um, or, or in the product or things like that. And then you start, you know, the real meat is when you start looking for, uh, constraints, right?

Anything we're talking about institutional inertia, right? The everything that's blocking us from achieving things. We want technical organizational, you know, decisions are maybe made or not made or made in efficiently, uh, existing contracts, existing, uh, legacy systems. Everything can be sort of, uh, a burden on what you're trying to achieve.

But then we go further and deeper. And this is, this is the hardest part to sort of explain, because I'm trying to get people to put stuff on the map that they would never think is something you would put on a map, right? Emotions are data, right. Somebody points to one of the items on the map and says, oh, that's that system or that feature.

I hate touching that thing. Um, I say, oh, what's that right? I, I observe, I try to pick, pick up on, on these things, these, these little stories, um, and say, well, let's, let's put that on the map. one.is I don't like this thing. Two dots is I, uh, I hate this thing. Three dots is if I ever need to touch it again, I'm quitting on the spot.

Um, and, and so everybody can put out, put on these dots. Yeah, it could be anything, right? This is just an example, but fears or, or, or skills or values, you can put social constraints on their limiting beliefs. Right? So the things we tell ourselves that may be very, um, unconsciously that prevent us from acting in a certain way.

Um, and so you S you start seeing sort of these clusters of constraints that, that, um, the cluster together constraints that maybe seem unrelated, maybe money and emotions seem unrelated. If, you know, if you try to imagine it, but if you see these things converse on the mapping in a certain location, you start sort of seeing where you might want to act or intervene or, or try to unblock things.

Um, I will. People see eristics on their right. I ask questions about, well, why did you decide that? What, what's your underlying belief, what's your underlying heuristic that you used to make this decision? And I'm using heuristics in the Billy Vaughn con sense of, well, basically everything is heuristic.

All decision-making is touristic. Um, you know, and, and you can start playing with, uh, especially if you do digital, uh, using a digital tool like Miro, you can change the sizes of things and move things, the relative importance of things. Um, Find ways to do layers, which is something mural doesn't really do well yet.

Uh, as far as I know, but this is sort of what I'm trying to get on this map as well. Um, because now all these things become very, very tangible. They become sort of items you can touch and move and point at and debate about and assign weight or value to it, or, uh, or, you know, um, contradict to each other.

If you see contradicting things, things you can put them together and point at them and say, well, is this is a problem. So these are, you know, that's, that's the role as a, as a facilitator, right? You're not doing the mapping yourself. You're sort of observing and listening for the stories. Uh, any, anything as a story, right?

Any, any unit of cultural transmission, any belief, any, any sort of like even body language I will try to pay attention to in, uh, in the, before times, at least, uh, when it was easier, um, you know, somebody is. Acting different when somebody says something, well, there's something there that maybe we should look for.

Um, sorry. I mentioned this to get people, to map the things they wouldn't think of work mapping, so it's a much, much less constraint. And what I think Wardley mapping is, but I can't really comment on that, uh, with, with knowledge. Um, but it's it's you, you don't know what you're mapping yet. You don't have a legend when you start, you don't have a notation.

When you start, you build that notation. In fact, I appoint a legend master and a legend enforcer to make sure that there's some coherence in the, the legend and the, at the map. And then, you know, getting through sort of the. What I'm trying to achieve there. Right. And I use a bunch of your risks, including your risks for, for, um, challenging assumptions and for exposing them.

Um, because a lot of the institutional inertia is of course, because people assume things because they believe things because things were a certain way and maybe they're not anymore, but they assume they still are, or the other way round. Um, so, and, and to find this, it helps to have your sort of backpack of, of reusable heuristics.

Right? So, so the method is not in the visible stuff in the, how you map or what you map or what notations, or what items end up on the map or not. Um, but it's sort of the process of, of going from very tangible, observable things like components or products to very sort of hidden things like, uh, information flow or decision flow or, or, or these kind of bottlenecks.

Um, so. You know, the, the, the first thing of course is, well, this, this mapping style. If there's one rule, it's make everything explicit, right. With everything on the map, it's easier to delete it from the map. If it turns out you don't need it than it is to see it when it's invisible. Right. So put it on there first and assumptions are like that as well.

Right? So we, we, we, we put them on a sticky. We put them on, uh, on, on a box, on a map and now we can move them. Now we can touch them and point at them and say, well, this assumption is this true. Do we need to challenge this? Uh, how do we know, can we do an experiment or approach, uh, to, to learn more about it, right?

As long as it buys us information, right? We keep doing this, uh, until the cost of, um, of finding out is higher than, um, the, the cost of not knowing. And so a heuristic I use is I call it assume assumptions. So assume that everything is really an assumption. We have our map there's there's constraints on there.

There's, you know, there's no money, there's no time. There's no skills. There's no, whatever. Right? All the, all the constraints that block us from doing something well, none of these they're all suspect, right? They're all guilty until proven. Otherwise I suspect all of them, uh, of being assumptions. And then I, and then I can challenge that, right?

Do we really have no money? Are we really not allowed to do this thing? Um, is, is really in the contract. Have you looked at the country? Um, even if it's in the contract, right? If, if the contract says we have to keep this thing online, 24 7, well, does it have to be a law online at. Three in the morning, uh, when it's just a part of it, maybe, you know, one chunk of data is offline for 30 minutes at midnight when our customers are sleeping and it's all data anyway, that they're not accessing.

And we have numbers to show that they, they only access that old data once and, you know, whatever. So there's this constraint of, we can't ever be offline while maybe we can have some downtime, right. A little bit as a, as a treat. And, um, this allows us to, um, to start creating some wiggle room, right?

Because now they're not hard constraints anymore. They very rarely are right there. There, there are hard constraints are physical constraints. You can only have, you know, uh, think and only be in one place or a person can only be in one place at a time, uh, that sort of thing. But apart from that, there's very few real, uh, hard constraints in mind.

And this is all anecdotal and my experience of course. Um, so another way for looking for four assumptions, right? You might be familiar with loss of , uh, thinking fast and slow. So loss of version is the idea of psychology that we, um, overvalue economic loss or the risk of economic loss versus the potential economic gain.

Right? Uh, so we would put in more effort or make choices that optimize for not losing something. So even just framing something as a loss or a wind can, can change people's perception on, on, uh, you know, what they want to do. And usually this ideas is, is, um, discussed as economic loss. And again, very, uh, No anecdotal, but I think it's, I think this exists for much more than just economic loss.

Right? So what, what, what I call this heuristic is the loss of version of version, right? We need to look for low loss of version, assume that everybody's afraid of losing something and find it and make it explicit. Right? It could be the work they've already done. Right? We've already invested in this well, sure.

But if it's going to be more expensive to keep doing it, then do the other thing or better to do the other thing. It doesn't matter how much work you've invested, but there's a, there's a sense of loss there, right? That you need to bring forward to make explicit. So they had ideas, they had plans. They had a way of doing things.

The, the comfort zone that they're in prestige, maybe they're worried about losing their job. Well, that would be economic loss, I suppose. But so there's, there's lots of these hidden things that people are afraid of, of losing, uh, for whatever reason it's not rational. Right. But we need to, we need to look for it.

And that's how you start exposing these assumptions because. There's something more, even more insidious happening, I think. And that is that we S we presuppose that other people will have loss of version. I think it's very ingrained in our, in our thinking, right? It's so deep in our psyche that, um, we, we, we, we, we can sort of see that we're doing this.

So we might not, I might not, um, offer an option to somebody because that option might cause them to lose something. And so I don't even consider it. I don't even. Give it the proper intellectual rigor to, to, um, to consider that option. Um, so a lot of options are never visible. They never become, you know, out in the open where, you know, as soon as an option is, is on the map, you can, again, Moving it and pointing at it and discussing it and deciding about it.

But if it's not even actively in somebody's mind, then it's, then it's, uh, you know, it's, it's unreachable. Um, I have a sort of a simple example maybe to make this, uh, more tangible. Um, we also organize conferences as sort of a side thing. And, uh, we had one last week actually, and, um, called event sourcing life.

And one of my colleagues who were. Uh, sort of on duty on, on, on the day that, well, my, my child is sick and we're all working from home and my child is sick. This might be a problem, et cetera. Um, and of course we were trying to deal with it, but I said, well, next time we should just get a childcare service maybe to, uh, pay them to, to, to take care of that.

So we, we are at peace and they said, well, no, because most of the time it's not a problem. Right. It's, it's, um, it's only if my child is sick, which doesn't happen very often, but it's, uh, you know, it's, it's, it's a, uh, a realistic thing, right? It's, it's plausible. So you need to think of it as insurance that we would pay child care, even though we don't need it.

But, you know, th the, I think this is sort of a case. Presupposed, lots of version that my colleagues were worried that I would not want to waste the money on something that we didn't need. Well, clearly we didn't need it even if only occasionally. So just as a, as an example, that this option sort of didn't appear valid to people that I might lose something as, as, as, uh, you know, or, or the business or a company might lose something, uh, which I wouldn't care about compared to the, to the potential gain of, uh, of, of hiring some money.

Oh, I think I actually accidentally unshared my screen. All right. Uh, how am I for time? Yeah, probably out of it. So yeah, negotiating constraint, every constraint is, uh, is negotiable. Um, you know, a negotiation is something don't be afraid of negotiation, right? That your superpower. To accept that you will have to negotiate learn to do it well, because, um, a lot of good outcomes don't happen because people are afraid of negotiation, right.

They don't know how to do, they don't have the skills to learn that teach others. Um, and you will find better solutions together. Uh, well, this, I feel that, uh, Liz covered it and I learned more about it already. What I would like to add here is, uh, I see my clients sort of wanting to do experiments, but they picked such a small impact to this thing that, you know, they might learn a little bit, but it means nothing.

And it just sits there then and everything else, no nothing else has changed as a result of the experiment. Um, you know, there's a time and place for these kinds of, uh, of experiments or probes, but you'll learn more when it, when there's actually a risk of failure, right. When it, when there's skin in the game.

Uh, so I wanted to mention that at least. Um, so, and, and what we're doing here really. By using experiments and interventions and trying to bring stuff in the open through mapping, uh, as a shift, the Overton window, not in the sort of political extremism kind of sense, but there's things that weren't debatable because people assume they weren't, you know, all we can to, uh, move from four releases per year to a hundred releases per year.

Uh, because it would be 25 times the effort. Well, no, but they, they, they, they don't see that. So we use mapping, we use intervention to sort of shift the ideas of what is debatable and, and, um, and, uh, what kind of behaviors are now possible within, uh, within the organization? You know, I now see as an essential part of design because it doesn't matter how good or great or beautiful or, or, or functional your designers if you can't actually make it happen.

So if there's any methods this right, we, we, we look for stories. Um, as a facilitator, you have to be the, the observer and the uncover of, of all these things. You know, these limiting beliefs, we mapped them, we put everything on the map, cross disciplinary, find all the people and the ideas and the concept is never going to be complete at all, doesn't matter.

But we see stuff happening. We see clusters, we see, you know, things happening in, in, in the same place. And now we know where to intervene, where to do experiments, where to sort of shift, you know, create. I like, I like doing minor victories. I call it. Do something with a team. And at the start of the day we start, we, we decide what we're going to do.

And it must be finishable by four o'clock before anybody believes. Right. And if I check, do you believe we can finish this by four o'clock? Uh, no, there's this. Okay. Let's make it smaller. But by four o'clock we have a victory, right? It's a very small one, but we have a victory and that's an, that's an experience that people haven't had often in these, you know, uh, organizations with this big inertia that you can finish something, decide on something and finish something in a day.

It changes how they see themselves. And then of course you can start scaling that up and do bigger intervention and bigger and bigger things and bigger experiments. Um, but it's sort of changing the, the, the story they tell themselves about what is achievable. Um, yeah. And that's, uh, that's my presentation.

Thank you. That's his first time speaking at map camp. Can you believe that? Oh, and, and my first time talking about this stuff publicly, uh, so it's a bit of an all prepared in, in a week. So, um, some of you have probably heard me talk about a personal interest that just in mapping negotiations, the specific thing of being able to visualize the components that you can use in a negotiation and to actually value them.

Because when we negotiate, of course, everything is emotional. And a lot of the time we negotiate in a way that doesn't actually mean value down to us. So really interesting. The idea that, um, Loss aversion and our interpretation of other people's loss of version, actually two distinct things that need to be measured separately.

And I just, some back into this negotiation, I'd love to talk on this more. Um, I I'm feeling by the sheer number of comments in chat that we're going to have. There's a lot of, uh, desire to talk about this. So I'm going to ask that we just dive into the next talk and then throw all of these questions into the big, um, sort of, uh, question announce a soup that we're planning.

Are you happy with that? Mathai's. Right. So, um, Fred and Liz, uh, I think this is certainly my first, uh, tag team talk. I want to see how this goes down. Um, would you like to introduce yourselves, um, introduce the idea and I suppose the inspiration behind what we're going to talk about? Yeah. So Simon approached me about doing something on challenging assumptions and I was like, aha.

I know exactly what I want to talk about. And I know I'm going to need help from Fred. Um, so Fred and I are both engineers at honeycomb. Uh, boop. Next slide please. And Bulli wanted to talk about today is the prominent we encounter with outages and how easy it is to fall into cognitive ruts, both when you're debugging a live production incident.

And then you're figuring out what happened while we were bugging the incident afterwards. And then we'll talk about some of the best practices that we've learned and what we think the evolving state of things is. And I'll maybe even show a wording that they made about what I think about the w. Um, so yeah, uh, Fred is a site reliability engineer.

How do you thumb, uh, would you like to introduce yourself? Oh yeah. I'm afraid of a site, reliability engineer at honeycomb. Uh, this is actually the first time I have the title of cyber liability engineer. Uh, I had about 10 years of carrier on the software engineering side, uh, heavy influence on operations.

Uh, but for the last few years, I've read everything I could about incidents, uh, the investigations of that, and decided to kind of shift Macquarie around and joined honeycomb specifically for that role. Cool. And I was, I've been in an SRA officially my job title for I think, 13 years now. Um, and I'm basically learning alongside the honeycomb engineering team and sharing with the world.

What I learned. So, this is we're going to tell two incidents that were really formative for both of us. This was the first formative incident. This was November of 20. Uh, this was November of 2019. So we got a, I got an alert that was like, Hey, you're having an exceptionally high rate of failures and people sending data to the high-income API.

And we only get one shot at getting that data. Right. If we drop into the floor, we drop into the floor. So it was like, that's not good. Why has never other monitoring goat gone off? And not just, why is none of our other monitoring gone off? Like why isn't, this is like a 2% of runoff that's happening every two hours.

And it's running up for 20 minutes taking about one to 2% of the traffic with it. And then it just comes back to normal. Like, what's wrong with that? And my first instinct here was, there is clearly a problem with the Amazon load balancers, but Amazon load balancers, not able to talk to her backends. I'm going to ring up the phone in the Amazon and get my contacts there and be like, is there a problem with the network load balancers, spoiler alert, you can scan with your code.

That is not at all. What was going on? What's happening was we were, we had introduced a memory leak. It was, you know, we'd restart all the processes at once. Consumer memory, consumer memory, consumer memory, crash, crash, crash, crash, crash, consumer, every consumer underneath, right? Like that's what was happening, but it took six hours for us to figure this out because we didn't have our system metrics in the same place as our application data at the same place as our AOB logs.

Like this was a problem where I jumped to the wrong conclusion early. And then another engineer who had just woken up who was fresh in the problem, like look at the obvious clinical obvious. Memory metric and was like, this is your problem. It's right there. So that was formative for me in that it caused me to recognize that you can have the most sophisticated tools in the room, but if you're not using them well, if you still have your cognitive biases and that was kind of one of the subjects they want to talk about.

So let's talk about some of the theory of this. Like why does this happen? Boop. So I think debugging is really challenging because we're faced with having to make seat of the pants decisions based off of incomplete information. So I think that this, the causes are kind of threefold. First of all, our tools lead us astray.

If you don't have tools that encourage you to kind of explore and dive into your data, you're going to have a lot harder of a time. Um, then, then if you have, then if you have tools that can let you actually. Pokemon and understand like, why is this happening? You know, not just this line, little, the same time as this other line, but like, this is actually a temporal, not just temporal, but like this is, there was causation here, right?

The reason that this thing took a long time is because it was weighing on the dependency. Right. If you can't see that, it's really, really challenging to, to be able to challenge your assumptions about what's going on and say the system. Boop. But I think, you know, in the case of the forum outage, right?

Like I, there was a spot of org incomplete. And I had been like mentally primed to think, you know, maybe there's a problem with, with Amazon. I think because we had just turned on the new Amazon application load balancers. Right. And then once I started going down this path, I had this confirmation bias because I was like, you know, it surely it's got to be some kind of problem with Lavazza, right?

Like it's getting a different era. It's returning an error code. That's different than what I got from the back end. Right. Like that's clearly a problem. So I think that's one area that kind of, so I think that's, that's the other area that kind of really messed me up.

Right. So one of the things that gets super interesting there, a society of a social technical system, uh, there are multiple components in the system. There's the technical ones, uh, the computers, the code that we ship, all of that. Um, but really the only way that that system tends to evolve and change over time is because people working with it to have an impact on it and change it and modify it and understand it.

And there's this sort of feedback loop that exists between, um, the people, the system and the people themselves, um, which really has this dynamic, even ecosystemic relationship in what happens, uh, which, um, is well coined by the term , which is the, you know, learning system that is made of learning parts itself and software teams, uh, are all that same mechanism of where the people on the team, uh, that run it, the software, the tools, and there's no.

Great ways to pin that into a snapshot in time, because the moment you look into it, that information feeds back into the system and it has an impact again. Um, so really one of the things that, uh, I love to do about these is that incidents really are a perfect opportunity for that sort of learning and all the things that could really, really be happening.

And most of the incident reviews in all companies, in all cases, uh, come back with this timeline that you have. It's the thing that you have to attach on to, to start looking backwards at what happened. Uh, and we have all these events that start at the top. It goes at the bottom decision points, things that happened, and then there's an incident at the end.

And really, uh, we have to walk back. Donald a tunnel in the order of things happen, not necessarily looking back with the idea of the hindsight that we have, and the reason for this is that the errors that we find are constructed, there's not something we discovered or not hinder into the system. They're an interpretation of things that we impose onto the system.

So we assume we have an early steady state. Things were looking fine. Uh, things were going fine. Uh, something happened that usually is approximate cause, right? This blew up and we were in a failure situation. And then we do things like a root cause analysis, where we walked backwards into why that this happened.

Why did this happen? Why did this happen until we find an explanation that feels. You know, satisfactory for multiple reasons, whether it is because it fits with previous biases or judgments that we had, uh, or because it's something that we feel is manageable to solve the problem and preventing from happening again.

Um, and so we come at it with a Lance already. And so if you think that humans are generally to blame and automation is super good, you will find human error as a cause. If you don't like a mechanical component or the people working on it, you are likely to highlight its flaws and whatever you investigate and whatever you figure out that way.

And so the reason why I want to walk back into the tunnel is that at each step of the way, there are multiple branches that might happen to a practitioner, an operator, and making calls all the time, because a decision could be going in a good way, in a bad way. There's multiple perspectives, not everything fits in everybody's mind.

The bigger the system is the harder it is for someone to have a picture that is accurate of what is going on. And so this causes these ideas of hindsight biases, where a decision was clearly bad because it led to a failure. Uh, the counterfactual is if we had did this, then it wouldn't have happened. And we study a thing that never happened.

This is a fictitious universe. We can talk about the future. We can talk about the past, but we can't really. Easily talk about the things that would have happened. If things were different, at least it's really, really hard to do, unless we have that understanding of all of these branches that were in someone's face, the pressures that were on them and all of that.

And that tends to come with normative judgements, the handbook or the manual, the procedure is this or that they didn't follow the procedure, therefore they were wrong. Um, and this causes a lot of things that are really, really tricky. You rather than, you know, asking the question of whether the procedures are adequate to the high pressure situation.

Liz, uh, mentioned a bit in her talk and was probed again with my question on this. This is something that interests me a whole lot. And so we get into the step of documenting what happened, right? This is a bit of a representation of the timeline you might have, but we start from these events and things that were happening.

Um, and this is something that has been mentioned in one of previous talks, uh, failures and successes use the same mechanism. When we're walking down the tunnel, we're trying to make these decisions about which branch, which action we make. And sometimes they're going to be successful and sometimes they're not going to be.

And so without knowing ahead of time, what the failure is going to be, um, then we have to assume that the failure and successes use the same mechanisms. If your idea is that something went wrong, we have to take the person away from there. Uh, you are taking a risk because you don't necessarily know what they usually do that make this successful.

And by taking them. Taking away the opportunity to make mistakes. You might be also taking the opportunity to have success at the same time. Uh, one of the things that I love to do is looking into what is not visible, uh, which interestingly ties a bit into the previous talk as well. Uh, when I see a timeline that looks like this, right, there's a big gap of takes a look.

The issue is found, the fix is written and the traditional way we like to build intuitively our reviews is to really focus on the issue was found. The fix is written, what were the mechanisms and how were they, you know, falling like dominoes and all of that event. Um, what is really interesting is that incidents to me are, uh, this clash between the mental models that everyone has a kind of implicit map that is in our head, uh, and what the system really is.

And there's no way for them to always be agreeing between the people. And if you see that issue where you have, like, this is an incident, there's a two hour gap where nothing is reported and no production, nothing productive is being done. If this is where the repair and understanding of the world is taking place.

This is one of the most interesting areas to dig into because this is where, uh, you know, the mismatch between our understanding of the world already existed long before then we've put the code in production. We took the actions, it was already in place. What we realized that that gap is, um, the drift that has happened.

And we are actively trying to bridge it, to understand the sequence of events that might have happened, the things that we could do to fix it. And this is where all of the very, very interesting cognitive work might actually be done. And it's not just the stuff that's like, you know, Hey, like, you know, here's what we found right here is the answer, all of the kind of twists and turns and the things you've discovered along the way that didn't lead to an answer.

Those are important too, right? Like, and if you leave those out of your timeline, you're sacrificing. Right. Yeah. There might be something like, uh, like there's mentioned, like we thought that this would be an ELB issue because we just changed that, like, this is important context that tents the entire investigation about that.

We call it recency bias, but it's often extremely useful because a failure that happened yesterday is very likely to happen today. Again, especially if we know we changed nothing since then.

Right. Yep. Uh, very happy that this all ties very well into the sort of progression we had in talks. And so in documenting what happened, um, one of the things I find is key is having a generative approach rather than classification into existing labels. One of the ways that we deal with complexity as people is that we invent categories and we assign them and we, which all kinds of facts and information in to them.

But there's a loss of definition and doing fat. And this is the sort of map, uh, that, uh, I create with some of the incidents that we have, that we find extremely interesting that there's something happening there and it's going to be, uh, it's not two dimensional, well, it's not a single dimensional. It's not just a timeline, right?

It has multiple dimensions of people working together at multiple times. Some of the things that we do is new information comes into sometimes it's things that annoy us hypothesis questions, surprises that you can see how this has similarities with the previous talks as well. Um, but really the idea there is that the map itself, uh, its objective is not to classify or provoke understanding it's to generate new questions that we might be asking in there are we highlighting things that we distrust procedures that we think don't work well in some circumstances, and we have to dig, uh, Into it's that way and create that sort of vision.

And then what we do is that we use that map to conduct the actual retrospective. We walk through it, we use it to put ourselves into right mindset of what was happening and discuss what was happening, what we were thinking, right? The map is not the product. It's the artifact that we use to do the actual retrospective at that point.

And you'll notice here that there's like references to what was going on in our system, how people interact with the system and how people interacted with each other. Right. I think that's to the core, what I think is important about when you're trying to map out a socio-technical system, you can't treat each part desparately or separately.

And so we get into the, uh, challenging assumptions of the things that we can do. And one of the practices that's real interesting there for me is the idea is, are facing the uncertainty that we have. Um, there's a tendency in hard sciences, physics, and everything. And I think software tends to borrow a lot from that to do a very large amounts of data being gathered and analyzed in depth.

Humanities is a very interesting approach, which is again, more focused on iPod offices generation. Pick a very interesting case where you think there's something interesting going through. We can learn from that. And then you do a single in-depth analysis of that. You don't necessarily do the large amount of data gathering, uh, because what happens is that you tend to, uh, remove all the messy details.

The surprising stuff gets polished off by averaging that data that you have in everything. When you do that, in-depth analysis of a single case, then you can focus on all the little messy details and over time as you do some of them, you can see themes coming up and whatnot. Uh, but it's a real interesting ones.

And there's the uncertainty in how we understand the system, but also how we communicate it to each other, which is where you compare and understand the mental models that people have in their psychological safety has been mentioned already. It's absolutely critical because if the idea. My understanding is not the same as yours.

There's a required vulnerability to communicate that properly. Um, and then, you know, you can test them. Chaos engineering is one of the approaches for that. And the quicker feedback you have, the better, uh, ability you have to correct and course correct. And change your models. And this actually ties in nicely with where we think there are problems with the state-of-the-art in incident analysis, right.

They kind of in-depth analysis, but Fred did have that one incident. He wouldn't do it all over incidents. Right. But it's important to do it for illustrative like one or two incidents. And it gives you a depth that you just cannot get from saying, you know, okay, here's the TTR for, you know, these five incidents through these times incidents, right?

Shallow incentive metrics. Don't surface these kinds of deep problems that are in the relationships in our. So I think that goes to where we stand with regard to the Wardley mapping of our space. Like, you know, my view is that we're still at the bespoken artismal phase of, uh, of doing incident analysis.

There are standards for doing this, right. It is regularized in the aircraft safety industry, right. This is exactly how you do an incident analysis. This is how you file a car's report with NASA, right? Like, and as much as I liked the folks at blameless and jelly, like I think that it's a little bit premature for QB in the mainstream just yet, um, that we're starting to, you know, do the knowledge sharing with things like the Verica open incident database, but like we're not yet at a stage where you can just productize something and saying, here's a process we'll run at your company.

It'll just automatically work like that. That's just not the reality of where we are today. So here's a math major. Um, what should we typically outlines kind of how all of these production technologies relate to each other? Right. When we think about software delivery and software reliability, you know, it's something that is bespoke to every organization, right.

We have some idea of how to do it, right? Like we're not inventing it from scratch, but you have to do a custom in-house and you might rely on, you know, your Lemnos or as kind of your utility hosting. Right. It's just a meter connection that you get your, you get your stuff through, but we don't think that the traditional ops model is working.

Right. You know? Yes. You can go in. You can go and get it off the shelf monitoring tools. They're not going to help you with the deep incident analysis and understanding exactly what happens inside your system. You can go and hire a network operation center from Tata consultancy services, right? They'll they'll have like, hate your pages, but that's not going to solve your liability problems.

So some of the things that have become utility or better have become commodities and standardize things like they're not helping us. But the things that we do think are helping, right? Like from most mature, at least mature right CSTD tools. Right. Everyone knows you need to be able to deploy your software to production consistently, whether you're using, you know, team cities or like whether you're using a, you know, circle CEI rate, it doesn't really matter.

There are tools that can do it. But I think that, you know, when we think about the SRE and DevOps practices, there still is a range of people figuring out, like, how do we do this? How do we do Esri and dev ops practices? Right. It's just as messy as software delivery. There are some norms in nursery, but they're not, you know, you cannot just go off the shelf, get an SRE or get a dev ops, right?

Like you're not going to get what you expect. And then what do those Esri's and demos practitioners need right. To be successful. I think the answer is you need observability tools to be able to figure out what's going on. And, you know, I think that we're at a stage where that is suddenly experiencing explosive growth.

It's experiencing kind of this traction in the market and the emerging standardization of it. Um, so this is why we're seeing like venture funding. We're seeing projects like open telemetry to claim to bring, to grips all the steps. That's not as bespoke. But I think incident and I'll ask, this is something that, you know, despite the fact that it is super standardized in the air, in the airline industry, you can't just pour it over and see if it's standard in one industry.

It must be standard in all industries. Right? I think here it is much more towards the Genesis of where we are. And then kind of towards the bottom of this, right, with regard to, you can have a little of, of, of downtime is treat right as slow tools are only as good as the incident analysis and observability that they were built on.

Again, you kind of can't quite standardize it yet. There's kind of no one formula for what makes for a great SLO such that you can stamp it out everywhere.

So that's what we have to share with you today, which was kind of argue into two, into two former outages and how we analyze them afterwards. And what we learned about her, about how other people, uh, can, can view incent best practices and challenging. Thank you.

Thank you both horizontally again, just really insightful. I love that. There's always a mix of big idea. Take this away. You don't know what to do with it, but if you'll be thinking about it for a couple of years, and then we have talks, which are really specific, and I think people can take something and actually apply it to their work the following days.

So thank you so much for that. Um, like we said earlier, I'm just going to open the question, announce it to everybody. Um, and for the next 20 minutes or 15 minutes, sod, we'll just have a, basically open shot around everything that we've heard. Um, we'll start just by asking the other panelists for any of their burning questions, ideas, you may have seen some themes that sort of relate to what you've spoken about.

Um, so who wants to go first and just give any thoughts or ideas?

I did. Um, so to, and Fred, uh, do you have any story around, what's the worst bug that you prevented with this approach or the worst problem you've uncovered before it actually hit you?

Oh, we have many of them. Uh, we were, I was writing reports recently. We had tough operational, uh, September and October. And I think we had something like 20 near incidents for every incident that hit the public. Like a lot of it is there's this delay between when something happens and when it becomes critical and noticeable to people, um, one of the noticeable ones we've had, um, at honeycomb specifically was one.

Or main Kafka cluster was nearly running out of memory. Um, and I was just, you know, a fly on the wall during this one, but it felt like these TV shows when there was a bomb and you have to cut the right wire at one second before it explodes. And it was literally that we started disk filling up 90%, 91, 92, 93, 94, and having the right come in with the right syntax hit at 99%.

And then disc was freed and bought us some time. So a lot of it felt, you know, it's not as exciting as the actual action movie, but it felt like a real equivalent to that. And nobody saw anything of it outside of the company. But the magnitude until after response is not necessarily, uh, you know, in direct relation with the risks that is seen and the actual impact that is seen, uh, in many cases, what we see is tons of near incidents, which means that to me, the process is working.

There's a lot of complexities, things bubbling up, but generally the notice, the notice of it is quick enough and the action is interesting. NFR mitigation. It doesn't become an actual incident. It's just very close to it. The same reaction, not the same impact. Yeah, although it does have an impact on our people, right?

Like we do set an upper bound for the number of incidents that we hope to have. Um, because that impacts the way, you know, the extent to which people feel burnt out the extent to which they feel tired. But at the same time, you know, you shouldn't be penalizing people and saying, you know, oh, I'm going to dock your bonus.

You declare more than 20 innocents this month. This means like people are not going to declare incidents. Right. The other really fascinating thing about that coffee thing, right, was that we had built in safeguards to the system, such that there was the wire you could cut to beef, use the bomb as close to seeing the bomb, just taking and taking and picking it, not being able to do anything about it until exploded.

Right? In this case, we have the safety option. Keep less data in, on local disk tier more that to Amazon S3 rather than having to completely rebalance our cluster and turn up new nodes. We had that safety margin to say, okay, cover retention from three hours to two hours on, on the local cash. Uh, and one of the attitudes that's interesting there, as well as this idea that the incidents are a normal outcome of complex systems working in a, and this ties into to blamelessness.

But you know, the operator or people doing operations, uh, are just, there are people sending on the tip of the iceberg, right. Of where all the previous decisions and to current running of the systems coming clash together. So anything to me that has to do with, uh, you know, the operator messed up something didn't go right.

Or that. Is looking into a tiny, tiny portion of all the organizational pressures and dynamics that exist. And we're only looking at the tip and seeing like, oh, you know, if we shaved off the top of this iceberg, then everything would have been fine. It's not necessarily effective. Um, and so incidents are a normal outcome that we have to learn from they're not necessarily preventable, uh, is just an opportunity to learn.

Thank you. I really love that observation of incidence as being something which are inevitable. If you're going to have any kind of innovation. I agree. Really great. Is there a story just around mapping? I suppose another level, the pace, uh, I suppose the incentive toys drive to go faster. Is there some things, as you were speaking, I was thinking about the act of mapping and the fact that you will record components, you will record influences, um, that perhaps are forgotten in that long run.

So when you are doing the sort of post-mortem, it's much easier to take a six month old map and say, oh, it's that thing than it is to go through all the logs and resurface that again. Um, and then they, you know, the larger piece of the, where we have a profit motive and we are always looking to move fast to move back to this.

Is there a lesson in how fast we should be moving, um, and practices that perhaps we could build in, uh, to make it easier to get back.

So one of the things there is, uh, how fast we are moving, I think tends to be a false solution, right? By the time the incident happens, the mismatch that we have between our mental models with, to me is a map. It's just not on paper. And the actual system is already existing. It's already there. The reason the incident might happen is that that clashes highlighted, but actual events stressing and trying the system going slower is not necessarily something that makes it safer.

It might just delay the inevitable of, you know, the map taken is the territory and the way it is right now and these decisions taking place. So, uh, counter-intuitively one of the good practices is trying to go faster. So you don't have the time to have your map H out as fast as possible. In this cases, chaos engineering tries to do it in a controlled way.

Continuous delivery, uh, is one of the approaches where you narrow your feedback loop. To me personally, one of the core elements is the duration of that feedback loop. The shorter it is the easier it is to say, like the way I took this action doesn't fit my mental model. And you can correct it gradually and continuously as a habit.

Right. Which is actually one of the things where one of our incidents than we had in the past month was caused by our automatic retirement of one cost per node per week. Right. And I think that, you know, yes, we could have not broken that Kafka node every week, but instead we would have discovered it at 3:00 AM, you know, on, on some Sunday, in November, like, you know, some months down the road, but instead of Sibley during another incident, possibly during another incident and then everything is worse.

Right. So yeah, you know, the more you continuously, you refresh that the better you will have knowledge to. Uh, I Mathias, I'm guessing that as that talk was going on, you're seeing a lot of commonality between what you were talking about, but you're the expert. What did you think of that? Um, and were there any areas of overlap?

Um, yeah, well, uh, definitely the sort of making the explicit, something I didn't do is express or talk about is expressing time through maps, but that's of course what you are doing all the time, but I do come from sort of, uh, you know, uh, the event storming community as well, where we're expressing everything as time, um, when it comes to, to do the sort of maps, um, and, and, you know, history, et cetera.

Uh, my, my general idea there is that, um, you express time simply by doing another map because keeping track of everything, is it. Yeah, it's a job in itself. People don't necessarily want to invest resources in, in maintaining history and keeping that organized and searchable and find-able. But if you, if you keep mapping regularly, you're, you're doing sort of a very deliberate act of, of knowledge transmission every time.

So the, th th the shared artifacts, you know, it's, it's the, the, the understanding of people have the shared knowledge shared mental models. Um, yeah, that's sort of what, what I was thinking about. Um, yeah, I'm, I'm usually SRE, I don't know anything about it, um, at all, except from what I read in books, like, uh, you know, drift into failure or, or rich book, um, But because as a designer, I'm sort of, uh, you know, especially one that they bring in from outside.

I'm the first session they cancel when there's a crisis, right? Uh, oh, we have no time for design or improvement or anything because we're dealing with this crisis. And the crisis, you know, as, as, as you were sort of saying, uh, listen, Fred, this is the end of the illusion, right? The situation was always there.

Um, but what I learned from, from, um, uh, Dave Snowden is that it's exactly the time when you need to bring in the designers. You have two task forces, one to solve the problem at hand, to put out the fire and immediately while they're sort of momentum and, and, you know, and motivation for people start putting up, you know, creating teams to, to solve the problems in a deep sense.

Right. And, and sort of look at where is our design room. Where's our understanding wrong. Um, if you wait until usually it's sort of, oh, first fix the crisis, and then we'll, we'll work on these things that we've known for a long time that we should have fixed, but we never did. But then the crisis is over and you're, you're actually, it's not really over, but you're still cleaning up stuff I imagine.

And then you're also trying to catch up doing the cleanup. Yeah. So you never get into the, but it's when the crisis is there, you know? Right. When, when, when Facebook was down, somebody should have put a lot of money into a new team to, I heard it, it was DNS, uh, uh, you know, solve DNS problems or, or think of a new way to do that, like foundationally.

Um, so yeah, that's, that's sort of a shame that the, the, the design is seen as the, the luxury as opposed to the necessity. So that's also something I was thinking about, uh, during, during your talk. Uh, I loved it by the way. I'm really good. We have loads of questions by the way. I know that Liz went through and diligently answered some questions already.

Um, In the chat. So it's worth reviewing those. If you haven't noticed they've been answered, you might find that actually you've got a question, um, that you wanted to ask. It's already been dealt with, uh, there, um, looking back so I could see so much going on in the chat, it's a little bit hard to track.

Some of your questions are in the main chat. Um, if you leave me a four minutes, if you leave me to look through that and surface them and talk amongst yourselves, perhaps you've seen a question that I haven't noticed yet. We've got one actually in the Q and a box right now, uh, for myself and for Fred. Uh, the question is how do you slate your team is to evolve those mental models continuously and ensure that there's coherence between different people's mental models.

Uh, there's plenty of ways. This happens naturally, right? The incidents and the review and the retrospective itself. If you are able to do the proper type of, uh, facilitation, it highlights this, this idea that someone comes in as like what was surprising at the time. And it's not necessarily surprising from everyone, but that's the kind of sharing context.

It gives it to you. If you don't have it as a proper pace with incidents and chaos engineering is one of the interesting ways to do that, to where you introduce or induce these errors on purpose, which is similar to probing and having these discussions is a great way to do that and frequently do simulation.

One of the really, really great things that honeycomb does is that every time there's a batch of new hires, uh, there's an architecture overview of the system and the people giving the architecture overview is always a different person. Usually someone who was in the last overview. So if you had your education last year, giving it next time around, it started forcing that discussion of what changed since last time.

Uh, and anyone is free to come there. So I really right back, we make the second newest person teach the newest person. And then there's kind of the bet on which dog will eat, forget over services or maybe after dogs. So, so for a lot of these, I think it's, you know, you, you can't transfer a mental model.

The thing you can do is force people to well force guide people into comparing and contrasting. And that's the way you kind of highlight the differences. And of course, correct. Uh, on the understanding that we have with systems.

Brilliant. And there was a question that's been answered in dealt with by me, but it was about adding the time as a third dimension, um, which from a graphing perspective, I thought, well, that's actually useful. Great idea until you think about what most two D maps look like. And I thought having a third dimension that you need to move through, unless you've got some way of only looking at one slices at a time, I think there's, you mentioned the stop motion, sort of where you're doing it as an animation.

Um, but we are coming up to time. I've got a minute and a half or so left. Uh, some people missing some messages. What I've tried to do, everybody is, um, take the text of what everybody's spoken about. So if we have some way late to sort of post up the actual questions, we will have a record of them, uh, color and shape as a way of noting time to, yeah.

Is that less complex as in. Three colors. Is it any, I suppose it depends on how you view color versus space or whatever, but interesting sort of visualization problem. Um, how could you express time? Um, over that? And actually one theme for me today has been that I've always seen, you know, I take a static and a static and I described the movie.

So seeing a change, isn't obvious, someone's got to describe this thing, that way to that point maps and all diagrams, inherently compress information and drop some. And so the question is, is there a point you're trying to make and what it is that you're keeping and trying to communicate, and you might need multiple of these maps because one map cannot contain itself right there.

Information, you keep information you're dropping. That has to be a conscious decision with some position you're taking. Indeed all maps are in. Perfect. And it's really about representing the most important thing or the most useful thing. Well, look, we are at time and thank you so much about loads of questions.

I will do my best to make sure everything is at least published and available. I'd encourage everybody follow look for the speakers, um, and you know, follow them on Twitter or whatever channels they're using purely because it's a much richer way to sort of follow. And what they're saying, look a previous work, um, future products, and hopefully we'll see all four of you here next year for the next week.

And by the way, um, let's say it's really wants a job at Facebook. So if anyone ever has any contacts

ingest to shut it down or mutagenic in case you do look at Facebook, somebody thank you so much. Thank you. Thank you. Bye bye.