Rationally Speaking #167: Samuel Arbesman on, “Why technology is becoming too complex”

Julia: Welcome to Rationally Speaking, the podcast where we explore the borderlands between reason and nonsense. I’m your host, Julia Galef, and with me is today's guest, Sam Arbesman. Sam is a complexity scientist and currently the scientist in residence at Lux Capital. He's also the author of the books the *Half-Life of Facts,* which we discussed on the show a few years ago, and more recently, the book *Overcomplicated: Technology at the Limits of Comprehension*.

Sam, welcome back to Rationally Speaking.

Sam: Thanks so much. Great to be back on the show.

Julia: A complexity scientist is … What is a complexity scientist?

Sam: A complexity scientist, essentially, it's a scientist who is focused on studying complex systems. Complex systems are any sort of system that has a huge number of diverse parts that all interact in often complicated ways.

That sounds super abstract. Really, it can be across many different domains. There's biological systems. These things are like the parts within a cell or even many organisms interacting within an ecosystem. These are all complex systems. Computers in a large network, that's a complex system. People interacting in an entire society; this is also a complex system.

It turns out there are a variety of mathematical and computational tools that you can use to understand complex systems, on both the details of each specific type of system as well as often stripping away all of these specific features, the domain-specific aspects of them, and saying, "Okay, these things are all specific. They're all just networks. They all have pieces. They're all interacting this way. Maybe there's some kind of mathematical framework for understanding all these different things.”

Complexity science essentially takes these tools and applies them to a whole variety of complex systems.

Julia: Great. I appreciate that you gave examples of complex systems because I think complexity science is one of those things where it's hard to define it without using the word complexity in the definition.

Sam: Exactly, yes. Frankly, I think a lot of the most interesting aspects of our world are complex systems and so it does have, yeah, touches upon pretty much everything that we see around us.

Julia: Right. Especially as someone who loves the idea that there is an underlying order or pattern or connections between seemingly disparate fields, or disparate things… Complexity science is very appealing to me. The idea that we can learn how to think about our economy from studying bird populations in the forest, that sort of thing. It's a very appealing way of looking at the world I think.

Sam: Yeah. It's very exciting. I think you have to caveat that with the fact that oftentimes these kind of models might give you maybe a first order of approximation to what's going on. Just because you can write a single equation that explains maybe how the metabolic process within organisms scales with the sizes of different creatures, these things are powerful and there's a lot of explanatory power in them.

At the same time, of course, they're going to strip away a lot of the details of why living things are the way they are, and why different creatures can have all the details behind them.

I think complexity science is always this balance between recognizing that there are details which, of course, a simple model might not capture. At the same time recognizing that there are these deep similarities in modes of behavior and ways of being between all these different areas. I think that's a really exciting thing, and shows that are certain models and modes of analysis that actually cut across lots of different domains.

Julia: Right. That is a tension that you touched on in the book and which hopefully we'll get back to in our conversation. Moving onto the thesis of your book, Overcomplicated -- What kinds of systems are you concerned with in the book, concerned that they are or are becoming too complicated? And what does it mean for a system to be "too complicated?"

Sam: Sure. I'm speaking about essentially all these technologies that we've built as a society. I'm using technology fairly broadly to mean anything that has been built by people for a specific purpose. It can include the traditional sorts of technologies that we think about, like large machinery, or pieces of software, or the computer on your desktop. It can also be the entirety of the internet. It can be our urban infrastructure. It can even be our legal systems and our legal codes. These are technologies of a certain purpose.

In the book, I shy away for the most part from the socio-technical systems, like things more like bureaucracy. Though I definitely actually discuss legal codes quite a bit because I think there's some very interesting similarity between how those technologies grow and evolve, and how more traditionally we think about technologies and how those things change.

Essentially, the argument I'm making in the book is that technology very broadly is becoming more and more complicated. Which I think intuitively people recognize. But increasingly, not only is it just too complex for a lay person to understand – it's one thing to say "I don't understand how my iPhone works, but there's somewhere an Apple genius who understands what's going on." But increasingly, many of the technologies that we're surrounded by and that we use on a daily basis, they're actually so complex that no one, whether you're an expert or otherwise, fully understands these things.

It's because the systems, they've essentially become very complex systems. They have enormous number of parts that are all interacting in highly nonlinear ways that are subject to emerging phenomena.

This is not just, "Oh! Biological things exhibit this behavior where it's truly hard to understand what's going on within a human body." Increasingly, the systems that we ourselves have built -- and we think ourselves as fairly rational individuals and we should be making logical constructions -- increasingly, these things that we build are not fully understandable.

The book looks at what are the forces that have led us to this point of incomprehensibility. The things where, for example, on the one hand, you want to continue adding sophistication to a technology over time. Adding to features to let's say a piece of software. That's great and each individual piece is good -- but over time, they accumulate and you have created a great deal of complexity. That force, as well as other kinds of forces, lead us ever closer to an increase in complexity.

Then on the other side, you have the fact that when these systems become complex enough, our brains really have not evolved to handle the kinds of systems that we're increasing and building. They're becoming more and more incomprehensible, where we don't fully understand these things.

Of course, understanding a system is not a binary condition. You can have different levels of understanding. But increasingly, we are having a reduced amount of understanding the systems.

The book looks at what are the forces that have brought us to this point, why our brains break down in the face of all of this incomprehensibility. What should we do? Should we freak out and say, "We're screwed?" Or are there more productive responses? I'm a fairly optimistic person by this position and I think there are ways of actually meeting these technologies even halfway.

Julia: Great. Just to emphasize, when you talked about our technologies increasingly becoming non-understandable, there's three levels of strength of that claim. Where the lowest level is that we, the individuals who use the technology don't understand it. Then the next level up would be: there's no person who understands this technology 100%. Then the higher level up, the third level is, it is not *possible* for a human to understand the system.

I guess there could even be a fourth level where you could say this is inherently not understandable. That even some being with more working memory and computational power and other forms of intelligence would not be able to understand it. Because it is inherently, there's not an order to be found there.

There are 4 levels that I just sketched out there. Which of those claims are you meaning to make?

Sam: I think I'm trying ... I would say depending on the specific technology, we're somewhere between 2 and 4. And depending on how you define understanding, there are certain situations where you're like, the ability to trace out all the different possible pathways within a piece of software or a computer program. Like all the different potential "if then" statements and all the ramifications. Once we get software of a certain size, even if we have far greater memory and far greater processing power than baseline humans, I think those things are actually impossible in the life span of the universe.

I think then, depending again the type of understanding, you can actually go all the way up to level 4. For the most part, I'm talking about levels 2 and 3.

Increasingly, I think it's not just practically we're getting, where it's like these systems no one fully understands them. I think we are verging more and more to a level of 3 simply because of the massive size and intricacy of these systems. As well as, often, the amount of expertise in multiple different domains that is required to understand these things fully. It's too much for a single person to know. It might even just take more than several lifetimes to actually gain all that mastery.

I think we are getting closer to level 3 where these things are, for all intents and purposes, impossible to fully understand.

Julia: Let's talk about why that's potentially a bad thing.

Sam: Why could be a bad thing, if we don't fully understand the systems that we are building?... We're going to have bugs and glitches and failures. And if we think we understand these things well and we don't, there's going to be tons of gap between how we think we understand the system and how it actually does behave. There's going to be these failures.

I think that can be very worrying for many people. The idea that there is this constant mismatch between what we can understand, and the complexity and the power and the actual behavior of these systems.

And I think for many people when that happens, if there's this concern that there's this mismatch, you immediately jump to, essentially, fear of these systems that are like super intelligent. Computers that are going to kill us all. Self-driving cars are going to crash uncontrollably. All these bad things are going to happen.

Sometimes people actually go the opposite direction, the other extreme. Where there's the system they cannot fully understand, and they almost have this undue veneration. Almost like a religious sense towards to it. Like, "Oh my god! This thing is so wonderful, so complex, I can never fully understand it. It must be perfect."

I think that's also a very dangerous extreme perspective. Both of them are not so great, because they end up cutting off questioning and actually trying to inquire how these systems work.

I think the better way of thinking about it is, yeah, there's always going to be this mismatch between how we think a system works and how it actually does work. There's this constant iterative approach to actually better understanding the system. As well as more effectively trying to reduce the gap between those things. And often that's what happens when you root out bugs and glitches, you make the system closer and closer to how you think it should operate. But you'll probably never get there.

And I think the healthy attitude is this, rather than have a company saying, "Oh our software is perfect. It has a little bug, we sweep it under the rug, we fix it. *Now*, this is perfect." There's always going to be errors, and we can constantly try to improve our understanding of the system and actually improve the performance of the technology.

Julia: Right. It sounds like there's at least a couple of different paths to this increasing complexity, or causes of the increasing complexity. Where one of them is just that we keep adding on to the systems that already existed, and we keep patching the bugs that crop up. Instead of saying, "You know, it's buggy. Let's start afresh, create a system that won't have these bugs in it."

Then a separate, a different path is that it seems like complex systems just do a better job at a lot of important problems than simple understandable systems do. Like the field of artificial intelligence that you mentioned has been moving over the last few decades away from these top-down approaches, where we program rules into the artificial intelligence to follow, for it to determine whether something is a cat or is not a cat, for example.

…To instead, a system that's more bottom-up, where the AI learns rules from mining data that we give it. And they're not rules that we could have even known about ourselves, or even that the algorithm could articulate in a way that we would understand. And that does a better job, it's just like a more effective form of artificial intelligence than the top-down, understandable systems of rules.

Does that dichotomy seem to capture most of the cost to you or is there another path I'm not putting out?

Sam: I think that's certainly another important factor. I think this speaks to the idea that the world is messy and complex. Therefore, often, in order to capture all that messiness and complexity, you need a system that effectively is often of equal level of messiness and complexity. Whether or not it's *explicitly* including all the rules and exceptions and kind of the edge cases, or a system that learns these kinds of things in some sort of probabilistic, counterintuitive manner. It might be hard to understand all the logic in the underlying machine learning system, but it still captures a lot of that messiness.

I think you can see the situation where in machine learning, the learning algorithm might be fairly understandable. But then the end result – like, let's say you have some network with millions of parameters, and a complex relationship between the inputted data and the actual output. You might be able to say, theoretically, I can step through the mathematical logic in each individual piece of the resulting system, but effectively there's no way to really understand what's going on.

I think that often is a result of the fact that the input, the world, is actually very messy and complex. You can see this even on a small level when you think about like, if you want to build like a calendar application of your iPhone. It's one thing to say, "Okay. There's 365 days in a year." Then you realize you have to deal with leap years, and you also deal with time zones, daylight savings time, and so on… and then you go, “Oh wait. This is actually a fairly complex thing."

The world is complex. And you see this also in self-driving cars. It's one thing to say, "I want to build a self-driving car that drives in low traffic conditions, on highways on sunny days." It's another thing to say, "Well, I have to deal with the messy real world,” like one-way streets and pedestrians, and maybe like a dog jumping out in the middle road, or a whole bunch of irrational drivers that you're surrounded by.

Suddenly, everything becomes much more complex. Whether or not you're manually hard-coding it in or allowing the systems to learn all these things organically, the end result is a massively complex system.

I would say that it is really the need for a system to capture all the edge cases and the messiness of the real world. That is a very important driver for how these systems become very complex and increasingly incomprehensible.

Related to the whole accretion, adding things over and over, is this idea of interconnection. That when you add bits and pieces they're not existing in a vacuum. Everything interacts with everything that came before it. Oftentimes the pieces that have come before it, they’re certainly foundational and they might have existed for years or decades prior.

You have situations with legacy code and legacy systems. The IRS uses computer systems that were developed, I think, during the Kennedy administration. You have many decades-old machinery that's still involved in a lot of very important technologies that we use on a daily basis.

And these things are so, they're so embedded within our technology. We could never really root it out and just start over. That would often be very prohibitive in time and resources.

In addition to that, as a related factor, sometimes, the people who built these original systems that we still rely upon, they might have been long retired. They might even be dead. And so therefore we can’t really talk to the people anymore who are involved in building these kinds of things. And seeing how those foundational systems interact with the more complex newer pieces, it becomes effectively impossible to fully understand what's going on.

Julia: I’m reminded of some of these cautionary tales where people looked at a pre-existing system that had developed somewhat organically, with pieces getting added on over time to meet new needs that cropped up. Maybe that's a city, maybe that's a social system.

And these people looked at these complicated messy systems and said, "Well, I can do much better than that. Let's raze this neighborhood to the ground and build a nice orderly city plan instead. Let's design a new social system that actually makes sense. Or a new language."

But in fact, that messy organic thing was serving a lot of important functions that were invisible from the top-down. And the nice, neat orderly new system actually failed in a bunch of hard to predict ways. Like it didn't like have spaces for communities to organically meet each other and gather, and create that sense of community, for example. It was all too sterile. That sort of thing.

There are all these cautionary tales of people trying to get rid of messy systems that are hard to understand, and then put in these legible systems instead. I wonder if it might in fact be sensible to *not* try to get rid of the complexity and put in place a new system that we think will work better.

Sam: Yeah, I totally agree with that. I think oftentimes when you build a new system, if you do it effectively, and you're trying to sweep away everything that was complex and messy before, oftentimes you end up with something of similar complexity. If you do it right. If you do something that actually works well, it often ends up just organically becoming as complex.

And I can see that thing happening... Or if we're not as complex, is what you're saying, it ends up failing in ways that we can't imagine.

I think there's this recognition of a humble tinkering approach with any system. Where rather than trying to sweep it away or fully understand it, recognizing that when you're confronted with a technology that might be new to you or you're trying to understand it, rather than trying to replace it with something very simple… Saying there's a lot of un-anticipated consequences. There's a lot of nonlinearities in the systems. A lot of feedback in ways that may be hard to understand. And therefore I'm going to have just play at the edges and try to see how it behaves.

Oftentimes in a company, someone like a new CEO would come in and imagine a lot of pressure to make their mark on this organization. Of course, an organization is also a very complex and nonlinear entity. To try to change it in large ways can often backfire in unexpected manners.

It's often probably better, maybe not necessarily from a PR perspective but from the perspective of actually trying to modify systems, to modify in these small ways at the edges. Rather than trying to sweep everything away and change it in big ways, where you're then prone to learning about the complexity of the original system, due to all of these unanticipated consequences.

Julia: Right. You made this really interesting distinction in the book -- this is starting to remind me of it -- between physics thinking and biological thinking. And this "tinkering on the edges" approach, would you call that an example of biological thinking?

Sam: Yeah. I think that's a great way to describe it. Yes, so this is the distinction between physics thinking and biological thinking.

Again, before I delve into it, I want to be clear that there are many physicists who employ biological thinking and many biologist employ physics thing. It's just a good shorthand way of thinking about it.

The physics approach, you see it embodied maybe in like an Isaac Newton. A simple set of equations explains a whole host of phenomena. So you write some equations to explain gravity, and it can explain everything from the orbits, the planets, the nature of the tides, to how a baseball arcs when you throw it.

It has this incredibly explanatory power. It might not explain every detail, but it maybe it could explain the vast majority of what's going on within a system. That's the physics. The physics thinking approach, abstracting away details, deals with some very powerful insights.

On the other hand, you have biological thinking. Which is the recognition that oftentimes in other types of systems, in certain types of systems, the details not only are fun and enjoyable to focus on, but they're also extremely important. They might even actually make up the majority of the kinds of behavior that the system can exhibit.

Therefore, if you sweep away the details and you try to create this abstracted notion of the system, you're actually missing the majority of what is going on. The biological approach should be that you recognize the details are actually very important. And therefore they need to be focused on.

I think when we think about technologies both approaches are actually very powerful. But oftentimes I think people in their haste to understand technology, oftentimes because technologies are engineered things, we often think of them as perhaps being more the physics thinking side of the spectrum.

When in fact, because they need to mirror the extreme messiness of the real world, or there's a lot of exceptions, or they've grown and evolved over time, often it's a very organic, almost biological fashion. They actually end up having a great deal of affinity with biological systems. And systems that are amenable to biological thinking and biology approaches.

I think we need to … I want to say "privilege" biological thinking over physics thinking when it comes to technologies. But at least, recognize how important it can be to say, "Okay, this is a system. It has a huge number of parts that have maybe been added and grafted on overtime."

It's evolved oftentimes in fairly similar ways to living things. Therefore, we need to actually use this more biological mode of thought, where we look at the details, and the exceptions, and the bugs in a system, to really get a better sense of how that system is working. I think that biological mode is very important when it comes to technology.

Julia: Another way I think to capture the difference between biological and physics thinking is that physics thinking relies more heavily on theoretical causal reasoning: "I predict this, because the system should work this way, based on my model."

Whereas biological thinking is maybe more reliant on empirical evidence. Like, "I will be confident that the system works this way when I have tested it, and confirmed that the system works this way. Whether or not that conforms to my expectation of how the system should work."

Actually, this same distinction I think, even with the biological and physics labels, was made by another recent guest on the podcast. Named Vinayak Prasad, I think, was his name. He co-wrote the book Ending Medical Reversal in which he was talking about all these medical results that were consensus among doctors, and put in practice for years. And then finally the solid gold standard long-term trial was done, and found that, "Oh actually, stents don't have the positive effects on mortality that we thought they did. Oops." Or "Oh, promoting handwashing in the hospital, or sorry, wearing gloves in hospitals, doesn't have the positive effects we thought it did. Oops."

He says he thinks that one reason for this, the reason this keeps happening, is that medical students are taught to think like physicists. Where the human body is this machine, and you can reason about what would happen if you do this thing or that thing. Instead of being taught to think like biologists, where you only really trust results if you have seen that, in fact, the evidence supports it and not just the theory.

Sam: Yeah. I think this goes back to where I was saying the idea of using this iterative approach towards understanding a system. Where we're like, if we think we understand a system well, and then there's a whole bunch of bugs or failures that make us realize that there's this gap in understanding, then that causes us to update our model of how we think the system works. I think that that is much more this biological mode, of constantly collecting bits of information to gain a better picture of what the system is doing.

Actually, in biology, one of the ways you can do that thing is: rather than just waiting for mutations in the system... You can actually inject the errors. You can mutate. You can use mutagenic chemicals or radiation on bacteria, in order to really try to understand the complex feedback within genetic networks. You can also do the same thing in technology. You can actually inject errors into our technologies, to really have this empirically-based understanding of a system.

Netflix actually has this software that they have released, called Chaos Monkey. The way it works is it essentially will randomly take out various parts of the system that you're working on, and render them inoperative.

The idea is that the system should be as robust as possible. When this thing happens, the system should respond robustly and be able to handle all the bugs. Of course, if it doesn't, you can improve the system so that when something does go wrong, there is as small a mismatch as possible between how you think it should respond and how it actually does respond.

I think that very empirically grounded approach to our technology -- like recognizing, "Okay. It's going to be very hard to understand how the systems work," but perhaps one of the ways of doing that like actually gaining insight into how something works is by seeing how it operates once something goes wrong. I think that is very powerful but also humbling approach to recognize that, "Okay.

Sometimes, the only way we can understand a system is actually when it malfunctions."

It teaches us about the gap in our understanding. It's a very different approach than saying I'm going to have this very mechanistic logical approach, where I have a single question that should explain everything and when it doesn't, be perplexed.

Julia: There's an expression - “It works in practice, but does it work in theory?”

Sam: Exactly. Yes. I like that.

Julia: Funnily enough, someone told me about Netflix's Chaos Monkey approach a while back. I misunderstood them at first and thought that there was someone at the company whose role was “Chaos Monkey.”

Sam: That would be a great job title.

Julia: Yeah, better than “scientists in residence!” Chaos Monkey for Netflix.

We've been touching a few times now on this idea that complexity makes systems less robust in some ways. It's definitely intuitive to me on some level. But I could also see an argument for complex systems being more robust.

For example, one of the things you talked about in the book is this feature of many complex systems that you call interoperability. Maybe that's not your personal word? But a word that's used for systems that are designed to work together, two different systems in different platforms.

For example, Uber uses Google Maps to tell the drivers where to go and how to get there. You make the case that this creates additional complexity. Which I can see. But I can also imagine this counterfactual world where every system is self-contained. In this world, Uber didn't rely on Google Maps, but instead developed their own mapping algorithm or app. There's less interoperability there -- but then there's also just lots of different systems that haven't already been tried and tested, where people developed familiarity with them.

It's not clear to me that that world of self-contained systems has less total complexity and less total downside risk. Do you think it does?

Sam: I think complex systems by and large are robust, just because if you look at a network, if it has this messy structure. It often means if you take out some component, oftentimes, the system reroutes around it.

I think also interoperable systems, when the system ends up being used by many other technologies, oftentimes, it ends up being more robust because more people view it. They've rooted out all the errors. I think in that sense, these systems can be more robust.

I think the converse of that is, oftentimes, when things are enormously interconnected as well as tightly coupled, you can often have a failure that can actually cascade in ways that it might be hard to anticipate and actually cause a huge failure.

If things are fairly small and maybe not so complex, fairly simple, then a single error might just take down a small subset of what's going on. Versus when things are all interconnected, then suddenly, these large cascades can happen.

A computer malfunction can actually ground an entire airline for a small amount of time, because of the possibility for these cascading failures. Or a small power outage can actually cascade through an entire electrical grid, and actually take down a huge amount of the power grid and affect millions and millions of people.

There's this idea within complexity science -- a mathematical framework for thinking about some of these complex systems. There's a concept referred to as "robust yet fragile," and the idea behind this is that a lot of these very complex systems are highly robust. They've been tested thoroughly. They had a lot of edge cases and exceptions built in and baked into the system. They're robust to an enormously large set of things --but oftentimes, they're only the set of things that have been anticipated by the engineers. However, they're actually quite fragile to the un-anticipated situations.

Oftentimes because of the complex structure and the tight coupling of the pieces, they can actually have these failure cascades, or other aspects of fragility that are maybe un-anticipated.

There's a great deal of power in complexity I think. Overall, we're going to continue building very complex sophisticated technological systems, because on the whole, they're very powerful. They're sophisticated. They can do many different things. Interoperability is great. It can allow information to pass from one system to another.

The downside, though, is that not only are these systems more difficult to understand, but as a result of the gap in understanding and this increasing comprehensibility, there's going to be the potential for failures in ways that are un-anticipated.

Julia: It reminds me of some of the financial models that were like, five sigma, incredibly confident that their algorithm would not fail massively. But they in fact did fail massively in the 2008 collapse, because real life just wasn't captured in the assumptions made by the model.

Sam: Right. I get it. There's a model, and then there's all the details the model maybe has swept under the carpet. But it turns out sometimes, these exceptions can actually swamp the general rules, and yield things that really can cause big qualms.

Julia: We've touched on artificial intelligence a few times. You briefly described some of the fears that people have about advanced artificial intelligences. But you distanced yourself a bit from those fears by saying, "Look, we're not talking about Skynet here, where the risk is the machines becoming self aware and declaring war on humanity. We're just talking about unintended consequences because of the complexity of the systems."

I guess it's not clear to me just how different those two scenarios are in practice. The thing that people who worry about risks from advanced superintelligence are afraid of, they're worrying about things like, "Okay. We program the AI with a particular goal like, I don't know, maximize the profits of my company." We don't give it specific rules that it's supposed to follow. It will learn those rules through various machine-learning things, deep learning, et cetera. And maybe it turns out that one of the strategies it develops is a clever way to assassinate the CEOs of the competing companies. Yay, profit maximization!” For example.

So I'm not sure there's this clear line between, "Oh, AIs will turn evil just for no particular reason other than we're robots and we like to be evil." That is clearly a silly risk. In practice, it seems to me that just unintended consequences of complex AIs could produce results that *look like* AIs being evil, and that we should actually be worried about that.

Sam: No, I think that's actually a very fair statement.

I'm not focusing on these distant risks like complex superintelligent AI. For me, the error of incomprehensibility is not something that's on the horizon, or maybe happening within the next 10 or 20 years. It is here now. We're seeing this thing.

It can be even as simple as the situation like Microsoft's chatboy Tay. The designers intended it to interact I think like an 18 or 19-year-old girl. When in fact, it ended up behaving like a white supremacist. In retrospect people realized the input data that it was going to receive was different than what they expected. There's this mismatch between how they thought it was going to respond to the input data, and how it actually did. At the same time though, that's almost like a trivial example in comparison to all these other systems.

We need a suite of tools and approaches for when we're never going to be able to actually fully understand the systems, but we still need to use them. We still need to handle them in some way.

For me, I'm more focused on those kinds of approaches for the here-and-now rather than the future scenarios. That being said, I'm sure that these approaches are still very useful for the future scenarios as well.

Julia: Do you have any other proposals or advice for the kinds of risks from complex systems that we are in fact facing today?

Self-driving cars might be a good test case. Is the answer just, “Test them way more than we think we need to, in as many different varied scenarios as we can think of?” Or are there other principles that we can use to increase robustness even if we can't predict exactly what might go wrong?

Sam: I think there is “engineering hygiene” that you can use to make sure you're making the systems more modular, so that they're more understandable. There are a lot of ways of reducing the trends towards incomprehensibility, so you can stave it off for as long as possible.

For me though, the situation that I'm interested in is like, "Okay. Assuming the systems are incomprehensible already -- which in many cases they are -- how do you then approach them?" I don't mean to sound like they are capped out -- but I almost think that to a certain degree, we need to take the approach of technological humility.

In the scientific world, we're recognizing that there are limits in terms of the things we can understand effectively in physics. I think in technology, we need to recognize from the outset that there's going to be limits in what we can understand – like, even theoretical limits to what we can fully understand.

…If we build our systems from the outset with this recognition, we're not going to have the same approach of, “We can understand these systems” and then suddenly be confronted by a failure, and really be shaken by the fact that we don't understand them.

If we can recognize this from the outset, we're going to continue to try to iteratively understand the systems, but in a more humble approach. There's going always to be this mismatch. We'll keep on addressing glitches and failures, and trying to test these things as best we can. But everything is going to be always a work in progress.

I think it will also change how a lot of us, even if you're not an expert, how you approach the technologies you deal with. Essentially, right now, a lot of people outsource technological understanding to the expert. If we recognize more explicitly that even the experts cannot fully understand these things, then I think it will ideally create a certain amount of responsibility for each of us to at least try to better understand these technologies.

We're never going to understand them fully. But it can be as simple as like maybe paying more attention to what a progress bar is doing when you're installing something. Even if there is like a tenuous connection to what is actually going out underneath.

Finding ways of seeing under the hood of our technologies. Like even the command line of your sleek user interface on your Mac, or whatever it is, and see a little bit of what's going on under the hood. I think those kinds of approaches are going to be increasingly important, if we recognize explicitly that we are in a world of our own making that we don't fully understand.

Whether or not we can always have these glimpses under the hood, that’s a totally different situation. I think we need this playful approach to our technology -- whether you're an expert or not, at least try to understand things, even a tiny bit, in the area where we cannot fully understand them.

Julia: I definitely like the idea of having a better grounding in what's going on under the hood of the technologies I use. But it's hard for me to see a causal mechanism between individual users having that understanding, and there being less serious downside risks from those technologies. Is there ... What's the mechanism?

Sam: I think that's actually a very fair point. I think there'll always be some amount of downside risk. I think it's just more just how each of us can approach it.

…Yeah. It might just become changing your expectations. It's this thing where right now, it's like very sexy to learn how to code. Most people who learn how to code, they're not going to be actually making applications or, like, large software packages. But it at least gives them a certain mode of computational thinking.

And I think this is related to that. Where having a computational thinking, or recognition of how these systems work -- or how they don't -- or even just the types of failure modes for these kinds of systems, I think it will give people the proper orientation in how they react to these systems.

Maybe we should actually ask for more. Ideally, we want to have systems that we fully understand, or ones that are completely safe. I think we'll never quite get there. It's just about a better understanding of risk, and how we understand trade-offs, and all these kinds of things when we build systems.

Julia: …Cool. We're about at time. So I'll just remind our listeners: *Overcomplicated*, highly recommended, we're going to link to it on the podcast website. For now, Sam will move on to the Rationally Speaking Pick.

[interlude]

Julia: Welcome back. Every episode of Rationally Speaking, we invite our guest to introduce the Rationally Speaking Pick of the episode. That's a book or article or website or something that influenced his or her thinking in some interesting way. Sam, what's your pick for today's episode?

Sam: Awesome. My pick is the book *Immortality* by the philosopher Stephen Cave. It's a book that looks at the 4 different modes that civilization has tried to, essentially, become effectively immortals. Everything from remaining biologically immortal, to immortality of the soul, things like that.

He actually looks at the ancient antecedents, like the ancient versions of these approaches, as well as the more modern versions. Then, he actually takes each of them and dismantles them. And shows how all these different approaches are doomed to failure.

Then -- this is my favorite part of the book, although actually the entire book is amazing -- he then says, "Okay. What do we do now? How can we come back from this?" If we're doomed to never have immortality, how can we have some optimism in the face of this?

He's been talking about both ancient and modern wisdom. He then goes into the ancient wisdom literature, the things of Ecclesiastes and Stoicism. There's a huge body of literature on the power of mortality and transience, and trying to understand how to create meaning and value in our transient lives.

It's amazing book. Jumps between ancient history and biology and philosophy… I highly recommend it.

Julia: I'm so torn about that subject. Because I definitely see the appeal of finding a way to come to terms with mortality. But at the same time, I worry that, like, wow, what if we could actually significantly extend our life spans? And we're just turning away from that possibility because we don't want to give ourselves false hope? It seems like a really tricky line to walk.

Sam: He actually discussed this. There's a very qualitatively big difference between drastically increasing our life spans and making us immortal. Maybe for most people, the difference between living a million years and living forever -- obviously, there's a mathematical difference, but for most people, it would not feel different.

He says, actually, this distinction is important. "Forever" actually is never going to be possible. I agree that drastically changing our life spans would actually change how we live. I think it would change how we find meaning and how to think about it.

I think it was Woody Allen, I think he says, rather than "his immortality is in his work," he would much rather have his immortality being in not dying.

Julia: Right. Exactly. I actually was also thinking of that quote. I just really appreciate when people turn up their nose at these, like, nice pretty, or comforting platitudes. And they're like, "No. Actually, the common sense thing is just ... Dying is just bad. I don't want to die."

Whether or not that's the most psychologically strategic or healthy approach to take… I just appreciate sometimes, people pointing out that the platitudes are platitudes.

Sam: Yeah. Actually, in this book, one of the cool things, and it's been a number of years since I've read it, but -- he doesn't give short shrift to those approaches. The subtitle of the book is I think, "how our quest for immortality has actually driven civilization forward."

He's not saying these drives are bad and we should abandon them. Thinking about these kinds of things have actually driven civilization forward. I think he actually recognizes there's a great deal of power in thinking about in human longevity, or trying to understand the nature of biology, and all these kinds of things.

It's a very complex stance. I'm not saying there is only meaning to human life because it is transient. I think that's a silly way to think to about it. I think you can still say, "Okay, given that life is fleeting, how can we still make it meaningful?" I think that's the better approach to that kind of thing.

Julia: Cool. Excellent. *Immortality*, we'll also link to that. We will link to immortality on the podcast website!

Sam: Sounds great.

Julia: ... as well as to your book. Sam, thank you so much for coming back on the show. It's been a pleasure having you.

Sam: Thank you so much.

Julia: This concludes another episode of Rationally Speaking. Join us next time for more explorations on the borderlands between reason and nonsense.