Episode Book

EPISODE #5954

Layered Safeguards and the Math Behind Them

PUBLISHED	14 October 2026 at 10:09 UTC
DURATION	27:59
LISTEN	myweirdprompts.com/episode/defense-in-depth-redundancy-math-75180/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.4
SOURCES	11 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:41 UTC

SYNOPSIS

Three safeguards that each work ninety percent of the time should fail together only once in a thousand tries. That arithmetic is correct, and it's also where the trouble starts. This episode works through the real mathematics of layered redundancy, why the independence assumption behind it so often breaks, and what reliability engineers call common cause failure. From the beta-factor model and the Humphreys checklist to the FAA's one-in-a-billion flight hour standard, from dissimilar redundancy on the Airbus A320 to physically separating critical systems, the episode explores why adding more of the same thing is not the same as designing genuinely independent layers—and why knowing when to stop adding layers may be the hardest question in the field.

TRANSCRIPT



Corn

Okay, so I want to start with a number, because the number is the whole problem. Three safeguards. Each one works ninety percent of the time. The probability that all three fail is one in a thousand. Which means the system is ninety-nine point nine percent reliable.



Herman

And that arithmetic is correct.



Corn

That's what worries me.



Herman

It should. Because Daniel sent us a prompt about exactly this, and the way he framed it is the way most people frame it, which is why it's worth doing properly. The question is why several individually imperfect safeguards combine into something that looks bulletproof. He wants the mathematics of layered redundancy, and he wants to know what happens when the independence assumption that produces that ninety-nine point nine falls apart. Then he wants it grounded somewhere real. Data protection, business continuity, safety-critical engineering, personal resilience. And the part I think is actually the sharpest thing in the prompt: the difference between adding redundancy and adding independent layers. Those are not the same move and the difference is the entire subject.



Corn

He also wants to know when you stop.



Herman

Right, and that turns out to be the hardest question in the whole field. Nobody has a clean answer to it. We'll get there.



Corn

So this is defense in depth. And the thing I want to flag early is that it isn't a cybersecurity idea that got borrowed. It's the other way round. Nuclear reactors were built in layers before anyone was patching servers. Households keep a torch and a first aid kit and a bit of cash for the same reason. Anywhere a system has to survive its own failures, you find this pattern.



Herman

The information security version is just the most recent and the most badly marketed. The principle predates it by decades.



Corn

So walk me through the arithmetic properly before we break it.



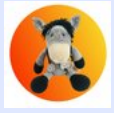
Herman

Three safeguards. Say a lock, an alarm, and a guard. Each succeeds ninety percent of the time. For the system to fail, all three have to fail at the same moment. If they fail independently, you multiply. Zero point one times zero point one times zero point one. Zero point zero zero one. One failure in a thousand. That's the calculation and it is not fake. It's real arithmetic and it works.



Corn

It works when the assumption holds.



Herman

And the assumption is independence. Which has a precise meaning that's worth saying out loud, because people use the word loosely. Two events are independent when knowing one happened tells you nothing about whether the other happened. Formally, the probability of the second given the first equals the probability of the second on its own. When that equality breaks, you have dependence. And positive dependence, where the two fail together more often than chance would predict, is exactly what reliability engineers call a common cause failure.



Corn

So the guard calls in sick because there's a flood, and the flood is why the alarm shorted out, and the flood is why the lock is underwater.



Herman

That's a common cause. One thing, and it takes out every layer at once. And this is the sentence that a 2004 paper in Risk Analysis put on the page, and I've been carrying it around for years. Some program managers share a common belief that adding a redundant component to a system reduces the probability of failure by half. This is true only if the failures of the redundant components are independent events, which is rarely the case.



Corn

Managers believe it because it's the arithmetic they were taught and nobody taught them the second half.



Herman

Nobody taught them the second half, and the second half is where the bodies are.



Corn

So how does anyone actually quantify this? Because you can't run a system with a hope in it.



Herman

You build it into the failure rate. The standard tool is the beta-factor model, and it's been the workhorse since Fleming introduced it in 1975. The idea is simple. Every component has a failure rate. You split it into two parts. One part is failures that come from inside the component and only affect that component. The other part is failures that come from a shared cause and take out everything sitting in the same boat. And the beta factor is what fraction of that component's failures are the shared kind. A beta of zero point zero five means five percent of the failures are common cause.



Corn

And the other ninety-five percent are honest.



Herman

Honest, local, uncorrelated failures. And you can see the trap immediately, because the model assumes that when a common cause failure happens, all the channels go. Every one of them. So the common cause contribution to the probability of failure on demand is identical whether you've got two channels, three channels, or five. Adding channels does nothing against a true common cause. You're just building more of the thing that's about to be destroyed.



Corn

That's the sentence I'd put on the slide. More of the same is not more protection.



Herman

And the beta values in practice are not small. There's a checklist from the late eighties, the Humphreys checklist, that scores a design on factors like separation, diversity, complexity, how often it's tested, how it's maintained. Score everything at its worst and you get a beta of zero point three. Thirty percent of your failures are common cause. Score everything at its best and you get zero point zero zero one. So the same nominal design can have three hundred times more common cause vulnerability depending on how it's built and looked after.



Corn

Three hundred times is a big spread for people writing the same spec sheet.



Herman

That's the point. The spec sheet doesn't determine it. The practice does. And the other big reference, IEC 61508, Annex D, has a thirty-seven question checklist, and it lands somewhere between half a percent and five percent for logic solvers, and between one and ten percent for sensors and final elements. So at the top end, one in ten failures of your sensor is a common cause failure.



Corn

Which means the second sensor is buying you much less than the invoice suggests.



Herman

Much less. And this is worth making concrete, because the abstract version drifts. Take a one-out-of-three system. Three channels, any one of them can do the job. Each has a dangerous undetected failure rate of five times ten to the minus six per hour. They're tested annually. There's a shock rate, a common cause event rate, of one times ten to the minus five per hour. And the probability that a shock takes out a given channel, given that a shock happened, is zero point two.



Corn

Spell out what the independent term comes to.



Herman

The independent contribution to the average probability of failure on demand is two point one times ten to the minus five. The common cause contribution is three point five times ten to the minus four. Add them and you get three point seven one times ten to the minus four.



Corn

So the common cause term is larger by more than a factor of ten.



Herman

Sixteen, seventeen times larger. The three-channel redundancy you paid for is sitting in the smaller term. Every additional channel you add keeps shrinking that small number while the big number sits there unmoved.



Corn

So the redundancy is doing something. It's just doing a fifth of what the brochure claims.



Herman

In that example, roughly. And that's a mid-range set of assumptions, not a hostile one.



Corn

What actually makes something a common cause? Is there a structure to it?



Herman

There is, and it's two pieces. Every common cause failure has a root cause and a coupling factor. The root cause is the underlying thing that, if you fixed it, the fault wouldn't happen at all. The coupling factor is the commonality that makes multiple items susceptible to the same root cause. Same design, same hardware, same software, same maintenance staff, same procedures, same physical location. And that second list is the one that should bother people, because look at what's on it.



Corn

Same staff. Same procedures.



Herman

The people are on the coupling list. And so is the maintenance schedule, and so is the room.



Corn

So your redundancy plan has a single point of failure and it's a Tuesday morning shift.



Herman

That's not a joke, that's the field. And there's a whole class of mitigation aimed at exactly that. You reduce the root causes, better components, environmental control, diagnostic testing. And separately you reduce the coupling. Physical separation, functional separation, electrical separation, hardware diversity, software diversity, simpler architecture. Note that most of that list is about breaking sameness, not adding more.



Corn

Which is the answer to Daniel's question, before we've even got to the examples. Adding redundancy is buying a second identical thing. Designing independent layers is asking what could kill both.



Herman

And most organizations do the first and call it the second.



Corn

Okay. So let's go where the stakes are highest, because aviation has been forced to solve this in public. What does the target actually look like?



Herman

The FAA sets it in Advisory Circular 25.1309-1B. The principle is one sentence. The more severe the hazard resulting from a system or equipment failure, the less likely that failure must be. And for the worst category, catastrophic failure conditions, the requirement is that they be extremely improbable, which is defined as an average probability per flight hour of one times ten to the minus nine.



Corn

One in a billion flight hours.



Herman

And Design Assurance Level A, which is the tier for the systems whose failure is catastrophic, requires you to demonstrate that number. Not assert it. Demonstrate it.



Corn

One in a billion. That's not a number you reach by being careful. That's a number you reach by deciding you need it and then engineering backwards from it.



Herman

And the way they engineer backwards is where it gets interesting, because the obvious move is to bolt in a second identical computer. And they found that doesn't get you there, because two identical computers running identical software have the same bug. Same source code, same compiler, same failure on the same input.



Corn

Same team wrote both.



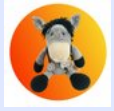
Herman

Same team, same spec, same assumptions. So aviation went to dissimilar redundancy. The space shuttle flew five computers. Four of them ran identical software. The fifth was an independent backup running different software, written by a different team, on different hardware, so that a design flaw in the primary software couldn't take out the backup.



Corn

The fifth one exists because the first four agree with each other.



Herman

The Airbus A320 uses five dissimilar computers running four dissimilar software packages. The Boeing 777 has three primary flight computers with dissimilar processors on independent channels, three unique control paths. And the reason for the word



Herman (continued)

dissimilar in every one of those sentences is common cause failure. Identical redundancy doesn't buy you the last two orders of magnitude. Dissimilar redundancy does.



Corn

And it's not just the electronics. There's a physical version of the same move.



Herman

There is. Fourteen CFR 25.795, paragraph C, sub 2. Redundant systems necessary for continued safe flight and landing have to be physically separated by minimum distances. So the two copies of a critical system aren't allowed to sit in the same bay, because a fire or an uncontained engine failure is a common cause that doesn't care how good your software is.



Corn

Separate rooms. That's the whole idea and it's not glamorous.



Herman

It's not glamorous and it's the one that shows up in every serious framework. And that gets us to nuclear, because nuclear is where defense in depth is a legal doctrine rather than a design preference.



Corn

And they've had to name the layers.



Herman

They have. The IAEA defines it as a hierarchical deployment of different levels of diverse equipment and procedures. Five levels. Prevention of abnormal operation, so the plant doesn't drift out of its envelope in the first place. Detection and control of deviations when it does. Control of accidents to limit releases. Mitigation of severe accidents, which is the core melt language. And mitigation of the radiological consequences of a release that's



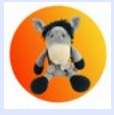
Herman (continued)

already happened.



Corn

Each level assumes the one above it failed.



Herman

That's the whole architecture. Level four is not there because they think level three is bad. Level four is there because level three might be. And there's a specific criterion that captures the operational version of this. N plus two. A safety function has to be performable even if any single component fails and another redundant component is simultaneously out of service for maintenance.



Corn

So they engineer for the failure and the maintenance at the same time.



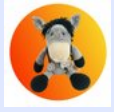
Herman

Because the maintenance is when you're actually vulnerable, and everybody in the industry knows it. And here's the twist that connects back to the beta factor. Nuclear has been shifting away from pure redundancy toward diversity specifically because digital instrumentation and control systems created new common cause vulnerabilities. When everything is analogue, two identical channels might fail independently. When everything is running the same software, they might not.



Corn

Same lesson as the shuttle. They learned it twice in different buildings.



Herman

And the diversity attributes they use are a list worth reading slowly. Human diversity, design diversity, software diversity, functional diversity, signal diversity, equipment diversity. Six different axes on which you can make two things not be the same thing.



Corn

Which means "independent layers" is a design property with at least six dimensions, not a checkbox.



Herman

And there's an admission buried in the sector's own documentation that I think is the honest heart of this whole episode. The CORDEL report from 2018 says no nuclear regulatory organization defines the specific term defense in depth and diversity, and that it is not used in any consistent manner.



Corn

After seventy years of building these plants.



Herman

After seventy years. They agree on the practice and they've never agreed on the definition. And on quantification, the same report says it is difficult to justify and separate from the qualitative aspects. Which means the most safety-critical industry on earth is running on a principle it can't cleanly measure.



Corn

I find that reassuring, weirdly.



Herman

Why?



Corn

Because it means the discipline is real and the scoreboard is fake. They know a good design when they see one and they can't put a number on it, and they've decided to live with that instead of inventing a number.



Herman

That's a generous reading and I think partly right. Now, data protection, where the stakes are lower and the discipline is much worse.



Corn

Give me the rule.



Herman

Three, two, one. Three copies of your data. On two different media or two different devices. With one copy off site. And you can hear the common cause analysis in each clause. Three copies handles hardware failure. The two-media clause handles the fact that a bad batch of drives or a bad firmware release can hit all your copies if they're all the same. And the off-site clause handles the fire, the flood, the theft, the burglary, and the flood that also happens to be the reason the alarm failed.



Corn

It's a redundancy rule that's actually a diversity rule wearing a number.



Herman

That's the right way to read it. And it's been extended, because the threat changed. Ransomware now targets backups first. So three, two, one, one, zero. The extra one is a copy that's immutable or offline, so the encryption can't reach it. The zero is zero errors verified on restore.



Corn

Which is the clause everyone skips.



Herman

Which is the clause that matters most, because a backup you've never restored is a hypothesis. And the whole point of the off-site copy is to survive the thing that took the primary. If you can't get the data back out, the copy was decoration.



Corn

And how many people actually do this?



Herman

Backblaze's own survey found eleven percent of computer owners back up daily.



Corn

Eleven.



Herman

Eleven percent. So roughly nine in ten people have a backup strategy that exists entirely in the space between "I should" and "I will".



Corn

So the human layer is the weakest and the least redundant. Which is exactly what the coupling factor list said. Same staff, same procedures, same maintenance.



Herman

The people are the common cause. And the industry knows it, which is why the checklist for common cause vulnerability includes things like training, supervision, and shift patterns, not just component quality. You can buy a perfect redundant system and staff it with one tired person who does two jobs.



Corn

And that's one person, one procedure, one point of failure. Now, the Swiss cheese model. I want to get to it because it's the metaphor everyone has heard and almost nobody knows has mathematics in it.



Herman

James Reason, 1990. The idea is that defenses are slices of Swiss cheese, and each slice has holes. The holes are weaknesses, and they vary in size and position. A hazard has to get through every slice to cause an accident. So the system is safe as long as the holes don't line up, and the accident happens at the moment they all line up at once, creating what Reason called a trajectory of accident opportunity.



Corn

Which sounds like a metaphor. You're telling me it isn't.



Herman

It's a metaphor with a theorem under it. Robert Frosch published a paper in 2006 that formalized it as percolation theory on a Bethe lattice. Same structure as how a fluid finds a path through a porous material. The holes aligning is a percolation threshold, and that's a real probabilistic object.



Corn

So "the layers failed together" isn't a story you tell after the fact. It's a phase transition you can in principle predict.



Herman

In a simplified model. The model still needs the correlations to be specified, so it doesn't rescue you from the hard part. But it tells you something the metaphor alone doesn't. There are systems where the holes are not just varying, they're correlated with each other. And once they're correlated, adding slices stops buying you safety.



Corn

So the metaphor and the beta factor are the same claim in two languages.



Herman

Same claim. Failures that share a cause break the multiplication.



Corn

So how does anyone decide when to stop?



Herman

This is where engineering runs into law, and honestly into ethics, and the answer is not a formula. The dominant framework in the UK is ALARP. As low as reasonably practicable. It comes out of the Health and Safety at Work Act of 1974. The idea is that residual risk has to be reduced until the cost of reducing it further is grossly disproportionate to the benefit.



Corn

Grossly disproportionate. Not "greater than". Grossly.



Herman

That word is doing real work, and it goes back to a 1949 case, Edwards versus the National Coal Board. The court held that risks have to be averted unless there's gross disproportion between the cost of averting them and the benefit. So it's a bias toward safety, not a neutral cost benefit analysis. The deck is deliberately tilted.



Corn

Which is the opposite of how most organizations actually behave.



Herman

Most organizations behave as though the burden of proof runs the other way. And the Health and Safety Executive's framing of ALARP is a diagram everyone calls the carrot. Three bands. At the bottom, broadly acceptable, where the risk is low enough that you don't have to do more. In the middle, tolerable, which is where the ALARP test bites, and you reduce until it's grossly disproportionate. At the top, unacceptable, where no amount



Herman (continued)

of benefit makes it acceptable and you have to stop entirely.



Corn

Three bands. Broadly acceptable, tolerable, and no.



Herman

And the famous unresolved difficulty is in the middle band, which is that you have to assign a financial value to a life. The framework forces you to, because you're comparing a cost against a reduction in fatalities. And there is no clean way to do it. The number exists in practice and everyone involved knows it's constructed.



Corn

That's the point where the mathematics stops and the judgment starts, and there's no paper that fixes it.



Herman

There isn't. And here's the thing that ties the whole episode together. The CORDEL report's conclusion is that quantifying defense in depth is difficult to justify, separate from the qualitative aspects, and that no regulator defines it consistently. So the field has a target it can set, like one in a billion flight hours, and a principle it can't measure, and a stop rule that rests on a value it can't defend.



Corn

So the answer to Daniel's last question, when has a system achieved enough redundancy, is that the answer is domain specific and there's no general formula.



Herman

There isn't. Aviation says one in a billion. Nuclear says n plus two and quote it to a framework. Data protection says three, two, one, one, zero. UK safety law says grossly disproportionate. Four answers, none of them derived from the others, all of them



Herman (continued)

ultimately a judgment.



Corn

Which means the honest position is that defense in depth is an engineering practice with a political conclusion.



Hilbert

I agree with all of that.



Corn

That's a first.



Hilbert

It's not a first. The part you missed is the layer nobody plans for. I've got one of those. It's sitting outside.



Herman

The generator.



Hilbert

Sixty kilowatt diesel. It's in a shed at the back because it doesn't fit through any door in the house, which I found out after I bought it.



Corn

You bought a generator you can't get into the house.



Hilbert

I bought it because the man selling it had a specification sheet and I had a checkbook. That's how these things happen. Anyway. The point is it's the third layer. Grid first. Battery second. Generator third. And the battery would carry the freezer and the router for about nine hours, so the generator's the one that matters.



Herman

So the layers are different. Grid, battery, engine. That's real diversity, different failure modes.



Hilbert

That's what I thought. Then the mains went down in February, and the battery kicked in, and I went out to the shed, and the generator wouldn't start.



Corn

Why?



Hilbert

Starter battery was flat.



Corn

Ah.



Hilbert

The generator had a battery. The battery that was supposed to charge the generator's battery was on the circuit that the generator was there to back up.



Herman

The third layer depended on the first two.



Hilbert

It did. And the man who sold it to me, Derek, he'd been telling me for two years to put a trickle charger on it, and I'd been telling him the whole point of a generator is that it doesn't need anything.



Corn

How long was the power out?



Hilbert

Forty minutes. The freezer was fine. Which is not the point. The point is that for two years I thought I had three layers and I had two layers and a shed.



Herman

That's the textbook version. It's a common cause failure through a coupling you didn't see, because the layers weren't independent at the level that mattered.



Hilbert

Derek said the same thing, in fewer words, and he charged me ninety pounds for the trickle charger and did it himself. I've not needed it since, which Derek says proves nothing.



Corn

Derek sounds like the most sensible person in this story.



Hilbert

He is. He also says the battery's not the real backup. The real backup is the manual pull start, which is under a cover, which is behind the generator, which is against a wall I can't get behind. So I've got a fourth layer I can't reach. But it's there. That's the one I count.



Herman

You count a layer you can't reach.



Hilbert

I count a layer nobody's tested, because that's what everyone does. I'm just honest about it.



Corn

Herman, he's just described the entire episode in a shed.



Herman

With a trickle charger and a wall.



Hilbert

One other thing. When I took the cover off to look at the pull start, I found a mouse had made a nest in the air filter housing. So the fifth layer is the mouse. I've kept him. His name is Kevin.



Corn

You kept the mouse.



Hilbert

He was there first. The generator's not going anywhere.



Corn

Right, so the thing I want to carry out of that is not Derek and not Kevin. It's the trickle charger. The failure wasn't the generator. The failure was that the backup to the backup was wired to the thing it was backing up.



Herman

Which is the common cause failure in domestic form. The layers looked different on the list and identical in the dependency graph.



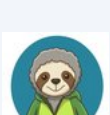
Corn

Let me try the general version. The ninety-nine point nine is not a property of the layers. It's a property of the assumption that the layers fail for their own reasons. And that assumption is doing almost all the work.



Herman

Almost all of it. Every time you break it, you lose most of the number. The one-out-of-three system went from roughly one in fifty thousand to one in twenty-seven hundred once the common cause term was included. Same hardware, same testing, same money. Different arithmetic because the independence was gone.



Corn

The industries that take this seriously have stopped trying to buy their way out of it. Aviation went to dissimilar software. Nuclear went to six axes of diversity. Backup rules went from three, two, one to three, two, one, one, zero. In every case the improvement came from making things not the same, not from making more of them.



Herman

The reason is that a common cause doesn't scale with how much you've built. It scales with how much you share.



Corn

When people ask how many layers is enough, the answer isn't a number. The answer is you can't answer it until you've drawn the dependency graph, and most organizations can't draw the dependency graph.



Herman

Which is roughly where the field is. Aviation can set one in a billion because they've spent decades mapping their dependencies. Nuclear can run n plus two because they have the staff dedicated to it. Everybody else is guessing and calling it a framework.



Corn

Which brings the last question back to us. If no regulator defines defense in depth consistently and nobody can cleanly quantify it, how does anyone know when they're actually safe versus just adding layers that feel safe?



Herman

You don't know. You make a judgment and you document the reasoning behind it, and you revisit it when the threat changes. The math is easy. The independence is hard. And knowing when to stop is harder than both.



Corn

Which is not a satisfying answer, but I think it's the first one all episode that's actually true.



Herman

The thought I want to leave people with is this. The nine in ten people who don't back up daily aren't making a math error. They're making an assumption error. They believe they have a backup because they've got something that looks like one, and what they actually



Herman (continued)

have is a copy with a shared dependency they've never looked at. So the question worth asking about any safeguard you own. What else relies on the same thing this relies on?



Corn

Which is the question Hilbert's trickle charger was answering all along.



Herman

It was.



Corn

My weird prompts is a human AI collaboration. The producer is Hilbert Flumingtop. The show is at my weird prompts dot com. Send us your own prompt on Telegram at t dot me slash MWP listener bot.



Corn

We'll be back soon.



Herman

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Rausand & Lundteigen, Chapter 10. Common Cause Failures (CCFs), NTNU SIS textbook, Version 0.1 [primary]**

   https://lundteig.folk.ntnu.no/Presentations/SIS-book/chapt10-1-CCFs-new.pdf

2. **World Nuclear Association CORDEL, Defence-in-Depth and Diversity: Challenges Related to I&C Architecture, Report No. 2018/003, April 2018 [primary]**

   https://world-nuclear.org/images/articles/CORDEL-Defence-in-Depth-Report-10-April.pdf

3. **FAA Advisory Circular 25.1309-1B [primary]**
https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_25.1309-1B.pdf

4. **Curtiss-Wright Defense Solutions, Optimizing Avionics Reliability With Dissimilar Redundant Architectures, December 6, 2018**
https://defense-solutions.curtisswright.com/media-center/articles/optimizing-avionics-reliability-with-dissimilar-redundant-architectures

5. **Wikipedia, Swiss cheese model**
https://en.wikipedia.org/wiki/Swiss_cheese_model

6. **Wikipedia, Defense in depth (computing)**
(computing)
https://en.wikipedia.org/wiki/Defense_in_depth

7. **Wikipedia, As low as reasonably practicable (ALARP)**
https://en.wikipedia.org/wiki/ALARP

8. **Backblaze, The 3-2-1 Backup Strategy of Data Protection, May 23, 2024**
https://www.backblaze.com/blog/the-3-2-1-backup-strategy/

9. **On the Limitations of Redundancies in the Improvement of System Reliability, *Risk Analysis*, 2004**
https://onlinelibrary.wiley.com/doi/10.1111/j.0272-4332.2004.00539.x

10. **Modeling Common Cause Failures in Systems with Triple Modular Redundancy and Repair, 2020 Annual RAMS**
https://dlnext.acm.org/doi/10.1109/RAMS48030.2020.9153662

11. **FAA, System Safety Assessments, Federal Register, August 27, 2024**
https://www.federalregister.gov/documents/2024/08/27/2024-18511/system-safety-assessments