DevOps: Transform Development and Operations for Fast, Secure Deployments Transcript

Part 1: Downward Spiral, Escalating Technical Debt, Security as Compliance, Burnout

Julia Allen: Welcome to CERT's Podcast Series: Security for Business Leaders. The CERT Program is part of the Software Engineering Institute, a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. You can find out more about us at cert.org. Show notes for today's conversation are available at the podcast website.

My name is Julia Allen. I am a principle researcher at CERT working on operational resilience. I'm very pleased today to welcome back Gene Kim. Gene is the founder and former CTO (Chief Technical Officer) with Tripwire. And today, Gene and I will be discussing some great research he's been doing over the past decade or so into the unique behaviors of high-performing IT and security organizations, including some real notaries like Netflix, LinkedIn, Google, Amazon, and Etsy.

Gene and his two coauthors have illustrated these ideas in story form in a recent novel about IT ops and DevOps, development and operations, titled *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. So we've also discussed some of Gene's foundational work in two previous podcasts, and I'll link to those when we post our show notes.

But with no further ado, welcome, Gene, it's really good to have you back on the postcast series.

Gene Kim: Oh, my gosh, Julia, I'm so delighted to be back. And I want everyone to know that Julia is one of my favorite people on the planet and has influenced my work probably more than anyone, so thank you so much, Julia.

Julia Allen: I appreciate those kind words, Gene. You've got a great value network, so I appreciate being included in your circle. So I had the good fortune to hear you speak about this -- what I consider to be groundbreaking work -- in a recent presentation you gave.

So let's use that as our structure. So, during that talk you stated that every company is an IT company. So from the point of view of your work, what does this really mean and why is it so challenging for most organizations?

Gene Kim: One of the motivations of why we wrote *The Phoenix Project* was we kept on hearing from organizations saying that, "We're not Google, we're not Etsy, so IT doesn't have to be a core competency." And yet when you look at the numbers, something startling shows up -- is that 95 percent of all capital projects have a technology component and 50 percent of all capital spending is technology related.

So it means that I think most people would say technology is the nervous system of the company. But it turns out that technology is actually the majority of the muscle mass as well. Whenever you have the organization saying, "Here's where we are and here's where we need to get to," Who's in the way? It's IT, and often if you go into the IT box, it's IT operations.

Actually it has two implications: one is that IT always appears to be in the way of the goals and aspirations of the organization; and secondly, not only is it impeding project work but it's

difficult to time your market needs to deliver new capabilities to the market and the opposite, which is how to deliver reliable, stable, secure IT service.
For decades we've thought that it's impossible to get both. In other words, you can't have fast feature flow into market while preserving stability and reliability. I think the biggest ah-ha moment, and the reason why I'm so excited to be a part of the DevOps movement, is that there's a pattern of behaviors that we're now calling DevOps where you can actually get astounding feature flow, hundreds or thousands of changes per day while achieving world-class security and reliability.

These are the practices that make the modern Internet companies run whether it's Google, LinkedIn, Twitter, Netflix. It's increasingly being adopted by enterprises as well, and I think it's going to become one of the most important competencies for organizations. And it has a huge implication for information security.

Julia Allen: Right, so we're going to talk about that in a little bit but let's stay with the pain point for another moment or two. So you talk about this core conflict between putting more products and services out faster and also, then, in the face of that constant churn and constant change, providing a stable, secure, foundational infrastructure within which all these new capabilities can operate. And you characterize this as a downward spiral, because it just seems like one thing -- it's a domino effect.

One thing happens and it exacerbates something else. And so what do you see as you've worked with different types of organizations as the behaviors, and conflicts, and actions that contribute to this downward spiral?

Gene Kim: It's sort of like in the movie Sixth Sense -- once you start seeing the dead bodies, you start seeing them everywhere. And so literally you start seeing the thumbnail of the downward spiral and it typically starts in IT operations. Whereas, IT operations our mission is to provide reliable, stable, and secure service to the customer but the problem is that so many of the assets that it relies upon for functionality are fragile. And so we call them fragile because every time we touch something, whether we try to patch it, or deploy change, or do a rollout, it blows up, and often catastrophically, and causes a huge amount of unplanned work for operations.

The tragedy is that where you find the most fragile artifacts are in the most critical business operations of the company or in the most critical business initiatives -- so both on the operations and project side. So when you have those important business initiatives and business operations relying on something that's fragile, the consequence is that the organization becomes unable to meet the commitments it makes to the markets, to the outside world, whether it's features promised to customers, or earnings per share, estimates to Wall Street, or promises about strategy and execution to analysts.

I think it’s part of human nature, and anyone, I think, who has been married can probably resonate with this. The way we compensate for missing our promises is to make even bigger, more audacious promises. And so when that happens, commitments are made to the outside world and now development sees urgent, date-driven projects put into the queue where a commitment has been made, and a date has already been set.

So now the problem is we have to cut whatever corner is required in order to hit the date. And so I think that has two consequences: one is that we focus only on the features as opposed to the non-functional requirements like reliability, stability, security; the second one is that we use up all the time in the schedule, leaving no time for the downstream work centers like QA, capacity testing, deployment, information security, operations. And so it means that at the end of every one of these projects, we have one more piece of fragile infrastructure in production. So that's one downward spiral. But there's actually one more insidious downward spiral, which is that as we accrue more and more technical debt, deployments start taking longer.

So the active deployment might take, in the beginning, like a minute or five minutes but then it starts taking an hour. It starts taking two hours. It starts taking a day. It starts taking two days. I've seen even deployments take six weeks. We estimated it took about 1,300 steps. When this is happening, no one is achieving their goals. Features aren't getting to market. Deployments are taking longer. We're getting more and more fragile artifacts. And now we're spending all of our time firefighting.

We spend more of our time on unplanned work, which is actually starving the entire organization to actually do more planned work to eradicate technical debt to reduce the surface area of risk for information security vulnerabilities. Once this happens, the organization becomes less and less able to not only make good on the promise it makes but also to survive as an organization. I think most people will resonate with some elements of the story, and I think the tragic part is that it happens everywhere.

Julia Allen: Right, and I'm just thinking about -- I have this vision of this snowball effect, and it just gets bigger and bigger, and more looming and more looming. And like you said, we keep making commitments that we can't keep which only exacerbates the situation.

But what I'd really like to spend a little bit of time on, I'm so glad you mentioned it, is with this constant focus on features, there's very little, if any, time left for all of the quality attributes that surround the product. So given this is a security podcast, can you say a little bit about why this behavior, and why this phenomenon that you've observed is relevant for information security. How does it affect them?

Gene Kim: Julia, I have to say it's so delightful to me that we're finding ourselves, again, in the same sort of research area of like how do we turn security into all the attributes like everything else in terms of non-functional requirement. But I think as an information security practitioner, and as a vocation, this downward spiral actually hits us worst of all, because we're the...

Julia Allen: Right, right, because there's no time left, right?

Gene Kim: Right, right, we're at the very last step. When people include us, if they include us at all, we're going to come at the very end. Right or Wrong, this is what happens. And, too, that means that the buffet table is empty by the time we get around to it.

I think more than most functional areas, information security is often pushed to the margins because we have a reputation of slowing things down. We have a reputation of blowing up the project due dates and the cost. I think we've created a learned behavior where the only way to get things done quickly is to bypass security. And so even when we're involved, we get engaged so late that there is no time to make a meaningful difference in outcomes.

I think the consequence of that is that as information security practitioners, we now get relegated to the role of compliance. So literally, our goal is to get all the boxes checked that were given to us from whether it's the regulators, or the external auditors, or the certification, the C&A (certification and accreditation) folks.

So I think for me that's very interesting, because at this point we're not making the organizations more secure. The only thing we're doing is not helping the organization achieve its goals. We're actually just trying to make an external third party happier, right?

A friend of mine, Josh Corman (he's a security researcher), he did a study of burnout in the information security industry. And the shocking finding was that the levels of burnout in info sec are actually higher than first responders, physicians in the emergency room, people doing multiple tours of duty in the military.

The three signs of burnout are fatigue, cynicism, not to say cynicism is our core competency, and the third one is illusions of self-efficacy. So we, as a profession, think that we're actually better than we are. So I think this is what happens when we live downstream, and you have to live with the consequences of decisions made far upstream.

I would just love to repeat the study to not only include just info sec but also IT operations as well. I think anyone who is downstream, having to live with compounding technical debt, because of decisions made upstream that have to live with it on a daily basis, I think you're going to see these characteristics. That's not great for the individuals, it's not great for the profession, and it's certainly not good for the organization.

Part 2: Improved Cycle Time, Early Involvement of Security, Ten Deploys Per Day

Julia Allen: Right, and not great for the product. I think you've painted a very challenging picture of what you've observed as the current state in a lot of organizations, so let's start discussing...

Gene Kim: Actually, can I add one more thing, Julia?

Julia Allen: Oh sure, sure, go ahead.

Gene Kim: I think the benefits, on the other hand, of a DevOp style workflow is that -- the reason why I'm so excited about DevOps for information security is that here finally is a way that we can get integrated into that daily work of both development and operations.

I think there are now an ever-growing number of case studies that are really showing that the cultural norms in a DevOps work stream is so consistent with what we want as information security practitioners. They want to focus on not just feature delivery rates but also quality scalability, deployability, manageability, security, visibility, all these things.

Julia Allen: Right, defect and vulnerability reduction, right?

Gene Kim: Absolutely, and I think one of the things that also is a huge benefit is the reduced cycle time. So when you're talking about doing tens, hundreds, or thousands of deploys a day, it means that the time from vulnerability detection to vulnerability fix generate to vulnerability fix deployed should be very, very short. We don't have to wait for that change window nine months from now.

So this, I think, inherently changes the working relationship between info sec and the other functional areas such as development and operations.

Julia Allen: Great, great, and I'm interested in exploring that with you further; because we've known for a long time that we need to address security earlier in the lifecycle but we've had a hard time getting uptake on that. And I think with DevOps, we really have an opportunity, and I'm sure you've seen many instances of making that so.

So let's talk about one. One of the examples that you cite, and granted this is their core competency so it may be difficult for a lot of smaller organizations to identify this, but you identify Amazon and what they've done as an example of a better way that takes advantage of the ideas in DevOps. So can you say a little bit about Amazon and some of their deployment statistics?

Gene Kim: There was a seminal presentation that I think really kicked the DevOps movement off in 2009 -- John Allspaw; he was the Head of Operations for Flickr and Paul Hammond was the Director of Engineering at Yahoo. Their startling presentation was that they're routinely doing 10 deploys a day, every day.

Julia, one of the things that you taught me when studying high performers is that the best always get better. The best always accelerate away from the herd. To see the acceleration in deployment rates is just, in my mind, breathtaking. Jon Jenkins at Amazon, he, at the Velocity Conference last year, went on record saying that they are now doing one deploy every 11 seconds.

So what is a deploy? A deploy is what is ever in the trunk of the source code repository -- is actually what's in production. So this is an extension of continuous builds, continuous tests, continuous integration -- definitely can be called continuous delivery. So that means code and infrastructure is being deployed at a rate of once every 11 seconds. And so that's a mean number of deploys of 22,000 per day. So they can be as small as...

Julia Allen: It's hard to even conceive of what that means, right?

Gene Kim: In contrast, most organizations have one deployment and release every year. So that's, what, four orders of magnitude difference? What a deploy can be, is it can be as small as a content change on the website. It can be a deployment of a new AB feature test. It could be as large as thousands of new environments going up being deployed behind a load balancer, so these are small changes and large changes.

I think one of the big ah-ha moments for me was realizing that I was often conflating deployments from releases. A prerequisite for deployment rates is that you are having developers check in once per day. So this often means that features are being deployed in production in a disabled state. This is as opposed to waiting until the very end of the project release to actually do the integration, the full feature testing.

So now what happens when you do that is that we are now combining the deployment risk, the inherent risk of making changes to production (which is one big batch of changes) and the release risk. We're doing it when it matters most on the day that we promised the feature to the market.

So in this state we have -- like in Operation Desert Shield, it's a yearlong project where thousands of men and materials are being put into place but do we have one release event? I think to separate deployment and release is just a wonderful thing because now development and operations can be held responsible for the deployment risk. The release risk we can now hold marketing responsible for. Did that business hypothesis actually pan out? Did that feature achieve the desired result?

One last comment, Julia, is that now suddenly if there's a sudden fix, a vulnerability, if we can develop a fix, chances are we can get it into production in hours or days as opposed to having to play that Shoots and Ladders game of trying to get all of the planets aligned to find a tentative window to shove it in production.

Part 3: Three Ways - Accelerate Lead Time, Shorten/Amplify Feedback, Practice Failure for Resilience

Julia Allen: Right, right, right, right. Well, let's talk about the DevOp's approach a little bit, and sometimes you call it Rugged DevOps when you add security considerations. So could you take us through?

You talk about different ways of thinking, the first way, the second way, the third way. I know in your presentation you walked through a structure of different perspectives and actions that are required to actually make this approach start to have some traction. So could you just start to walk us through that and I'll interject from time to time?

Gene Kim: In *The Phoenix Project*, this is closely modeled after the book *The Goal*. Julia, you know this is actually one of my favorite books that influenced me so strongly.

Julia Allen: Right, by Eliyahu Goldratt.

Gene Kim: Yes, the late Dr. Goldratt. They both feature a Yoda-like character who speaks in a gnomic, enigmatic ways. And so the language that our Yoda talks in is in a language of the three ways, which are the set of principles that we believe actually derive almost all the DevOps patterns from.

So the first way is a way of flow. So it's the flow from work as you go from left to right, from development into operations. And so why dev and ops? Because dev and ops are what's in between the business and the customer, so when the business says, "Here's where we need to go," that gets translated into work in dev that lets it successfully transition into service into operations. And then and only then have we actually delivered value to the customer. The goal is always to accelerate flow, so like in manufacturing.

According to Mike Rother, who wrote the book *Toyota Kata*, the most important metric for performance is lead-time. How long does it take to go from raw materials to finished goods? So the principles behind the first way are understand the flow of work, the notion that work should flow quickly in one direction. It shouldn't ping pong back and forth between dev and ops. Heaven forbid it shouldn't be caught by info sec who wants to drag it all the way back to the beginning. It should be smooth flow.

The second impetus would be the need to accelerate flow. So in the DevOps community, our favorite metric typically tends to be cycle time. In other words, how many releases do we do, deploy do we do a day? But, again, I think the most important metric is lead-time.

The metric I love to look at is how long does it take to go from code committed to code successfully running in production? We now know that for the benchmarking-- I work with an organization called Public Labs. We benchmarked 4,200 organizations. And we found that the high performers, the lead-time is measured in minutes or hours where the non-high performers the mean is somewhere between months and quarters.

Julia Allen: Wow, what a difference.

Gene Kim: Exactly. Interestingly enough, not only are they deploying code three times more quickly, the lead-time is much shorter but they're getting far better outcomes.

Julia Allen: You're talking about in terms of more stability, less disruption from the changes that just got introduced, right?

Gene Kim: Exactly right. Twice the change success rates. And when the changes fail (because Murphy does exist), the meantime to repair for those is 12 times shorter. So it's just stunning evidence that you can actually get both stability and agility. It's numbers. These high performers are out performing in both agility and stability.

Julia Allen: Well, Gene, why don't we move to the second perspective where we talk about the right to left -- how we get feedback going from the actual operations side of the house back into development? Can you say a little bit about that?

Gene Kim: Yes, absolutely. I think the concrete outcomes is that in order to achieve the first way is having ops and security work together to make environments available on demand. That means that whenever someone wants an environment, whether its developers or testers, they can get one with one click and it's an automated process. You don't have to open up a ticket. You don't have to wait 42 weeks.

That's available to anybody. And I think that one thing we found in the benchmarking is one of the behaviors that was almost universally present in every one of the high performers. I think the second way, whereas the first way was all about flow from left to right, the second way is all about the feedback from right to left. So how do you take those key learnings that we learn only in the failure modes in operations and info sec, and get that fed back earlier in the line? I think achieving one of the things that we've always aspired to do as info sec -- how do we get those things designed in and create quality at the source?

So the principles in the second way are shortening and amplifying feedback loops. Oh my goodness, I think it was you and Eileen Forester who taught me that, right? That the heart of every process improvement methodology is just desire to shorten and amplify feedback loops.

Secondly, is really create the capabilities and knowledge where we need it. I think the best example, the paragon of this behavior, is like the Toyota Andon cord where at the top of every work center is a cord that every employee is empowered and expected to pull the cord whenever we deviate from the expected behavior. So if a part isn't there, if a part is defective, if the work takes longer than specified, we pull the cord, which actually stops the entire production line so trucks no longer come out the other end. A friend showed me this article that in the Alabama plant, the Andon cord is pulled 3,000 times per day.

Julia Allen: Wow!

Gene Kim: It's just astounding. And so as info sec practitioners, how many times do we see something in a given day where we really think or suspect that the right thing to do is actually to stop the line? I think in a DevOps work stream that's not only acceptable, that's demanded.

I think the DevOps paradigm I just love so much in this second way are this notion of waking developers up at 2 a.m. There is this wonderful quote from Patrick Lightbody. He founded a company called BrowserMob. He said, "We found that when we woke up developers at 2 a.m., defects got fixed faster than ever."

Julia Allen: Sure. Sure.

Gene Kim: I think that's so wonderful, because it's shared pain and you need to have shared pain in order to have really shared goals. As opposed to the functional silo thing - “it's not my problem” -- for dev to just throw it over the wall and have ops and security live with the consequences. Having dev live with daily consequences about decisions made upstream I think is absolutely required to get these kind of amazing outcomes like increased stability while improving agility.

Julia Allen: Well, yeah, and some of the comments that you made that really resonated for me is have developers develop in an instance of the production environment. When they throw the code into operational deployment, have them be responsible for maintaining their own service.

So actually put developers into what is typically an IT ops role, and as you said, let them feel some of the pain. Let them be called at two o'clock in the morning. Let them really see what it's like to receive some of the code that they've developed downstream and have to deal with it, right?

Gene Kim: Exactly right. I just saw this quote from Werner Vogels -- he's the CTO at Amazon, and he said, "It goes beyond just that." He said, "It also gets developers closer to the customer," which is actually another one of these important things to increase feedback. Now developers can see the entire value stream and the ones that matter the most, not only just the downstream customer, the internal customer, but also the external customer as well.

Julia Allen: Right - well, Gene, we're coming up on time and I certainly don't want to give short shift to the third way, which is really what makes this stuff all stick, makes it all be part of the way that the organization does business as part of the culture.

So can you say a little bit about this cultural dimension of experimentation, and learning, and pulling the cord when something needs to be done? Can you say a little bit about that?

Gene Kim: Yeah, absolutely. The first way is flow, the second way is feedback, and the third way is creating a culture that I think does two things. One is this demand that everybody has to take risks, try new things, learn from successes and failures, and I think this is so -- I don't want to say synonymous.

There are so many movements right now, whether it's the Jim Collins “choosing to be great,” whether it's Eric Ries's “lean startup,” whether it's Scott Cook, the founder of Intuit, his rampant innovation culture -- this notion of learning from failure, constant experimentation. They talk about it a lot from this business side but I think it applies just as much if not more to the IT value stream as well.

So to give a concrete example, just because we've had the same software development lifecycle process, the same security-testing schedule, the same change management processes for the last 20 or 30 years, doesn't mean that we should be afraid to change them as long as we understand the goals, right? You can't achieve goals without understanding risks. I think if we understand what the global goals are, there has to be this culture that says, “Be allowed and expected to experiment.”

The second piece of this is that in order to do this safely, we have to understand the link between repetition as a prerequisite to mastery. I just love this emerging school of thought to saying, whether it's learning to play a musical instrument, or Special Forces training in the military, or training a sports team, it's always better to practice 15 minutes a day, every day. So it's always better to practice 15 minutes a day everyday than it is to practice once a week for three hours. What comes to mind is the annual security pen test. We have this ritual we go through where we tell everybody that the pen testers are coming. We get scheduled in a timeslot, and we have this theater that we're going to actually pretend that we're going to improve anything.

Whereas really what we want is something more like what Netflix does with Chaos Monkey where they were the only organization to survive the large- scale Amazon EC2 failures in 2011.When everyone asks why, why were they the only organization that survived, and what were they doing differently, one of the things that they talked about was a design decision that they made that said, "Because we can never depend on Amazon for availability, because it will never be there when we need them most, we made the decision that in order to survive failure we have to fail all the time."

And so they unveiled Chaos Monkey, which they described as running on every production server in the cloud, which randomly kills processes and entire compute instances. And they do this in production all the time -- large-scale fault injections in production. You talk about a way that creates more resilient code, resilient infrastructure, and resilient organization, that's it. That's the opposite of this annual theater event that we have called security pen testing. I think the benefits of that to info sec are just huge.

Julia Allen: Right, in other words you actually make failure, regular failure, small, medium and large-scale failure, a part of everyday operations. And so you develop a culture and a set of skills and competencies in your staff, and a robustness in your system assets, that just expects that that is the norm. And it turns out, that is the norm, right?

Gene Kim: Right, right, right, exactly. I think this is what I find so wonderful about where I work is overlapping it, again -- is this notion of like these are the attributes of resilient organizations. You can't get good. You can't get resilient if there's not some element of cost and adversity.

Julia Allen: Right, right. Well, listen, Gene, I know we've barely scratched the surface, and there's so many rich and robust concepts in the work that you've done that I hope our listeners are intrigued enough by our short conversation today to go pursue this topic further. So do you have some places where our listeners can learn more?

Gene Kim: There's a reading list that we've put together around *The Phoenix Project* book. There's a blog post -- it's called “Learn More About the Concepts of The Phoenix Project.” That's part one of at least two or three that's written. That's one resource. There's another presentation and that's at the itrevolution.com site.

The second one is what I think is one of the best info sec presentations I've seen in, oh my goodness, at least five years. And that was how the Twitter info sec team, what they did to create more secure applications and operations after the Barack Obama account was hacked and they got slapped with an FTC injunction to have a security program for the next 25 years.

The presentation is from the team that made that happen, which I think is amazing. It’s just showing how much they're influencing and integrating into the daily work of development. I think it's just magnificent. In my mind, 10 years from now, this will be the way every information security's going to be operating.

I think the third resource that I would point people to is if people are interested in a description of the downward spiral, and story of how does the transformation look like, *The Phoenix Project* -- we're making available for free the first half of the book. So I'll give a link there if you put in your e-mail address, we'll send you a PDF file. And that's really designed to be shared.

I think researchers have shown over and over that the most effective mode of persuasion and mobilization is storytelling. And so the book is just really designed to show the downward spiral. One of the most gratifying things of the journey is that we're getting so many comments. Just like Dr. Goldratt did, is that he got so many letters that said, "You have described our plant. You have described our characters. You've been hiding in our plant for the last six months." We get those same sort of comments.

In order to transform like this, you have to have fellow stakeholders in development and operations. And so the book is really designed to create empathy and shared pain showing that we're all in this together, right? And there's actually a bigger problem that needs to be solved. And so I'm hoping that the excerpt will be used as a way to find those fellow travelers and those other people required to create the coalition to make DevOps transformation happen, because I think it's urgent and important.

Julia Allen: Excellent, Gene. Well, thank you. I can't thank you enough for your time, for your preparation, for this incredible body of work that you've committed your professional life to. You mentioned kudos at the top of the call to me, but I'd like to say to you, it's been my pleasure to work with you, and collaborate with you and to move this whole conversation forward. So thank you so much for your time today.

Gene Kim: Oh no, thank you, Julia, and I'm looking forward to the next one.