accenture

Video Transcript

Greg Bledsoe,
Host, and Gene Kim, **CTO of IT Revolution and co-author of The Phoenix Project, The DevOps Handbook, and Accelerate**

Q&A with The Phoenix Project author Gene Kim

Greg Bledsoe: Greetings and welcome to another edition of Agile Amped, I'm your host Greg Bledsoe we are podcasting today from the DevOps enterprise summit 2018 in Las Vegas. Today my guest is the inimitable and illustrious Gene Kim, multi award winning CTO, researcher and author. He is the founder of Tripwire and served as a CTO for 13 years, his books include the Phoenix Project, the DevOps Handbook, Beyond the Phoenix Project, Visible Ops Handbook, Visible Ops Security and I think most recently co-author of Accelerate with Dr. Nichole Forzwin and Jess Humble. A very important book for the entire industry and if you haven't read it shame on you, go get it.

Gene is a huge fan of IT operations and how it can enable developers to maximize throughput of features from code complete to in production without causing chaos and disruption in the IT environment. Gene welcome and thank you so much for your time.

Gene Kim: I'm delighted to be here and hey thank you for being here at DevOps Enterprise.

Greg Bledsoe: Our pleasure, truly, truly our pleasure.

So let me just jump right in, so today someone, I was trying to prepare for this and I was asking people "if you had a few minutes with Gene Kim, what would you want to know?" And somebody was, we were trying to figure out what kind of questions to ask, and someone suggested that you are the de facto curator of the DevOps ecosystem and so it led me to the real question I wanted to ask, is how do you see your role in this ecosystem?

Gene Kim: Hmm that is interesting. I like the word curator, but I would maybe shrink that to the DevOps enterprise community. I mean I think that's something that is where I'm just passionately interested and the reason why I'm such a fan of Mirco Hering’s where one of your colleagues is at, I think for many years people thought that DevOps was not possible in large, complex organizations and so in the early years we actually had this kind of rule "no unicorns allowed." Only large complex organizations, the horses if you will. And so I do think they face unique challenges and so the goal was really to capture those stories and I think over the last five years we've really been looking for the most representative samples of these amazing journeys, these heroic, courageous journeys and yeah I would say the curator, collector, admirer because those experience reports give other organizations hopefully the courage to do the same thing right?

And in the age of digital disruption, it's increasingly not optional to modernize terminology because software's eating the world and what organization is not reliant on software these days to do what they need to do?

Greg Bledsoe: All business is cyber business.

Gene Kim: Yeah exactly.

Greg Bledsoe: And if you don't understand that, you're probably already half dead. So I always tell people agile is not, so this is the Agile Amped podcast, so the regular audience is a lot of people who are really passionate about Agile principals and that sort of thing. And I always tell people that agile is not a framework, agile is not scrum, agile is not, you can have agility without doing any of those things. But there's one thing if you don't do it I would have a hard time understanding how you're agile and that's the retrospective.

The retrospective to me is the heart of agile because that's the learning component and so I was wondering if you'd do a quick retrospective for us and if you had to go back in time and do it all over again, what would you do differently and why?

Gene Kim: Gosh that's a great question so I'm just going to think out loud here. So if I had to go back in time and redo some of those things, what would be? You know I think we did something different this year and I think we were a little more explicit in terms of framing the programming objectives. Right so you know at the highest level the goal is to capture team support but I don't think that we did a great job in clearly articulating how here are the program objectives, for each one of those what percentage of the program do those talks constitute and really expose the thinking about why those things are important.

Yeah, I think each year we had a difference in our program objectives and I think there's a clarity that comes when you actually have to explain it to other people. There's a columnist he said, "in order to speak clearly you actually need to be able to think clearly." And usually that involves writing it down right? So I found that within the programming committee and incidentally Marcus Mardell for example was part of the programming committee in the early years just to be able to broadcast that out and scrutinize it together and really kind of come to a shared understanding and agreement that these are the objectives that we're setting out to do for the next year was super helpful.

And just I think not only does it help in the selection process but also I think sharing it more explicitly with everybody, it helps to know where we're going right? What are we doing in the next three days and why?

Greg Bledsoe: Okay. Really interesting so let me jump to my next question because your time is limited here. So there's a theoretical maximum to what we can achieve with the methodology in its current form, one of the ways I describe DevOps is that it's really an open source methodology and that sort of has grown out of the sharing component of the culture of DevOps. And so my question for you is how close do you think we are of reaching the full theoretical benefit available through DevOps?

Gene Kim: Oh, great question I mean here’s the way I think about it. As good as we think we are, I think that we're a long way from like really exploiting the true, miraculous powers that technology enables. And I think about two things often.

Instagram was acquired by Facebook for a billion dollars and that was ten engineers. Pokémon Go was the fastest property to a billion users that was 25 engineers, so you know I think that the real goal is to be able to take that kind of productivity represent to any modern business context and use everything that is, compared to ten years ago, miraculous. Right? And to be able to solve those problems in any business context yeah so I think in my mind that's kind of an interesting benchmark. When you compare that to most organizations is that really the level of productivity that we see? I think we would say that we're in the single digit percentages of where we would ideally like to be. Low single figures.

Greg Bledsoe: Alright. So what do you think is the bridge between here and there?

Gene Kim: Oh my gosh. I think part of it is the, you know there's as Courtney Kisser from Nike and many other people have said is; "there's a level of technical that is dragging us down." And for complex business processes right, Scott proofing CSG is a great example, there is 45 years of business rules that are encoded in software but that doesn't mean that it can't be modernized. And that it can't be decoupled and so forth, so I think the winners are able to do what needs to get done and yet somehow unshackle themselves. Not by necessarily killing legacy applications, that's rarely the answer it's how you create the architectural practices that allow small teams of developers to be able to work independently without having to coordinate with 10, 20, 50 other teams right?

So I think that's the barrier and maybe just to put that into context I like chief architects as much as the next person but the way I was trained at Tripwire many years ago, 15 years ago. We were always trained it's always safe to ignore the architects. Especially chief architects because we all knew that they lived in their ivory tower, they came out once a year, they would publish a power point slide and a Visio diagram, email it to everyone and then go back to the ivory tower never to be seen again for another year.

I think that was just a funny way of saying they didn't impact how daily work was performed. But these days how daily we perform is entirely or to a significant extent dictated by the architecture that we work within. So if it was ever true that chief architects were just Visio jockeys that's certainly not the case now.

Greg Bledsoe: I couldn't agree with that more and you know one of the early learnings for me was that I could architect away all my emergencies and that's when life got good and that's why I started kind of, a lot of the DevOps principles I has kind of already implemented before there was a name for it, and I was trying to come up with my own catchy name and then Patrick DuBois beat me to it but so there's ...

Gene Kim: By the way this is one of the reasons why I love Mirco's book, "DevOps For The Modern Enterprise," is just there's so many examples that you are just wouldn't come to mind as things that you could architect around and whether it's mainframes or SAP systems or whatever right. So I love the book just because it is a mature, and I don't mean in like a mature you model, but I mean it's a book written through a point of view that understands that these are the things that have been built up over the last 40 years right that it's not going away right?

Greg Bledsoe: They don't go away in a day.

Gene Kim: Exactly.

Greg Bledsoe: So true. And I'll tell you here's like the number one question that people asked me to ask you and I was questionable on whether I wanted to ask it or not because-

Gene Kim: Oh now I can't, now I've got to hear it.

Greg Bledsoe: ... because I'm sure you get it all the time, but I have to start every engagement with I have to give, when I say DevOps what do I actually mean? What is the functional definition of DevOps? And everybody sort of has their own and I say donkey and people hear zebra. And I'm saying we're going to saddle up this donkey and ride it up this hill and people are like "you can't domesticate zebras, you're going to die."

So it's a definition problem and you know ITSM back in the day had very static definitions, this is the set of things that are contained and it may change and evolve but we can always go an point our finger to say what it is.

Gene Kim: Yeah.

Greg Bledsoe: DevOps has a very evolving, it's an open source methodology which we're throwing new things in their all the time, the problem from my perspective is if it means everything it doesn't mean anything. And so you have to definition that excludes what it isn't. What is your current functional definition of DevOps? Without nailing yourself into one.

Gene Kim: Sure, I think that was one of the many aspects of genius right, one is just to create such an easy name. I mean names are so powerful right and I think he very much resisted any sort of orthodoxical definition of DevOps. In my mind I need a definition and I'm going to try revising, I was going to say something until the Chris O'Malley talk this morning, that was a sea of copy ware and it was really great to get someone from outside the bubble, outside of our own echo chamber and really I had asked him here to help teach us how to appeal better to more conservative business leaders maybe who always over delegate technology.

Gene Kim: So anyway, I'm going to try to on the fly recast my definition.

Greg Bledsoe: Well once you give me yours, I'll give you mine and you see if you like it.

Gene Kim: Okay yeah so I think it is the set of architecture, technical practices and cultural norms that allow us to innovate and ideate and quickly delivers that to customers so that they're getting value as speedily as possible. So that means that we need fast, and it allows experimentation and innovation, it allows simultaneously the fastest flow through the technology values while preserving world class reliability, security and stability. So somewhere around there, I shuffled around a little bit just to make myself sound more like Chris O'Malley.

Greg Bledsoe: One of the problems is that the definition does change fast because we're learning, at first we didn't know that what we were really doing was implementing and giving score team points in software delivery. And then we said okay well let's break down barriers between departments, works really well let's try driving out fear and then kind of progress enumerating through those lists and so I think the underlying principle is of really lean thinking.

Greg Bledsoe: Right and so my definition of DevOps is; it's the realignment of IT around business value.

Gene Kim: That's very good. In fact, it reminds me of another definition I also love which comes from John Smart who for many years was the head of ways of working at Barclays. And his is; better value, sooner, safer, happier.

Greg Bledsoe: Something like that.

Gene Kim: Something like that.

Greg Bledsoe: And it incorporates, see like these definitions incorporate things by reference.

Gene Kim: Yeah.

Greg Bledsoe: And so I was really trying to incorporate the right things. So I feel like I drew in all of lean by reference by the realignment, or you got flow around value.

Gene Kim: And I think the characteristics of that definition are important because it doesn't actually talk about the practices, right?

Greg Bledsoe: Which will change.

Gene Kim: It's about the outcomes.

Greg Bledsoe: It's about the principles that were trying to guide us to the outcomes.

Gene Kim: And I think what's great about outcomes is that it really says, "I don’t really care how you got there, I don't care that these are the practices that enabled you to do it 15 years ago that predated the language." If you were achieving those outcomes, then that is achieving the goal.

Greg Bledsoe: The goal. Which is another great segue that I would have loved to have gotten into if we had a little bit more time. I'll hit you with one last question here; so is DevOps just a set of engineering practices?

Gene Kim: Is engineering just ...

Greg Bledsoe: Is it restraint in engineering practices?

Gene Kim: I don't think it can be. I think one of the program major objectives is how do better span the business and technology divide and that's really grounded in the observation that if you look at the journey of these technology leader in this community, increasingly the barriers they face are not within the classic technology value stream. Increasingly the barriers are outside of technology, it's business leadership, it's powerful project management officers, it's security and compliance, legal. So just by existence proof clearly it has to involve something bigger than technology because that's in the way so I would I guess categorically say it can't be just restricted to engineering principles.

Greg Bledsoe: And I agree with that, I'm glad you said that I wanted to get that on the record because I deal with that assertion quite a bit. And one of the things I say is that there is no technical system, it's really a human system that's layered on top of the technical system. And you can't change the technical system without changing the human system.

Gene Kim: Right.

Greg Bledsoe: Those are all the constraints so we're at the 15 minute mark so our guest today has been Gene Kim, Gene I really appreciate your time today. Really enjoyed our talk as short as it was. So why don't you tell people how they can keep up with you and what you're doing?

Gene Kim: Oh yeah wow that's a great question, probably twitter is actually one of the best ways to reach me I'm @realgenekim. And you can always hit me up on LinkedIn or whatever and what am I working on now? No, I think it really is helping, learning how to more predictably get senior business leadership onboard and to get them not to delegate the problem away. I think that was very much in the context of why we brought Chris O'Malley. Because he's spanned both sides both being very much responsible for creating processes and accountability and performance and holding people accountable as well as the other side right which is someone who came from a computer science background right? Has been in the ideation mode and so I thought he was fantastic because what better person to each us than one who has time in both roles?

Greg Bledsoe: Great point, a great point. So to our listeners thanks again for listening to this edition of Agile Amped, if you learned something new, if you found this interesting please tell a friend, a coworker, a client about the podcast and subscribe to hear more inspiring conversations. On behalf of Gene and myself thank you very much for listening.

Gene Kim: Thank you.

Speaker 1: Thanks for listening to Agile Amped, find more inspiring conversations at AgileAmped.com, iTunes and your favorite podcast app. If you have an idea for a topic or feedback on an episode reach out to us on Twitter, Facebook and Instagram or send an email to agileamped@accenture.com.