**Why everyone needs DevOps now by Gene Kim -** [**https://www.youtube.com/watch?v=vTxAR8EOc8A**](https://www.youtube.com/watch?v=vTxAR8EOc8A)

**Questions:**

What did Gene find was happening in 2007? ~1:30

Downward spiral in IT organizations independent of organization kind, non/for-profit, etc. shown in two acts:

1) Operations fixing fragile artifacts found in critical revenue generating operations or in critical transformative project. Organization ceases to be able to regain the momentum, people without the accurate grasp of technology and its ability.

2) Developers is another problem, urgent projects in queue and without input in product development that must cut things in order to meet project deadlines set by people without a clear understanding of capable technology. Accumulation of technical debt, to clean up when there is a little more time.

Increasing time in deployments, conflict in developers and operations.

What is the usual goal of operations? ~2:05

Ensure that products are pushed out at the right deadline and ensuring that it matches what

the investors want. Keep things running to provide a secure and reliable service to customers. Problem is fragile artifacts.

What are the most fragile items in a business? ~2:35

Critical revenue generating operations or in critical transformative project.

What is technical debt? ~4:30

Accumulation of technical debt, to clean up when there is a little more time. Technical debt is when extra development work arises when code is easy to implement in the short run. Decisions made that become increasingly more difficult to fix over time due to increasingly large number of promises made to promises.

Why is every company an IT company? ~8:15

Pressured simultaneously, respond more quickly to urgent business needs. Provide stable, secure, and predictable IT service. 95% capital projects have an IT component. 50% of capital spending is technology related. At any point, if a change needs to be made, it usually requires IT input or perspective.

What 8 companies are listed as providing unusual levels of stability, availability, and features? ~10:35

Netflix, Flickr, Amazon, Google, Spotify, Etsy, Facebook, Spotify(?), Twitter

Do you use any of these 8 companies’ software or services? (Personal question)

Amazon, Google, Spotify, Facebook

What was the two types of audience response to the conclusion of the John Allspaw 2009 presentation? ~11:15

People knew that something historically was happening, other reactions were breathtaking skeptical, i.e ill-advised, etc.

What is Amazon's code deploy rate according to the slide? ~14:10

11.6 time between deployments. 1,100 deployments in an hour. Max 30,000 hosts receiving a deployment. Mean, 10,000 mean number of hosts receiving deployment per day.

What did Intuit do during Tax season to it's software that normally would not seem like a good idea? ~14:50

165 changes in peak 3 months of tax season, production changes. Conversion rate of website is up 50%.

What are the three takeaways from Intuit's plan? ~15:40

Time to make changes is during peak traffic, great operations/development skills to make it happen and lastly, experimentation of ideas and learn from failures.

What is the first way? ~19:30

DevOps, more agile and more reliable. Successful changes and reparations, as well as faster deployments all together. Flow from developers into operations, what’s between business and customer, engineering requirements -> service -> operations -> customers. Lead time is best predictor for quality, customer satisfaction, and employee happiness. Code committed to code successfully running in production. Working and shippable code, demonstrated in an environment that resembles production.

What is the outcome of the first way? ~31:00

Deploy smaller changes, more frequently. Reduces risk and improves MTTR. Reduce batch size by 50%. Great for DevOps, Organization, and Customers. Creating single repo for code and environment, determinism in release process, consistent production environments, features being deployed delay without catastrophic failures, decreased lead time, faster cycle time.

What is the second way? ~ 31:40

Feedback, reciprocal feedback from Ops back to Dev. Learning from mistakes to prevent it from happening ot enable quicker detection and recovery.

What is the Toyota Andon Chord and the number of times it is pulled per-day relate to the second way? ~32:00

Stops production immediately, when things go wrong until issue has been resolved. They go back and ensure that everything is fixed, 3500 times per day. Despite the disruptive issue it helps with 2,000 vehicles a day, technical debt accrues downstream and ensures that the problem is fixed at the top.

What feature allows Google to safely let their engineers commit code to the direct source for all production projects? ~ 36:09

15,000 engineers on 4,000+ projects. All code checked into one source tree, 5,500 code commits a day, 75 million test cases. Transforms fear into boredom.

How many production metrics does the company Etsy generate each day? ~39:20

New developers must deploy within the first day. 250,000 production metrics every day. Low trust to high trust, more changes with less supervision. Lead time predictors, low trust, messengers are trust, difficulty in communication, failure is covered up, new ideas are crushed. High trust, information is shared, messengers are trained, etc. Trust model.

What is the conclusion of the second way? ~46:41

Defects and issues getting fixed. Disciplined automated testing enables many small, agile teams to work productively. All groups communicating and coordinating better. Everybody is getting more work done.

What is the third way? ~47:05

Continual experimentation and learning. Break things early and often. Expect and tolerate failure. Meant to allocate 20% of cycles to technical debt reduction. Opportunity cost of wasted IT spending, more than 2.6 trillion USD.

What is the outcome of the third way? ~55:55

By learning from issues and having a high-trust culture, it allows organization improvement through risk-taking. Value organizational input and ensuring that there is an environment that improves daily work, and converting local learnings into global learnings. Requires iterative and scientific approach.

Do you agree with the presenter or disagree with his thesis relating to development?  Why or why not?

I do agree in a sense, that DevOps is a methodical and practical concept about streamlining processes between the development and operations stages. There is no formal release handoffs, no waterfall-through-various environments, and the general strict and low-trust environment that most IT organizations were built upon. I agree that it is important to have a good understanding and open perspective in listening to both the developers and the operations side through small batch projects for efficient processes.