Margaret Shimerdla

Module 12 Assignment

CSD380-DevOps

Compliance

The case studies from our books discuss the importance of auditing code and using telemetry to audit code. Auditors did not have to worry about code changing in a game or software that had been created. An audit of a game or software was done by getting screenshots of code being used with that software. However, as the coding rules change, the auditing rules must be changed too. Even though it might not have been easy for auditors to check code, they knew what they needed to look for. They also knew that the code would not be updated for another year. However, more developers are updating their code regularly through cloud, GitHub, or other ways. Developers do not just update code once a year like in the past. Code is being updated regularly. This means that it needs to be audited more often. These case studies explain ways to audit codes more regularly.

"Providing Compliance in Regulated Environments"

The case study is trying to show how auditing is quite different today than it has been in the past. Auditors are supposed to request certain information from a DevOps team. However, they do not necessarily understand how to read code. This means the auditor must work with a programmer to better understand what they are looking for. In this case study it talks about creating a place where information about how the code works is placed. This information is placed into a telemetry system, such as Splunk or Kibana. When this information is placed on the web page, it helps prove that the processes they use for creating code are working and up to date. Splunk or Kibana is a website that is much used like a chat page. Creating it to be a chat page is a fantastic way to show auditors that everything is working well. Also, auditors can work freely in the chat room and access information that is needed. This case study helps to bridge the gap between DevOps teams and auditors.

"Relying on Production Telemetry for ATM Systems"

This case study talks about how it is important for auditors to start testing the diverse ways that code can be changed. Instead of auditors just looking to see if the company is doing what they are supposed to be doing, they should also look at the code regularly to make sure that it is correct. This should be done regularly to ensure the code is changed correctly and check who approves it. If the code is changed, then there should be a coding signature to see who made this change. If someone approves a code change, there should

be a way to tell who made it. These are things auditors need to know and understand to check code.

In conclusion after reading about these case studies it is important to note that technology is being used more in depth. Obviously, developers are always using code and trying to find new and improved ways to use it (AI—aka Scary Robot Takeover). However, this means auditors must be on top of their game to ensure all the code put out there is safe for people to use. Since code is changing regularly in programs these days then auditors need to be able to check the code whenever they want (limit AI—aka Stop Scary Robot Takeover).

Source

The DevOps Handbook

How to Create World-Class Agility, Reliability, & Security in Technology Organizations

by Gene Kim; Jez Humble; Patrick Debois; John Willis; Nicole Forsgren

The DevOps Handbook