Patrick Guerrette (PWG48)

SYSEN 5260

5/19/2023

cslds

1. At the very beginning of the Evernote SLO Story, McCormack refers to "undifferentiated heavy lifting." What does he mean by this term? Why did he want to move his engineering team away from it?

Undifferentiated heavy lifting refers to the IT work that is done that does not add value to the service Evernote provides. The engineers were spending a large amount of time and effort working on the backend portion of the service that the customer does not see. McCormack wants to move the engineers' focus to improving the product rather than continuously maintaining servers.

2. How do the goals of development teams differ from operations teams?

The developer team's goals were to implement new features while the operations team's goals were to maintain the product and ensure it is ready for production. The operation goals are easier to achieve if the product receives no new features. Developing new features can cause issues with existing code that the operations team will have to handle.

3. What was Evernote's first SLO? How did they choose it? How did they measure it?

The first SLO was an uptime measurement of 99.95% for certain services and methods. They chose uptime because it directly affected their customer base. The SLO is simple, and the customer will understand it. Uptime was measured by using probes independent of their environment. Evernote used frequency probes to ping their frontend nodes every minute. They used location probes to ping their specific servers/locations. If a ping fails, another probe will ping the same node. If the ping from the second probe fails, then the node is marked down.

4. When Evernote's services missed their SLOs, how did their team respond?

Missing their SLOs is an indicator of what Evernote needs to prioritize. They also meeting with Google to review their SLOs and do a deep dive into the root causes of the missed SLOs. Evernote then creates a plan for the improvements needed.

5. In The Home Depot's SLO Story, Bonnell writes about Home Depot's shift toward agile development and microservice architecture. How did this change the software reliability landscape?

Essentially by using microservice architecture and shifting to agile development, the software work was broken down into much smaller chunks. This allowed for the developers the freedom to commit their

changes, but also be responsible for any issues those changes cause. If an operations member discover an issue, they can immediately create a ticket for the developer that implemented resolve the issue.

6. What SLOs did Home Depot choose? How did they choose them? How did they measure it?

- Volume Traffic
    - Chose volume so operations and software can communicate this value.
    - Measured by requests to a service in requests per second, peak requests per second, and volume of requests over time.
- Availability
    - Chosen because uptime is important to customers.
    - Measured by pinging the server
- Latency
    - Chosen because of microservice reliability requirements
    - Each service determined the best endpoint to measure latency, but as a black box performance monitoring for issues caused from outside the service.
- Errors
    - Chosen so THD can track their services' performance.
    - THD chose to track 4xx and 5xx errors, but 5xx errors were strictly for the SLO measurement.
- Tickets
    - Chosen because THD historically measured software performance through tickets.
    - Measured based on the number of tickets.

7. When Home Depot's services missed their SLOs, how did their teams respond?

The Home Depot's response to a missed SLO is similar to Google's error budget culture. If an SLO is missed, the team would stop implementing new features and focus on fixing issues related to the SLO failure.