

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?
 - Meaning: Refers to the tasks and processes that are necessary for the operation of a business but don't differentiate the business in the market. For the engineering team this meant that they shouldn't be spending time managing the physical datacenters and the manual backend that don't affect the value that customers get out of the Evernote platform.
 - Why Move Away: McCormack wanted to shift the engineering team's focus from these routine, non-core tasks to more value-added activities, specifically product engineering that directly impacts customers. By moving to a public cloud, Evernote could offload these routine tasks and focus on enhancing product features and user experience.
2. How do the goals of development teams differ from operations teams?
 - Development Teams: Typically focus on creating new features and functionality for products. Their goal is to innovate and bring new ideas to life.
 - Operations Teams: Are generally responsible for maintaining the stability and efficiency of existing systems. Their focus is on reliability, uptime, and optimizing existing processes.
3. What was Evernote's first SLO? How did they choose it? How did they measure it?
 - The SLO: Evernote's initial Service Level Objective (SLO) was centered around uptime, specifically aiming for 99.95% uptime measured over a monthly window for certain services and methods.
 - Choice and Measurement: This SLO was chosen based on discussions with internal teams and feedback from users. It was measured using a third-party service (Pingdom) to probe their service status page, thus ensuring an independent and accurate assessment of uptime. The first ping was an unconfirmed outage flag if downtime occurred, which was then confirmed by another geographically located pinger.
4. When Evernote's services missed their SLOs, how did their team respond?
 - When Evernote missed their SLOs, the team used this as a feedback mechanism to prioritize fixes and improvements. The SLO/error budget approach helped them allocate resources to areas that required attention, ensuring continuous improvement in service quality.
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?
 - The shift towards agile development and microservice architecture allowed The Home Depot to increase the velocity and quality of software development. This change led to a more dynamic, modular, and scalable IT environment, requiring a more nuanced approach to software reliability and inter-service dependencies. Previously the teams would be so detached and non-communicative that new services that were being

maintained or upgraded would severely impact other services that depended upon the prior.

6. What SLOs did Home Depot choose? How did they choose them? How did they measure it?
 - Chosen SLOs: The Home Depot focused on SLOs for availability, latency, traffic volume, and errors for API calls.
 - Selection and Measurement: These SLOs were selected based on the nature of their services and the need to support agile and microservices-based architectures. They used automated metric collection platforms to track these SLOs with TPS reporting framework, ensuring consistency and accuracy.
7. When Home Depot's services missed their SLOs, how did their teams respond?
 - Similar to Evernote, when services at The Home Depot missed their SLOs, the development teams used this information to make informed decisions about prioritizing fixes and improvements using the VALET framework. The approach of using SLOs helped them maintain a balance between innovation and reliability, ensuring that the software development aligns with the overall business objectives.