Margia Rounok (mr879)

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?

   In the context of the Evernote SLO story, the term "undifferentiated heavy lifting" refers to the operational tasks and responsibilities that are common across many organizations and do not provide a competitive advantage or directly contribute to the unique value proposition of the product or service. These tasks are necessary for the smooth operation of the infrastructure and systems but do not differentiate the company from its competitors.

   Ben McCormack wanted to move the engineering team away from undifferentiated heavy lifting for a few reasons:

   1. Focus on customer value: By shifting the engineering team's focus away from tasks that don't directly impact the customer experience, they could redirect their efforts towards delivering features and improvements that customers care about. This means prioritizing work that adds value to the product and enhances the user experience.

   2. Increase engineering velocity: Undifferentiated heavy lifting often involves repetitive, time-consuming tasks that don't require deep domain expertise. By reducing the time spent on these tasks, the engineering team can allocate more time and resources to product development, innovation, and addressing customer needs. This can lead to increased engineering velocity and faster delivery of new features and improvements.

   3. Leveraging cloud services: Moving away from physical data centers and adopting a public cloud infrastructure allows the engineering team to offload many of the undifferentiated heavy-lifting tasks to the cloud service provider. Cloud providers offer managed services for various operational tasks, such as server provisioning, database management, and scaling, relieving the engineering team from the burden of managing these tasks themselves. This frees up their time and expertise to focus on higher-value activities.

   By reducing the focus on undifferentiated heavy lifting, Evernote aimed to create a more efficient and effective engineering organization that could deliver better products, improve the overall quality of service, and better serve the needs of its customers.

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

   The operations team was responsible for ongoing service delivery whereas the development team focused on future service extension and evolution.

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

Evernote's first SLO was focused on achieving 99.95% uptime measured over a monthly window. The selection of this specific number was based on discussions with their internal customer support and product teams, as well as user feedback. These stakeholders played a crucial role in determining an acceptable level of uptime that would satisfy customer expectations.

To measure the uptime SLO, Evernote implemented a system that involved testing a service endpoint to verify its expected functionality. They utilized a status page built into their service that exercised most of their stack and returned a 200 status code if everything was functioning properly. To ensure independent and comprehensive testing, they employed a third-party company specialized in building and running such tests, specifically Pingdom.

Pingdom acted as a prober, periodically calling the status page to check the service's performance. The frontend nodes were polled every minute, and multiple probes were placed in different locations across North America and Europe to gather data. If a prober check failed, a second geographically separate prober would perform another check to confirm the failure. Only after the second check also failed would the node be marked as down for SLO calculation purposes. The node would remain marked as down as long as consecutive probe requests continued to register errors.

In summary, Evernote chose a 99.95% uptime SLO based on inputs from internal teams and user feedback. They measured it by employing Pingdom as a third-party prober to periodically check the service's functionality from multiple locations, ensuring independent verification of uptime.

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

   When Evernote's services missed their SLOs, the team responded by using the SLO/error budget concept as a method to allocate resources and prioritize relevant fixes, improvements, and bug fixes. They conducted monthly reviews of SLO performance, involving teams from both Evernote and Google. During these reviews, they analyzed the SLO performance from the previous month and performed a deep dive into any outages or failures.

   The purpose of these reviews was to identify the root causes of the SLO misses and understand the impact on customers. By analyzing the data and conducting thorough investigations, the team aimed to identify areas for improvement that might not have been captured through regular root-cause analysis processes.

   Based on the analysis and findings from the SLO reviews, the team set action items to address the identified issues. These action items were aimed at making software and operations changes that would improve the service and enhance customer satisfaction.

   The response to SLO misses emphasized a continuous improvement mindset. By utilizing the SLO data and feedback, Evernote's team aimed to learn from their mistakes, prioritize customer-centric improvements, and refine their systems and processes over time.

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 toward Agile development and microservice architecture at The Home Depot brought about significant changes in the software reliability landscape. Here are some key aspects of this change:

1. Microservices Architecture: The Home Depot transitioned from supporting large, monolithic software packages to a microservices architecture. In this approach, software systems are broken down into smaller, independently operated services. This shift brought modularity, allowing for better scalability, maintainability, and fault isolation. It also reduced the impact of changes to specific services, minimizing the risk of failures propagating across the entire system.

2. Full-Stack Ownership: The company introduced a "freedom and responsibility culture" of full-stack ownership, where developers were empowered to push code when they deemed it appropriate. They also shared joint responsibility for the operations of their services. This change encouraged developers to have a deeper understanding of their services, leading to more proactive monitoring, troubleshooting, and accountability for the reliability of their code.

3. Service Level Objectives (SLOs): To ensure effective collaboration between development and operations teams, The Home Depot implemented service level objectives. SLOs provided a common language and metrics to measure and communicate the reliability and performance of services. Teams could now have transparent and consistent answers regarding the reliability, latency, and capacity of their services. This enhanced visibility into dependencies facilitated better communication, trust, and accountability between teams.

Overall, these changes in the software reliability landscape at The Home Depot fostered a more agile and resilient development environment. The adoption of microservices architecture, coupled with full-stack ownership and the implementation of SLOs, improved system reliability, scalability, and the ability to manage complex dependencies efficiently.

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

The Home Depot (THD) chose a set of SLOs (Service Level Objectives) for their systems based on the following criteria and considerations:

1. Availability and latency for API calls: Each microservice had to define availability and latency SLOs for its API calls that were invoked by other microservices. The SLOs were

published by the service being called, allowing the consuming services to determine if the service could meet its reliability requirements.

2. Infrastructure utilization: THD decided against setting utilization SLOs due to several reasons. Microservices were primarily concerned with handling traffic volume and meeting user demands rather than focusing on utilization. Additionally, their move to the cloud reduced the need for capacity planning based on utilization.

3. Traffic volume: THD needed to communicate the capacity of their services to handle traffic volume. They tracked metrics such as average requests per second, peak requests per second, and the volume of requests over a reporting time period. An SLO was set for traffic volume to ensure the services could handle peak capacity, particularly during high-demand periods like Black Friday.

4. Latency: Each service defined its own SLO for latency and decided where to measure it. Services were encouraged to use both white-box performance monitoring and black-box monitoring to capture issues caused by network or other external factors. Percentiles were preferred over averages, with user-facing services having a target of 95th and/or 99th percentile latency.

5. Errors: THD standardized error handling for web services, using HTTP response codes to indicate different types of errors. They tracked both 4xx and 5xx errors, but used 5xx errors to set SLOs. This allowed different applications to leverage the error dimension for various contexts.

6. Tickets: Tickets were historically used to evaluate production software at THD, so they continued to track them alongside other SLOs. Tickets were considered analogous to "software operation level."

To measure these SLOs, THD implemented automation and reporting systems:

- TPS Reports: A framework built on GCP's BigQuery platform that automatically captured VALET data for deployed services. It transformed logs into hourly VALET metrics stored in BigQuery, enabling automated reports and alerts.
- VALET service and dashboard: A service and UI created to store and report on SLO data. It allowed service registration, setting SLO objectives, and adding metrics types. The dashboard provided visualization and reporting capabilities, allowing users to query and analyze VALET metrics across services and time frames.

The proliferation of SLOs at THD led to the adoption of SLO-based approaches for various applications, testing, and future aspirations. They aimed to establish an error budget culture, refine VALET tracking for endpoints and consumers, extend VALET data to application deployments, analyze service dependencies, and involve business owners in setting SLOs based on the criticality of their services.

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

   When Home Depot's services missed their SLOs, the teams responded by taking action to address the issues and return the services to their target levels of reliability. Here are some of the ways they responded:

   1. Regular SLO Reviews: Developers conducted regular SLO reviews of their services, typically on a weekly or monthly basis. During these reviews, they analyzed the SLO metrics and identified areas where the services were not meeting their objectives.

   2. Action Items: Based on the SLO reviews, developers created action items to address the issues and improve the reliability of their services. These action items could include bug fixes, performance optimizations, infrastructure adjustments, or other measures aimed at resolving the underlying problems.

   3. Adjusting SLOs: In some cases, developers realized that the SLO objectives were unrealistic or needed adjustment. They could modify the SLOs to better align with the capabilities and constraints of their services. This allowed them to set more achievable goals and focus on improving within those revised targets.

   5. Automation and Monitoring: Home Depot invested in automation and monitoring systems to capture and track SLO data. This allowed them to have a real-time view of service performance and identify deviations from the SLOs. Monitoring systems would generate alerts when SLO breaches occurred, enabling quick response and remediation.

   Overall, Home Depot's teams took a proactive approach to SLO breaches, treating them as opportunities for improvement rather than accepting subpar performance. By conducting regular reviews, setting action items, adjusting SLOs as needed, and prioritizing reliability over new features, they worked towards building and maintaining reliable services for their customers.