Case Study

The Launch and Hand-Off Readiness Review at Google (2010)

Matthew R. Howard

Bellevue University

What are Site Reliability Engineers (SREs)? SREs are the cross between “treating operations as a software problem.” (Google, n.d.) Their mission is to “protect, provide for, and progress the software and systems behind all of Google’s public services — Google Search, Ads, Gmail, Android, YouTube, and App Engine, to name just a few — with an ever-watchful eye on their availability, latency, performance, and capacity.” (Google, n.d.)

Site Reliability Engineers help keep the high consistency of quality in both staffing and hiring which holds teams to standards. This is done by making sure that only the most important teams are assigned Site Reliability Engineers. And even if a team has an SRE, they still must self-manage and self-report on their service for a minimum of six months before it becomes eligible for an SRE to be assigned to it.

In order to help the self-managed process along, Google created two checklists as a safety item to use during the two critical stages of releasing new services: Launch Readiness Review (LRR) and the Hand-Off Readiness Review (HRR). The Launch Readiness Review is the first step in the process. It’s developer-managed and self-reported by the product teams themselves and must be signed off on before any new Google service is released to the customer or receives live production traffic (Kim, G., Humble, J., Debois, P., & Willis, J., 2016).  This is a great way to ensure only production-ready items are passed through and limits possibility of bugs and errors getting out of the production room and into the customers hands (i.e. quality control). Hand-Off Readiness Review holds a bit of a higher standard and is stricter in acceptance.

Whatever the checklist, the team going through them, has an SRE assigned to help them through the process. They assist in helping the teams understand and pass the requirements set by the checklists. And like many DevOps processes, the LRRs and HRRs have evolved and matured over the last 14-years and have helped to make Google the company they are today. When you force product teams to be responsible for the self-management of their own products makes service transition easier and more predictable as well as helps to create understanding between upstream and downstream work centers (Kim, G., Humble, J., Debois, P., & Willis, J., 2016).

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). Case Study: Creating Self-Service Metrics at LinkedIn (2011), Pages 207 - 208, *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* (First ed.). Portland, OR: IT Revolution.

Google (n.d.), Site Reliability Engineering: How Google Runs Production Systems, Retrieved August 10, 2018 from: https://landing.google.com/sre/

Case Study

Creating Self-Service Metrics at LinkedIn (2011)

Matthew R. Howard

Bellevue University

The innovative thinking of then-Intern Eric Wong, shows how thinking out-of-the-box can be of great benefit to any company. Because he didn’t want to waste precious time going through a sloth-like ticketing process to pull CPU usage off all the hosts or go digging through a cumbersome web interface, so he decided to write his own code to streamline the process and pull metrics quicker than the incumbent system was currently doing.

Because LinkedIn is a networking platform for better job opportunities, they need to ensure they can handle the growing amount of traffic they receive every day. To do this, they need to be able to make the full functionality of the site available every day for every user. This meant that from the off, they needed to catch bottlenecking and failures as they begin to happen.

The overwhelming amount of telemetry data what making it difficult for the engineers to access and analyze the data. This is where Eric Wong’s innovative thinking came in. And even though he still had to go through a manual set up to collect the metrics, the amount of time he spent in navigating Zenoss was reduced. By adding more functionality to InGraphs, engineers were able to get the exact information they needed across multiple datasets. InGraphs also gave them to ability see real-time performance of a provider’s major web-mail sever which enable them to fix it promptly.

The takeaway I got from this is that there are always improvements that can happen. Sometimes it takes re-inventing the wheel and sometimes, in the case of InGraphs, it takes creating something to aid the existing system. The support that Eric Wong had from LinkedIn leadership shows their belief in the culture of innovative and creative thinking as well as the agile method. Change can come from anyone and the fact that LinkedIn celebrated him by plastering his graphs front-and-center on the walls engineering offices shows their faith in the interns they hire and freedom they give them to be innovative and creative thinkers.

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). Case Study: Creating Self-Service Metrics at LinkedIn (2011), Pages 207 - 208, *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* (First ed.). Portland, OR: IT Revolution.

Wong, G. (2011), Eric the Intern: The Origin of InGraphs, Retrieved August 7, 2018 from: <https://engineering.linkedin.com/32/eric-intern-origin-ingraphs>