Principles of sustainability to create viable systems

During my internship at Cisco Meraki, my first major project involved improving the automatic machine lifecycle by optimizing the way firmware images were hosted and distributed across our Private Cloud infrastructure. Previously, updating firmware on physical machines relied on servers which are planned to be decommissioned for inefficient data retrieval and extensive resource consumption.

To address these inefficiencies, I designed a solution that automated firmware installation and updates, ensuring that machines could retrieve the required images seamlessly. The initial plan was to pull firmware images from a backend using a proxy server hosted in each data center (DC). However, to further enhance sustainability, these servers would also maintain a local cache in each DC, aligning with sustainable engineering principles that emphasize resource conservation and process optimization (Dowling et al., 2016). This ensured that once a file was requested, it would be stored locally, allowing future requests for the same file to be served from the local cache rather than repeatedly querying the backend. This reduced bandwidth usage, lowered operational costs, and improved system responsiveness, directly contributing to the sustainability of our infrastructure.

As I worked on implementing this solution, I encountered challenges related to documentation. Given Meraki's vast and complex infrastructure, I had to navigate through extensive internal documentation to understand the existing firmware update process and the broader lifecycle of physical machines. However, I soon discovered that much of the available information was outdated, deprecated, or entirely missing critical details related to my project.

This lack of documentation made troubleshooting and development more difficult, forcing me to rely on trial and error, debugging, and consulting senior engineers to fill in the knowledge gaps. Through this experience, I realized that sustainability in system design isn't just about automation and efficiency—it also requires knowledge preservation. Hicks et al. (2000) emphasize that designing systems with sustainability in mind reduces long-term operational inefficiencies and prevents unnecessary redevelopment, reinforcing the importance of maintaining accurate documentation for future engineers (Hicks et al., 2000). A well-documented system ensures that future engineers can build upon existing solutions without unnecessary repetition or reliance on tribal knowledge.

By automating firmware updates and implementing local caching, I helped create a system that not only reduces resource consumption but also improves long-term

efficiency. This aligns with the principles of sustainability, ensuring that our infrastructure remains scalable, cost-effective, and resilient.

Additionally, my experience with documentation gaps highlighted another crucial aspect of sustainability—knowledge management. Without proper documentation, teams waste time rediscovering solutions, leading to inefficiencies that hinder long-term system viability. By actively updating and creating documentation, I contributed to making our processes more sustainable for future engineers.

Moving forward, I plan to integrate sustainability-focused thinking into all my projects by considering not just immediate efficiency gains but also long-term maintainability and resource optimization.

I will continue to emphasize the importance of documentation in every development effort, ensuring that knowledge is properly captured for future reference. Additionally, I will advocate for proactive documentation practices within my team, reinforcing the idea that a well-documented system is as critical to sustainability as automation and optimization.

This experience has deepened my understanding of how Site Reliability Engineering (SRE) principles contribute to building viable, self-sustaining systems—where automation, efficiency, and knowledge continuity work together to ensure long-term success.

Professional Practice within intercultural and global contexts

During my internship at Cisco Meraki, I experienced a significant shift in work culture compared to my previous corporate roles, particularly at the Australian Taxation Office (ATO). The work environment was far more casual and flexible, contrasting sharply with the rigid structure I was used to. Additionally, working with colleagues across different time zones (UK and US) introduced collaboration challenges, requiring adaptation to asynchronous workflows and flexible scheduling.

The casual work culture at Cisco was immediately noticeable. There was no strict dress code, no micromanagement, and no rigid work schedules. Senior staff interacted casually with everyone, and meetings were more relaxed. At first, I found this lack of structure unsettling since I was used to the strict scheduling at ATO. However, as I adapted, I realized that this flexibility reduced stress and allowed me to excel in my role. Research shows that flexible work environments can enhance job satisfaction and reduce burnout, especially in tech-driven fields (Bloom et al., 2015).

Another challenge was working across multiple time zones. My manager and half my team were in the UK, while some colleagues were in the US, meaning meetings were often outside normal working hours. While I was comfortable with this due to the flexibility I had during the day, it sometimes led to missed meetings or rescheduled discussions when schedules didn't align. This made asynchronous workflows more important, but I was fortunate to have two senior engineers in Sydney who were very knowledgeable and helpful, ensuring I could get guidance when needed.

This experience helped me appreciate how global teams operate effectively. I learned that a casual work environment doesn't mean inefficiency, and flexibility can improve engagement. At the same time, working across time zones requires strong communication, adaptability, and local support networks to avoid delays. According to Ford et al. (2021), successful remote collaboration in distributed software teams depends on clear communication, time zone awareness, and shared documentation practices.

Moving forward, I plan to:

- Find a balance between flexibility and structure to maintain productivity.
- Leverage asynchronous communication and clear documentation to collaborate efficiently with global teams.
- Build local support networks within international teams to ensure smooth workflows.

References

Dowling, D., Hadgraft, R., Carew, A., McCarthy, T., Hargreaves, D., & Baillie, C. (2016). *Engineering your future: An Australasian guide* (3rd ed.). Wiley.

Hicks, D. I., Crittenden, B. D., & Warhurst, A. C. (2000). *Design for decommissioning: Addressing the future closure of chemical sites in the design of new plant. Institution of Chemical Engineers, Trans IChemE*, **78**(Part B), 465–476. https://doi.org/10.1205/095758200531014

Bloom, N., Liang, J., Roberts, J., & Ying, Z. J. (2015). Does working from home work? *Evidence from a Chinese experiment.* The Quarterly Journal of Economics, 130(1), 165–218. https://doi.org/10.1093/qje/qju032

Ford, D., Storey, M. A., Zimmermann, T., Bird, C., Jaffe, S., Maddila, C., ... & Butler, J. (2021). *A Tale of Two Cities: Software Developers Working from Home during the COVID-19 Pandemic.* ACM Transactions on Software Engineering and Methodology (TOSEM), 31(2), 1–37. https://doi.org/10.1145/3487560