

## SERVICE MANAGEMENT AND SITE RELIABILITY ENGINEERING

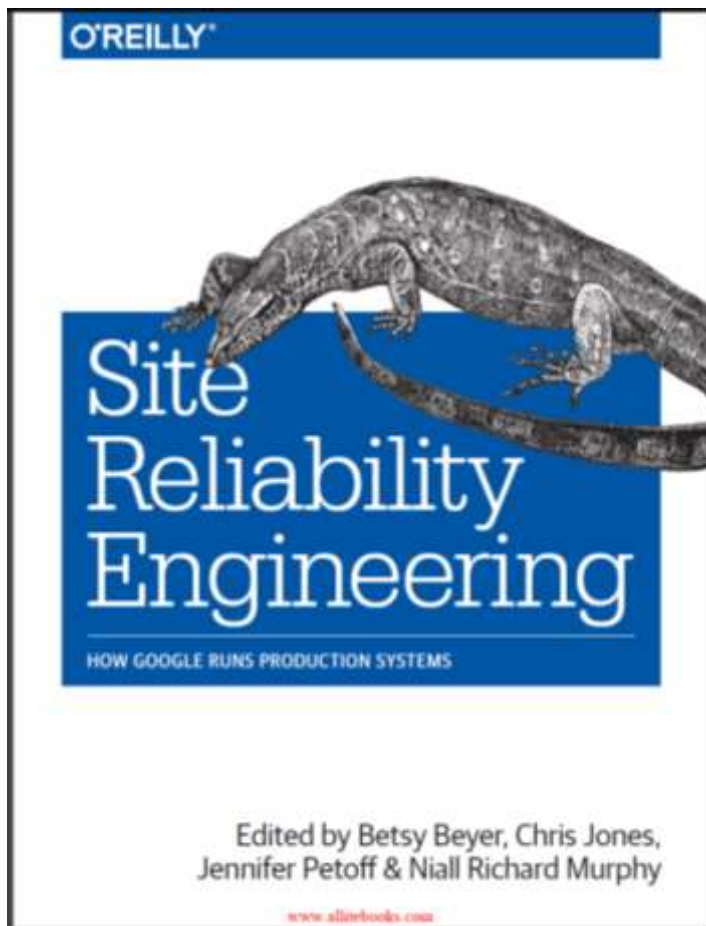
**NOTE:** You will be able to find this, and all my previous articles at [www.sm4all.com.au/history](http://www.sm4all.com.au/history)

*This is the **third** in a series of four postings on some new frameworks and methodologies around Service Management. How do they work together, and what really matters?*

The frameworks under discussion:

1. ITIL 4
2. The new DevOps and Agile Service Management
3. **Site Reliability Engineering**
4. What really matters

### A new player in the game



There is an exciting new IT role coming out of Google. The need for service and site reliability in such a global player is obvious. This has led to a specific role focused on expanding the operations space. The **Site Reliability Engineer (SRE)** is a skilled software developer driven to improving existing services. Building support systems and tools well beyond the batch job realm of traditional operations.

Staff at Google have released a tome (500+ pages) on Site Reliability Engineering. A new role in IT fast growing in importance. The concept originated in Google, but is applicable to enterprises of all sizes. How does this new role fit in the service management world? How does it relate to ITIL practices?

The book itself is a heavy read. There are chapters that are quite technical and specific to the Google environment. However, there is a lot of material which makes use of service management processes to ensure greater reliability.

Various chapters of this book deal with ITIL practices familiar to us all.

- Incident Management (Chps 11, 12 and 14),
- Problem Management (Chps 15 and 16),
- Availability and Continuity (Chp 13),
- Validation and Testing (Chp 17), and
- Capacity Management (Chps 18 -22)

These are a deep dive into the practical means of delivering on these practices. Certainly worth the read. I suggest much of the content will find its way into the various Practice Guides of ITIL 4.

## **The First Rule of SRE**

Google applies an SRE rule of 50% operations (Service Support) and 50% development (Service Delivery). Remember ITIL v2! The Development in order to improve the services from a customer reliability perspective (Service Warranty) and Operations to keep the SRE in touch with the real/operations world.

The SRE is an ideal fit in the DevOps space. Engage SREs early in new service development to plan and simplify the launch of that service. All services will undergo many releases as future enhancements pass through service value streams. The SRE can make this transition a whole lot smoother. In fact, there is a specialist role – Launch Coordination Engineer – within SREs, especially for this purpose.



## Rule 2: Toil is bad

One of the major goals of SRE is, “the elimination of toil.” The definition being given as

***“Toil is the kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as a service grows.”***

Some great points here:

1. Tasks that can be automated
2. Devoid of enduring value (backups, housekeeping)
3. Scales linearly as a service grows.

## From automation to autonomous

Removing the human element of supporting services improves the reliability of systems. Yet automation is more than automating current processes. Review activities, in line with Service Level Objectives (SLO), to Lean processes. Identify complexity, both Essential Complexity and Accidental Complexity, and look to remove the accidental component. Simplified services make automated recovery procedures easier to spot and design. A balance between Stability and Agility can only be achieved through Simplicity.

The downside of highly effective automation is human operators become progressively more relieved of useful direct contact with the system. Automation covers more and more daily activities over time. Inevitably, a situation arises in which the automation fails, and the humans are now unable to successfully operate the system. Hence, the need for the 50% operational element.

Throw in some AI and we start to have services with reliability at levels hard to imagine, let alone achieve. The book then answers the interesting question of, How much reliability do we need? Look at the Service Level Objective on reliability. Is it 99.99% or 99.999%? Where is the weakest link in the reliability chain?

## Devoid of enduring value

All human activities should be focussed on adding value. This is perhaps more so in the operations than in development. Where the tasks are not adding value, either eliminate it (if it is accidental complexity), simplify and automate it (if essential complexity).

## Scales linearly as a service grows

This aligns with another principle in the SRE space - Service support must grow at a lesser pace than linear, as the service grows. Otherwise, you will never have the human resources to support the services. SREs are given the development element of their role so as to address this. Removing the toil will remove the demand to constantly recruiting more support staff.



## Some other interesting concepts in SRE

### Error Budgets

Not actually a monetary budget. Error Budgets are a means of managing release agility with reliability expectations.

$$\text{Service Uptime} - \text{Service Level Objective} = \text{Error Budget}$$

If the service availability exceeds the availability objective (little or no downtime) we have a positive Error Budget and we can continue to release more service updates. Should the availability fall to the SLO, or lower, service releases must be put on hold. Recovering from a service failure 'spends' some or all of the Error Budget. The effect is that service development now have 'skin in the game' to produce a more reliable service or suffer not being responsive to their customer.

## Launch and Iterate mantra

Aligned to agile development, agile reliability improvements should be delivered in regular, small, flexible packages. Have customer feedback sessions to confirm the enhancements are delivering what was planned.

## Production Readiness Review

The equivalent of Operational Readiness checking. An agreed structure of review to be applied to decide whether the quality and warranty aspects are acceptable. Three approaches presented, the strongest being **the SRE Platform** – a skeleton service structure (standardising reliability, security, availability, infrastructure, etc) with the service specifics being ‘plugged into’ this skeleton.

*This reminds me of the ITIL v2 Application Management superstructure development, where service functionality is plug and pay. Releases become small and localised, allowing flexibility and reduced risk.*

## Terminology mapping

“Problem Management investigation” is called “Post Mortem review”. The essential element is still a ‘No Blame’ culture.

“Capacity Management” is called “Service Intent Planning”.

Still, the mapping to ITIL practices is not a hard one.

## Conclusion

I love the innovation of the new Site Reliability Engineering role. Rather than being concerned about the ever-changing IT world, Service Management professionals should embrace this new role as it aligns so well with existing (and future) ITIL thinking.

**Bring it on!**