✓ 200 XP

Key SRE principles and practices: The human side of SRE

10 minutes

A successful operations process, one that achieves the desired reliability and sustains it, is as much dependent on how we treat the machines as it is how we treat the humans responsible for that environment. Site reliability engineering acknowledges this truth in a number of ways that are crucial to its practice.

Toil

The first is a focus on the notion of “toil”. In an SRE context, toil refers to operations work being done by a human that has certain characteristics. Toil has no long term redeeming value. It doesn't move the service forward in any meaningful way. It is often repetitive and largely manual (even though it could be automated). As the service or systems gets bigger over time, the number of requests for that system will also probably increase in quantity at a proportional rate and require even more manual labor.

For example, if a service requires the SRE team to reset something every week, or to provision new accounts and disk space by hand, or repeatedly restart it by hand this is operational load that is toil. Completing those actions hasn’t made the service better in any long-term, persistent way. These actions will likely have to be repeated over and over.

> ⓘ **Note**
>
> Even if you keep requests of this nature in some sort of ticket system as many places do, performing the action and resolving a ticket is still toil. It's just well-tracked toil.

SREs hate toil. They work to eliminate it whenever possible and appropriate. This is one of the places automation comes into play in SRE. If these requests can be handled automatically, that frees up the team to work on more rewarding and impactful things than draining the request queue.

It should be mentioned that the use of the word “appropriate” here is similar to its use around reliability. There are situations where toil elimination work is of lower priority than other work, but on the whole, stripping toil from a service is a key focus for an SRE.

Project work vs. reactive "ops" work

To do the work necessary to remove toil, or improve the reliability of a system, an SRE’s time has to be allocated in such a way that they aren’t spending all of their time firefighting, replying to pages, or just processing a ticket queue. They need to have the time set aside to write code to eliminate the toil, construct self-service automation so tickets aren’t necessary, build projects that make the service and the people more efficient. The figure usually cited (which comes from the original Google model) is one of no more than 50% operational load on a team.

> **Note**
>
> 50% is a somewhat arbitrary number, but in practice it seems to work as a reasonable goal for many people.

There are moments in an SRE’s life when all of their time is devoted to firefighting, but that can’t be a steady state. If a team’s reactive "ops" work (much of it toil) takes up more than 50% of their time for an extended period, that’s a recipe for burnout and poor reliability. In this situation, the virtuous cycles we discussed before cannot operate or be constructed. SRE similarly pays attention to poorly balanced on-call load because that too has the potential for a very strong negative impact on the team.

Now that we’ve had a chance to see some of the core practices and principles of SRE, let’s talk a bit about how to get started.

Check your knowledge

1. Which of these things is *not* a characteristic of toil (in the SRE context)?

○ The work is repetitive

◉ The work is tracked in a ticket system ✓

How the work is tracked is not a characteristic of toil. While toil is often tracked this way, ticket systems could be used to track other, more productive kinds of work.

○ The work provides no lasting value

○ The work is largely manual in nature and not automated

2. What's an SRE's relationship to toil?

○ SREs work to eliminate toil as their first priority.

○ SREs attempt to manage toil the best they can.

● SREs work to eliminate toil wherever and whenever it is appropriate ✓

3. What is a suggested breakdown of work for an SRE?

○ 80% reactive ops work/20% project work

○ 10% reactive ops work/90% project work

● 50% reactive ops work/50% project work ✓

This is the usual balance SREs strive for (50% is seen as an upper bound on operational load--smaller would be better).

Next unit: Getting started with SRE

Continue ›