



A guide to joining **operational work** in your new *DevOps* team

Recommendations for writing **runbooks**

Kazuki Higashiguchi
Conf42 Site Reliability Engineering 2022
June 09, 2022

@hgsgtk



Kazuki Higashiguchi

ID: [@hgsqtk](#)

Senior backend engineer at Autify

- Web application development using Ruby on Rails
- Infrastructure development using AWS, containers, Terraform, etc.
- Participate in incident handling and be in on-call rotations ...etc

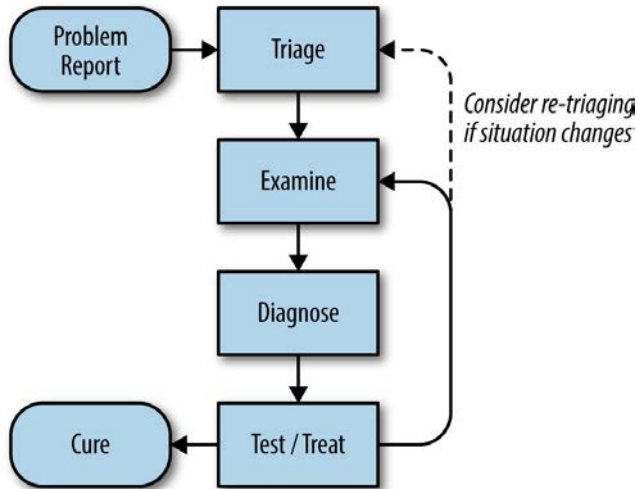
Autify: AI-based software test automation platform

- Start-up company founded in San Francisco
- Autify for Web / Autify for Mobile

A person is seen from behind, standing in a dark doorway. They are looking out into a brightly lit yellow space. The person is wearing a dark jacket and shorts. The scene is dramatically lit, with the person's silhouette against the bright light.

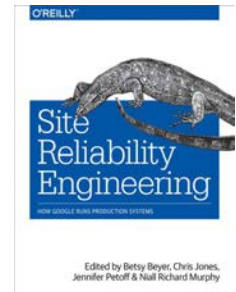
**What makes you difficult to join
service operations?**

A process for troubleshooting

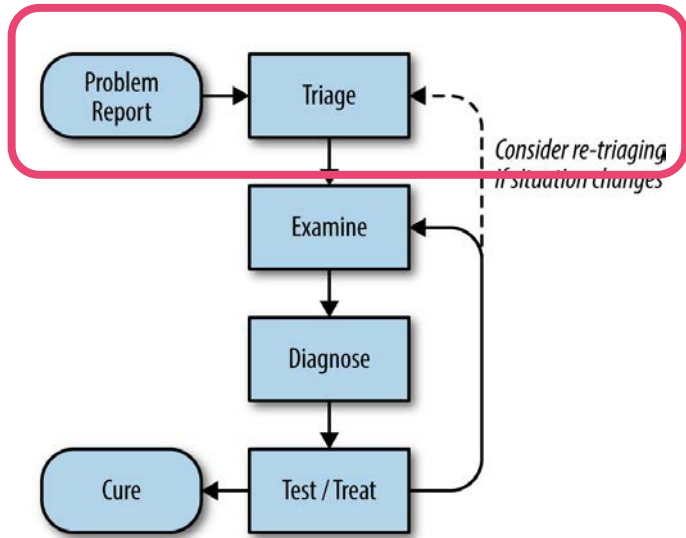


1. Start with a problem report
 - e.g. alerts, customer inquiries, etc
2. Look at system's telemetry and logs
3. Understand current states
4. Identify possible causes
5. Treat the system, change the system
6. Observe the result

["Site Reliability Engineering"](#) / Chapter 12. Effective Troubleshooting



Challenges in the Triage phase



e.g. Alerted that CPU utilization is over 90%

You need to answer following questions to consider whether it requires your action immediately

- Is this alert the first time for your team has?
- Which workflow is the server used for?
- Does users use the service? Or for internal use?
- Is it a known issue for the team?
- ... etc

However, **you don't have knowledges yet** when you just joined

Two contexts of alerts

1. Alerts meant to wake someone up

- Require action to be taken immediately or else the system will go down (or continue to be down).
- e.g. all web servers are unavailable

2. Alerts meant as an FYI

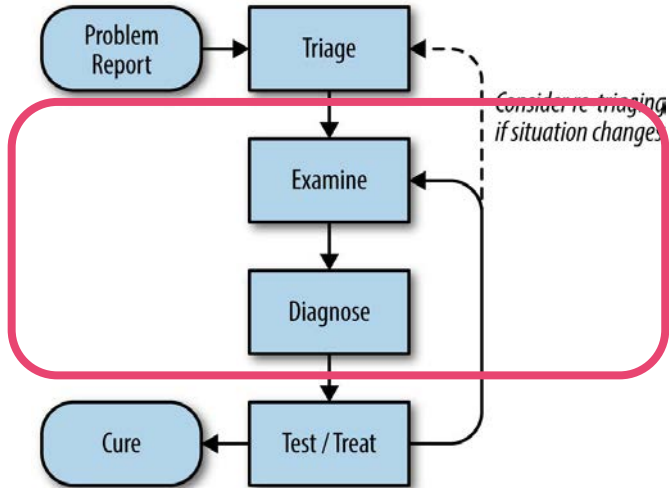
- Require no immediate action, but someone ought to be informed that they occurred.
- e.g. an overnight backup job failed



["Practical Monitoring"](#) / Chapter 3. Alerts, On-Call, and Incident Management

Contextual judgement is one obstacle to join operational work.
It highly depends on knowing its failure patterns.

Challenges in the Examine/Diagnose phase



Need knowledge of how the system is built, how it should operate and its failure modes

The exercise depends upon two factors

1. an understanding of how to troubleshoot generically
2. **solid knowledge of the system**

e.g. queue processing has stopped and the number of waiting events has increased...

- What event the system handles
- Possible causes
- Whether the retries are implemented or not
- ...etc

A long, straight asphalt road stretches into the distance under a clear blue sky. The road is flanked by dry, golden-brown grass and scattered trees. A dashed white line runs down the center of the road, and solid white lines mark the edges. The road curves slightly to the right in the distance.

How to start participating in service operation work

Tip 1. Look at problem reports even if you're not sure

When you get problem reports:

- Click links to problem reports, even if you're not sure about them
- Set the timebox (e.g. 30 minutes)

The more you jump into problem reports, the more knowledge you gain about problem patterns

Tip 2. Leave what you've learned in documents

After going to the detail of alerts:

- Create a blank page in the internal documentation system
 - e.g. create a page in Notion, create an investigation note in Datadog
- Leave what you've learned in the page
 - e.g. system architectures, similar cases in the past, related metrics

Make your learning visible to get trust from the peers, show your knowledge of the system

- How your system works
- How to diagnose atypical system behaviors
- ...etc

Document how experts solve real problems

"In general, the best way to facilitate skill transfer is to watch experts in action. Ideally, you're working alongside them. Watch them solve real problems and document how they mitigated operational surprises: you see how they interpret signals, which tools they use, and you ask them how to make their decisions."

"97 Things Every SRE Should Know"

Chapter 36. Making Work Visible by Lorin Hochstein



Tip 3. Write Runbook

Runbook

- A detailed “how-to” guide for completing a commonly repeated task or procedure
- Step-by-step instructions followed by the operator
- Sometimes known as a **Playbook**

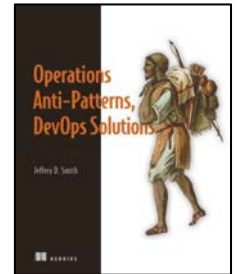
Becomes a **shared wealth of knowledge and expertise** that would otherwise be kept solely in the heads of Subject Matter Experts (SMEs).

Once you put it on, you will be able to take over its operation.

Operation Anti-Pattern: *Only Brent knows*

*“Unless purposeful action is taken, information tends to **coalesce around key individuals**. It makes those individuals incredibly valued but also equally **burdened**.”*

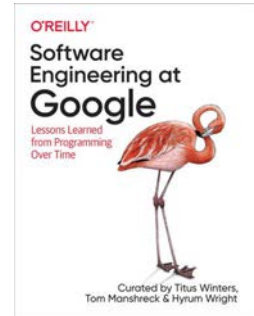
[“Operations Anti-Patterns, DevOps”](#) / Chapter 10 Information hoarding: Only Brent knows



The best time the documentation can be improved

The first time you learn something is the best time to see ways that the existing documentation and training materials can be improved. By the time you've absorbed and understood a new process or system, you might have forgotten what was difficult or what simple steps were missing from the "Getting Started" documentation.

["Software Engineering at Google"](#) / Chapter 3. Knowledge Sharing

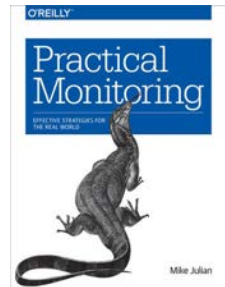


Good runbooks answers these questions

A good runbook is written for a particular service and answers several questions:

- *What is this service, and what does it do?*
- *Who is responsible for it?*
- *What dependencies does it have?*
- *What does the infrastructure for it look like?*
- *What metrics and logs does it emit, and what do they mean?*
- *What alerts are set up for it, and why?*

“[Practical Monitoring](#)” / Chapter 3. Alerts, On-Call, and Incident Management



Write shitty first draft

Learn to embrace what Anne Lamott describes as the “shitty first draft”: an imperfect document is infinitely more useful than a perfect one that does not yet exist.

[“Seeking SRE”](#) / 19. Do Docs Better: Integrating Documentation into the Engineering Workflow



Key takeaways: 3 tips to participate in service operation work

1. Look at problem reports even if you're not sure
2. Leave what you've learned in documents
3. Write Runbook

Resources

Books

- [“Operations Anti-Patterns, DevOps”](#) by Jeffery D. Smith
- [“Practical Monitoring”](#) by Mike Julian
- [“Seeking SRE”](#) by David N. Blank-Edelman
- [“Site Reliability Engineering”](#) by Betsy Beyer, Chris Jones, Niall Richard Murphy, Jennifer Petoff
- [“97 Things Every SRE Should Know”](#) by Emil Stolarsky, Jaime Woo
- [“Software Engineering at Google”](#) by Titus Winters, Tom Manshreck, Hyrum Wright

Blog posts

- [DevOps runbook template](#) by Atlassian
- [What is a Runbook?](#) by PagerDuty
- [Common Attributes of a Good Runbook](#) by Transposit
- [Stack Overflow Developer Survey Results in 2016](#) by Stack Overflow

Thank you for your listening

Autify for Web



Simply interact with your browser

Effortless **no code** testing **for everyone**

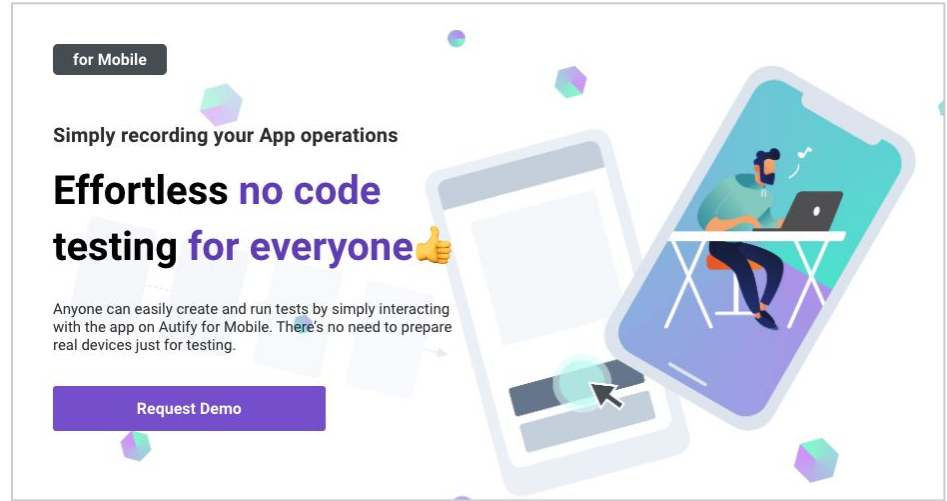
https://autify.com

Sign up

Request Demo

Autify for Web banner showing a sequence of browser interactions: a globe icon with the URL https://autify.com, a 'Sign up' button, a 'Request Demo' button, and two input fields with masked text. Below the sequence, three stylized illustrations of people working on laptops are shown against a teal background.

Autify for Mobile



for Mobile

Simply recording your App operations

Effortless **no code** testing **for everyone** 👍

Anyone can easily create and run tests by simply interacting with the app on Autify for Mobile. There's no need to prepare real devices just for testing.

Request Demo

Autify for Mobile banner featuring a dark grey 'for Mobile' button, the text 'Simply recording your App operations', and the main headline 'Effortless no code testing for everyone' with a thumbs-up emoji. Below this is a paragraph of text and a purple 'Request Demo' button. The background is white with faint illustrations of mobile devices and a person working on a laptop.

We are taking demo requests <https://autify.com/>