

Part IV - Management



1. [Table of Contents](#)
2. [Foreword](#)
3. [Preface](#)
4. [Part I - Introduction](#)
5. [1. Introduction](#)
6. [2. The Production Environment at Google, from the Viewpoint of an SRE](#)
7. [Part II - Principles](#)
8. [3. Embracing Risk](#)
9. [4. Service Level Objectives](#)
10. [5. Eliminating Toil](#)
11. [6. Monitoring Distributed Systems](#)
12. [7. The Evolution of Automation at Google](#)
13. [8. Release Engineering](#)
14. [9. Simplicity](#)
15. [Part III - Practices](#)
16. [10. Practical Alerting](#)
17. [11. Being On-Call](#)
18. [12. Effective Troubleshooting](#)
19. [13. Emergency Response](#)
20. [14. Managing Incidents](#)
21. [15. Postmortem Culture: Learning from Failure](#)
22. [16. Tracking Outages](#)
23. [17. Testing for Reliability](#)
24. [18. Software Engineering in SRE](#)
25. [19. Load Balancing at the Frontend](#)
26. [20. Load Balancing in the Datacenter](#)
27. [21. Handling Overload](#)
28. [22. Addressing Cascading Failures](#)
29. [23. Managing Critical State: Distributed Consensus for Reliability](#)
30. [24. Distributed Periodic Scheduling with Cron](#)
31. [25. Data Processing Pipelines](#)
32. [26. Data Integrity: What You Read Is What You Wrote](#)
33. [27. Reliable Product Launches at Scale](#)
34. [Part IV - Management](#)
35. [28. Accelerating SREs to On-Call and Beyond](#)
36. [29. Dealing with Interrupts](#)
37. [30. Embedding an SRE to Recover from Operational Overload](#)
38. [31. Communication and Collaboration in SRE](#)
39. [32. The Evolving SRE Engagement Model](#)
40. [Part V - Conclusions](#)
41. [33. Lessons Learned from Other Industries](#)
42. [34. Conclusion](#)
43. [Appendix A. Availability Table](#)
44. [Appendix B. A Collection of Best Practices for Production Services](#)
45. [Appendix C. Example Incident State Document](#)
46. [Appendix D. Example Postmortem](#)
47. [Appendix E. Launch Coordination Checklist](#)
48. [Appendix F. Example Production Meeting Minutes](#)
49. [Bibliography](#)

Part IV. Management

Our final selection of topics covers working together in a team, and working as teams. No SRE is an island, and there are some distinctive ways in which we work.

Any organization that aspires to be serious about running an effective SRE arm needs to consider training. Teaching SREs how to think in a complicated and fast-changing environment with a well-thought-out and well-executed training program has the promise of instilling best practices within a new hire's first few weeks or months that otherwise would take months or years to accumulate. We discuss strategies for doing just that in [Accelerating SREs to On-Call and Beyond](#).

As anyone in the operations world knows, responsibility for any significant service comes with a lot of interruptions: production getting in a bad state, people requesting updates to their favorite binary, a long queue of consultation requests...managing interrupts under turbulent conditions is a necessary skill, as we'll discuss in [Dealing with Interrupts](#).

If the turbulent conditions have persisted for long enough, an SRE team needs to start recovering from operational overload. We have just the flight plan for you in [Embedding an SRE to Recover from Operational Overload](#).

We write in [Communication and Collaboration in SRE](#), about the different roles within SRE; cross-team, cross-site, and cross-continent communication; running production meetings; and case studies of how SRE has collaborated well.

Finally, [The Evolving SRE Engagement Model](#), examines a cornerstone of the operation of SRE: the production readiness review (PRR), a crucial step in onboarding a new service. We discuss how to conduct PRRs, and how to move beyond this successful, but also limited, model.

Further Reading from Google SRE

Building reliable systems requires a carefully calibrated mix of skills, ranging from software development to the arguably less-well-known systems analysis and engineering disciplines. We write about the latter disciplines in "The Systems Engineering Side of Site Reliability Engineering" [\[Hix15b\]](#).

Hiring SREs well is critical to having a high-functioning reliability organization, as explored in "Hiring Site Reliability Engineers" [\[Jon15\]](#). Google's hiring practices have been detailed in texts like *Work Rules!* [\[Boc15\]](#),¹³⁴ but hiring SREs has its own set of particularities. Even by Google's overall standards, SRE candidates are difficult to find and even harder to interview effectively.

¹³⁴Written by Laszlo Bock, Google's Senior VP of People Operations.

[previous](#)

[Chapter 27 - Reliable Product Launches at Scale](#)

[next](#)

[Chapter 28 - Accelerating SREs to On-Call and Beyond](#)

Copyright © 2017 Google, Inc. Published by O'Reilly Media, Inc. Licensed under [CC BY-NC-ND 4.0](#)