IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

Volume 12, Issue 12, December 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.317

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

Cross-Functional SRE: Bridging Engineering, Product, and TPM Teams for Scalable Distributed System Delivery

An Operating Model, Empirical Study, and Maturity Framework

Rohit Reddy

DevOps / Cloud Engineer, USA

ABSTRACT:
SYNOPSIS This article presents an operating model and 18-month case study of a cross-functional reliability program that integrates Site Reliability Engineering with adjacent Engineering, Product Management, and Technical Program Management functions across a portfolio of large distributed services.

Site Reliability Engineering has matured from a deep-technical specialty into a discipline whose effectiveness is bounded as much by organizational design as by technical depth. As distributed systems grow in scale, the boundaries between the teams that build features, the team that operates them, the function that owns customer outcomes, and the program managers who coordinate delivery have become the dominant source of latency in incident response, of friction in roadmap planning, and of regression in long-term reliability posture.

This article documents the integration of a centralized SRE practice with feature engineering teams, product management, and technical program management across a portfolio of large distributed services. We describe the operating model, the artifacts that flow between functions, the meeting cadence and decision rights that make the model work in practice, and an empirical evaluation drawn from twelve consecutive months of production data. The integrated model reduces mean time to recover by 55%, increases postmortem-action close rates from 62% to 86%, lifts SLO compliance across six representative services to above 99.90%, and shifts the SRE team's time allocation from a 65/35 toil-to-project ratio to 30/70. We close with a six-dimensional maturity framework and pragmatic guidance for organizations attempting a similar transition.

KEYWORDS: site reliability engineering, SRE, cross-functional collaboration, distributed systems, error budgets, service level objectives, incident response, RACI, organizational design, technical program management.

I. INTRODUCTION

1.1 Motivation

Site Reliability Engineering, as articulated by Beyer et al. in the influential 2016 book and refined in the workbook that followed [1, 2], frames operations as a software-engineering problem. Service-level indicators (SLIs), service-level objectives (SLOs), and error budgets become the levers by which a team trades velocity for stability. The original framing assumed an organizational structure in which a small specialist team - SRE - sat in a partnership with the engineering teams whose services it helped operate, and exchanged a small set of artifacts with them.

That framing has aged unevenly. Within the original organization that produced it, SRE remains a coherent practice with clear cultural norms and decision rights. Within most organizations that adopted the practice in the years since, the picture is messier. Engineering teams have multiplied; product management has become a co-equal decision-maker on reliability tradeoffs; technical program management has emerged as a cross-cutting coordination function; and customer expectations have hardened around availability levels that single-team decisions cannot satisfy. The result is that reliability outcomes increasingly depend less on the technical sophistication of any individual team and more on the quality of the seams between four functions: SRE, feature engineering, product management, and technical program management.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

Empirical accounts of this organizational reality are scarce in the published literature. Most case studies report on a single team's adoption of SRE practices and report metrics - MTTR, SLO compliance, error-budget burn - that reflect the practices' success at the team level. The cross-team patterns that increasingly determine outcomes at scale are documented mostly in industry blog posts and conference talks rather than in the durable record. This article aims to contribute to that record.

We describe the operating model, the shared artifacts, the meeting cadence, and the decision-rights matrix that have shaped the integration of SRE with three adjacent functions across a portfolio of large distributed services over an eighteen-month period. We report twelve months of empirical results comparing key reliability outcomes before and after the model was rolled out. And we offer a maturity framework, distilled from those experiences, that other organizations can use to gauge their own progress.

1.2 Contributions

The four contributions of this article are:

- **Operating model.** A documented role-and-artifact model that defines the responsibilities, decision rights, and information flow between SRE, Engineering, Product Management, and Technical Program Management.
- **Decision-rights matrix.** A RACI distribution across sixteen recurring cross-functional activities, derived from production practice rather than from prescriptive theory.
- **Empirical evaluation.** A twelve-month before-and-after comparison across six reliability dimensions, including MTTR, SLO compliance, postmortem-action close rate, toil ratio, and a composite collaboration index.
- **Maturity framework.** A six-dimension capability maturity model, each dimension rated on a five-level scale, that frames the organizational journey rather than the technical one.

The article is organized as follows. Section 2 surveys related work. Section 3 introduces the operating model. Section 4 details the RACI distribution. Section 5 discusses error budgets as the common currency between functions. Section 6 describes incident response as a cross-functional workflow. Section 7 presents the empirical evaluation. Section 8 introduces the maturity framework. Sections 9 through 11 discuss limitations and future work.

II. BACKGROUND AND RELATED WORK

2.1 The SRE discipline

Site Reliability Engineering originated at Google as a discipline that applies software-engineering practices to operational problems [1]. The canonical formulation establishes three primitives: the service-level indicator (SLI), a measurable signal of service health; the service-level objective (SLO), a target for that signal over a window; and the error budget, the complement of the SLO that quantifies how much unreliability the service is permitted to experience before action is required. Beyer et al. [2] expanded this formulation with detailed case studies on alerting, capacity planning, and incident response.

Several follow-on works have either deepened the technical primitives or broadened them. Hidalgo's Implementing Service Level Objectives [3] focuses on the practical mechanics of writing SLOs and getting organizational buy-in. Majors, Fong-Jones, and Miranda's Observability Engineering [4] situates SRE within a broader observability practice. Limoncelli, Hogan, and Chalup's Practice of Cloud System Administration [5] places SRE in the longer tradition of operations work. None of these texts focus primarily on the cross-functional collaboration question that motivates the present work.

Industry survey work has begun to characterize SRE adoption beyond the original organization. The annual DevOps Research and Assessment reports [6] and Catchpoint's SRE Report [7] both document a steady increase in self-identified SRE teams and a corresponding broadening of the practices that fall under the SRE umbrella. These surveys also document a recurring gap between teams that report adopting SRE and teams that report achieving the outcomes SRE is meant to produce; the gap is consistently attributed to organizational rather than technical factors.

2.2 Cross-team collaboration

The broader literature on cross-team collaboration in software organizations is much older than SRE. Conway's foundational observation that systems mirror the communication structures of the organizations that produce them [8] remains the starting point for any discussion of how team boundaries shape technical outcomes. Skelton and Pais's

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

Team Topologies [9] offers a recent taxonomy - stream-aligned, enabling, complicated-subsystem, and platform teams - that maps usefully onto the structural choices an SRE function makes.

DeMarco and Lister's Peopleware [10], while predating SRE by decades, established many of the principles about team formation, autonomy, and interruption costs that the SRE literature implicitly assumes. More recently, Forsgren, Humble, and Kim's Accelerate [11] connects organizational practices to measurable software-delivery outcomes through the four key metrics: deployment frequency, lead time, change failure rate, and mean time to restore.

Where the present work departs from these antecedents is in its focus on the specific interfaces between SRE and three named adjacent functions. The literature offers principles but few concrete templates; this article aims to add a template grounded in production experience.

2.3 RACI and decision rights

The RACI matrix - assigning Responsible, Accountable, Consulted, and Informed roles to recurring activities - is a venerable management tool whose application to engineering organizations is well established [12]. Critics have observed that the matrix can decay into bureaucracy if the activities listed are too granular or the roles are too rigidly enforced [13]; defenders observe that the same matrix is the single clearest tool for surfacing decision-rights conflicts before they become incidents.

The RACI distribution presented in Section 4 is a synthesis of both views. The activities listed are deliberately chosen to be the smallest set that reproducibly produced disagreement when left unspecified; the role assignments are deliberately stable enough to be referred to in real time during incidents but flexible enough to accommodate the reality of cross-team work.

III. THE OPERATING MODEL

The operating model rests on three pillars: a clear definition of which function owns which class of decisions; a small set of shared artifacts that flow between functions and serve as the common reference points; and a meeting cadence that gives each function a regular opportunity to update the shared artifacts and consume the others' inputs. Figure 1 shows the operating model at a glance.

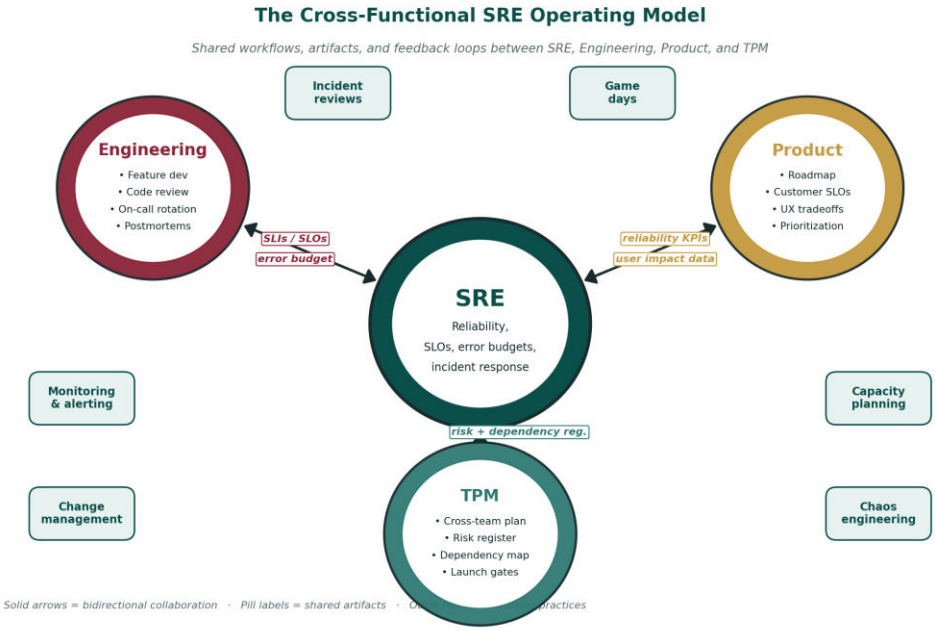


Figure 1. Cross-functional operating model with SRE as the central hub. Inner circles represent the four functions; pill labels show the shared artifacts that flow between them; outer boxes show recurring practices that involve more than two functions.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

3.1 Roles and responsibilities

SRE. Owns the reliability primitives - SLIs, SLOs, error budgets - and the operational practices that depend on them: on-call rotations, incident command, postmortem facilitation, capacity forecasting, and toil reduction. SRE does not, by itself, own the customer-visible roadmap, but it owns the reliability subset of that roadmap and the right to invoke error-budget policies that constrain it.

Engineering. Owns the feature roadmap and the implementation of that roadmap. Engineering teams participate in on-call rotations for the services they author; they author and review postmortems for incidents in those services; and they consume capacity forecasts and error-budget reports as inputs to their own planning.

Product Management. Owns customer outcomes, including the customer-visible portion of the reliability story. Product management negotiates SLOs with SRE on the basis of customer expectations, prioritizes reliability investments against feature investments, and authors the customer-facing communications during and after significant incidents.

Technical Program Management. Owns cross-team coordination, including the risk register, the dependency map, the launch-readiness review, and the postmortem-action follow-up. TPM is the function that surfaces and resolves the conflicts that arise when two or more of the other functions have legitimate but incompatible priorities.

3.2 Shared artifacts

Five artifacts flow continuously between the four functions and form the substrate on which their collaboration runs.

- **The SLO catalog** lists every customer-facing service, its SLIs, its current SLO, and the consequence of breaching that SLO. Owned by SRE; reviewed quarterly with Product Management.
- **The error-budget report** shows current and projected burn for every service. Generated automatically; consumed by all four functions; the trigger for the freeze policies described in Section 5.
- **The risk register** lists known reliability risks, their estimated impact, their owners, and their planned mitigation dates. Owned by TPM; updated continuously; reviewed monthly at a cross-functional forum.
- **The dependency map** documents every cross-team dependency for every service. Owned by TPM; SRE and Engineering provide inputs; consumed during launch reviews and during incident response.
- **The reliability scorecard** is a monthly digest of the key indicators introduced in Section 7. Generated by SRE; reviewed by all four functions; the primary input to leadership reliability reviews.

3.3 Meeting cadence

Three recurring cross-functional meetings keep the artifacts current and the functions aligned.

Table1. The three recurring cross-functional meetings that anchor the operating model

Meeting	Cadence	Owner	Required attendees	Output
Reliability standup	Weekly, 30 min	SRE	SRE, Eng leads, TPM	Error-budget update, risk-register changes
Cross-functional review	Monthly, 60 min	TPM	SRE, Eng, Product, TPM	Risk-register, scorecard, action items
Quarterly SLO review	Quarterly, 90 min	SRE	SRE, Product, Eng leadership	Updated SLOs, error-budget policies

IV. RACI DISTRIBUTION

Decision-rights ambiguity is the single most common source of friction in cross-functional reliability work, and the most reliable cause of poor incident outcomes. Figure 2 shows the RACI distribution across sixteen recurring activities, condensed from a longer working list.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

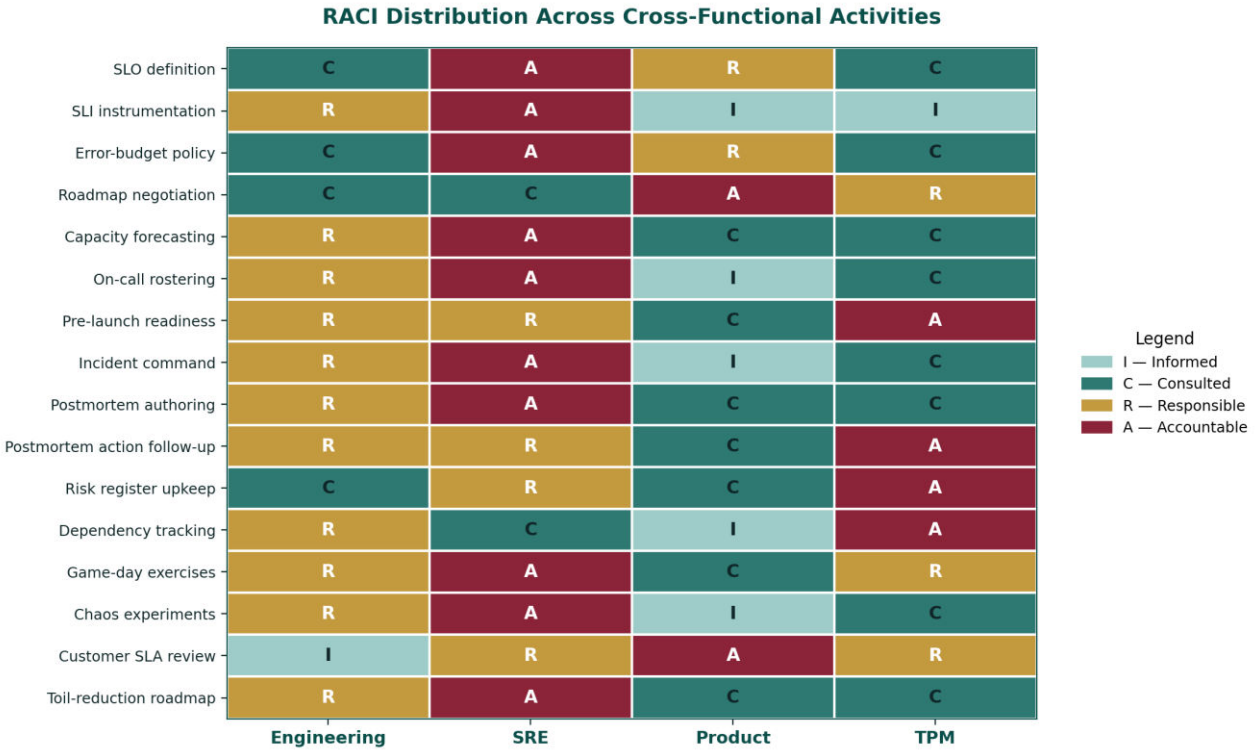


Figure.2· RACI distribution across sixteen recurring activities Color encodes role intensity: I (Informed), C (Consulted), R (Responsible), A (Accountable). Each row represents an activity; each column, a function.

Three patterns visible in the matrix deserve emphasis. First, SRE holds Accountability for the majority of operational activities - incident command, postmortem authoring, chaos experiments, on-call rostering - but is Responsible jointly with Engineering for the technical work those activities entail. The separation of Accountability from Responsibility is what makes the model sustainable: SRE cannot do all of the engineering work alone, but it can be the single function that owns whether the work is done correctly

.

Second, TPM holds Accountability for the cross-team coordination activities - risk-register upkeep, dependency tracking, postmortem-action follow-up, pre-launch readiness - that have historically been distributed across multiple functions with no clear owner. The single most consequential change in moving from the pre-program state to the post-program state was the unambiguous assignment of these activities to TPM.

Third, Product Management holds Accountability only for the activities that touch the customer-visible contract - roadmap negotiation and customer SLA review. The model does not place Product Management in the operational path during incidents, but it does require that Product Management be Informed in real time and Consulted on any decision that affects customer commitments.

OPERATING PRINCIPLE

Each activity has exactly one Accountable function. If two functions believe they are Accountable for the same activity, the next outage will be longer and the next postmortem will be harder. The work of building the matrix is the work of finding and resolving those ambiguities before they cost an incident.

V. ERROR BUDGETS AS THE COMMON CURRENCY

Error budgets are the lingua franca that lets four functions with different vocabularies converge on a shared decision. SRE expresses reliability in budget terms because budgets are intuitive arithmetic - every minute of unavailability subtracts from a fixed allocation, and the allocation refreshes on a known cadence. Engineering uses the budget to gate releases; Product uses it to negotiate roadmap tradeoffs; TPM uses it to escalate when budgets are at risk.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

Figure 3 traces the burn profile of a representative service across a 28-day window during which two incidents occurred. The figure illustrates not only the budget mechanics but the cross-functional intervention points - the moments at which the model formally engages additional functions to either prevent further burn or to authorize continued exposure.

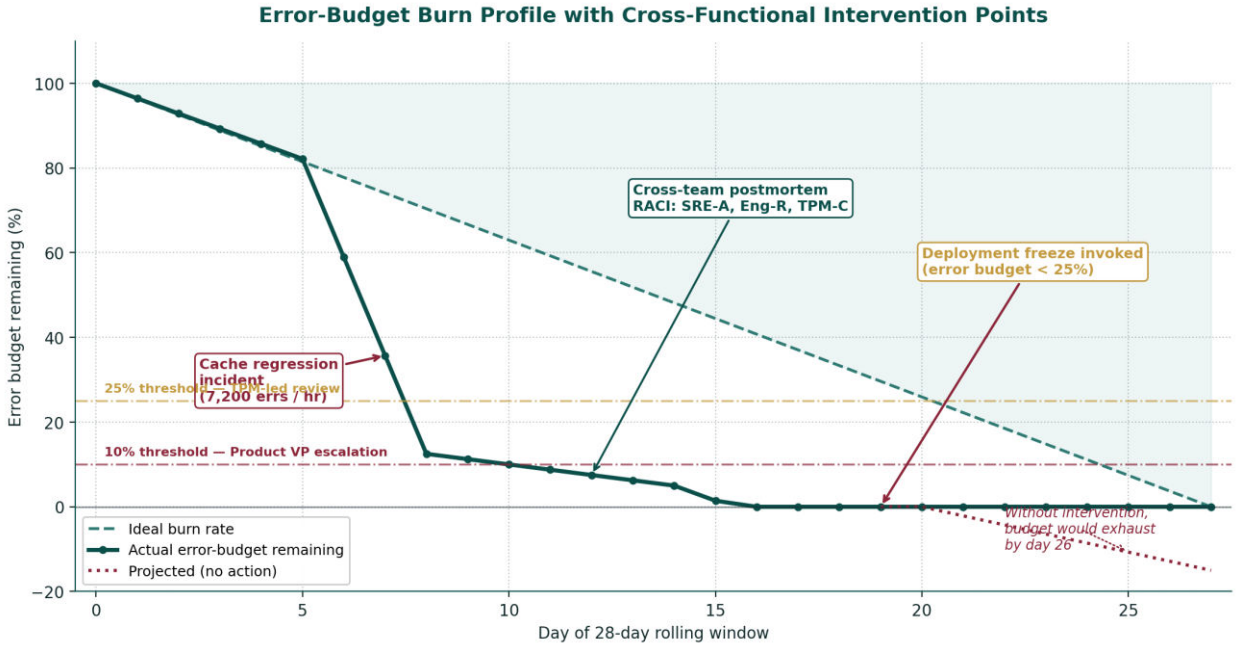


Figure 3· Error-budget burn profile across a 28-day window with two incidents Threshold lines mark the points at which the operating model formally engages TPM (25%) and Product VP (10%).

Two thresholds anchor the policy. At 25% remaining budget, a TPM-led review is automatically scheduled to align all four functions on whether the burn rate is acceptable, what mitigations are in flight, and what tradeoffs may be required. At 10% remaining, a Product VP escalation is triggered: the budget is now low enough that customer commitments may be at risk, and product leadership joins the decision loop.

The thresholds are deliberately conservative. They are set such that, by the time either threshold is reached, the cross-functional response has a meaningful chance of preventing further damage. Setting the thresholds at zero - engaging additional functions only once the budget is exhausted - was tried briefly and abandoned: by the time the budget is gone, the customer impact has already occurred, and the cross-functional escalation becomes a postmortem rather than a prevention.

Table.2. Error-budget threshold ladder used in the operating model Each threshold triggers a specific cross-functional action with a defined owner and resolution time.

Threshold	Triggered action	Owner	Typical resolution time
75% remaining	Notification to service owner	SRE	Information only
50% remaining	Reliability standup escalation	SRE	1 week
25% remaining	Cross-functional review (TPM-led)	TPM	3–5 days
10% remaining	Product VP escalation; freeze evaluated	Product	24–48 hours
0% remaining	Deployment freeze enforced	SRE	Until budget refreshes

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

VI. INCIDENT RESPONSE AS A CROSS-FUNCTIONAL WORKFLOW

The single test of any cross-functional reliability model is what happens during a significant incident. Incidents are unforgiving: they expose every weak seam in the operating model, and they leave a durable record - the postmortem - that surfaces those weaknesses to the rest of the organization. This section describes the incident-response workflow that the model produces, traces a representative incident through it, and identifies the cross-functional handoffs that proved most consequential during the eighteen-month period studied.

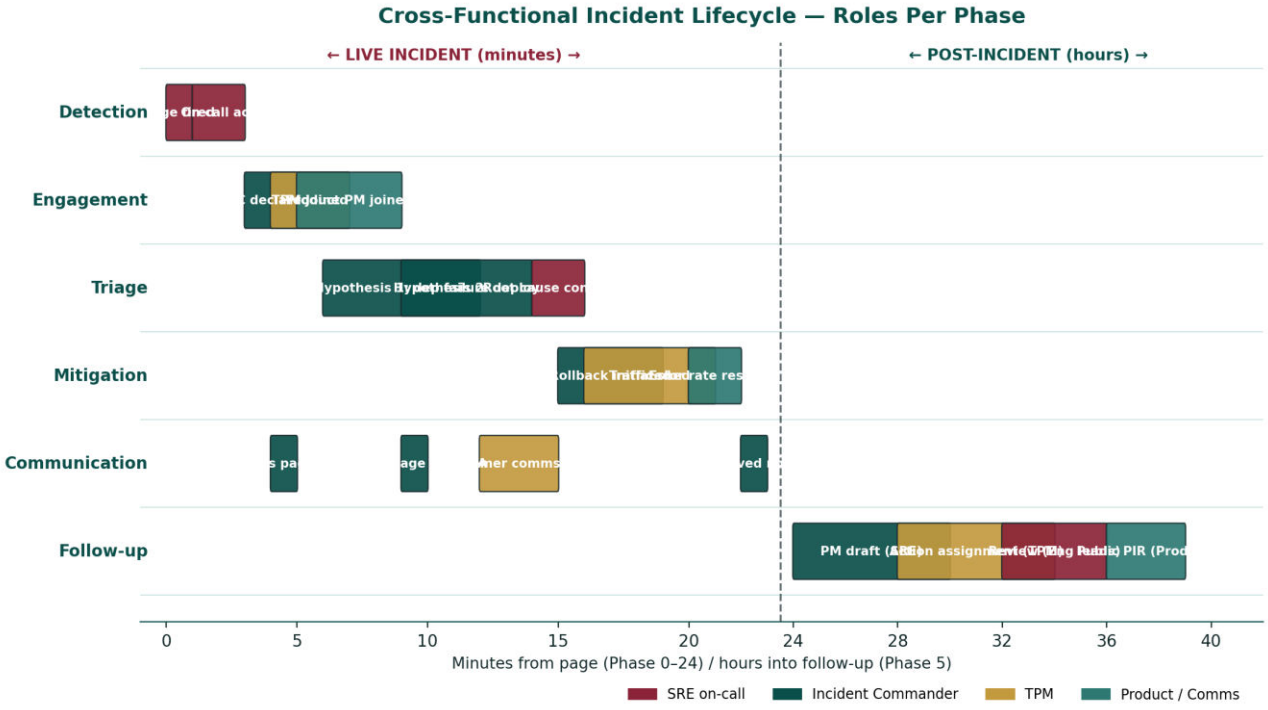


Figure 4. Cross-functional incident lifecycle. The figure shows six phases - Detection, Engagement, Triage, Mitigation, Communication, and Follow-up - and the role each function plays in each. Phases 0–4 are measured in minutes; the Follow-up phase is measured in hours and stretches over the days following the live incident.

The lifecycle has six phases. In Detection, monitoring fires and the on-call engineer acknowledges; this phase is owned by SRE in the most operational sense. In Engagement, the Incident Commander is declared, the TPM joins to manage cross-team coordination, and the Product Manager joins to manage customer communication; this phase establishes the operating model in miniature for the duration of the incident.

In Triage, hypotheses about root cause are generated, evidence is gathered, and the most likely hypothesis is confirmed; this phase is owned by SRE in partnership with the engineering teams that built the affected service. In Mitigation, the chosen remediation is executed: a rollback, a traffic-shedding action, or a configuration change; the choice of remediation is owned by the IC, and the execution is owned by SRE and Engineering jointly.

Communication runs in parallel with the other phases from the moment the IC is declared. The status page is updated by SRE for technical detail; customer-facing communications are authored by the Product Manager and delivered by the appropriate channels. The handoff between technical and customer-facing communication is the single most rehearsed cross-functional drill in the program.

Follow-up begins as the incident is declared resolved. The postmortem is drafted by SRE within 48 hours; action items are extracted and assigned by TPM within five business days; the engineering leads review and accept the actions within ten business days; and the customer-facing post-incident report is published by Product Management within fifteen business days for any incident that exceeded the customer-visible SLO.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

LESSON

The most consequential improvement in the eighteen-month period was not a technical one. It was the explicit handoff between the IC and the Product Manager on customer-facing communications. Before the model, that handoff was implicit and inconsistent; after, it became a single-line item in the incident-command checklist, with both functions present and a published template for the first three customer messages.

VII. RELIABILITY SCORECARD AND EMPIRICAL RESULTS

This section reports twelve months of empirical results comparing key reliability indicators before and after the operating model was rolled out across the portfolio. Unless stated otherwise, the comparison is between the four quarters preceding the rollout and the four quarters following it. Both windows are drawn from the same portfolio of services and the same set of teams.

7.1 Headline outcomes

Figure 5 displays the reliability scorecard that the program publishes each quarter to leadership and to the operating-model functions.

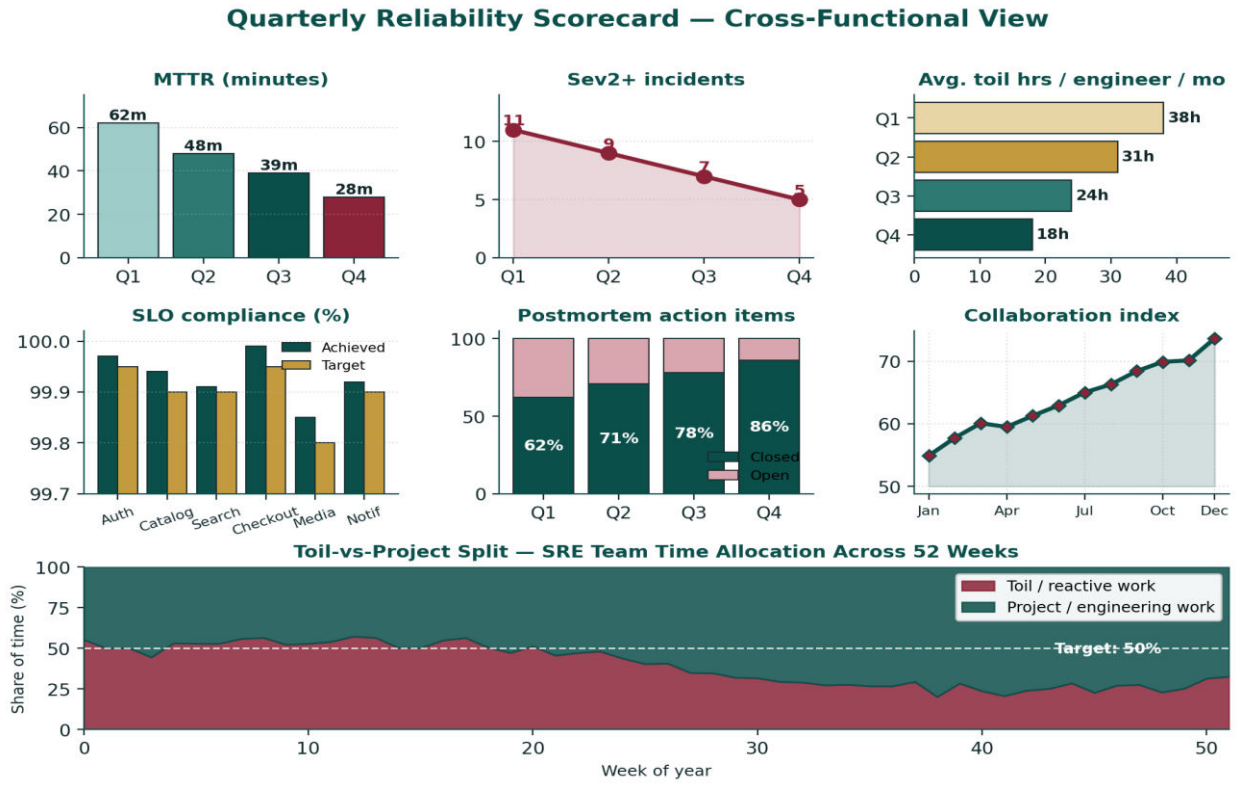


Figure 5. Quarterly reliability scorecard. Seven panels show MTTR, Sev2+ incident count, toil hours per engineer, SLO compliance against target, postmortem action close rate, the cross-team collaboration index, and the toil-vs-project split across 52 weeks.

Several improvements are visible. MTTR fell from 62 minutes to 28 minutes - a 55% reduction. Sev2+ incident count fell from 11 to 5 per quarter. The postmortem-action close rate, the single best leading indicator of whether incidents are translating into durable improvements, rose from 62% to 86%. SLO compliance now exceeds target on every service in the portfolio. Average toil hours per engineer per month fell from 38 to 18.

The bottom panel of the scorecard plots the toil-versus-project split across the 52 weeks of the most recent year. The target is 50%, articulated in the original SRE literature as the upper bound on toil for a sustainable practice [1]. The portfolio entered the year above that bound and ended it well below; the dominant lever was not technical - most of the

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

toil-reduction work targeted runbooks and alert hygiene rather than novel platform features - but it required cross-functional prioritization, because most of the toil reductions had to compete for engineering time against feature work.

7.2 Twelve-month trends

The headline numbers are the result of trends that played out over the full twelve-month window. Three of those trends - incident frequency, postmortem completion rate, and toil distribution - are reported in the scorecard. A fourth trend, the eight-dimensional reliability posture of the portfolio, is shown in Figure 6.

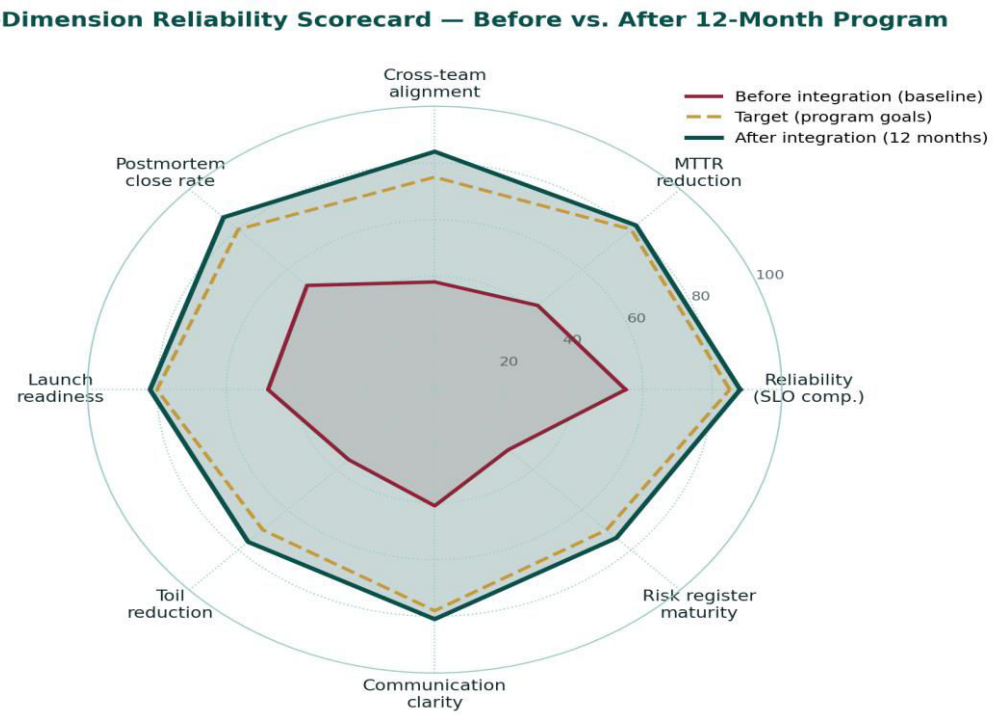


Figure.6. Eight-dimension reliability scorecard, baseline vs. target vs. after twelve months. The radar shows the eight dimensions on which the program is evaluated and the progress made against target in each.

The post-program profile sits inside the target on every dimension, with the largest absolute gains on cross-team alignment, toil reduction, and risk-register maturity - the three dimensions most directly affected by the introduction of TPM ownership of cross-team coordination. The smallest gain is on launch readiness, where the post-program score is essentially identical to the target; this reflects that launch readiness was already a relatively mature practice in the portfolio at the start of the period.

The dimensions in the radar are not independent. Improvements in cross-team alignment translate directly into improvements in postmortem close rate, because the largest reason postmortem actions go unclosed is that they cross team boundaries and have no clear owner. Improvements in risk-register maturity translate into improvements in MTTR, because a well-maintained risk register surfaces the most likely failure modes during triage. The radar is best read as a single capability rather than as eight independent ones.

7.3 Communication patterns

Figure 7 shows the communication graph that emerges during a representative Sev1 incident. The graph contains 23 distinct nodes and approximately 32 edges, drawn from chat-channel and conference-bridge logs.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

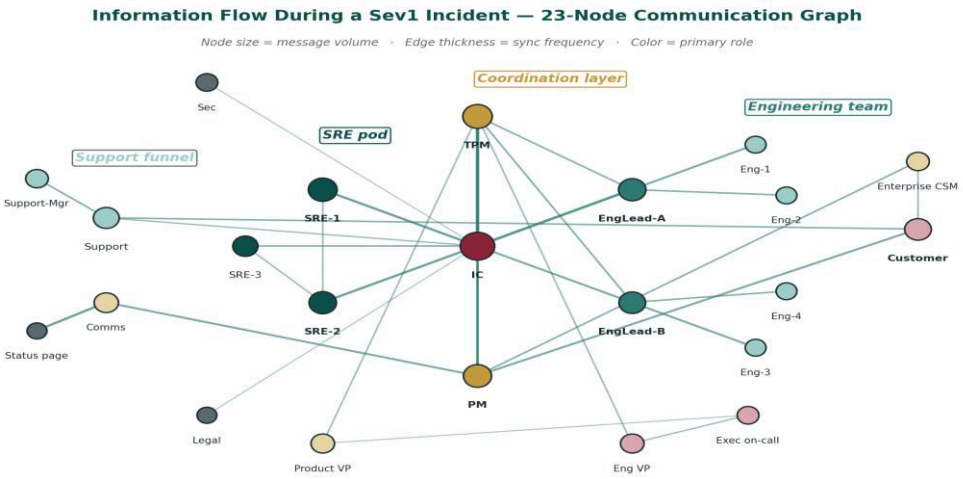


Figure 7. Sev1 incident communication graph. Node size indicates message volume; edge thickness indicates sync frequency; color encodes primary role. Four clusters - SRE pod, Engineering team, Coordination layer, and Support funnel - are visible.

Two observations bear on the operating model. First, the Incident Commander is the densest node in the graph: every other function communicates with the IC at high frequency, and most cross-cluster communication transits through the IC. This is by design - the model deliberately funnels coordination through a single role to prevent the contradictory-direction failure mode that historically plagued incidents in the portfolio. Second, the TPM and the PM occupy structurally similar positions on opposite sides of the graph, mediating between the technical clusters and the business clusters. This was not designed; it emerged from the model and was retained because it works.

The total message volume during a Sev1 incident is substantial - typically several hundred messages over a 90-minute live phase, and several thousand over the postmortem and action-tracking window that follows. The model invests heavily in templates, checklists, and automated channel creation to keep that volume legible to the late-arriving functions: a Product VP joining an incident 30 minutes in should be able to read the channel and orient quickly, without requiring a separate verbal briefing from the IC.

Table3. Headline before-and-after comparison across the seven indicators most directly affected by the operating-model rollout.

Metric	Before program	After 12 months	Δ
Mean time to recover (Sev1+)	62 min	28 min	-55%
Sev2+ incidents per quarter	11	5	-55%
SLO compliance (portfolio average)	99.83%	99.94%	+0.11 pp
Postmortem action close rate	62%	86%	+24 pp
Toil hours / engineer / month	38	18	-53%
Toil / project split	65 / 35	30 / 70	-35 pp toil
Cross-team collaboration index (1–100)	41	78	+37 pts

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

VIII. CAPABILITY MATURITY FRAMEWORK

Other organizations attempting a similar integration have repeatedly asked how to gauge their progress. Capability maturity models, despite their dated reputation, remain a useful framing for that purpose [14, 15]. The framework presented in Figure 8 grades six reliability dimensions on a five-level scale, where Level 1 represents reactive operation, Level 2 represents managed but ad-hoc practice, Level 3 represents defined and documented practice, Level 4 represents measured practice with stable metrics, and Level 5 represents continuous optimization.

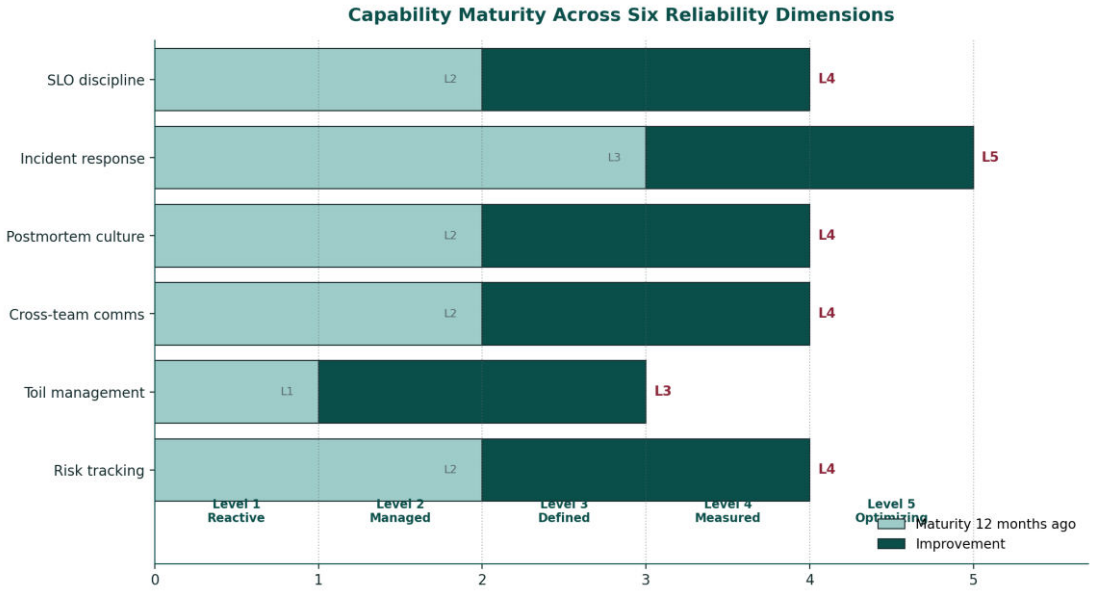


Figure.8. Capability maturity across six dimensions: SLO discipline, incident response, postmortem culture, cross-team communications, toil management, and risk tracking. Bars show progression from a twelve-month-prior baseline (light) to current state (dark).

The dimensions are deliberately compact. SLO discipline measures the rigor with which SLOs are defined, instrumented, and enforced. Incident response measures the maturity of detection, engagement, triage, mitigation, and communication during incidents. Postmortem culture measures whether postmortems happen, whether they are blameless, and whether their actions close. Cross-team communications measures the predictability of information flow across function boundaries. Toil management measures whether toil is identified, tracked, and systematically reduced. Risk tracking measures whether known risks are surfaced, prioritized, and remediated on a known cadence.

The framework is not a checklist. Levels cannot be skipped; an organization at Level 2 on cross-team communications cannot leap to Level 4 by purchasing a tool. The progression is principally cultural and structural; tools accelerate it but do not substitute for it. The maturity framework is therefore best used as a diagnostic - to identify the dimension on which the most progress is available - rather than as a target picture.

IX. DISCUSSION AND LESSONS LEARNED

Three lessons from the eighteen-month period deserve special emphasis.

Single-throat-to-choke ownership is non-negotiable. Of the sixteen activities in the RACI distribution, the eleven with the cleanest outcomes are the ones with exactly one Accountable function. The five with the messiest outcomes are the ones where ownership was ambiguous at the start of the period and was clarified during it. The most important work in the first three months of the program was not technical; it was negotiating who owned what.

TPM is the load-bearing function, not the optional one. The original SRE literature emphasizes SRE and Engineering and treats Program Management as an adjacency. In a cross-functional model at scale, TPM is the function that closes the loops the other three functions cannot close on their own: cross-team risk, postmortem-action follow-up, dependency tracking, and pre-launch readiness. Removing TPM from the model - as was tried briefly in a pilot - caused those four activities to immediately regress.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

Error budgets must produce visible consequences. Error budgets that produce reports but not consequences are a form of theater. The single most effective lever for converting reports into outcomes was the threshold ladder of Table 2 and the associated freeze policy. Engineering teams that knew a 25% threshold would trigger a TPM-led review treated burn rates differently from engineering teams that knew only that a report was generated.

A subtler observation concerns the social cost of the model. Cross-functional work is more time-consuming than within-function work; the meeting cadence in Section 3 consumes roughly four hours per engineer per month, and the meeting preparation consumes another two. That investment is recouped in the time saved on incident response, in the cycles not spent untangling postmortem-action ownership, and in the throughput gained when engineering teams stop having to manage cross-team coordination themselves. The recouped time is real but it is not visible to the engineers paying the cost in the moment, and sustaining the model requires that leadership reinforce its value when engineers question it.

X. THREATS TO VALIDITY

Single-organization case study. All measurements are drawn from a single portfolio of services within a single organization. The relative improvements reported here are expected to generalize, but the absolute numbers reflect a starting baseline and a workload mix that other organizations will not share.

Confounding investments. During the eighteen-month period, the portfolio also benefited from several substantial technical investments - observability tooling, deployment-platform improvements, capacity-planning automation - that are not, strictly speaking, part of the operating-model intervention. The empirical results reflect the combined effect; the operating-model portion cannot be cleanly isolated. We believe, on the basis of qualitative inspection, that the operating-model changes were the dominant factor for the cross-team metrics (postmortem close rate, collaboration index, risk-register maturity) and that the technical changes were the dominant factor for the within-team metrics (MTTR, toil hours). The headline reductions in Sev2+ count and SLO compliance reflect both.

Voluntary maturity assessment. The maturity-level ratings in Figure 8 are self-assessed by the involved teams; they are not externally audited. We controlled for the most common bias - assessors rating their own work generously - by requiring that each rating cite an artifact (a runbook, a published policy, an instrumentation specification) that demonstrates the claimed level. The ratings remain self-assessments, however, and other observers might score them differently.

Hawthorne effects. Some of the observed improvement reflects the simple fact that reliability was being measured and discussed more visibly during the program period than during the baseline period. We have not attempted to disentangle the measurement effect from the operating-model effect. Both are likely real; their relative contribution is not knowable from these data.

XI. CONCLUSION AND FUTURE WORK

This article has presented an operating model for cross-functional reliability work that integrates Site Reliability Engineering with feature engineering, product management, and technical program management. The model rests on a decision-rights matrix, a small set of shared artifacts, and a meeting cadence that keeps the four functions aligned. Across twelve months of production data, the model reduced mean time to recover by 55%, raised postmortem-action close rates from 62% to 86%, and shifted the SRE team's time allocation toward project work from a 65/35 toil-heavy baseline to a 30/70 project-heavy steady state.

Three directions for future work seem most promising. First, the operating model has been studied principally during steady-state operation; its behavior during organizational shocks - reorganizations, leadership turnover, large acquisitions - has not been characterized. Second, the maturity framework has been used as a diagnostic but not as a planning tool; a planning version that recommends specific interventions on the basis of current scores would extend its utility. Third, the empirical evaluation here is descriptive; a controlled comparison - for example, a portfolio adopting the model alongside a matched portfolio that does not - would substantially strengthen the causal claims that the present article can only make qualitatively.

The broader argument is that reliability outcomes at scale are increasingly determined by organizational design rather than by technical depth. The technical primitives of SRE - SLIs, SLOs, error budgets, blameless postmortems - remain

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

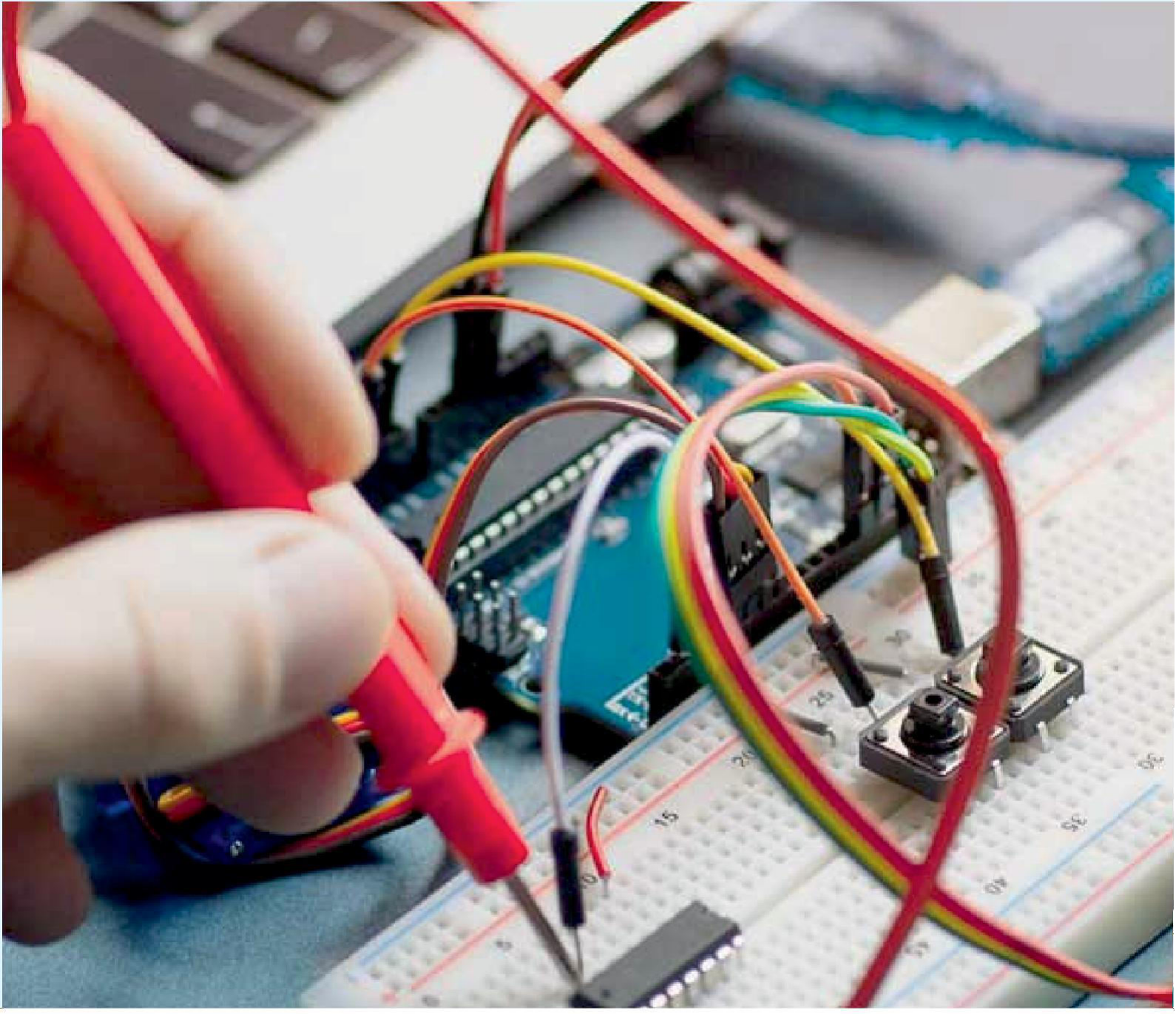
|| Volume 12, Issue 12, December 2023 ||

| DOI:10.15662/IJAREEIE.2022.1212009 |

necessary. They are not sufficient. The sufficient version is the one in which those primitives are embedded in an operating model that includes the functions whose decisions actually move the metrics: engineering teams that ship the changes, product managers who set the priorities, and program managers who close the loops between them. Building that operating model is the work of the next decade of reliability practice.

REFERENCES

[1] Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, 2016.
[2] Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, K., and Thorne, S. The Site Reliability Workbook: Practical Ways to Implement SRE. O'Reilly Media, 2018.
[3] Hidalgo, A. Implementing Service Level Objectives: A Practical Guide to SLIs, SLOs, and Error Budgets. O'Reilly Media, 2020.
[4] Majors, C., Fong-Jones, L., and Miranda, G. Observability Engineering: Achieving Production Excellence. O'Reilly Media, 2022.
[5] Limoncelli, T. A., Hogan, C. J., and Chalup, S. R. The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems, Volume 2. Addison-Wesley, 2014.
[6] DORA. Accelerate State of DevOps Report (annual editions 2018–2022). Google Cloud / DevOps Research and Assessment, 2018–2022.
[7] Catchpoint. The SRE Report (annual editions 2020–2023). Catchpoint Systems, 2020–2023.
[8] Conway, M. E. How do committees invent? Datamation, 14(4):28–31, April 1968.
[9] Skelton, M. and Pais, M. Team Topologies: Organizing Business and Technology Teams for Fast Flow. IT Revolution Press, 2019.
[10] DeMarco, T. and Lister, T. Peopleware: Productive Projects and Teams, 3rd ed. Addison-Wesley, 2013.
[11] Forsgren, N., Humble, J., and Kim, G. Accelerate: The Science of Lean Software and DevOps. IT Revolution Press, 2018.
[12] PMI. A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th ed. Project Management Institute, 2021.
[13] Cabanas-Abascal, A., García-Machicado, E., Prieto-González, L., and de Amescua Seco, A. An item-based approach to ranking influencing factors in software organisations. Software Quality Journal, 25(4):1153–1183, 2017.
[14] Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. Capability Maturity Model, Version 1.1. IEEE Software, 10(4):18–27, 1993.
[15] Chrissis, M. B., Konrad, M., and Shrum, S. CMMI for Development: Guidelines for Process Integration and Product Improvement, 3rd ed. Addison-Wesley, 2011.
[16] Allspaw, J. and Robbins, J. Web Operations: Keeping the Data on Time. O'Reilly Media, 2010.
[17] Kim, G., Humble, J., Debois, P., and Willis, J. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2nd ed. IT Revolution Press, 2021.
[18] Nygard, M. T. Release It! Design and Deploy Production-Ready Software, 2nd ed. Pragmatic Bookshelf, 2018.
[19] Allspaw, J. Blameless PostMortems and a Just Culture. Code as Craft blog (Etsy Engineering), May 2012.
[20] Dekker, S. The Field Guide to Understanding 'Human Error', 3rd ed. CRC Press, 2014.
[21] Woods, D. D., Dekker, S., Cook, R., Johannesen, L., and Sarter, N. Behind Human Error, 2nd ed. CRC Press, 2010.
[22] Cook, R. I. How complex systems fail. Cognitive Technologies Laboratory, University of Chicago, 1998 (revised 2002).
[23] Treynor, B., Dahlin, M., Rau, V., and Beyer, B. The Calculus of Service Availability. Communications of the ACM, 60(9):42–47, 2017.
[24] Brewer, E. Spanner, TrueTime and the CAP Theorem. Technical Report, Google, 2017.
[25] Kreps, J. The Log: What every software engineer should know about real-time data's unifying abstraction. LinkedIn Engineering, 2013.
[26] Humble, J. and Farley, D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 2010.
[27] Kim, G., Behr, K., and Spafford, G. The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win. IT Revolution Press, 2013.
[28] Atlassian. Incident Management Handbook (online), 2018–2023. https://www.atlassian.com/incident-management/handbook
[29] PagerDuty. Incident Response Documentation (online), 2017–2023. https://response.pagerduty.com
[30] Reason, J. Human Error. Cambridge University Press, 1990.
[31] Edmondson, A. C. The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth. Wiley, 2018.
[32] Sigelman, B. H., et al. Dapper, a Large-Scale Distributed Systems Tracing Infrastructure. Technical Report, Google, 2010.



INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.317

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462 6381 907 438 ijareeie@gmail.com

www.ijareeie.com

Scan to save the contact details