KPMG

Unlocking Hidden Value

Modern IT Transformations powered by SRE (Site Reliability Engineering)

Introduction: Why SRE in IT Operations?

As organizations adopt distributed system architecture and cloud in both hybrid as well as pure public cloud set-up to scale their software systems (including products and applications), they are gaining competitive edge in their business and markets by virtue of gaining higher productivity, achieving faster time to market and greater innovation using agile principles. However, along with greater flexibility to use best-fit platforms in modular systems design and build greater products or services to cater to their consumers, organizations also face increasing level of complexity to ensure reliability of such complex systems at scale.

While modern day DevOps practices have been a boon in enabling organizations in breaking silos and increasing development velocity, faster deployment and reducing go-to-market cycles, thus increasing the competitiveness of IT for numerous clients, they are also facing increasing level of complexity to manage the operations and ensuring reliability of these systems at scale. Often, organizations in their quest to achieve higher velocity and agility to release better products and services, fail to maintain system availability and falter on their service level commitments thus impacting their business.

These challenges are rightly solved by Site Reliability Engineering (SRE). SRE practice visualizes IT infrastructure as a code, and a software engineering approach typically used by developers is applied to run IT operations. By removing some of the complex burdens in balancing scalability and maintaining uptime in distributed systems, SRE model allows developers to focus on feature development instead of the nuances of achieving and maintaining service reliability.

What does Site Reliability Engineer (SRE) do?

While traditional ITSM approach to operations uses procedures and processes to solve operational problems, SRE uses an engineering approach to solve the same. It involves a fundamental cultural change in mindset of approaching an operations function. The broad focus areas of SRE are as follows:

- Administration tasks and automating them as far as possible
- Minimizing toil through continual service improvement measures
- Address operational bottlenecks through capacity planning
- Improving system availability and line of sight through Observability
- Reduce failure rates and down-time, and making the environment highly stable
- Embrace risk in the environment within tolerable thresholds to improve environment while maintaining system performance
- Transform delivery by infusing engineering service metrics such as SLO (Service Level Objectives) and Error budgets, and tracking them in operations for improvement in service delivery
- Cultural transformation in ways of working where focus is to identify gaps in the system responsible for bottlenecks and vulnerabilities rather than hold people responsible for problems/issues
- Closely collaborate with developers to ensure system design is highly reliable, scalable, and resilient, while also enhancing developer experience

How is SRE model different from traditional ops model?

In SRE model, there are lot of interaction points between SRE and Product development teams which foster efficiency and reliability of system. Here are some scenarios where SRE solves key challenges for organizations which is not possible in a traditional operations function:

Scenario #1: Context-aware service delivery

Traditional model: Ops team has no option but to maintain the system forcefully because the service has already been contracted.

SRE model: SRE team also gets involved in the 'Acceptance' phase helping in defining the service levels which is closer to reality.

Scenario #2: Manage cloud-native and modern services

Traditional model: Ops team cannot create and maintain Infrastructure-as-Code (IaC) templates and manage cloud native solutions that are immutable, require version management and source code control.

SRE model: SRE team is competent to manage modern cloud services using Infrastructure-as-Code (IaC) and in addition, supports and enhances other cloud-native solutions as part of their charter, once they are set-up in the build phase.

Scenario #3: Developer experience enhancement

Traditional model: Developers not only have to build the product feature, but also must ensure that it is rightly integrable and deployable on the platform for the value stream deliverable to function.

SRE model: Developers do not have to bother about the platform or its technical enablement function and only need to focus on the product feature or value stream deliverable that impacts the business. In this model the developer experience is enhanced, as the SRE takes the responsibility of managing the product platform and its corresponding integrability on the cloud, thus freeing up developers' bandwidth to solely focus on core value stream development and enrichment.

Scenario #4: Automation and self-service focus

Traditional model: Ops team juggle between administrative tasks and issue fixing all the time thus de-prioritizing system reliability.

SRE model: One of the key objectives of SRE team is to 'automate' maximum work and use 'self-service' as much as possible, as improving system reliability is one of the key performance metrics as part of their charter.

Scenario #5: Observability – beyond pure monitoring

Traditional model: Ops team is given delivery mandate with service levels to maintain. If system has issues, then every issue fix follows a reactive approach thus eating up the team's bandwidth in just finding tactical fixes with no room for improvement.

SRE model: SRE team proactively monitors the environment and remediate issues rather than providing reactive support. In the event of a critical issue, SRE team will use Observability platform to trace the anomaly to its root-cause and provide rapid response.

Scenario #6: Synergy between developers and SRE to solve design issues

Traditional model: The system architecture may not be suitable for the level of scale that the Ops team wants to run the system.

SRE model: SRE lead invites the Product development lead(s) to get involved in Capacity planning phase to experience the production environment and understand the performance bottlenecks in the product value stream that can affect the capacity requirements. This understanding will help the Product development team in addressing the performance requirements in the subsequent sprints.

Scenario #7: Cultural transformation

Traditional model: When a critical incident arises, post the temporary fix and problem ticket creation, there is lot of anxiety and blame-game among the team members to point towards an individual or team who is responsible for the critical incident.

SRE model: Blameless postmortems are conducted and driven by the SRE, in which focus is on identifying the root cause of the problem and the systemic lapses that resulted in the issue, rather than indicting any team member. This ensures the right focus on making the system more reliable and resilient, without degrading the team morale.

Scenario #8: Value stream visibility and insights

Traditional model: Ops teams that run the systems have little insights of how it is designed or delivered, or little control over what they need to manage.

SRE model: SRE team get involved in design and development phase to provide their expertise. Using their 'run the system' knowledge in production phase, they provide inputs that can help improve the architecture of the system during design and development phases. They also gain visibility on the system that needs to be managed, which leads to higher system reliability.

Service Metrics used in operationalizing SRE

As we now understand how the SRE model shifts away from the traditional model of ITSM in managing the IT environment, let us now look at some of the key service metrics that enable service improvement and operationalizing the SRE model into practice in any engagement.

Service Level Agreements (SLA):

These are contractual levels of service which are to be delivered to the client, below which the provider is liable to pay penalty to the client.

Service Level Indicators (SLI):

These are quantitative measures of certain aspects of service that is currently provided to client, and those that have impact on client experience and service improvement. These are the actual numbers that are indicative of the system health.

Service Level Objectives (SLO):

These are target levels of service that are measured by a combination of SLIs. These are internal quantifiable targets for SRE teams to keep the service reliable while ensuring client experience and service improvements.

Error Budget (EB):

This is the "permissible unreliability" of the system within the contractual service levels while maintaining the system performance. Hence it provides SRE teams with sufficient space to innovate and improve existing systems without affecting user experience.

Observability Strategy – Technology foundation platform of SRE

While we have discussed how SRE results in a cultural transformation in managing production operations of an IT landscape from people and process standpoint, there is also an underlying technology foundation which needs to be designed/set-up to enable SRE function to drive this operational transformation. This technology foundation is known as the 'Observability' platform, and this is architected in such a way that it enables an SRE to have a **360-degree view** of the entire complex IT landscape, provides a line-of-sight into the breadth of the environment, can be used to **trace** any **anomalies** both reactively and proactively and help SRE achieve the objective of system 'reliability' by acting as the necessary **technology enabler**.

An Observability platform does not necessarily warrant a greenfield build or a rip-and-replace strategy that require clients to dispose all the existing investment made in a traditional operations environment. The observability platform can also be optimally designed by leveraging investment made by the client in existing tool stack, thus protecting the current client capex and augmenting the tool stack with best-fit minimal COTS or open-source tools and, integrating these additional tools effectively with the current tooling architecture. The SRE journey is incomplete without a definitive Observability roadmap.

The key features that an architecture design must entail to build an Observability platform are as follows:

- Collect metrics data as inputs from all data sources of an IT landscape to provide a unified view
- Map all system and application dependencies that help in correlated monitoring
- Provide holistic monitoring from technical, user-experience and business process standpoint
- Perform proactive monitoring of alerts and anomalies thus minimizing potential incidents
- Provide traceability feature to track-back the anomalies and help identify root-cause of incidents which have already occurred
- Provide visualization based on real-time (or near real-time) data-driven insights for easy executive consumption and better decision making
- Enable provision in system design for scalability to provide higher order use-cases viz. self-healing, predictive analytics, AIOps based automation etc., as the engagement scales up in maturity cycle

Continuous Improvement through SRE framework

One of the unique aspects of our SRE adoption framework is the focus on continuous improvement that is embedded into the operations model. Based on the platform environment, SRE creates a list of **reliability principles** which form the basis for defining the **Service Level Objectives (SLO)**. Based on the contracted Service Level Agreements (SLAs), SLO levels are pre-set (generally higher than SLAs) internally to track the reliability of the engagement. These SLOs and in turn the **Error Budgets**, at any given point of time act as **feedback inputs** and **denote the risk embracing capacity** of the business, which are then analyzed by the SRE to focus their attention to the areas which need improvement, and in tandem with product developers **calibrate** the overall team's bandwidth between **product feature/development velocity** and **reliability/stability** of the product.

Given below is the depiction of the framework which embeds continuous improvement into our SRE operations model.

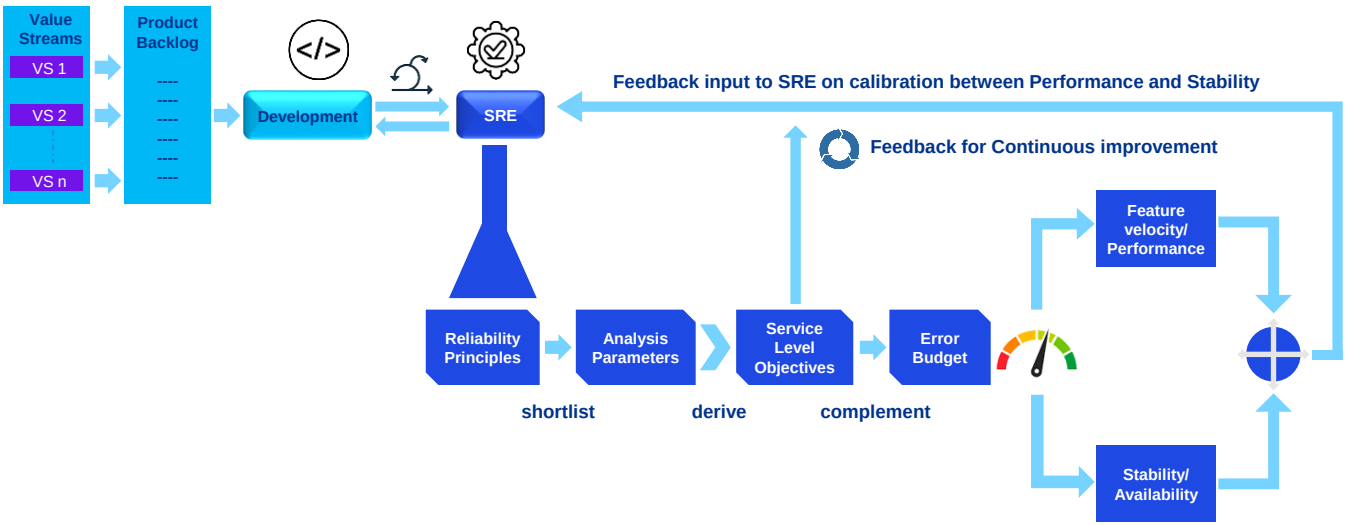

Fig 1: KPMG SRE framework for continuous improvement

As we now understand, SRE model not only ensures optimal reliability of IT landscape on the cloud, but also **enhances developer experience** in product development by eliminating the bandwidth required for platform configuration and integrability.

Despite overwhelming benefits that SREs provide, that clearly outweigh the costs associated with executing SRE model in managing the environment, we see that surprisingly fewer organizations have adopted SRE model in their engagements.

Challenges in executing an effective SRE model

Despite understanding the immense value realization that SRE model infuses in IT operations in managing products and/or value stream delivery, we see that organizations still fail to implement SRE primarily due to one or more challenges and roadblocks faced in adopting SRE approach effectively and exploiting its full potential. Some of the key challenges are as follows:

- **Misplaced sense of understanding** of SRE philosophy and how to leverage it to transform the current state
- Implementing SRE involves strong '**change management'** element which needs to be **executed** carefully & requires specialist guidance for **executive buy-in** and **strategizing roadmap**
- **No clarity on 'where to start'** and **'how to begin'** in the journey of SRE adoption
- **Absence** of business units' **involvement** while executing this key change in operating model
- Fundamental **shift in culture**, role changes, and ways of working will **encounter resistance** from stakeholders
- **Restricting SREs' visibility and involvement to only operations scope** hinders the full potential of SRE model
- **Absence of Observability roadmap** results in weak tech foundation platform to enable SRE
- **Confusion** about defining relevant Error Budget (SLOs and SLAs) to measure improvement
- Complex tooling landscape without a multi-cloud tooling roadmap aligned with Ops Model will lead to **ineffective service delivery**

Addressing the challenges to successfully execute SRE model

Irrespective of the current state of operations set-up in organizations, i.e., traditional hybrid cloud or public cloud set-up, KPMG SRE enablement framework, which is built based on our wide experience of consulting numerous clients, offers a comprehensive approach to adopt a robust SRE centric service delivery model and provides a detailed transitionary as well as stabilized run operations enablement framework to enable organizations make a seamless switch over to a transformed SRE mode of operations.

Our framework encompasses multi-pronged levers that ensure seamless and comprehensive transformation of IT in a hybrid/cloud environment from current state to a mature SRE engagement model as shown in the diagram. Each of the levers are carefully analyzed based on assessment of client culture, landscape and ways of working.

KPMG SRE Enablement framework

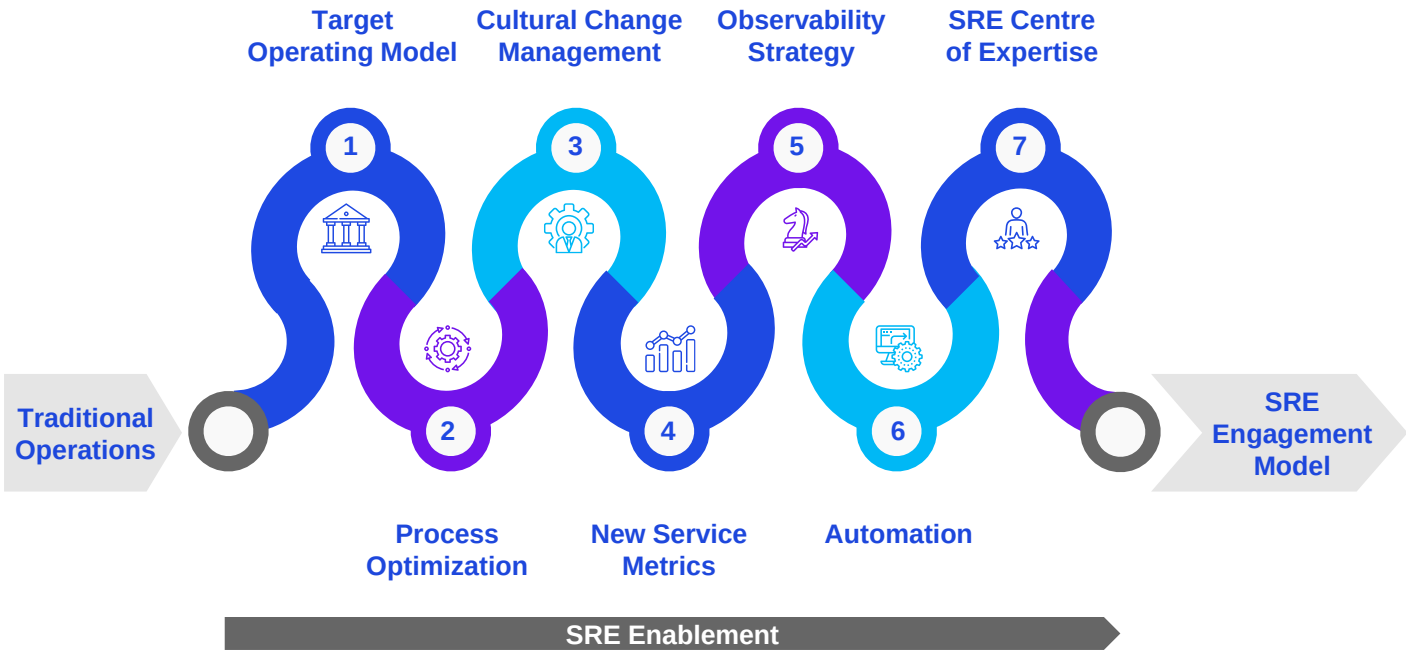

Fig 2: KPMG SRE Enablement framework

The objectives of our SRE enablement framework are achieved through the following:

- Design a target operating model with SRE alignment to client's value stream model
- Create a transitionary model to enable seamless switch-over from current ops model on day-1
- Continually evolve the transitionary model until operating model attains maturity
- Evaluate client's existing processes and redesign them in-line with SRE mode of operations
- Redesign teams and create new roles in-line with client's value stream delivery
- Map traditional roles to new roles for potential up-skilling/cross-skilling
- Design roles and responsibilities with clear boundaries to eliminate grey areas and discrepancies
- Create a robust Governance framework and Cultural Change management tenets to drive the adoption of SRE effectively in the organization
- Define new service metrics to measure the performance of service delivery and enable SREs to balance system performance & feature velocity with system reliability & resilience
- Design a custom fit Observability Architecture to unlock the full potential of SRE

- Create a self-sufficient value stream model where teams can self-serve and manage their own requirements guided by a consistent framework
- Build strong continual improvement capabilities in a self-sustaining mode
- Establish a Centre of Expertise (CoE) function that will guide the team/stakeholders, assist in organization change management, overcome challenges in SRE adoption, and ensure evolution of SRE model until maturity
- Create a comprehensive roadmap to move from traditional ops to a mature SRE engagement model

This framework when executed to operationalize SRE model of engagement results in delivering key outcomes, viz. delighted client experience, robust system reliability & resilience, enhanced developer experience and superior organization value realization.

Fig 3: Outcomes delivered through KPMG SRE Enablement framework

These outcomes are also complemented by the enablement provided by SRE to the product developers in building contextual system design/architecture that are not only relevant in terms of product features but also in terms of capacity, performance, and consumption.

Once an engagement transforms to adopt SRE model in its environment, it can be further matured to a state where SREs, by using their expertise in running systems in production phase, work closely with developers during the design/build and develop phases to improve the system architecture, make the system more scalable for desired business requirement, define better service levels, and recommend measures to reduce debt. This helps in building robust systems, attaining greater efficiencies and in turn achieving client excellence.

For more details on how to effectively adopt Site Reliability Engineering in your organization and transform your IT in a comprehensive and seamless manner, tailored to your current set-up, maturity level, culture, and ways of working, please contact us.

KPMG Contacts

Adrian Bradley

Partner – Cloud Transformation

Adrian.Bradley@KPMG.co.uk

As Head of Cloud Transformation, KPMG UK, Adrian leads our teams who transform and manage the technology platforms which enable our clients to deliver on the promise of digital. Adrian's roles have included migrating FTSE100 companies to infrastructure leading one of the world's largest Office 365 implementations, and delivering digital transformations across front and back offices. Primarily, Adrian's background is in the Energy and Consumer Goods sectors.

Chris Astley

Partner, Cloud Enablement

Chris.Astley@kpmg.co.uk

Chris has over 16 years experience in building and managing enterprise workloads in multiple scenarios. For the past 10 years he has focused on deploying applications and services in public cloud environments whilst retaining a strong security posture. Chris is responsible for the connected engineering practice for KPMG UK

Rohit Sinha

Associate Director, Cloud & DevOps – CIO Advisory

rohitsinha5@kpmg.com

Rohit leads our KGS India Cloud Engineering team. He specializes in inculcating engineering practices in DevOps & SRE engineering and leads their adoption in clients' organizations. He has extensive experience in rolling out automation engineering and capability building from scratch, and has delivered several major transformations involving technology refresh, and organization change management.

KPMG Contacts

Chhavi Saluja

Associate Director, Cloud Tx – CIO Advisory

chhavisaluja@kpmg.com

Chhavi is an Associate Director in KPMG's Cloud Transformation team. Thought Leader and strategist known for innovative and disruptive approach in driving cloud transformations, digital transformations, operating model alignment and developing scalable practices for cloud. She is best in applying complex technologies to business strategies and continue to be a creative thinker with high energy and enthusiasm.

Suharsh Paldewar

Manager, Cloud Tx – CIO Advisory

suharshpaldewar@kpmg.com

Suharsh is part of KPMG’s Cloud Transformation team. He is passionate about solving clients' problems by using a blend of technology, business appreciation and commercial acumen to drive transformations in enterprise cloud, operating model, finops, SRE adoption, IT service management, and provide compelling solutions in managing IT operations.

Some or all of the services described herein may not be permissible for KPMG audited entities and their affiliates or related entities.

kpmg.com/uk

Document Classification: KPMG Public