WHITE PAPER – **NOVEMBER 2018**

SITE RELIABILITY ENGINEERING (SRE)

SRE with VMware Professional Services

vmware®

Table of Contents

1. Introduction 3

2. SRE—Key Concepts 5

- 2.1 Definitions 5
- 2.2 SRE and DevOps 6
- 2.3 SRE Core Tenets and Responsibilities 8

3. SRE Applied to VMware-Supported Environments 10

- 3.1 SRE for the Software-Defined Data Center 10
- 3.2 SRE for Hybrid Cloud and Multi-Cloud 12
- 3.3 SRE for Cloud-Native and Hybrid Applications 12

4. SRE—Operating Model Considerations 13

- 4.1 People Perspective 13
- 4.2 Process Perspective 17
- 4.3 Evolving to a Site Reliability Engineering Model 21

5. Resources 25

Content Contributors and Acknowledgements

Content Contributors:

- Kevin Lees, Chief Technologist
- David Leith, Practice Manager
- Chad Nale, Staff Architect
- James Wirth, Technical Solutions Architect

Reviewers:

- Kai Holthaus, Business Architect
- Louise Ng, Senior Manager, Advisory Services
- Roman Tarnavski, Principal Architect
- Steve Tegeler, Senior Director
- Paul Wiggett, Senior Technical Operations Architect

1. Introduction

Companies seeking to increase velocity and reliability of solutions within their digital business should shift their software development efforts "further to the right" into infrastructure and operations (I&O) teams by adopting tenets of Site Reliability Engineering (SRE). The SRE ethos was conceived at Google to help them run their products and services smoothly, efficiently and reliably at scale.

SRE is defined as "what happens when you ask a software engineer to design an operations team."[1] SRE practitioners analyze business services to determine their actual required availability (which in actuality is seldom 100%) and then specify the operational strategy, including deployment frequency, to meet the availability requirement. This is often a fine balancing act between maintaining the desired availability and getting new features to users faster.

VMware CEO Pat Gelsinger talks about "the gap" between infrastructure, the teams that manage infrastructure, and the "crazy application folks." The developers are concerned with creating new features and bringing them to market as quickly as possible. The I&O team is concerned with operational requirements: security, compliance, governance, and the reliability of the virtual environments used (VMs, containers) to reduce risk and maintain stability.

This gap slows the business in meeting its desired outcomes and generating shareholder value. DevOps has long been hailed as the solution to these problems, and SRE, as a superset of DevOps principles, promises to provide a prescriptive and holistic approach to doing so.

[1] Benjamin Treynor Sloss. "Introduction." Site Reliability Engineering: How Google Runs Production Systems. Edited by Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Murphy. O'Reilly Media, 2016.

The proliferation of software-defined environments has expanded the breadth of activities to which SRE concepts can be applied because they encourage and accommodate far higher levels of programmability and automation. From a VMware field perspective, SRE concepts should be applied equally to addressing IT service reliability. Services provided by IT can include application-based business services, the "traditional" SRE area of focus, as well as include:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Containers as a Service (CaaS)
- Other IT services such as desktop services or data analytics services

As with applications that make up business services, SRE practitioners analyze IT services to determine their true reliability requirements, and then develop a resulting operations strategy, including a new capability deployment frequency to meet those requirements. SRE practitioners also proactively define service frameworks addressing operational considerations such as instrumentation and logging as well as for building reliability into the application itself, to help developers deliver applications that support operational reliability.

The primary premise of this white paper is to discuss the application of SRE concepts to maintaining IT service reliability in VMware® software-defined environments.

Note: The concepts in this white paper are VMware adaptations of the original Google Site Reliability Engineering concepts and definitions.

2. SRE—Key Concepts

2.1 Definitions

Following are a set of foundational definitions for commonly used terms associated with SRE:

TERM	DEFINITION
IT service	An IT service is composed of one or more software and software-defined infrastructure components and configurations that, combined, provide business value. An example of an IT service is a customer relationship management (CRM) service based on off-the-shelf CRM software or a cloud-based software-as-a-service (SaaS) offering, an intranet site, or an as-a-service platform that provides infrastructure-based services. SRE then relates to deploying, running, and continually improving these IT services with a reliability mindset.
Software-defined environment (SDE)	• An SDE optimizes the entire computing infrastructure—compute, storage, and network resources—so it can adapt to the type of work required. Currently, resources are assigned manually to workloads; this happens automatically in an SDE." • By dynamically assigning workloads to IT resources based on a variety of factors, including characteristics of specific applications, the best available resources, and service-level policies, an SDE can deliver continuous, dynamic optimization and reconfiguration to address infrastructure issues.
Virtual environment	Virtual machines or containers in which IT service components are deployed, as well as the IT service–specific, software-defined infrastructure configurations deployed with them in the SDE.
Service level indicator (SLI)	SLIs are metrics over time, such as request latency, throughput of requests per second, or failures per request. These are usually aggregated over time and converted to a rate, average, or percentile that can be subject to a threshold.
Service level objective (SLO)	SLOs are targets for the cumulative success of SLIs over a window of time agreed-upon by stakeholders. Unlike traditional environments where SLOs may be measured over a 30-day period, these should be measured at least daily in an SDE to account for the increased agility of these environments.
Service level agreement (SLA)	• An SLA is a commitment by a service provider to provide value to the consumer based on an agreed contract for availability—and what the costs are for failing to deliver the agreed-upon level of service. SLAs are typically defined and negotiated by whoever owns the business relationship with a customer and promises a lower availability than the SLO. NOTE: An SRE practitioner's goal for site uptime will be just slightly better than the minimum level of availability, defined in the SLA, that customers will accept.
Error budget	Error budgets and availability measures are determined by SLOs and SLIs. For example, if the service must be working and available 99.99% of the time, it could be unavailable 0.01% of the time. This 0.01% allowance for downtime is the error budget for the service.
Toil	Toil is a kind of work tied to running a production service. It tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and linearly scalable as the service grows. Not every task deemed toil has all these attributes, but the more closely work matches one or more descriptions, the more likely it is to be toil.
Mean time to recovery (MTTR)	MTTR is the amount of time it takes to bring a service back to a healthy state.
Canary release	A technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.

TERM	DEFINITION
Blue/green deployments	A technique that reduces downtime and risk by running two identical production environments called **blue** and **green**. At any time, only one of the environments is live, with the live environment serving all production traffic.[2]
Immutable infrastructure	Immutable infrastructure is composed of immutable components replaced for every deployment, rather than being updated in-place. The components are started from a common image that can be tested and validated. The common image can be built through automation. Immutability is independent of any tool or workflow for building the images.
Infrastructure as code (IaC)	A method of writing and deploying machine-readable definition files that generate service components, thereby supporting the delivery of business systems and IT-enabled processes.

2.2 SRE and DevOps

SRE practitioners embrace the DevOps model, but not all who follow the DevOps model are necessarily SRE practitioners. SRE practitioners do a lot more than produce software, but when they do code, they follow DevOps principles.

It's useful to understand the differences and similarities between SRE and DevOps to lay the groundwork for future conversation. SRE's evolution at Google took place in the early 2000s, independently of the DevOps movement. However, SRE shares the spirit of DevOps, while being much more prescriptive in measuring and achieving reliability through the engineering of operations activities. In short, SRE defines how to succeed in DevOps. The table below lists the DevOps Guiding Principles—as defined in the "DevOps and Agile Development: A VMware Field Perspective" white paper—and related SRE activities:[3]

DEVOPS GUIDING PRINCIPLE	RELATED SRE ACTIVITIES
Dev and Ops as one team	• Understand the delivery pipeline and customize it to meet the needs of the IT service developers. • Actively and continuously collaborate with IT service developers from planning through testing and release in addition to ongoing operations to ensure they develop the service with built-in reliability, monitoring, and other ongoing, proactive operations in mind. • Understand the IT service architecture and its reliability implications for the virtual environment in which it runs. • Share ownership with IT service developers by using the same tools and techniques across the stack, such as when developing automated workflows.

[2] Cloud Foundry Foundation, "Using Blue-Green Deployment to Reduce Downtime and Risk." Cloud Foundry Documentation. Last updated December 8, 2017.

[3] Kevin Lees, John Gardner, and Peg Eaton, "DevOps and Agile Development: A VMware Field Perspective." VMware white paper. 2017

DEVOPS GUIDING PRINCIPLE	RELATED SRE ACTIVITIES
Teams embracing a DevOps mindset are accountable and responsible	• Understand security, audit, and compliance requirements for the delivery pipeline as well as the virtual environment in which the IT service will run in production. • Understand the need for workload placement based on stacked criteria such as risk, compliance, availability zone, security, service levels, and costs/budget. • Understand the impact of deploying the IT service and its virtual environment into production from a service reliability perspective. • Have a formula for balancing service reliability with new IT service releases. • Define prescriptive ways for measuring availability, uptime, outages, toil, etc.
Shift to the left	• Automate security, compliance, performance, and operational readiness tests for early inclusion in the delivery pipeline. • Actively work with IT service developers to integrate operational considerations directly in their service. • Provide open access by IT service developers to monitoring and logging throughout the delivery pipeline and as allowed by compliance policies in production. • Work with IT service developers to define operational metrics and provide customized service-specific monitoring throughout the delivery pipeline and in production. • Proactively provide data to IT service developers that continuously improves the service delivery and optimizes customer experience. • Actively monitor the IT service's virtualized environment throughout the delivery pipeline, and provide feedback to service developers. • Move from reactive to proactive, self-healing IT services using automation techniques.
Automate the delivery pipeline	• Participate in delivery pipeline automation definition, design, and implementation. • Build compliance controls and audit capabilities into delivery pipeline automation. • Provide self-service, on-demand, or programmatic access to software-defined infrastructure, blueprints, and policies. • Use the same tools and code for operations activities in production as used in the delivery pipeline. • Encourage "automating this year's job away"[4] and minimizing manual delivery pipeline as well as production operations work to focus on efforts that bring long-term value to the IT service.
Immutable infrastructure	• The only way to modify production is through the delivery pipeline. • Never update, patch, or modify anything in-place in production.
Infrastructure as code (IaC)	• Ensure ability to declare infrastructure as code and put it under version control. • Always define the virtualized environment as code and maintain its definition under version control.

[4] Seth Vargo and Liz Fong-Jones, "SRE vs. DevOps: competing standards or close friends?" Google Cloud Blog. May 8, 2018.

DEVOPS GUIDING PRINCIPLE	RELATED SRE ACTIVITIES
Smaller more frequent releases	• Work with IT change management and compliance auditors to focus governance on the version-controlled definitions (for example, application code, infrastructure as code, blueprints, and policies) of what is to be deployed as well as the automated delivery pipeline itself rather than each application release unit. • Encourage moving quickly by reducing costs of failure.

2.3 SRE Core Tenets and Responsibilities

While the day-to-day activities, priorities, and ways of working vary among SRE practitioners responsible for different services, there is a defined set of core tenets and responsibilities for the services they support, to which SRE practitioners should adhere. This core set of tenets and responsibilities are based on those originally defined by Google but have been modified and supplemented as needed to support services delivered in VMware-supported environments. This section describes these core responsibilities and tenets. The next section describes how they are executed in commonly found VMware-supported environments.

2.3.1 Core Tenets

SRE practitioners satisfy their responsibilities while adhering to a couple of core tenets. These core tenets focus on a set of work practices that help SRE practitioners maintain a level of focus on measuring and automating activities for executing their operational responsibilities with minimal human intervention

2.3.1.1 50/50 Time Utilization

SRE practitioners should strive for spending no more than 50% of their time on operational work supporting their services and the other 50% on automating and optimizing activities to minimize manual operational work. The 50/50 time-utilization targets must be negotiated and accepted by management and the service teams the SRE practitioner is a member of or supports, reflected in the SRE practitioner's annual review criteria, and monitored. This is important to both ensure appropriate behavior as well as support offloading the SRE practitioner's excess operational work to others in the service team.

During their 50% non-operational time, some of the tasks SRE practitioners should be focusing on are as follows:

- Developing automated service deployment workflows, including integrating with other systems involved in the service's deployment workflow, using tools like VMware vRealize® Orchestrator™
- Developing operations-focused automated service testing capabilities including, for example, security, compliance, and operational readiness testing
- Using modern configuration management tools like Puppet, Chef, Ansible, or Salt to automate the service's configuration for deployment throughout the CI/CD pipeline
- Using the integration of VMware vRealize Automation™ and VMware vRealize Business™ or the cloud automation services in VMware Cloud™ services and CloudHealth for cost/performance management and for refining workload placement policies based on SLAs, especially in multi-cloud environments

- Using VMware vRealize Operations Manager™ to optimize service monitoring and to automate remediation wherever possible while also developing troubleshooting and remediation best-practice playbooks where automation isn't appropriate
- Identifying and proposing architectural changes and new technology areas to assist with increasing service performance, availability, reliability, and recoverability such as availability zones, distributed resource scheduling (DRS) and high availability (HA)

2.3.1.2 Maximizing Service Change Velocity While Minimizing Stability Impacts

To satisfy their change management and availability responsibility, SRE practitioners need to address the conflict between supporting the required pace of innovation as reflected in service feature release velocity and meeting SLOs. This is where the error budget comes into play. The SRE practitioner works with the rest of the service team to optimize service releases to maximize the number of service releases within the error budget while minimizing the impact on service stability and meeting availability SLOs.

2.3.2 Responsibilities

The following table contains the set of core responsibilities SRE practitioners have for the services they support as originally identified by Google:

RESPONSIBILITY	DESCRIPTION
Availability	The percentage of time during which the IT service is available for c onsumption by an end user of the service as defined in the SLA. SRE practitioners are responsible for ensuring the agreed level of service availability is met and sustained.
Latency	The time it takes to service a request. SRE practitioners are responsible for monitoring service-request latency and proactively taking action to minimize it.
Performance	A measure of how a service responds under load. SRE practitioners are responsible for monitoring and improving the response of their service under load.
Efficiency	A measure of how a service uses its underlying resources to accomplish its performance target. SRE practitioners are responsible for ensuring a service maximizes its resource usage typically through understanding forward-looking demand, provisioning capacity, and optimizing how the service components consume resources.
Change management	Minimizing the impact of a change to the service. SRE practitioners can positively influence the change management of their service by minimizing the amount of change introduced in a rollout, automating how service changes are rolled out, detecting rollout problems, and safely rolling back changes when a problem is detected.
Monitoring	Watching your service and its underlying resources to ensure it's achieving the target level of availability and performance. SRE practitioners are responsible for developing and implementing a monitoring strategy for their services. They make use of tools like Wavefront™ by VMware, VMware vRealize Operations Manager, VMware vRealize Log Insight™, VMware vRealize Network Insight™ and VMware Cloud services in a VMware-supported environment to proactively detect and resolve issues before they become service impacting.
Emergency response	Activities undertaken in response to an unplanned service disruption. SRE practitioners are responsible for minimizing the MTTR during emergency responses by minimizing the impact of human intervention through automation and remediation best-practices playbook creation.
Capacity planning	Determining the capacity a service needs to satisfy projected demand while continuing to meet availability targets. SRE practitioners are responsible for ensuring sufficient capacity and redundancy are always available to satisfy projected demand while continuing to meet service availability targets.

3. SRE Applied to VMware-Supported Environments

According to Benjamin Treynor Sloss, Vice President of Google Engineering and founder of Google SRE, "SRE is what happens when you ask a software engineer to design an operations team."1 VMware-supported environments are based on API-driven, software-defined infrastructures utilizing a common cloud management platform. This is true whether in on-premises, hybrid-cloud, or multi-cloud contexts. In all three cases, being software-defined and API-driven help reach the levels of flexibility and automation needed. Given this description, applying SRE concepts is a natural fit for VMware-supported environments.

Having a standardized, software-defined approach to "infrastructure anywhere"—on-premises, hybrid cloud, and multi-cloud—enables SRE practitioners to more effectively and consistently implement operations from a software engineer's perspective. Instead of having to address on-premises, hybrid-cloud, and multi-cloud environments separately, they can take a common approach to all three, knowing that the difference is more about where to focus in each environment, without having to create separate approaches and use different tools for each. Across VMware-supported environments, common, reusable approaches and implementations can be deployed by PSO's DevOps for Infrastructure services for:

- Infrastructure-as-code definitions and provisioning tools
- Automating network and security overlay provisioning and operations
- Application, service, and virtual infrastructure monitoring, logging, troubleshooting, and remediation tools
- Application, service, and virtual infrastructure release management

The following sections describe how SRE concepts apply to the most commonly found VMware-supported environments.

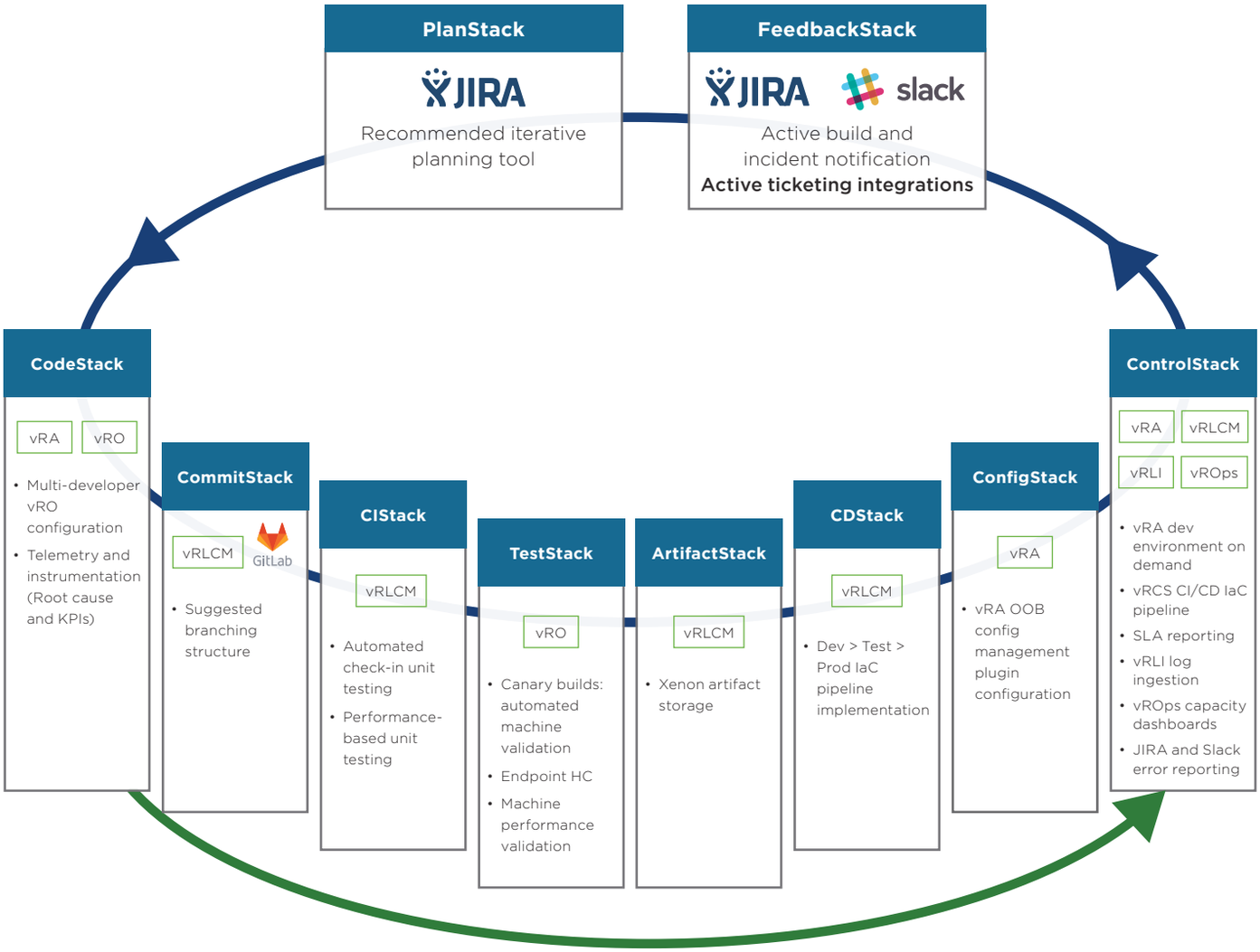
3.1 SRE for the Software-Defined Data Center

In an on-premises cloud environment based on the software-defined data center (SDDC), SRE concepts have broad applicability. IT can provide and support a wide array of services in an on-premises cloud environment, including:

- Cloud infrastructure services, from core infrastructure services through IaaS
- PaaS based on product offerings such as VMware Pivotal Container Service (PKS) or Pivotal Application Service (PAS)
- Database as a Service (DaaS) based on a product offering such as Amazon RDS on VMware
- Digital workspace services based on product offerings such as VMware Workspace ONE® and VMware Horizon®
- Application-based services such as migration to VMware Cloud on AWS with HCX

SRE concepts apply in all cases, and SRE practitioners are expected to satisfy their responsibilities across these service areas. Being able to apply a common set of approaches and tools across service areas is the key to SRE practitioners' success. A common IaC development, test, and release pipeline as shown in the following diagram is an example of how this can be applied across all services.

Example Opinionated DevOps Stack: Site Reliability Engineering for IaaS and PaaS with vRA



vmware® vRealize Lifecycle Manager

Not only can this pipeline be used for IaC, but for virtually all SDDC objects, such as dashboards, workflows, and software-based policy development, thereby providing a single approach to developing, testing, and managing SDDC changes.

Furthermore, a common set of monitoring and troubleshooting tools—such as Wavefront by VMware, vRealize Operations Manager, vRealize Log Insight, and vRealize Network Insight or their VMware Cloud services equivalents—can be used across all the infrastructure services. This enables the sharing of customizations, automation workflows, and a common basis for troubleshooting and remediation best-practice workbooks. Using a common set of monitoring and troubleshooting tools helps with reusability, thereby accelerating the effectiveness of SRE for emergency response, performance, availability, and latency across SRE practitioners and the services for which they are responsible.

Very importantly, vRealize Orchestrator can be employed as a common automation workflow tool for everything from virtual infrastructure provisioning fulfillment through its integration with vRealize Automation and day 2 operations automation through automated remediation workflows via the management pack for vRealize Operations.

3.2 SRE for Hybrid Cloud and Multi-Cloud

Applying SRE concepts in VMware-supported hybrid-cloud and multi-cloud environments differs from applying them to an on-premises environment only in lack of access and visibility to the physical infrastructure itself. Because public cloud infrastructure is provided and managed by the cloud provider, the SRE practitioner's focus is further up the stack. The same approaches and tools, especially those provided through VMware Cloud services, can be used to satisfy the SRE practitioner's responsibilities for their services hosted in VMware-supported hybrid and multi-cloud environments above the virtual infrastructure level itself. As a result, accommodations have to be made by SRE practitioners for cloud provider SLAs and processes as they relate to providing and managing the virtual infrastructure, especially related to troubleshooting and remediating service issues. This can be relatively easy in the case of cloud services like VMware Cloud on AWS wherein VMware, as the provider, proactively vacates customer workloads and replaces a failed host.

3.3 SRE for Cloud-Native and Hybrid Applications

Applying SRE concepts to cloud-native and hybrid applications is viewed from two perspectives:

1. The platform, such as PKS, on which cloud-native applications and the cloud-native portion of hybrid applications are developed and run
2. The applications themselves

In the context of the platform, SRE concepts and responsibilities are applied to the reliability of the platform itself. In this context, you might find a reference to a platform reliability engineer, who is concerned with the reliability of a specific platform. A platform reliability engineer would employ SRE concepts and satisfy SRE-defined responsibilities to ensure platform reliability.

Regarding cloud-native and hybrid applications, an SRE practitioner leverages the same tools for other services running in VMware-supported on-premises, hybrid-cloud, or multi-cloud environments. The difference being SRE practitioners are working directly with developers as an extension of the IT service team.

4. SRE—Operating Model Considerations

SRE concepts provide a new framework to view how operational activities supporting IT service release and consumption are performed. It places a heavier emphasis on reducing risks and errors through the automation of operational activities. As a result, it emphasizes spending the majority of an SRE practitioner's time on developing automation and applying it to service release and operations. This is undertaken with the primary objective of maintaining service reliability. While SRE concepts originated in supporting large applications delivered as services at Google, VMware sees SRE as something that can be applied just as easily to VMware-supported environments.

VMware recommends employing a service-oriented operating model to fully leverage the software-defined capabilities present in VMware-supported environments. This involves viewing everything delivered for consumption in a VMware-supported environment as a service. These services range from providing cloud infrastructure services for consumption by other services teams as well as end-customers, to platform services, including PKS, supporting consumption of Kubernetes, to digital workspace services utilizing Workspace ONE or applications themselves packaged in virtual machines or containers for direct consumption by end-users. SRE concepts applied in this context have a direct impact on aspects of the optimization of capabilities as part of a service-oriented operating model, namely from both the people and process perspectives.

4.1 People Perspective

4.1.1 The SRE Role

As described in a previous section, SRE practitioners have a defined core set of responsibilities. The following table describes the required skillset and responsibilities, with examples, in the context of VMware's service-oriented operating model.

RESPONSIBILITIES	SKILLSET
Maintaining the level of service reliability required to meet the target SLOs. Depending on the service(s) the SRE practitioner is supporting, this could include the reliability of: • On-premises cloud infrastructure services including the cloud management platform and underlying SDDC components as well as IT infrastructure support services hosted in the environment • Cloud infrastructure services delivered in a hybrid-cloud environment including the VMware NSX® overlay network as well as IaaS consuming both on-premises and public cloud resources • A PKS environment consuming services from underlying cloud infrastructure services • An application-based service packaged in containers running in a hybrid cloud where the containers can reside either on-premises or in a public cloud Supporting service development, testing, release, and continuous improvement, for example by: • Customizing the VMware vRealize Code Stream™ workflows using vRealize Orchestrator • Developing Puppet-based service configurations to act as Service as Code • Developing automated operational readiness tests for the service and virtual environment in which its packaged • Monitoring service behaviour as it moves the test pipeline to inform operational considerations once in production as well provide feedback to the service developer(s) Optimizing how the service(s) for which the SRE practitioner is responsible is monitored, such as: • Customizing vRealize Operations dashboards and super-metrics, vRealize Log Insight dashboards and filters, and Wavefront by VMware dashboards and filters for optimal availability and performance monitoring of the application and VMs in which its components are packaged in a hybrid, VMware Cloud or native cloud environment Work with business stakeholders or business relationship managers to understand forward-looking demand and perform service-based capacity planning, such as: • Based on business stakeholder or business relationship managers discussion, develop vRealize Operations intelligent analytics to monitor capacity usage of the application and VMs in which its components are packaged, as well as automated workload-placement policies that can make use of both on-premises and VMC on AWS resources	• Sufficient cross-domain skills to have a service-specific, full-stack functional understanding • SME skills in select, service-specific functional domains. This assumes multiple SRE practitioners are responsible for a given service, with complementing SME skillsets • Software development, which may include, as needed: vRealize Orchestrator, NSX API, vRealize Automation API, vRealize Operations™ API, VMware Cloud API, public cloud APIs, JavaScript, PowerShell, Python, configuration management tools (Puppet, Chef, Ansible, Salt) • Release Management tool customization, such as for vRealize Lifecycle Manager™ and vRealize Code Stream; release management techniques such as canary and blue/green release techniques • DevOps concepts and Agile-base development methodologies • Monitoring tool customization, such as for Wavefront by VMware, vRealize Operations Manager, vRealize Log Insight, and vRealize Network Insight as well as for VMware Cloud services and integration with other monitoring and tracing tools as needed

RESPONSIBILITIES	SKILLSET
Working with the rest of the service team to develop a strategy for supporting frequent service feature releases while maintaining service stability, such as: • Work with the service team to determine the service release frequency and scope based on error budget as well as develop automation to automatically update the NSX load balancer fronting the service in production to effect canary releases. Minimizing the impact of human intervention during all operational activities including, for example, emergency response, service releases, and optimizing resource utilization by, for example: • Customizing vRealize Operations Manager guided root cause analysis to reflect remediation best practices identified during blameless post-mortems • Automating service release to facilitate a canary release mechanism and rollback process to make use of pre-identified failure criteria • Developing workload placement policies for use during service provisioning in VMware vRealize® Automation® as well as policies for ongoing workload optimization in vRealize Operations Manager	

4.1.2 Measuring the Value of SRE

Implementing an SRE-based approach will result in service operational improvements. The following table describes some key metrics for measuring SRE effectiveness.

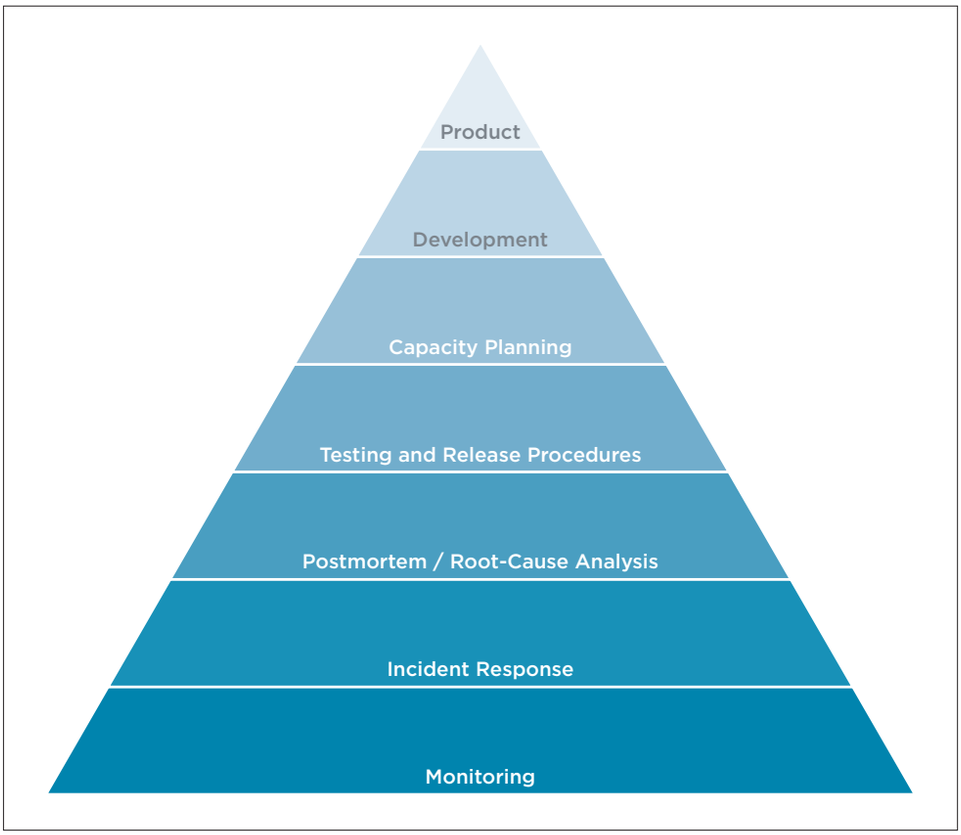
METRIC	DESCRIPTION
Release frequency	• How often are we able to launch new consumer capabilities such as new code or a new blueprint, day 2 action, or automation capability. • Release frequency should trend up or remain stable from week to week.
Change volume	• How many new service features are being deployed over a 30-day period? What is the complexity of the changes? • While we want release frequency to trend up or remain stable, this has to be balanced against the amount of change occurring during releases which should remain low, thereby supporting the concept of smaller but more frequent releases.
Change impact	• Average number of cycles through the continuous testing component of the CI/CD pipeline per change. • Measures the quality of changes as well as automated tests. • Average number of cycles should decrease over time reflecting a reduced cost of change as well as increased “speed to market” for changes.
Average change rollback time	• How long does it take, on average, to roll back a “bad” change or release? • Average change-rollback time should trend downward or at least remain stable week to week as detection and automated rollback continuously improves.

METRIC	DESCRIPTION
Lead time (from development to deployment)	• This is the cycle time from when development begins for new service features to when it successfully gets deployed into production. Cycle time is an important indicator of efficiency in the process—when tracked using value stream mapping, it can help the team to visualize areas in the process which need improvement, such as automated testing. • Lead time should decrease as the service team continuously improves the process.
Mean time to recovery (MTTR)	• When failures occur, how long does it take the team to recover from the issue? • Spikes in MTTR are fine for complex issues which the team has never encountered before, but the overall trend for this metric should decrease over time as remediation is automated and remediation best practice playbooks are developed. • System generated tickets triggering automated response should trend upwards while the frequency of manual intervention should trend downward. • The success rate of automated response as measured against the total number of automated responses should trend upward.
Customer ticket volume	• How effectively is proactive issue detection and remediation being implemented? • The number of system-generated (detected by the monitoring service) tickets should trend upward while user generated ticket volume should trend downward.
Availability	• How well are the agreed availability targets defined in SLOs (or SLAs, as applicable) being met? • SLO (or SLA, as applicable) availability targets should be met 100% of the time
Performance (Response Time)	How well are performance targets for critical service activities being met? These could include but are not limited to: • Service response time to user request or access • Service provisioning time • Time to complete a self-service, day 2 operation provided as part of a service Defined and agreed service performance targets should be met 100% of the time regardless of % change in user volume or any new deployment.
Service efficiency	How is the service using its underlying resources to meet its performance target? This can be reflected by, for example: • Percent of time performance targets are missed due to lack of infrastructure capacity • Percent of time performance targets are missed resulting in having to optimize service component configurations or other service component aspects Defined and agreed service performance targets should never be missed due to service inefficiency.
Error budget	• How well is the service's error budget being spent? • The amount of error budget spend should trend upward or remain stable month after month but never exceed the error budget.
Toil	• How much time is being saved automating manual operational processes? • Time spent performing manual operational tasks to support the service should trend downward toward the 50% target. • Time spent automating manual tasks and performing higher-value tasks related to increasing lead time and release frequency should trend upward toward the 50% target.

4.2 Process Perspective

4.2.1 Service Reliability Hierarchy

SRE practitioners are ultimately responsible for the health of the services they take ownership of. Google took an interesting approach to characterizing the health of a service by modeling it after Maslowe's hierarchy of needs pyramid.[5] In much the same way Abraham Maslow categorized human needs, the service-reliability hierarchy categorizes the health of a service from the most basic requirements for functioning as a service to the higher levels of "self-actualization" by taking proactive control rather than the reactive mode of constantly fighting fires. This section uses the model to describe recommendations, from basic to advanced, for SRE activities in the context of VMware-supported environments.



[5] "Part III. Practices." Site Reliability Engineering: How Google Runs Production Systems. Edited by Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Murphy. O'Reilly Media, 2016.

4.2.1.1 Monitoring

Monitoring is basic to understanding the health of any service. Without monitoring you have no way of knowing if a service is even responding, short of help desk calls from end users. SRE practitioners are responsible for not only developing the monitoring strategy for their services and their delivery pipeline but also continuously refining the implementation of that strategy. In the context of VMware-supported environments we recommend full stack monitoring for a service. This includes not only the service, the applications composing it, and its usage—using a tool like Wavefront by VMware for real-time, analytics-driven monitoring—but also the infrastructure on which the service is running, using tools like vRealize Operations Manager for intent-based, time series based analytics and alerting, vRealize Log Insight for intelligent analysis of structure and unstructured log data, and vRealize Network Insight for intelligent network overlay and underlay monitoring, or their VMware Cloud services equivalents for multi-cloud environments. VMware also recommends instrumenting the service components for more service-specific monitoring, such as applications composing the service and the vRealize Orchestrator–based service workflows, and making the data available to Wavefront, vRealize Operations Manager, and vRealize Log Insight, or their VMware Cloud services equivalents.

4.2.1.2 Incident Response

On-call duty continues to be a necessary evil even for SRE practitioners. One difference is how SRE practitioners should view and use on-call duty: as another way to continuously learn how their service works and improve its reliability and efficiency. It's also about using reactive incident and proactive issue resolution to understand how to continuously decrease the amount of MTTR latency due to human intervention—ultimately by implementing self-healing IT services.

In a VMware-supported environment context, this is about leveraging the intelligent analytics and guided root cause analysis capabilities of vRealize Operations Manager along with its integration with Wavefront by VMware, and vRealize Log Insight as well as its ability to accept time series–based data from external systems. SRE practitioners can take advantage of these capabilities in several ways, including:

- Speeding up troubleshooting and remediation—whether in response to an incident that has occurred or to proactively resolve an issue before it impacts customers
- Implementing automated remediation where feasible
- Providing insights that can be turned into building more resilience and self-healing capabilities into the service's code and configuration
- Creating troubleshooting and remediation best-practice playbooks that can be built right into the guided root cause analysis dialog
- Integrating with trouble ticketing systems and Agile planning tools to promote an active feedback loop with everyone on the service's team for continuous service improvement

4.2.1.3 Blameless Postmortem / Root Cause Analysis

A key tool for SRE practitioners is the blameless postmortem. Traditionally, operations teams avoid postmortems as they are usually more about placing blame than productively deconstructing the incident or issue and understanding how to avoid it in the future. For SRE practitioners, it's yet another learning opportunity. For postmortems to be effective, they have to be blameless and focused on understanding the root cause and how to reduce the likelihood and impact of its recurrence. The goal is always to prevent repetitive failures.

From a VMware-supported environment perspective, this is again about leveraging the intelligent analytics, guided root cause analysis, and automated remediation capabilities of vRealize Operations Manager. These capabilities are leveraged to fully understand the root cause and then either develop automated remediation callouts or use the detailed root cause analysis to understand what additional resilience can be built into the service and its supporting workflows.

4.2.1.4 Testing + Release Procedures

Agile methodology calls for more frequent but smaller releases. While this has many advantages such as releasing new service features to customers more frequently, increasing the agility with which customer feedback on a feature can be incorporated, and limiting exposure during any given release, it does put increased pressure on SRE practitioners both for test coverage and reducing total testing time. SRE practitioners also need to worry about frequently releasing service changes into production without adversely impacting overall service reliability. This is true not only for the service itself, but also for changes SRE practitioners make to automation workflows, IaC configuration changes, and tool customizations, such as to vRealize Operations dashboards.

For SRE practitioners in a VMware-supported environment, vRealize Lifecycle Manager, vRealize Orchestrator, and API access to underlying VMware technologies for on-premises or hybrid cloud and the VMware Cloud services VMware Cloud Assembly™ and VMware Code Stream™ for multi-cloud are critical tools. For example:

- vRealize Lifecycle Manager and VMware Code Stream provide software artefact release management that makes use of rule-based progression between custom definable stages and inherent version control. It can be used to automate the testing and release of service components as well as support vRealize Orchestrator-based automation workflows, IaC, and vRealize suite content such as dashboards and policies.
- vRealize Lifecycle Manager and VMware Code Stream integrate with many of today's leading CI/CD pipeline tools and automated testing tools.
- vRealize Orchestrator can be used to develop automated operational readiness testing capabilities and automated rollback controlled by vRealize Lifecycle Manager or VMware Code Stream (planned).
- APIs to underlying technologies such as NSX provide the opportunity to automatically update software-defined load balancers and firewalls to facilitate canary and blue/green release testing approaches.
- The same tools can be used for releases and rollouts of service components themselves as well as its supporting automation workflows, IaC definitions, and vRealize Suite customizations

4.2.1.5 Capacity Planning

As described in the SRE Responsibilities section, capacity planning is another critical SRE defined responsibility. SRE practitioners are responsible for ensuring sufficient capacity and redundancy to satisfy projected demand while maintaining required availability. Unlike traditional capacity planning, which relied heavily on historical capacity trending, in the highly dynamic VMware-supported environments, capacity planning relies more heavily on forward-looking demand prediction. This requires a better understanding of how business units intend to use the services or see their customers consume the services. SRE practitioners typically access this information through either the service owner or business relationship managers who own the relationship with one or more business units.

In a VMware-supported environment, capacity monitoring, management, and planning is accomplished with vRealize Operations Manager. Using vRealize Operations Manager, SRE practitioners can:

- Monitor service-specific capacity usage as well as utilize forward-looking capacity forecast analytics
- Fully automate service and workload balancing across the data center and hybrid cloud deployments to ensure performance targets are adhering to SLOs
- Use predictive DRS to pre-empt resource contention and take automated action
- Understand service-specific capacity reclamation and right-sizing opportunities
- Implement threshold-triggered workload auto scaling
- Use forecast-based planning scenarios to model upcoming service demand and capacity requirements

4.2.1.6 Development

SRE practitioners impact development from two perspectives in a VMware-supported environment:

1. Supporting service developers, or application developers if the service consists of an application
2. Developing service automation workflows, Infrastructure or Service as Code configurations, service-specific software defined policies, and operations tool customizations, for example

SRE practitioners support service and application developers, for example, by:

- Creating service-specific test and release workflows as well as interstage promotion rules in tools like vRealize Lifecycle Manager or VMware Code Stream
- Developing automated operational readiness tests to be run during service and application testing stages
- Using Wavefront by VMware and vRealize Operations Manager to monitor service and application components during testing stages, and providing feedback to the developers
- Working with service and application developers to integrate operations considerations, such as resilience, directly into their code
- Developing and maintaining version-controlled IaC to support immutable infrastructure and idempotency of the environments used throughout the development, testing, and release process

In the second instance, SRE practitioners also directly develop automation workflows, configurations, software-defined policies, and operational tool customizations, for example. When doing so, SRE practitioners should use the same DevOps concepts, Agile methodology, and CI/CD pipeline tools used by the service and application developers.

4.2.1.7 Product

At the top of the Site Reliability Hierarchy is the product or service for which the SRE practitioner is responsible. Everything below the product or service contributes to its reliability. In a VMware-supported environment, this includes:

- All aspects of the product or service being consumed are instrumented for and monitored with self-healing and automated remediation capabilities using tools like Wavefront by VMware and vRealize Operations Manager.
- Where automated remediation isn't possible, troubleshooting and remediation best-practice playbooks are ideally instantiated in vRealize Operations Manager's guided root cause analysis dialogs.
- After an appropriate change impact risk assessment, any changes made for whatever reason are implemented, tested, and rolled out through the CI/CD pipeline using vRealize Lifecycle Manager or VMware Code Stream as a release management orchestrator and using techniques such as canary and blue/green releases.
- Automated continuous delivery or continuous deployment and rollback capabilities using workflows developed in vRealize Orchestrator are in place.
- Metrics are in place, as described in the Measuring the Value of SRE section above, to measure and continuously improve product or service reliability.

4.3 Evolving to a Site Reliability Engineering Model

We realize that it may not be feasible for all organizations to have a dedicated team or role that solely performs the SRE function. This is not a barrier to implementation of SRE.

4.3.1 The Person vs. the Function

In Google's model, SRE practitioners are part of a dedicated function in the organizational model. SRE employees and small teams of SRE practitioners are dedicated to Google's most critical services, for instance Borg,[6] Google's large-scale cluster manager, which runs at the core of most Google services.

Dedicated SRE practitioners are concerned with the reliability of the service to which they are assigned. Small SRE teams split their time between making improvements and keeping the lights on. Typically, a single SRE practitioner is assigned the role of keeping the lights on for a limited time, with other SRE team members rotating into that role on a schedule. This allows the remaining SRE practitioners to dedicate their time to reliability improvements.

[6] Abhishek Verma et al. "Large-Scale Cluster Management at Google with Borg." Google AI. 2015.

There is another type of SRE practitioner, which is the "SRE practitioner on loan." These SRE practitioners are dispatched to service teams without dedicated SRE practitioners, and who have met documented criteria for reliability and criticality. The "on loan" SRE practitioners assist service teams with taking a software engineering mindset into the operations domain. They demonstrate good practice to the team and transfer knowledge the team can use to improve their service delivery. Often, the SRE-practitioner-on-loan approach is used as part of an evolutionary approach to implement SRE more broadly and mitigate the risk of investing too quickly in dedicated SRE practitioners or teams to specific IT provided services.

This illustrates that the main factor in the success of SRE is not necessarily the organizational structure Google or others have adopted. The principles are what matters—and these principles should be adopted no matter the organization structure. The models could include:

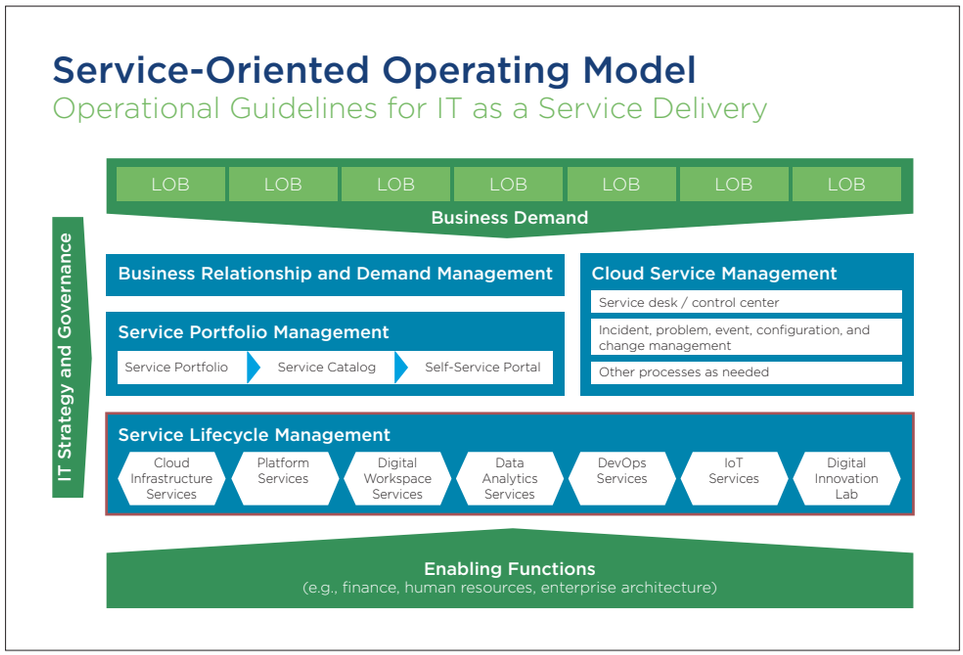
- Dedicated SRE team or teams
- Dedicated SRE practitioners as part of service teams
- SRE defined responsibilities shared among team members of service teams

4.3.2 A Service-Oriented Model as a Prerequisite for SRE

VMware is a recognized leader and innovator in "software defined." Taking a software-defined approach enables the agility and speed IT needs to consistently and continuously deliver value to the business at the speed the business requires. This in turn allows IT to move away from simply providing capabilities to becoming an internal service provider supplying the business-enabling solutions that drive innovation and deliver value. Doing so positions IT as a true business partner rather than an increasingly irrelevant, cost-centric technology supplier. From a VMware perspective, therefore, employing SRE concepts requires IT to adopt a service-oriented approach to defining, developing, delivering, and operating what IT provides for consumption in VMware-supported environments.

At the core of VMware's service-oriented operating model lies the concept of service lifecycle management teams. These teams are responsible for their services from definition through end of life. As shown in the diagram below, these services can consist of, for example:

- Cloud infrastructure services that can span software-defined infrastructures and cloud management platforms hosted on-premises as well as in hybrid-cloud and multi-cloud environments
- Platform services containing, for example, PKS
- Digital workspace services such as Workspace ONE or Horizon, either on-premises or cloud-based
- Data analytics services such as providing data lakes or a business intelligence application hosted on-premises or cloud-based
- IoT services providing everything from onboarding to managing, monitoring, and securing devices, gateways, and access in an end-to-end IoT solution, perhaps even consuming data analytics services for analyzing IoT-generated data
- A service containing one or more applications such as a customer-facing mobile application or internal business application



Service lifecycle management teams are self-sufficient, exhibit a DevOps mindset, and work in an Agile way. They contain the core skills and capabilities required to address the Plan (strategy to services architecture), Build (engineering), Deliver (request to fulfillment), and Run (operations) aspects of their services as well as the critical technologies comprising the service. Service lifecycle management teams always include a service owner role along with service-specific architect, engineer, administrator, analyst, and developer roles. An individual may fill multiple roles or a single role. Depending on business criticality and scale, there may be multiple people with different technical skills filling a single role. In addition to these common, core roles, a service lifecycle management team can contain service-specific roles such as data scientist in the case of data analytics services, for example.

SRE concepts are a natural fit in this service-oriented operating model.

4.3.3 Adopting SRE Concepts

Specifically, SRE concepts are a natural fit for service lifecycle management teams. These teams are responsible for everything related to the development, delivery, and operations of their service and therefore the service's reliability. They are already expected to exhibit a DevOps mindset and work in an Agile way as well as place a heavy focus on automation.

Incorporating SRE core tenets and responsibilities as well as the Service Reliability Hierarchy represents a natural progression for service lifecycle management teams. Per the treatment of "The Person vs. the Function" topic above, this is the critical step. Whether a specific SRE role is included or the SRE responsibilities are shared across multiple roles in a service lifecycle management team becomes secondary and subject to a company's preference.

That said, it also depends on IT's level of maturity. Likely, the bigger challenge will be adopting a service-oriented operating model if one is not already in place. If you are not yet service-oriented but intend to shift to a service orientation, you should consider incorporating SRE concepts and an explicit SRE role in your service lifecycle management teams. This role could consolidate the analyst, administrator, and developer roles. If you have already transitioned to a service-oriented approach, it might be easier to initially share SRE responsibilities across existing team members and decide over time if it makes sense to consolidate them into an explicit SRE role.

In either case, when a decision is made to adopt SRE concepts, take the following steps based on the Service Reliability Hierarchy:

1. Start small by picking a single service for which to initially implement SRE concepts.
2. Identify individuals to fill the SRE role, or whom to spread SRE responsibilities across.
3. In support of service reliability issues:
   - Identify service level metrics, measure, and report them.
   - Create a service level baseline for the service.
   - Identify service reliability gaps based on the metrics, develop a sprint-based action plan to close them, and execute sprints until service level targets are met.
   - Implement service provider principles based on your customer's demand.
4. Build a service feature backlog with consideration to continuous improvement of workload management for automated request to fulfillment capabilities.
5. Review the monitoring strategy to include a reliability perspective, develop a prioritized sprint-based plan to implement, and implement highest priority items while placing the remainder in backlog.
6. Review incident–response process, postmortems, and root cause analysis. Implement blameless postmortems and a full root cause analysis process. Develop automation as well as troubleshooting and remediation best-practice playbooks over time.
7. Review testing and release procedures, develop a prioritized, sprint-based plan focused on automated testing and reliability "aware" release procedures, and implement highest priority items while placing the remainder in backlog.
8. Review capacity and scalability planning, develop a prioritized, sprint-based plan focused on capacity monitoring and identifying forward-looking demand, and implement highest priority items while placing the remainder in backlog.
9. Determine an initial error budget and work with other service lifecycle team members to arrive at a target release size and frequency.
10. Identify similar individuals for other services, enabled based on experience with initial service so they can begin implementing for their respective services
11. Create an SRE "chapter"[7] consisting of SRE practitioners or those responsible for SRE activities across service lifecycle teams to develop standards, share knowledge, raise awareness within the organization, and provide enablement.
12. Develop and execute an SRE communication strategy to drive further adoption of SRE concepts and principles within the organization.

[7] Henrik Kniber and Anders Ivarsson. "Scaling Agile at Spotify with Tribes, Squads, Chapters, and Guilds." Crisp. October 2012.

5. Resources

The following resources are useful for further understanding and exploring SRE concepts:

DESCRIPTION	RESOURCE
Site Reliability Engineering: How Google Runs Production Systems	https://landing.google.com/sre/book/index.html
DevOps and Agile Development: A VMware Field Perspective	http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/vmware-devops-agile-development.pdf

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents. VMware is a registered trademark or trademark of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. Item No: VMW-WP-SRE-USLET-101
11/18