The Four Golden Signals

The four golden signals of monitoring are latency, traffic, errors, and saturation. If you can only measure four metrics of your user-facing system, focus on these four.

Latency

The time it takes to service a request. It's important to distinguish between the latency of successful requests and the latency of failed requests. For example, an HTTP 500 error triggered due to loss of connection to a database or other critical backend might be served very quickly; however, as an HTTP 500 error indicates a failed request, factoring 500s into your overall latency might result in misleading calculations. On the other hand, a slow error is even worse than a fast error! Therefore, it's important to track error latency, as opposed to just filtering out errors.

Traffic

A measure of how much demand is being placed on your system, measured in a high-level system-specific metric. For a web service, this measurement is usually HTTP requests per second, perhaps broken out by the nature of the requests (e.g., static versus dynamic content). For an audio streaming system, this measurement might focus on network I/O rate or concurrent sessions. For a key-value storage system, this measurement might be transactions and retrievals per second.

Errors

The rate of requests that fail, either explicitly (e.g., HTTP 500s), implicitly (for example, an HTTP 200 success response, but coupled with the wrong content), or by policy (for example, "If you committed to one-second response times, any request over one second is an error"). Where protocol response codes are insufficient to express all failure conditions, secondary (internal) protocols may be necessary to track partial failure modes. Monitoring these cases can be drastically different: catching HTTP 500s at your load balancer can do a decent job of catching all completely failed requests, while only end-to-end system tests can detect that you're serving the wrong content.

Saturation

How "full" your service is. A measure of your system fraction, emphasizing the resources that are most constrained (e.g., in a memory-constrained system, show memory; in an I/O-constrained system, show I/O). Note that many systems degrade in performance before they achieve 100% utilization, so having a utilization target is essential.

In complex systems, saturation can be supplemented with higher-level load measurement: can your service properly handle double the traffic, handle only 10% more traffic, or handle even less traffic than it currently receives? For very simple services that have no parameters that alter the complexity of the request (e.g., "Give me a nonce" or "I need a globally unique monotonic integer") that rarely change configuration, a static value from a load test might be adequate. As discussed in the previous paragraph, however, most services need to use indirect signals like CPU utilization or network bandwidth that have a known upper bound. Latency increases are often a leading indicator of saturation. Measuring your 99th percentile response time over some small window (e.g., one minute) can give a very early signal of saturation.

Finally, saturation is also concerned with predictions of impending saturation, such as "It looks like your database will fill its hard drive in 4 hours."

If you measure all four golden signals and page a human when one signal is problematic (or, in the case of saturation, nearly problematic), your service will be at least decently covered by monitoring.

Why are these three pillars the most important?

When talking about unified observability, it is in the context of metrics, logs, and traces, but why are these so important? Let's explore.

Metrics provide real-time insight into the health and performance of applications or infrastructure. With observability into metrics, you will see the greater context of system health and be able to proactively identify performance issues.

Traces provide insight into the flow of the application. With observability into traces, you can see the source of the problem and identify the root cause, even in distributed systems like microservices and containers.

Logs provide insight into all events and errors within a software environment. With observability into logs, you can see when the problem occurred and which events correlate with it.

IT service management (ITSM)

The implementation and management of quality IT services that meet the needs of a business. IT service management is performed by IT service providers through an appropriate mix of people, processes, and information technology.

Incident management

The process responsible for managing the life cycle of all incidents. Incident management ensures that normal service operation is restored as quickly as possible and the business impact is minimized.

Problem management

The process responsible for managing the life cycle of all problems. Problem management proactively prevents incidents from happening and minimizes the impact of incidents that cannot be prevented.

Change management

Transitioning something newly developed (i.e. an update to an existing production environment or something entirely new) from the service design phase into regular service operation, all while aiming to ensure that standardized methods and procedures are used for efficient handling of all changes.

Asset management

A generic activity or process responsible for tracking and reporting the value and ownership of assets throughout their life cycle.

Information Technology Infrastructure Library (ITIL®)

A set of best-practice publications for IT service management. ITIL® gives guidance on the provision of quality IT services and the processes, functions, and other capabilities needed to support them. The ITIL® framework is based on a service life cycle and consists of five life cycle stages (service strategy, service design, service transition, service operation, and continual service improvement), each of which has its own supporting publication.

Software as a service (SaaS)

A software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted by the vendor. It is sometimes referred to as "on-demand software."

Key performance indicator (KPI)

A metric that is used to help manage an IT service, process, plan, project, or other activity. KPIs are used to measure the achievement of critical success factors. Many metrics may be measured, but only the most important of these are defined as KPIs and used to actively manage and report on processes, IT services, and activities. Help desks should select KPIs to ensure that efficiency, effectiveness, and cost-effectiveness are all managed.

Service-level agreement (SLA)

An official commitment that prevails between a service provider and a client. When defining SLAs, the service provider and service user agree on particular aspects of the service, including quality, availability, and responsibilities.

On-premises

Software that is installed and runs on computers within the premises (in the building) of the person or organization using the software, rather than at a remote facility such as a server farm or cloud.

Cloud

Refers to using a network connection to access applications and data stored in other locations, often by accessing data centers using wide area networking (WAN) or regular internet connectivity.

Self-service portal

A self-service portal is a website or app that enables users—whether they're customers, employees, suppliers, or partners—to perform high-value transactions, from simple account updates to paying bills, managing support tickets, and more.

Service catalog

A database or structured document with information about all live IT services, including those available for deployment. A service catalog is part of a service portfolio and contains information about two types of IT services: customer-facing services that are visible to the business, and supporting services the service provider requires to deliver customer-facing services.

Knowledge base (KB)

A logical database containing data and information used for knowledge sharing and management.

Incident

An unplanned interruption to an IT service or reduction in the quality of an IT service. Failure of a CI, even if it has not yet affected a service, is also an incident (e.g. failure of one disk from a mirror set).

Root cause analysis (RCA)

A methodology used in problem management to analyze the core issue (root cause) that led to a series of incidents. The root cause is an element or factor that, when removed, restores normalcy and prevents the problem from reoccurring.

Known error database (KEDB)

A repository of previously identified and recorded errors or root causes with tested workarounds, which can be used as reference for similar problems in the future.

Workaround

A temporary solution to a known error that minimizes or eliminates the impact of an incident or problem.

Response time

The time taken to respond to a logged ticket. The first response time refers to the time taken to respond to a ticket for the very first time after it was logged.

Resolution time

The time taken to resolve an incident or problem and bring it to closure.

Project management

An organized process that involves the planning, organizing, managing, and controlling of IT projects to accomplish specific IT goals.

Enterprise service management (ESM)

Managing a service organization with a suite of ITSM tools and applications applied to ITSM software to optimize its performance. ESM enables departments within an organization to stay connected to each other and to external resources, thereby functioning as a single unit across all facets of service management.

Change Advisory Board (CAB)
A change advisory board (CAB) is a board made up of representatives from important areas within an organization like IT, finance, and facilities, including technical staff and key decision makers.

These members advise the Change Manager on the assessment, prioritization, and scheduling of changes within the IT environment.

Business Impact Analysis (BIA)
A business impact analysis (BIA) is an activity that identifies business functions and their dependencies. Usually delivered in the form of a report, this type of analysis is helpful in the service management world for developing a strategy in case certain business units experience disruption.

This type of analysis can also reveal vulnerabilities and areas of concern to help prevent problems and reduce risk.

Key Performance Indicator (KPI)
A KPI is a measure of regular assessment used to indicate the performance of an IT process. KPIs are usually accompanied by an agreed-upon threshold of quality and performance standards an organization sets.

Though performance indicators vary among organizations, industries, and IT infrastructures, there are KPIs to signify strategies, design, operations, and more. An example of a KPI in service management is Average Wait Time.

KPIs are typically derived from what are called critical success factors (CSF).

Critical Success Factor (CSF)
Critical success factors (CSFs) are often used to denote an organization's business strategy. In an IT service management (ITSM) context, there are a number of fairly general critical success factors that must be present for an ITSM implementation project to be successful.

Some examples of those are clearly defined roles and responsibilities for the project staff, developing a workforce with the necessary knowledge and skills required to be successful in their roles, and an understanding and management structure for different stakeholder perspectives.

For measuring the necessary knowledge and skills for employees, for example, this CSF is usually paired with a KPI for measuring progress.

Service Level Agreement (SLA)
A Service Level Agreement (SLA) is one of the most important ITIL terms to know and can be defined as an agreed-upon level of service between a service provider and a service requester.

SLAs identify what level of service has been agreed upon, what metrics should be used, and any penalties that may be imposed if the agreement is breached.

Check out our other blog posts to learn more about SLAs and other important ITSM ITIL metrics.

Business Relationship Management (BRM)
Business relationship management (BRM) is an important ITIL practice that assesses the needs of customers to help provide the level of service required.

When IT understands and closely aligns with the needs and goals of the business, the two operate much more seamlessly in tandem.

Service Configuration Management (SCM)
Service configuration management provides information about the configuration of services and the configuration items (CIs) that support them. This helps give leaders a clearer picture of an organization's service lifecycle.

Service Value System (SVS)
In ITIL 4, the service value system (SVS) is an overarching concept that represents how components and activities within an organization work together to achieve value creation.

To create this value, the SVS within an organization as outlined by ITIL can be broken down into five key components: guiding principles, governance, service value chain, practices, and continual improvement.

Service Request Management (SRM)
Service request management (SRM) is a key component of an ITIL service catalog that enables service requests to be handled appropriately. Requests can come in many forms, whether it is a request for access, information, or even feedback.

From submission and routing, to service request approvals, monitoring, and delivery, SRM enables leaders to properly handle requests and meet their quality of service standards.

Request for Change (RFC)
A request for change (RFC) is documentation that is submitted prior to a change being implemented. This formal request outlines the details of the change itself, before the change process begins.

This is not to be confused with a change record, which documents the lifecycle of a change that has already occurred.