MEAN TIME TO RESOLVE (MTTR) AS A SERVICE DESK METRIC

The service desk is a valuable ITSM function that ensures efficient and effective IT service delivery. A variety of metrics are available to help you better manage and achieve these goals. These metrics often identify business constraints and quantify the impact of IT incidents. Of course, the vast, complex nature of IT infrastructure and assets generate a deluge of information that describe system performance and issues at every network node. The challenge for service desk? Identifying the metrics that best describe the true system performance and guide toward optimal issue resolution.

We've talked before about service desk metrics, such as the cost per ticket. Another service desk metric is mean time to resolve (MTTR), which quantifies the time needed for a system to regain normal operation performance after a failure occurrence. In this article, we'll explore MTTR, including defining and calculating MTTR and showing how MTTR supports a DevOps environment.

What is MTTR?

Beyond the service desk, MTTR is a popular and easy-to-understand metric:

- DevOps professionals discuss MTTR to understand potential impact of delivering a risky build iteration in production environment.
- Business executives and financial stakeholders question downtime in context of financial losses incurred due to an IT incident.
- Customers of online retail stores complain about unresponsive or poorly available websites.

In each case, the popular discussion topic is the time spent between failure and issue resolution. So, let's define MTTR.

'Mean time to recovery' is the average time duration to fix a failed component and return to an operational state. This metric includes the time spent during the alert and diagnostic processes, before repair activities are initiated. (The average time solely spent on the repair process is called 'mean time to repair', also shortened to MTTR.) MTTR can be mathematically defined in terms of maintenance or the downtime duration:

$$MTTR = \frac{Total\ Hours\ of\ Maintenance}{Total\ Number\ of\ Repairs}$$

$$MTTR = \frac{Total\ Hours\ of\ Downtime}{Total\ Number\ of\ Incidents}$$

In other words, MTTR describes both the reliability and availability of a system:

- Reliability refers to the probability that a service will remain operational over its lifecycle. It can be described as an exponentially decaying function with the maximum value in the beginning and gradually reducing toward the end of its life.
- Availability refers to the probability that the system will be operational at any specific instantaneous point in time.

The shorter the MTTR, the higher the reliability and availability of the system. From a practical service desk perspective, this concept makes MTTR valuable: users of IT services expect services to perform optimally for significant durations as well as at specific instances. For example, Amazon Prime customers expect the website to remain fast and responsive for the entire duration of their purchase cycle, especially during the holiday season. If the website is down several times per day but only for a millisecond, a regular user may not experience the impact.

MTTR encourages DevOps

MTTR is a valuable metric for service desks on its own, but it also encourages DevOps culture and practices in a variety of ways:

- **Low impact of incidents.** The primary objective of MTTR is to reduce the impact of IT incidents on end users. If an issue is resolved before a customer's online activity is disrupted, the service will be accepted as efficient and effectively delivered.
- **Resilient system design.** The service desk goals associated with MTTR are achieved by developing a resilient system or code. For example, a website feature could be developed as a separate code with web service called independently from other features. Any repairs or changes to a specific feature may not impact the performance of other website features. This would make the entire website resilient, with each feature easy to repair.
- **Feedback loop.** Any improvement to the software build requires a fast feedback mechanism that informs developers early during the SDLC pipeline. MTTR can be reduced when the bugs are small in scope, easy to fix, and identified during the early development stages.
- **Reduced dependencies.** MTTR increases when fixing a single issue requires the fix of multiple functions and systems tightly integrated and dependent to each other. By reducing such dependencies, the MTTR is reduced. From a service desk perspective, the services are carefully evaluated to ensure low dependencies.

- **Active monitoring.** The resolution process can only begin after a fault is identified. Actively monitor the infrastructure logs to identify patterns of anomalous behavior and the underlying problem root cause. With this information, the service desk can perform appropriate problem management or incident response actions, thereby reducing downtime.
- **Rapid iterations.** Any fix applied to a system can have negative consequences. The focus on reducing MTTR encourages small and fast fixes that can be easily deployed—and rolled back—in response to a negative outcome.
- **Designing for failure.** Reducing MTTR encourages the service desk to design for, prepare for, and embrace failure. Downtime and service outages are inevitable, but the success of the service desk depends on how well it can respond and mitigate the impact of fast-fail product development and service delivery initiatives.
- **Velocity, quality and performance.** The DevOps goals of velocity, quality, and performance are achieved when build iterations are released rapidly at high quality, reducing waste processes. Incidentally, this is also the purpose of reducing issue resolution time for service desk activities and processes including incident and problem management.
- **Continuous improvement.** Repetitive issues downgrade overall system performance and any possibility of resolving issues in a timely manner. Finding the problem's root cause and reducing repetitive issue resolution requests is a sign of continuous improvement.
- **Automation and intelligence.** To resolve issues quickly, service desk must gain end-to-end visibility and control into the IT network and assets. Advanced AI capabilities to eliminate alert noise and proactively identify problem root cause help reduce the issue resolution time, among other ITSM objectives.

By following the DevOps philosophy, service desk can achieve the wider ITSM objectives of efficiently and effectively delivering IT services. MTTR is one among many other service desk metrics that companies can use to evaluate for deeper insights into IT service management and operations activities. With any technology or metrics, however, remember that there is no 'one size fits all': you'll want to determine which metrics are useful for your organization's unique needs, and build your ITSM practice to achieve real-world business goals.