Command Center Handbook

Proactive IT Monitoring

Protecting Business Value Through Operational Excellence

Abdul A Jaludi

abby@tag-mc.net

www.tag-mc.net

Chapter Five

Command Center Interactions

Monitoring and incident, change, and problem management are very tightly intertwined and must work closely together to ensure issues are prevented or corrected as quickly as possible, and that they do not recur.

Monitoring process

Monitoring should be proactive, not reactive. Its purpose is to sound an alert when something has the potential for failing, not to notify when something breaks. It alerts the right people that a failure may occur if an action is not taken to correct the event that just occurred.

Early warning alerting and monitoring can be done in various ways and through numerous tools. One way is by setting thresholds so that alerts are generated when an event reaches a specified value. For example, an alert would be generated if the hard drive on a server with 500 MB of space gets down to 75 MB free, in other words if 85% of the available space is used. The first alert would be sent to the command center (and hopefully the support staff) when available space on the drive reaches 15%. When the first alert shows up, someone should be dispatched to investigate and correct the issue. If the incident is not corrected, a second alert would be generated and sent when available space reaches 10%. A third alert would be sent when the available space reaches 5%. If the condition is not fixed, when the available space reaches zero, the server will crash and your customers will be unable to perform the functions that generate your income. If you let that happen often enough, your customers will take their business elsewhere.

Some support team members will want to know when available space reaches, let's say 15%, but will not act until it reaches 10%. They will instruct the command center member calling them to ignore the alert and to call back when available space reaches 10%. That is a mistake, and any request to ignore an alert should be rejected. Situations like that can be addressed by sending the 15% alert only to the support team and the subsequent alerts to the command center. Ignoring alerts on the command center monitoring screens for any reason is a recipe for disaster. Before long, the monitoring screens will be flooded with non-actionable alerts, increasing the likelihood for a real alert to be missed and an outage to occur.

Available space on a hard drive is one example, but the monitoring team will receive hundreds of types of alerts. Everything that might affect the integrity of the domain for which the command center is responsible must have a way to send a notification if a potential failure may occur, before the situation becomes critical.

Another way for generating early warning detection alerts is through applications internal error reporting routines, picking up critical messages and displaying them on the monitoring screens. Those types of alerts are implemented with the aid of the application development team.

Incident management process

An incident management team with a properly defined and executed process helps to ensure problems are well-documented, broadcast to all required parties, and resolved as quickly as possible. One of the most important functions of the incident management team is to document every action taken during the incident management process. Who is called, who responds, who does what, and what happens during the process must all be documented in the trouble ticket that's created for each specific event.

Never, ever combine the incident management and problem management processes within a single team. It is a major conflict of interest and a recipe for disaster! In fact, the problem management function should not fall within the command center but should be an autonomous team reporting to a senior level in order to prevent any conflict of interest between it and the groups with which it interacts.

In addition to managing the incident resolution process, the incident management team also performs monitoring, usually at a high level that allows them to correlate global-type impacts to a specific event or problem, helping to prevent global issues or to reduce outage duration.

Change management process

Command centers by their nature are heavily involved in the change management process. That happens for two reasons. First, in many cases the change control teams operate only during business hours, so the command center management assumes some of the oversight function during emergencies in order to resolve an ongoing incident that affects users. Second, many of the changes will be performed on systems they manage, and as a result they're an interested party in the success or failure of those changes. In some cases, command center staff involvement is required in order to implement certain changes.

It is in the best interests of the command center to develop a healthy relationship with the change management teams. There are many instances where the watchful eyes of an alert change management staff member catch pending changes that may negatively impact a managed system. Another benefit of a maintaining a healthy relationship is that the change management teams can help implement and enforce standards defined by the command centers through process improvement initiatives.

Problem management process

A well-implemented problem management process helps to ensure problems have a permanent solution and actions have been taken to prevent the problem in the future or to resolve it quicker if there is a recurrence.

As part of the problem management process, tasks may get assigned to the event management team to implement monitoring or correct alerting for problems that could have been prevented if the proper monitoring was in place.

Tools and procedures should also be in place for all problems and corrective actions to be documented and available for the incident management team to utilize in order to reduce outage durations for known errors.

A good practice is to use ITIL recommendations as a basis to build on for each of those processes.

> ITIL (IT Infrastructure Library) is a framework of best practices for the five core processes used by information technology to identify, plan, deliver, and support IT services. The five core processes are service strategy, service design, service transition, service operation, and continual service improvement. Additional information can be found on the ITIL website: http://www.itil-officialsite.com

Do's and Don'ts

Do have well-documented procedures for the command center's role in monitoring, change management, disaster recovery, escalation, incident management, security management, event management, and problem management. Those procedures must be followed by everyone. There should be no exceptions, unless specifically accounted for in the procedures.

Do have well-documented guidelines for the different severity levels. They will determine how each outage is treated, the types and frequency of notifications, and the escalation points. Ambiguous definitions may result in insufficient notifications and escalations leading to extended delays in resolving a major problem. Frequent reviews should be performed to make sure the guidelines are properly implemented.

Do centralize all of your monitoring. All systems and applications that are critical to the operation of the businesses and divisions supported should be monitored centrally by the command center. That allows the command center to quickly determine whether a problem is global in nature and speeds up the recovery process. Knowing which alerts are outstanding when a major problem occurs helps to identify the proper support teams to engage and reduces the length of the problem.

Do standardize your monitoring tools and create best practices profiles so that certain functions can be fully automated, such as the addition of monitoring on a new production server.

Do act on every alert. Every alert that the command center receives should be acted upon. If a support staff member says to ignore an alert for whatever reason, the response should always be no. The alert should be documented with a corresponding trouble ticket. That should be a onetime-only alert. The support teams must correct the alerting system so that the alert is triggered only when there is an actionable issue.

Don't allow individual teams to utilize stand-alone products if there is a standard monitoring tool defined for a given platform.

Don't allow manual checks of an applications function. Relying on someone to manually check application functionality causes delays and has the potential for extended outages. Operators should be automatically alerted when an exception occurs that may impact application functionality.

Don't let support teams perform their own monitoring exclusively. Some teams may want to perform their own monitoring; that is not acceptable. It is OK if they want a view of what the command center sees. It is not OK to let them perform monitoring instead of the command center. Departments and teams that perform their own monitoring with no oversight tend to hide their dirty laundry. They also tend to apply temporary fixes. As a result, you will have more outages, longer outage times, and a higher percentage of recurring outages. Because there's no oversight, senior management is usually unaware of the number or scope of the outages. Customers usually take their business elsewhere and senior management is left wondering why they did so.

> When the command center performs the monitoring, it serves two purposes.
>
> First, the command center is able to quickly determine whether a problem in one area is affecting other areas. If a communications line goes down and several businesses suddenly go red, command center staff will know which support teams to notify first. A team monitoring its own area may not be aware of the communications failure and will spend valuable time diagnosing the problem.
>
> The second purpose is transparency. Teams that are being watched by others tend to run better, with fewer problems.

When something breaks, they tend to find a permanent solution so that the same issue doesn't recur.

Don't ignore alerts. Ignoring alerts is a sure way of getting into trouble. At some point a mistake will be made and someone will ignore the wrong alert. Avoid the problem altogether by making sure non-actionable alerts are not even seen.

Don't let the monitoring screens get flooded with alerts. A monitoring screen that's flooded with alerts makes it very easy to miss a critical alert. Alerts need to be acted upon and corrected as soon as possible. It is not acceptable to let alerts sit in the queue until the support teams' regular work hours. If the condition is acceptable during off-hours, then the alert should only be generated when it is not acceptable and when someone is available to work on the problem. Alerts that don't get addressed within a specific timeframe should get escalated to the next level of management.