REACTIVE VS PROACTIVE PROBLEM MANAGEMENT

Problem management in IT is rarely discussed, but it is certainly practiced daily—in a variety of ways, some of which are successful. More often, problem management looks like a group of sys admins arguing about who's to blame for the latest episode of company-wide slow-down.

When done well, however, problem management has the potential to catapult the IT unit from a fire-fighting position to one that offers a clinical focus on improvement and innovation—precisely the value and ROI that your company expects from IT. In fact, the way IT goes about identifying, managing, and eliminating problems plays a major role in how the unit is viewed by other business units and the company at large. Atlassian reported that high-performing IT teams are nearly 2.5 times more likely to practice problem management proactively, instead of waiting to put out fires.

In this article, we'll take a look at problem management and compare reactive and proactive approaches.

What is problem management in IT?

We already know from ITIL that any problem is an underlying cause of one or more incidents. Problem management, then, refers to how you manage the lifecycle of problems. IT can approach problem management in two ways: reactively or proactively.

- **Reactive problem management** is concerned with solving problems in response to one or more incidents.

- **Proactive problem management** is concerned with identifying and solving problems and known errors before further incidents related to them can occur again.

Both approaches are key to ensuring a holistic and comprehensive tackling of the underlying issues that negatively impact IT services, but it is the reactive approach that is usually the first port of call for most support teams. Balancing the two approaches must be ingrained throughout your organization and should be one of the leadership's imperatives.

Defining reactive problem management

Reactive problem management is triggered directly after an incident that is deemed worthy for a root cause investigation, such as one major incident or a series of incidents which are significant in totality. It complements incident management by focusing on the underlying cause of an incident to prevent its recurrence and identifying workarounds when necessary. Reactive problem management considers all contributory causes, including causes that contributed to the duration and impact of incidents, as well as those that led to the incidents happening.

The swarming technique is a strong approach in reactive problem management: different units come together to examine an incident, then brainstorm and identify the source and the potential root causes. Take, for example, an application that has crashed. Incident management would restart services that have stopped or reload a recent version, while reactive problem management would investigate the source of the crash by analyzing logs or getting information from a developer or vendor. The problem would be logged as a direct reference to the incident and workarounds, as identified by incident resolution, would be documented alongside it. If the fix requires a patch, then change management process would be used to permanently resolve the problem.

Other techniques for reactive problem management include chronological analysis, Kepner and Tregoe, 5-Whys, and fault isolation.

One of the main drawbacks of reactive problem management is its defensive nature, not unlike closing the gate after the horse has bolted. Secondly, technical teams are usually under pressure to find the incident's root cause instead of focusing on restoring service as quickly as possible. However, the benefits of reactive problem management are clearly visible to stakeholders once it is proven that a fix, whether permanent or temporary, will prevent recurrence or reduce impact should the incident resurface.

Understanding proactive problem management

Proactive problem management is driven from a continual improvement perspective. The trigger is not the result of an active incident, but rather the result of identified risks to service. These risks may include warnings, errors, or potential breaches to thresholds that indicate potential problem areas. As such, proactive problem management activities take place as ongoing activities targeted to improve the overall availability and end user satisfaction with IT services. The main techniques of proactive problem management include trend analysis, risk assessment, and affinity mapping.

Let's use the same example as above to demonstrate proactive problem management. The monitoring unit detects errors in the application—they aren't causing downtime, but they may indicate problem areas.

The sys admins take time to document the errors and research potential causes. This may indicate

that that the errors occur whenever the application calls to a particular database, routed through certain interfaces. The sys admin can elevate this issue to the network admins and the database admins, who can then identify the exact issue and shut down the effected interfaces, ending the errors. Depending on the situation, the admins may opt to reconfigure the ports or replace the affected components in order to permanently eradicate the problem before it becomes serious.

The clearest benefit of proactive problem management is a significant decrease in the number of critical incidents. An IT team can never prevent all incidents, so reactive problem management is something all teams will have to deal with. However, proactive problem management is the mark of a truly mature IT unit.

Putting in place metrics that measure proactive problem management and placing a reward on the same from an innovation perspective will serve to motivate IT to focus on such opportunities. Interestingly, proactive problem management can have a negative side effect, at least from an IT marketing perspective: the company may not fully appreciate a problem that was addressed as it never caused an issue in the first place.