Be Your Company's IT Hero: Be Proactive - Not Reactive

IT heroes learn to identify where IT operations incur risk, and take steps to mitigate

Everyone in IT should know how much their specific organization has at risk from IT service outages. Figuring this out can be illuminating – bespoke numbers can be deeply scary, and are much harder to dismiss than de-contextualized industry analyst estimates (e.g., average cost of an IT outage = $5,600 USD/minute).

Disciplined risk calculation compels breaking down silos between IT and other parts of the business (e.g., Finance, Legal, etc.) and forces leadership to acknowledge the enterprise's absolute dependence on IT for continuity (never a bad thing, though coming into the spotlight certainly raises the stakes). Within IT, it compels connecting the operations playbook, and its model(s) for IT risk with a fully-fleshed-out picture of business downsides, resulting (usually) in a much-improved understanding of where risks lie, and a clarified mandate for risk mitigation.

Though IT wonks might naively think that such calculations are regularly performed up in the C-suite, perhaps in collaboration with insurers, legal counsel, risk analysts and other experts, that's not the case for the majority of large enterprises. According to a fairly recent survey report from North Carolina State University's Enterprise Risk Management (ERM) Institute, only 30%-40% of enterprises are applying a fully-baked ERM process internally. The rest are treating risk more or less aspirationally: biz leaders are aware of the problem, know they want to hire more IT risk and security pros, etc., but are still, themselves, unsure about how to couple risk with strategy, how to compensate risk professionals, and of the overall value of ERM to risk reduction.

Help a Business Leader Out!

Given all the above, it might freak these folks out to understand that most IT organizations have a largely reactive posture with respect to risk. In other words, a lot of us are basically waiting for outages to occur before dealing with them.

Obviously, this doesn't mean we're not working hard to reduce risk. Much of the complexity we're all dealing with, these days, involves reducing risk exposure: by building highly-available and fault-resilient services, for example, or by exploiting far-flung datacenter locations, public cloud regions, carrier footprints, etc., to make services available despite localized issues. And of course, the constant round of patches and updates and DNS tweaks and access controls and storage encryption configurations and backup schedules designed to ensure the integrity of data at risk and in motion.

But there's lots more IT Heroes can, and should be doing, here. Both because it reduces our chance of being held at fault if issues occur, and because being known as a business-risk reducer (which is how you should be thinking about this stuff) can get you promoted.

IT Risk for IT Heroes

Fun fact: Most IT outages happen because someone screwed up a config. (Or at least, that's what happened to these amateur Bozos.) For this reason, the sad and scary fact is that it's not possible to predict many outages. What's possible (see below) is to limit the damage misconfigurations can do, and also speed up

your team's ability to recognize impending failure and intervene to prevent outages outright, or limit their scope.

The riskiest way to make configuration changes, of course, is when you do it manually. Three key IT Hero to-dos emerge from this understanding:

Identify any production systems on which manual changes are being made, discuss these beloved "pets" openly, praise their long-suffering owners, and schedule them (the workloads, I mean, not the owners) for dismantling as quickly as possible. Replace them with "cattle" VMs, container flights, public cloud services, etc. – vehicles for service delivery identified by the tattoos in their ears, whose passing you will not mourn.

Automate everything, so you never again need to make a manual change. While automation doesn't magically prevent config errors, it systematizes change-making, enables its rigorous documentation, provides (or gateways into) the "single source of truth" for configuration, enables easy testing of what are effectively "production" deployments in staging, vastly facilitates rollouts and testing in production (e.g., green/blue, canary, etc.), and (most important), enables rapid rollbacks when things don't work out well. Bottom line, automation speeds up deployment of the above-described "cattle" workloads, so that, when they fail, you can shoot them in the head without qualm. It also enables ubiquitous monitoring.

Discover and monitor all the things. Monitoring gaps are recognized by Google, among other amateur Bozos, as a major contributor to cascading failures.

Create and task a team to track and maintain systems preventively, and resource them to succeed. Recommended maintenance of IT systems (e.g., databases), orderly retirement of legacy hardware, replacement of components after a given number of power-cycles -- all this work needs to be done on schedule and tracked scientifically to continually improve your model of how systems/infrastructure fails at your organization (and explore the limits of that model to make operations more cost-efficient). Even if you're running on clouds, you need to pay attention to service generations, region histories, provider SLA compliance, etc., and keep your workloads on what (for you) represents the best compromise between resilience and cost efficiency.

Beyond this, your constant task is to understand better how systems fail, and apply that knowledge rigorously in your organization. One of the best introductions to this topic is a two-year-old talk by Luke Stone, Director of Customer Reliability Engineering at Google. Strongly recommended, along with Google's SRE Book and its associated workbook, which goes deeper into the mechanics of SLIs, SLOs, making things observable, and some of the key characteristics of impending application failure.

How does this Connect with Monitoring?

Here are a few ways:

Enterprise-class monitoring comes to the table with much understanding of impending application failure built right in. When you implement pre-packaged monitoring of, for example, a SQL cluster, a mature enterprise monitoring system will have included the metrics and thresholds DB makers have determined give the most-complete and actionable picture of system health. So, simply put, you can (after appropriate testing) rely on the monitoring to tell you when to proactively intervene: for example, to take a somewhat-malfunctioning node offline (which, of course, you've already tested doing) so that it doesn't cause cascading failure.

Enterprise monitoring shows you the full stack. So you get indication of issues with hardware and/or fundamental resources (e.g., CPU, RAM, storage, etc.) as well as OS, virtualization, and application layers above, and can drill into what's happening to determine root causes of issues.

Enterprise monitoring notifies your team and helps you escalate. It also – if properly configured – minimizes false positives and helps prevent alert exhaustion.

Enterprise monitoring helps you visualize impending failure before it causes a service outage. Features like Opsview Monitor's Business Service Monitoring let you model the interdependencies and behavior of resilient infrastructure blocks, like load-balanced server tiers and clusters, and shows whether (and how much) the inevitable failure of lower-level sub-elements impacts availability of your applications. If outages don't affect availability, you can go back to sleep and deal with it in the morning.

Enterprise monitoring lets you send data to external tools, like data analysis, SIEM, and machine learning platforms, that can discover patterns and clue you to security anomalies and incipient failures.