MARCH 2015

New Thinking for Old Problems:

The Executives' Guide to Common IT Issues

FORWARD

Modern company IT responsibilities have changed dramatically in the last 20 years. Where IT used to be a support function to the business, now in most businesses, IT is a critical part of the system. Gone are the days of waiting till morning to fix a problem with a database, network or server. Today's IT teams are on the front line 24x7.

The following guide provides high-level suggestions from professionals that have been in the IT support business for over 20 years. The first section contains some general advice for IT Managers. The second talks about the challenges in dealing with incidents that are now all too frequent in modern IT infrastructures. The third section gives some tips on optimizing response to issues to reduce time to resolution.

While this guide is not all-encompassing, it addresses common rules of thumb found in most modern businesses and how to avoid those issues.

TABLE OF CONTENTS

FOCUS ON WHAT YOUR COMPANY DOES BEST & LEAVE THE OTHER STUFF TO SOMEONE ELSE.

Historically, companies have built their own on-call response systems in-house. The reason for this? There was no product available for purchase in the market to solve the problem.

Building your own solution to manage critical alerts and on-call processes may not be the best use of your company's limited resources. Internally-developed solutions are typically very expensive to build and support, even if they are quite simple. Even if you build your own system, it would unlikely record the statistics you need to help manage teams & improve process.

According to **The 2014 State of On-Call Report**, 70% of respondents use homegrown tools or processes to solve on-call problems. While the vast majority of respondents had homebuilt systems, they were not happy with them.

If you're looking for a better solution, **we can help**.

MAKE SURE YOUR IT TEAMS KNOW THE GOALS OF THE BUSINESS.

Much of what an IT person does (architecting networks, building data centers, running email servers) tends to look superficially similar from one company to the next. And an IT person's job is often sufficiently abstracted from a company's line of business that it feels unrelated.

But the attitude hurts Operations teams in two ways: New business initiatives can catch IT flat-footed and unprepared if the team isn't paying attention. And assuming that a "cookie cutter" approach to infrastructure architecture and development is best, means missing out on opportunities to do things more efficiently, or to embrace new technology and approaches that can give your business a competitive advantage.

Your IT team is likely the backbone of what makes your business run. If they're not connected with the larger company goals, you're missing an opportunity to break down silos, increase transparency and get your most vocal technology advocates on board with what your business is trying to accomplish.

YOUR UPTIME SHOULD CORRELATE WITH YOUR BUSINESS NEEDS.

It's not uncommon that the technology parts of the company are sometimes disconnected from the business side. However, when your company's revenue is closely coupled with the performance of your website, it's best to make sure everyone understands the common goals.

In order to have a real uptime strategy, it's important for engineering, management and business management to agree on what constitutes downtime, how it's measured and what the expectation is around it.

REALIZE THAT A NEW DASHBOARD ISN'T ALWAYS THE ANSWER.

"A dashboard is just another way to admire a problem."

Dashboards deliver general situational awareness, which is great for teams. But often they show the "state" of the system rather than the "events" in the system.

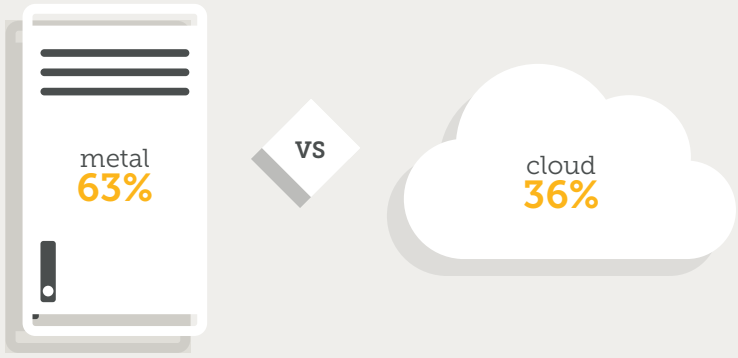
Don't use a dashboard that simply shows busy work but rather choose one that simplifies complex relationships in data that can help to inform team members. If your dashboard displays are not synthesizing multiple pieces of data to draw new results, then take another look.

For instance, rather than just showing user load over the last 12 hours, show user data over the last 12 hours as compared to last week at the same time. This provides a view that team members would not normally be able to process in their brains automatically.

USE THE RIGHT ARCHITECTURE FOR THE RIGHT PROBLEM.

While there are many benefits of using the cloud (faster time-to-market, streamlined processes and flexible infrastructure costs), it may not be better for every task that your IT infrastructure needs to perform.

For example, some highly-transactional businesses, like Ad Tech, suffer performance issues in the cloud. Taking a hybrid approach - with a mix of old-school hardware & new-school cloud options - might actually be the best choice for some architecture and companies. The appeal of PaaS is strong, but optimizing architectures is always a good idea, especially when the business depends on it.



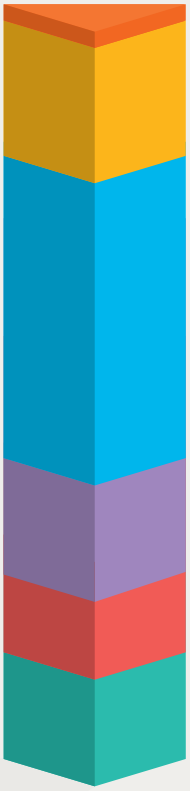
According to **The 2014 State of On-Call Report**, the majority of respondents are still using physical infrastructure but more are moving less performance-oriented systems to the cloud every year.

WHEN AN INCIDENT STRIKES, DON'T ASSUME THAT ALERTING WILL SOLVE THE PROBLEM.

If you break down a typical "incident resolution" into phases, you see that generally a small portion of time is spent "being alerted to the problem". VictorOps internal data shows that, at most, 5% of the total TTR has anything to do with alerting or escalation of problems. There are incidents where a team member does not respond but this is generally more about the team member than the platform finding him or her.

The Alerting phase historically has been a longer portion of TTR back when teams actually carried "pagers," as those systems were quite slow. Now that team members have smart phones, human behavior has changed to be more "always-on" and engaged with that device. Alerting has come along for the ride with more ubiquitous connectivity and overlapping networks of WAN, LAN and SMS data.

Find a solution that supports all aspects of the incident lifecycle and can accurately track Mean Time To Resolution. You can't optimize what you don't have insight into.



INCIDENT LIFECYCLE

- 5% Alerting
- 18% Triage
- 40% Investigation
- 15% Indentification
- 10% Resolution
- 12% Documentation

Estimates based on data from current VictorOps customers

BE OKAY WITH NOT SOLVING THE PROBLEM – RIGHT THIS SECOND.

With everyone these days measuring downtime in seconds, it seems obvious that if something is wrong with your platform, you should do anything to end the downtime as quickly as possible.

However, there are a few situations where it might not be in your best interest to rush in:

1. **THE PROBLEM IS NEW OR UNIQUE.**
   If you haven't seen a particular problem before, giving your server a quick reboot might clear up the problem, but it might also destroy any forensic evidence that would give you a hint about root cause. At least take a moment to get a handle on the state of a system before "pushing the button."

2. **YOU'RE CONSIDERING A "QUICK FIX" OR A "HACK".**
   Doing a "band-aid" fix may get your site up in half the time, but if it fails again a week later, was it worth it?

3. **YOUR FAILED COMPONENT IS PART OF A CASCADING FAILURE CHAIN.**
   If a given dies because of excessive load, and is restored to service without addressing the load that caused the problem, then you could cause an immediate repeat of the failure, or a failure of a downstream component. If you don't know, or haven't addressed, the root cause of the problem, then you're taking that risk.

DON'T WAKE EVERYONE UP WHEN THERE'S AN ISSUE.

Sometimes the problem is so serious that, even when the root cause is found, it's clear that it will take hours or even days to get it fixed. A great example is database corruption. You may have a backup but when that turns out to be corrupt also, you may now be looking at a longer unexpected outage.

The 2014 State of On-Call Report found that, in most companies, it takes an average of 5 people to solve an incident.

Get enough people on the problem to fix the problem, and let everyone else keep on with whatever they're doing. When you reach hour 12 of the recovery process, you'll be grateful that there are fresh, rested people ready to take over and give the first responders a break.

All you need to resolve an incident, in some cases, is getting the right eyes on the problem. If you have an on-call solution that provides context around an incident within a mobile setting, then remote collaboration (for all) becomes not only possible, but highly effective.

GET YOUR DEVELOPERS ON-CALL.

Why should Ops be the only ones woken up at 4am because there's a problem in production? With the advent of DevOps practices, it's now becoming more popular to include devs in the on-call rotation.

Complex IT systems are hugely multidisciplinary in nature. It has become unrealistic to expect a frontline IT person to be able to debug the majority of problems and as a result, on-call responsibilities are reaching deeper into the organization. This is a trend that will continue.

There are many ways you can introduce the concept of expanding on-call coverage, everything from pairing people of different disciplines together while on-call to making sure that the on-call person knows exactly where to find the runbook for dealing with problems they may not be able to solve.

See how other companies are **successfully making the transition** to putting devs on-call.

STOP ALERT FATIGUE BEFORE IT STOPS YOUR TEAM.

Alert fatigue can happen when someone is exposed to a large number of frequent alerts and therefore, becomes desensitized to them. This can lead to a longer response time or to missing important alarms altogether.

In the 2014 State of On-Call Report, 63% of survey respondents reported that alert fatigue was an issue. Additionally, 64% believe that up to a quarter of all alerts are false alarms.

Too many alerts can actually lead to being less prepared for an outage because false alarms rob the valid alarms of the value they were intended to add. Ever heard of the boy who kept crying wolf?

Companies need to have practices in place to review alerting and incident data in order to adjust thresholds. This lessens the load for everyone. False alarms do not go away — they get worse with time.

USE SMART(ER) ALERTING TO SOLVE PROBLEMS FASTER.

In the heat of the firefight, it can be hard to find the resources you need. Sometimes it's an issue of not having the proper access to internal documentation but oftentimes, it's a matter of not having any updated documentation available. How can anyone be expected to resolve an incident at 4am without the right help?

Imagine if you could add annotations to alerts so that the first responders have the exact information they need, exactly when they need it. The documentation necessary to solve the problem (whether that's a link to a runbook, a Graphite graph or simply a note explaining what to do) is attached to the alert, making for happier on-call engineers and a reduction in your TTR.

See for yourself how VictorOps can make on-call suck less.

RESOURCES

http://www.rackspace.co.uk/cloud-computing/myths

http://victorops.com/knowledge-drop/guides/call-firefight-survival-guide/

http://victorops.com/knowledge-drop/devops-docs/putting-devs-call-empower-team/

http://victorops.com/knowledge-drop/reports/state-of-on-call-2014/

http://attackwithnumbers.com/the-laws-of-shitty-dashboard