

Be Your Company's IT Hero: Optimize Ruthlessly

Iteration, measurement, and reflection help IT teams make consistent progress towards goals

Almost nothing (kittens and chocolate ice cream being the only known exceptions) is good from the very start. Much more often, good is the eventual product of doing things over and over, accumulating gradual improvements, integrating feedback, test results, and other data. The old joke: "How do you get to Carnegie Hall?" (Answer: Practice, practice, practice!) has a required corollary: it's the third "practice" that really starts moving you forward.

Optimization isn't just about iterating, either – especially in IT. It's also a concentric process, where improvements radiate outward from the deliverable (application software, deployment, whatever) to inflect surrounding stuff like documentation, sample configurations and other worked examples, automation, monitoring detail, associated procedures, etc. – all of which influence broader consumability, utility, and impact.

IT Heroes need to cultivate the willingness to optimize ruthlessly and continually in order to push back against daunting complexity and constant change. And they need to know how to optimize strategically, investing cycles for maximum ROI.

Optimization for IT Heroes

Here are some tips for IT Heroes looking to step up their optimization game:

Monitor, monitor, monitor. Improving products and processes begins with measuring the characteristics you want to improve. The first generation of workers and theorists on process efficiency conducted what used to be called "time and motion studies," where they followed workers around with stopwatches and noted times on (paper) spreadsheets. Today, at least in IT, that's not usually needed, since we all work inside tools that



can (if properly configured and used) track what we're doing. The systems we work on, meanwhile, should be monitored as a matter of course, and the monitoring itself should be subject to fairly frequent review and optimization.

Don't monitor naively. MIT's Katie Bouman is much in the news these days because her algorithms recently enabled creation of the [first-ever visualization of a black hole](#): an amazing achievement, credit for which she graciously and appropriately shared with her team. But online jerks used her Github commit record as evidence that she hadn't contributed much to the project. Her colleague Andrew Chael, of Harvard, who manages the project repo and was primary developer, wrote an [impassioned series of Tweets](#) chastening the community for this toxic, sexist dumbness, and setting the record straight. Your takeaway is to not be these tech bros: measure the right signals and be sure you're drawing the right conclusions from them.

Exploit ratchets to incrementally claim and hold territory. Converting static deployment configurations to active automation and using them in production solves problems, provides you with tools (saving time, money, attention, etc.) and preserves information painfully gained through experience. Writing coherent documentation from a perspective of empathy for end-users does a similar job. In general, such encodings work as ratchets to preserve hard-won forward progress. Bonus: they

“

IT Heroes need to cultivate the willingness to optimize ruthlessly and continually in order to push back against daunting complexity and constant change.

”



can be stored in generalized, human-readable ways in code repositories, where they can be socialized, commented on, forked, and pull-requested, facilitating web access for readers and downloadability for code-consumers.

Always pluck low-hanging fruit and beware letting the perfect become an enemy of the good. Be sure you're spending time and energy wisely, by first improving what's easiest and will produce the largest gains, leaving more incremental gains and harder-fought battles as "nice to haves." A working solution (ideally, working code) is always better than nothing, and in many cases, the MVP is good enough to stick with until compelling reasons emerge to change it. (Also remember never to mix metaphors.)

How Does This Connect with Monitoring?

Monitoring is your main tool for measurement, hence for optimization of everything running in your hybrid IT estate. In addition, mature enterprise-class monitoring offers administrators many features and degrees of freedom for optimizing monitoring itself: improving the quality of data gathered, its utility, and your own ability to visualize it.

Many fine-grained adjustments. Mature production monitoring solutions may support many thousands or tens of thousands of service checks, each with notification thresholds and other details that can be changed to suit the requirements of specific target systems or desired measurements. The best platforms also support comprehensive automation, making it easier to capture minute configurational optimizations and propagate them efficiently over thousands of hosts.

Central take-off point for analytics. Your monitoring solution is collecting large numbers of performance-related data points, constantly. The best solutions make it easy to export this data to log servers, search and visualization platforms like Elastic Stack, and data analysis platforms like Splunk Enterprise, all of which can provide insight and additional value.

Sophisticated visualizations. As noted in previous solution briefs, a mature monitoring solution offers a range of easy to use tools to build dashboards, generate reports, and model systems in ways that let operators focus on and find important information, while avoiding distractions.

[FIND OUT MORE](#)

USA: +1 866 662 4160

sales@opsview.com

EMEA: +44 1183 242 100