Your non-linear problem of 90% utilization

"You've got a breakfast meeting, meetings all morning, two lunch meetings, your afternoon meetings, a dinner meeting, and then it looks like you've pencilled in 'self-immolation' after that?"

credit[1]

Suppose a web server is running at 50% of its full capacity. Browser traffic doesn't arrive in regular, smooth amounts; it comes in spurts and occasionally large spikes. Because the server is under-utilized, when a spike arrives there are spare resources to handle the increase. If the spike is sufficiently large, performance will degrade, and if larger still, many of the requests will be rejected rather than answered; after all, there's *some* limit past which the server cannot do any additional work.

Now suppose someone looks at a report that shows "50% utilization" and says "Hey now, this is a server, not a person! It costs us the same whether we drive it at 50% utilization or 90% or 99%. So let's get our money's worth and drive it into the 90s!"

What happens? Even normal variations in traffic will drive the server past its capacity. The average time to respond to a request will skyrocket, and often requests will be dropped altogether. Not due to an unusual event, but all the time. **The system is now brittle**—not good for costs, not good for the quality of the product or customer experience—just bad all around.

Maybe we can drive high utilization by having multiple servers work as a team. Suppose we have three servers, all serving traffic for the same website, all at 70% capacity. That sounds like a happy medium between 50% capacity (wasting money) and 90% (brittle). The total amount of utilization is 2.1 servers (3 x 70%), so we're nicely over-powered for traffic spikes.

But what happens when one server runs into problems? Suppose it crashes, or the power in its data center cuts out, or someone else breaks the network with a glut of garbage traffic. The 2 remaining servers now have to deal with 2.1 servers' worth of traffic. Both are at 105% capacity, and we're back to broken and brittle.

This isn't really about servers; it's about you and your teams. It's about how your "busy" life not only diminishes your productivity, but how your whole team is hectic, yet bringing itself to a crawl.

We all have a capacity, whether you want to measure it in hours, in energy, in focussed attention, or if you don't want to measure it at all. Instead of web-requests, we have life-requests, whether those are in-bound emails, Jira tickets, Zendesk tickets, Salesforce leads, requests from co-workers, requests from a friend, or families that need our time and attention even more than they need our paycheck.

90% utilization is causing more failure than you realize, not just in burn-out, but in productivity and output. Of course you'll burn yourself up, sacrificing sleep, health, friends, family, and other things you mistakenly take for granted, but I suppose you knew that already. You're trading that for super-human productivity, right?

But you won't even receive outsized professional gains as a reward. **This condition is a combination of frequent context-switching and interruption—the Twin Enemies of productivity.** Work-completion will drag out because it's constantly interrupted. Some will be abandoned.

"I'm sorry, I was sure there was something over there that I needed to bark at. Please continue."

credit[2]

Worse, in many organizations *everyone* is operating at 90%, which then reacts like the three-server system, where the inevitable hiccup from any one person causes a ripple effect that hurts several other people or projects. Since *they* are over capacity, rather than absorb the spike, they too will ripple the problem to others—a cascade like the run-away chain reaction of an atom bomb.

The key word here is *inevitable*. People get sick or die or leave or change or have to run an errand or want to do even one minor piece of work that wasn't mapped out weeks in advance. True emergencies arise that deserve to interrupt work. This is not something you can "architect out" of the universe; rather, you need to build a system that assumes variation and interruption, and design your personal and team's work-style to be resilient to that variation.

The ideal is probably a situation where most of the time you're in the safe zone, with occasional surges into high gear **for a short period of time** and **for good cause**. For example, a brand new product launch is usually attended by some extra time fixing bugs, especially post-launch where it hits real customers and a few issues are discovered that we all agree should be fixed swiftly before more customers encounter it. Or there could be a clear-and-present danger to the company that requires a special, time-bounded rally. Or you could use infrequent and brief surges in a fun way, like a Hack-a-Thon or a Bug Squash Competition or a Ticket Kill Day.

We're erring on the side of over-utilization, and rather than providing the benefits of competitive advantage through higher productivity, it's creating needless turmoil and lower productivity.

Don't let yourself, or your team, fall into the trap.

The current version of this article:

https://asmartbear.com/utilization/

More articles & socials:

https://asmartbear.com

References

1. https://andertoons.com/meetings/cartoon/7019/youve-got-meetings-and-then-pencilled-in-self-immolation
2. https://andertoons.com/dog/cartoon/6812/i-was-sure-something-over-there-needed-to-bark-at