Unplanned Work

The Silent Killer

Introduction

You can't see it. You can't smell it. But it's deadly, and it may be in your IT organization's basement, silently killing your company. It's called unplanned work, and CIOs and CISOs are losing their jobs because of it.

Kevin Behr, co-author of *The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps*, and I recently discussed unplanned work and how it affects an IT organization. This silent killer is so hard to recognize that many IT professionals don't even realize it exists. Some will challenge, "If things are really so bad, why aren't airplanes falling out of the sky?"

It's a fair question. In effect, they are asking, "If unplanned work is so deadly, where are all of the dead bodies?" And the answer is, like in the movie *The Sixth Sense*, the dead people are everywhere. And once you see them, you'll see that they are one step removed from the root of virtually all IT problems.

It's difficult to overestimate the impact of unplanned work on an IT organization. Here's a back-of-napkin calculation: According to a Forrester Research estimate from 2002, 10% of the U.S. gross domestic product is spent on IT, comprising 50% of company capital expenditure. But IT projects are like a free puppy—the lifecycle cost of the puppy is dominated by the "operate/maintain" costs, not the initial acquisition costs. The U.S. GDP in 2004 was approximately $10 trillion; if 10% of that is spent on IT, and suppose we conservatively estimate that 50% of that IT spending is on "operate/maintain" activities, and that at least 35% of that work is unplanned, that's $350 billion. That's a lot of dead bodies. (For many, the IT controls work for Sarbanes-Oxley Section 404, which AMR Research estimates will exceed $6 billion in 2006, is an unplanned activity—more dead bodies.)

What is the precise definition of unplanned work? It is any activity in the IT organization that cannot be mapped to an authorized project, procedure, or change request. Any service interruption, failed change, emergency change, or patch or security incident creates unplanned work.

Why is it interesting to know how much unplanned work you have? Because it's a remarkably accurate indicator and predictor of IT effectiveness. In 2002, early in our research of high-performing IT organizations, Kevin Behr and I developed a 75-question assessment to determine whether an organization is high-performing or not. We look back at this assessment now with some embarrassment, because now we believe we can make conclusions about an organization's maturity and their needed prescriptive steps just by asking one question: What percentage of your IT organization's work is unplanned? We found that those organizations that spend less than 10% of their time on urgent and unplanned work also usually have extremely high levels of operational excellence, compliance, security, and have a good working relationship with auditors.

The dead bodies of unplanned work are everywhere. While CIOs aspire to focus on strategic issues, they must master the tactical issues, because operational unplanned work comes at the expense of strategic planned work[1].

Measuring Unplanned Work

Before we begin measuring unplanned work, we must first define the goal of an IT organization. A common view is that IT has two business functions:

1. Build and complete new projects for the business: The ideal IT organization is completing projects on time, with reliable quality, and delivering needed capabilities to the business. These IT projects are planned work, and anything that detracts from completing it is unplanned work. If developers spend 30% of their time on emergency break/fix issues escalated from IT operations, project commitments suffer, often resulting in late projects.

2. Operate/maintain existing IT services and assets effectively, efficiently, and securely: IT services are performing as advertised and promised, with a reliable level of quality; customers are satisfied; controls exist so that IT management detects variance early and can repair it in a planned and orderly manner; and controls exist to foster a culture of compliance, helping IT management achieve business goals and satisfy auditors.

[1] Hagerty, John, and Scott, Fenella: "SOX Spending for 2006 To Exceed $6B" November 29, 2005. AMR Research.

For the purposes of discussion, let's suppose that there are two extremes of IT organizations: high- and low-performing IT organizations. In our research, high performers have the lowest amount of unplanned work (less than 5%). Low performers typically have poor service quality, with constant service outages, break/fix work, and firefighting; unhappy customers that seem to see every mistake; and auditors constantly bombarding them with more documentation requests, tests, and archaeology projects—and of course, high amounts of unplanned work (often higher than 50%).

The sources of unplanned work are very different for high and low performers. I asked Bud Campbell (one of the consultants cited extensively in *The Visible Ops Handbook*, Greg Downer (principal consultant at Pepperweed Consulting, LLC) and Kevin Behr (CTO of IP Services and co-author of *Visible Ops*) for their top contributors to unplanned work in lower performing IT organizations. Their answers were virtually identical:

- Failed changes: The production environment is used as a test environment and the customer is the quality assurance team.
- Unauthorized changes: Engineers do not follow change management process, making mistakes harder to track and fix.
- No preventive work, making repeated failures inevitable: MTTR (mean time to repair) may be improving, but without root-cause analysis, the organization is doomed to fix the same problems over and over.
- Configuration inconsistency: Inconsistencies in user applications, platforms, and configurations make appropriate training and configuration mastery difficult.
- Security-related patching and updating: Inadequate understanding and consistency of configurations makes applying security patches extremely dangerous.
- Too much access: Too many people have too much access to too many IT assets, causing too many preventable issues and incidents.

On the other hand, for high performers, this group of consultants cites "product and environment failures" as the top cause for unplanned work in mature IT organizations. Campbell says, "Things break. This is a fact of life, and should be the only true cause for unplanned work." "Release failures" and "people mistakes/user errors" rate a distant second and third on his list, as these causes for unplanned work are much rarer in mature IT organizations. According to Campbell, "Mature organizations have proper checks and balances to keep these things from happening and catch them when they do. This is the linkage that tests the integration between process, people, and technology."

Quantifying Unplanned Work

Now that we've examined the two goals of an IT organization—delivering new projects and operating and maintaining IT assets—we can recognize how unplanned work detracts from these goals by pulling IT professionals away from activities that achieve them. But can we quantify the costs of unplanned work to justify the ROI of putting in controls to reduce it?

We sometimes hear the question, "How can you justify the cost of implementing IT controls? Show me a business case for us to buy testing servers and the tools to enforce our change management process." It's a fair question, and one that can be addressed with a simple example:

> Suppose someone changes an IT asset, but the change fails catastrophically due to lack of preproduction testing and change management authorization. The failed change results in an "all hands on deck" situation for the IT operational staff; IT drops planned work to remedy the results of the changes. The service disruption causes an incident that takes 4 hours to repair and involves 25 IT staff from all functional roles: application developers, QA staff, database administrators, network and system administrators, and security. Lost IT staff productivity is the first cost of this episode of unplanned work.

Unplanned work also comes at the cost of planned project work. In this case, the application developers and QA staff are taken from the critical path of an important sales support project, and the project ship date slips one week. Additionally, to address this project delay, IT has to employ a team of contractors longer.

The costs continue to mount. While the IT staff works to restore service, external customers call the service desk to find out why they can't access their billing information. Because of the large customer base, thousands of customers call the service center. The excess calls require the service center to activate the overflow call center, which costs tens of thousands of dollars. Revenue is also disrupted because the service center staff cannot take orders while processing the customer incidents.

Downtime and IT project resource costs run in the thousands of dollars; service center costs, lost revenue, and the delayed IT project costs are in the tens of thousands. Let's take it one step further. Maybe customers become so unhappy that 2% of them leave. The business now has to spend hundreds or thousands of dollars to recapture each of those customers.

Now that your single rogue change impacts customers, costs increase almost exponentially. With unhappy customers, you now have marketing and public relations problems. Your marketing department has to both gain new customers and win customers back—a feat more difficult and more expensive than gaining brand new customers. With any business process that is close to the customer, unplanned work can quickly and easily rack up huge costs. After looking at our scenario, how can you justify not implementing change controls and testing? Try the following exercise: Look at your top ten unplanned outages in the last quarter or year and determine which ones were caused by failed changes. Of the failed changes, which ones were untested or unauthorized? Calculate the cost of unplanned work for each of those episodes. If any of those failed changes resulted in disruption similar to our scenario, there's your business case for IT controls.

It's easy to see how one failed change can quickly add up to hundreds of thousands of dollars—and how implementing IT change control processes can easily pay off tenfold.

The Visible Ops Approach to Solving Unplanned Work

The Visible Ops approach to solving the problem of unplanned work is to create a culture of change management and causality. We outline a four phase implementation for Visible Ops; Phase One involves dealing with uncontrolled change in the IT organization. Uncontrolled change is the highest contributor to unplanned work, and unplanned work is the best indicator of whether our transformation is succeeding or failing.

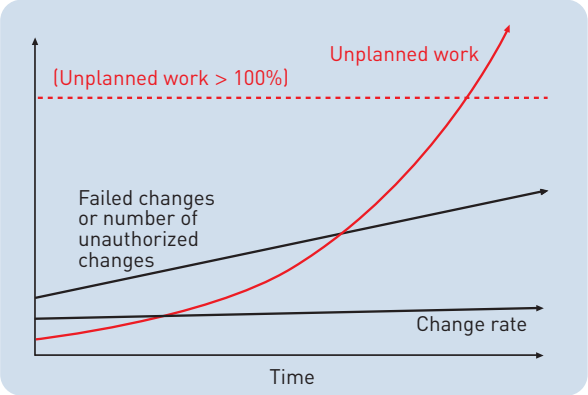
We'll illustrate the four phases outlined in *The Visible Ops Handbook* using only up and down arrows. As we examine each phase, we will describe the dysfunctional behavior that leads to a downward spiral, and then we will reveal the Visible Ops intervention which leads to a virtuous spiral.

In the first scenario, we have an IT organization in a mode of "Ungoverned Change." We ask our hypothetical IT executive, "What happens when this IT organization makes a given number of changes, and we have an increasing number of failing or unauthorized change? Does unplanned work go up or down?"

Most IT executives will say, "Of course, the number of outages causes more firefighting, and unplanned work goes up." Unplanned work increases exponentially as the number of unauthorized and failed changes steadily increases. Unplanned work will continue to increase until it hits 100%; above that, it becomes overtime, 2:00 a.m. phone calls, burnout, and staff turnover.

Unplanned work is not only the most expensive kind of work, but it also interferes with planned work. When application developers spend 20-30% of their time on break/fix activities, instead of working on the next release, completion of planned work goes down, meaning late, incomplete, or poorly completed projects.

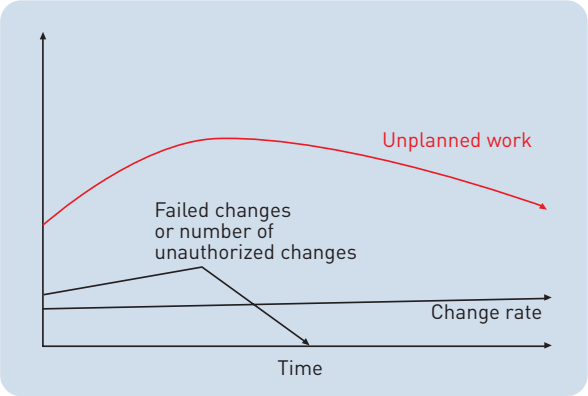
With a given change rate, when the number of failing or unauthorized changes increases, then unplanned work also increases until it dominates the entire IT organization.



Phase One of Visible Ops is called "Stabilizing the Patient."Again, we ask our hypothetical IT executive, "What happens when we take the same number of changes, but we reduce failing and unauthorized changes? We can't always prevent failed changes, but we can set the expectation that the only acceptable number of unauthorized changes is zero. What then happens to unplanned work?"

Again, most IT executives will immediately point out, "If we bring down the number of failing and unauthorized changes, we reduce the number of unplanned outages, and unplanned work goes down!"

When you bring down the failing and unauthorized changes to zero, then unplanned work is reduced, allowing more time for completing planned work.



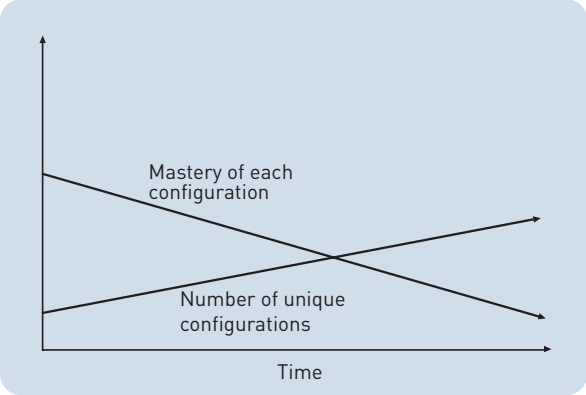
Like clockwork, when an IT organization starts to effectively control changes, we see a dramatic decrease in the amount of unplanned work in the IT organization. IT can then redirect resources to work that contributes to the business.

How Ungoverned Change Effects Unplanned Work

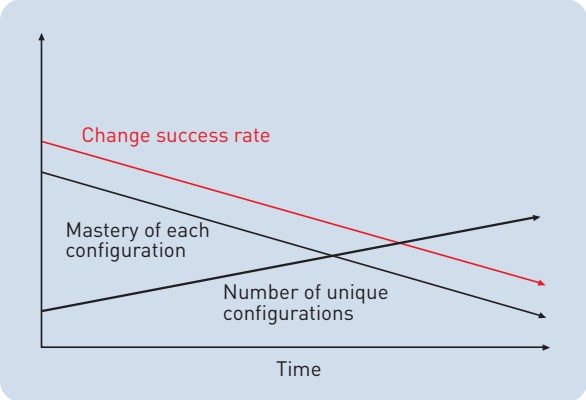
We now look at what happens after we stabilize the IT environment by controlling change. In Phase Two of Visible Ops, the highest contributor to unplanned work becomes drifting configurations and what we call "fragile artifacts".

Fragile artifacts are the IT assets that everyone in the IT organization is afraid to touch, because when they do, the asset blows up and causes a catastrophic amount of unplanned work—and the organization rarely knows how to build a new one! Low configuration mastery, low change success rates, and high mean time to repair (MTTR) create ideal conditions for protracted firefighting and downward spirals of unplanned work.

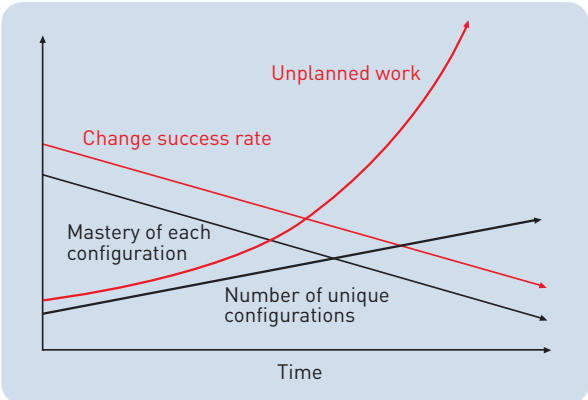
Returning to our hypothetical IT executive, we ask, "You started with 1000 servers that all looked the same, but now they are like snowflakes: no two alike. When you have this ever-growing number of configurations, what happens to your average level of configuration mastery?"



Our IT executive recognizes immediately that as the number of unique configurations increase, the mastery of each configuration decreases.



Our next question is, "When your average level of configuration mastery goes down, what happens to your change success rate?" Again, our IT executive usually points out that as configuration mastery goes down, the change success rate goes down.



We continue, "And when your change success rate goes down, what happens to your unplanned work?" As the success rate drops, unplanned work goes up.

Visible Ops Phase Two ("Catch and Release" and "Find Fragile Assets") focuses IT management on proactively finding fragile IT assets that cause this downward spiral and flagging them to prevent and deter risky changes. We prescribe putting yellow Post-It Notes on these fragile assets, boldly warning "Do not touch!" Change the login banners to scream, "Do not touch!" Flag these assets in the change management process as "Do not touch!"

This prescription is very important. In our experience, changes to a fragile IT asset have an average change success rate of 23% and an average mean time to repair of 12 man-weeks. We've just averted a tremendous amount of unplanned work.

Visible Ops Phase Three ("Create Repeatable Build Library") focuses IT management on creating repeatable, stable, and secure builds for each fragile IT asset. This step helps the IT organization gain mastery of each configuration, create maximum time limits for mean time to repair, and reduce the skill level needed for repairs, human error, and unplanned work.

Implementing Visible Ops

In the previous sections, I claimed that unplanned work is a powerful and extremely accurate indicator of availability, service levels, and ability to complete projects, as well as the compliance and security postures of the organization.

To test this conjecture, let's step through a thought experiment. Suppose you are an IT manager and that you must choose between two scenarios to be parachuted into—neither of which is very desirable. In Scenario A, you have 1000 servers supporting a business process, configured identically but insecurely. In Scenario B, you have 1000 servers supporting that same business process, but each server is configured randomly, of which 50% are configured securely. Which scenario would you choose?

Security practitioners usually choose Scenario B. They give many reasons, most centering on the monoculture argument. In biological systems, increased homogeneity in crops results in increased risk of catastrophic crop failures. They may observe that virtually all bananas are of the Cavendish variety, and are therefore all prone to disease from one strain of fungus, so when one gets sick, they all get sick. Security practitioners conclude from the biology analogy that the risk of disease is similar to the risks of unpatched and "insecure" infrastructure, so therefore randomness is better than consistency.

To fully appreciate the negative consequences of Scenario B, let's explore Scenario A, which every high performing IT organization will choose instead. High performers emphatically point out that when every configuration is identical, then:

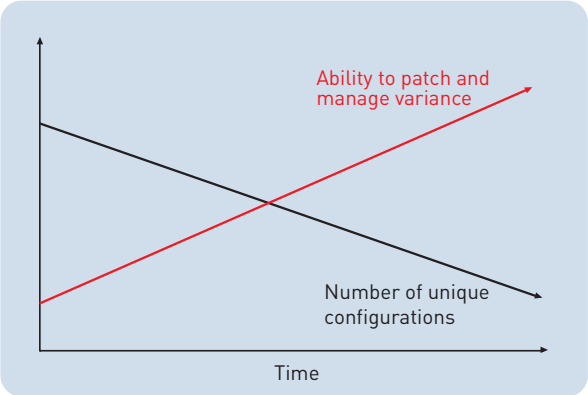
- Your "mean time to generate fixes" is lower since you have only one fix to generate
- Your fix requires a lower level of skill to engineer, since you have high configuration mastery of that one approved configuration
- Your "mean time to test fixes" is lower, since you can build one testing environment that faithfully matches the configuration of the production environment
- Your change success rate is significantly higher, because changes are tested and you have a higher level of configuration mastery
- Your "mean time to deploy fixes" is likely much higher, because having one configuration deployed across 1000 systems implies that you have some automated software distribution mechanism

One of my favorite and most compelling reasons to choose Scenario A is that under it, the organization shows that it can defeat entropy. The fact that all servers are identical shows that the organization can keep systems in a defined state, as opposed to letting them drift apart over time.

Just how much more expensive in terms of time, effort, and cost is Scenario B over Scenario A? I've heard answers spanning from 10x to 1000x, but I agree that the difference is around 500x. That is an astonishing difference in effort and in work (whether it is planned or unplanned), and shows how much more desirable Scenario A is when you have one well-understood configuration.

We can create a strong case in our conversation with our hypothetical IT executive that…

…as we reduce the number of unique configurations we dramatically increase our rate to manage patches and variance!



If you believe that Scenario A is more desirable, then it suggests that a more appropriate analogy to use in IT operations and security is not biology, which relies on luck, but manufacturing processes, which rely on skill. While having 1000 random configurations may make life more difficult for the attacker, it also makes life much more difficult for the defender. Remember, IT is like the free puppy—the cost of operations and maintenance dwarfs the capital cost. To be effective, efficient and secure, management must focus on building the processes that control all aspects of change—the changes that you make, as well as the changes that are made to you.

We call Phase Four of the Visible Ops methodology "Continuous Improvement." Organizations that achieve this stage have mastered change, and consistently spend 5% or less of their time on unplanned work. By following the Visible Ops methodology, the IT organization progresses from immature to mature, and can fulfill those two ultimate business goals: building and completing new projects for the business, and operating and maintaining existing assets.