



(Un) Due Diligence

A few months ago, I wrote an article about transitioning to a new machine. In that article, I pointed out that I create frequent backups. And then backup the backups. And then – you get the idea. I indeed AM a bit obsessive compulsive about backups. In my office, I have lots of convenient backup options. I have network access to a RAID drive (Redundant Array of Inexpensive Disks, sometimes Redundant Array of Independent Disks). My RAID cluster gives me fault-tolerant storage of over a terabyte. By fault-tolerant, I mean that the RAID cluster consists of four disk drives, and any two of them can fail without losing any data. It's EXTREMELY reliable. Plus, we have it on a UPS. I also back up weekly to various USB thumb drives and two USB disks. I also create a hot backup on my laptop. Do I need so many backups of my backups? Not really. But when I DO need my backups, it pays off.

My fabulous new machine that I set up back in October failed a few weeks ago. Spectacularly. In fact, the magic smoke escaped from both the mother board AND the hard drive. (For those of you unfamiliar with the magic smoke concept, it's the mysterious substance that powers electronics. The magic smoke is held captive inside of a chip. If the chip ever breaks, the magic smoke leaks out, and the device will no longer work. This theory appears sound – in that whenever I see the magic smoke escaping, the device never works again.) Luckily, my total downtime due to yet another machine failure was about 20 minutes. Find unused machine, login to network, connect to RAID drive, access all files.

Does this mean that I will slack off on the excessive USB additional backups? Probably not. It has been said (Mark Twain) that if a cat accidentally sits on a hot stove once, it will never sit on a hot stove again. In fact, it will probably never sit on a cold stove again either. I have been burned on poor backups before – and lost a lot of work that took weeks to recover. And the problem with being burned is that you're scared of fire for a long time. It helps me sleep well at night knowing that there is redundancy in my backup process.

Back during World War II, the story goes that a B-17 returned from a bombing mission and was severely damaged. The Colonel had a meeting with his staff to discuss how the damage analysis could help them protect B-17s better. Looking at the damaged wings, everybody commented on how shot-up the airplane was and how additional plating was needed on the wings. Everywhere there was damage; somebody recommended additional modification to help strengthen the aircraft. After all of the staff had spoken, one lone lowly Lieutenant spoke up, and said, "This aircraft represents one of the bombers that made it back. What we should do is strengthen it wherever it is NOT damaged – because the damage we see is obviously survivable." Now THAT is thinking outside of the box.

That's the problem with processes – sometimes you are probably strengthening them where they have failed in the past. However, having been burnt, you might be ignoring weak points. What you need are processes that are adaptable and provide you with feedback. What kind of feedback? First of all, you need to know what your current status is. How do you get this status? Well, if you are a small development effort, you need to TALK. And I do not mean weekly status reports or End-Of-Month

reports. I mean honest, open, frequent communication. This is the basis for ANY good process. Capability Maturity Model, Personal Software Process, Team Software Process, Scrum, Agile; call it what you want.

I personally like the concept of a daily stand-up meeting – where you literally stand up for the entire meeting. REALLY cuts out the long-winded talkers. In fact, my personal favorite method is when the person talking stands on one foot (not balancing the whole time – they can switch feet – but one foot should always be off the ground). In 10-15 minutes, issues get discussed, problems identified, resources quickly reallocate. While not every issue is resolved, at least proactive planning can occur.

What if you are too big for daily stand-up meetings? Well, you can still emphasize daily meetings for the programming teams, but somebody has to wrap up the important issues and elevate them. That is where metrics come in. Metrics are like the temperature of a project. You want to know whenever the project starts to run a fever. What do you measure? Whatever you need to measure that will allow you to reduce the fever. Errors. Fix times. Testing time. Source of errors. Rework.

Don't run a project based on how you got burned last time. Yes, it is important not to make the same mistakes again. It is also important to think ahead, and try not to make any critical new mistakes, either.

Backup often, but don't become obsessive. Use due diligence, not undue diligence. Instead of making one small part of your process bullet-proof, how about strengthening all (or at least most) of the weak points. Collect meaningful and useful metrics, and use the metrics to find out what the weak points are, and reallocate resources as necessary.

And try not to get burnt.

— David A. Cook

Senior Research Scientist and
Principal Member of the Technical Staff
The Aegis Technologies Group, Inc.
dcook@aegistg.com

Can You BACKTALK?

Here is your chance to make your point, even if it is a bit tongue-in-cheek, without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CROSSTALK, we also accept articles for the BACKTALK column. BACKTALK articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BACKTALK article should be entertaining and clever or original in concept, design, or delivery. The length should not exceed 750 words.

For a complete author's packet detailing how to submit your BACKTALK article, visit our Web site at <www.stsc.hill.af.mil>.