Drowning in Chaos?

Pull Yourself Out!

Make Sense Of The Standish Reports And Take Control Of Your Projects

By Robin Goldsmith

Most IT projects are late, over budget and deliver something other than what was expected. Such outcomes are the consequence of arbitrarily mandated budgets and schedules, inadequately defined business requirements and too-little/too-late reactive attention to quality and testing. Learn to avoid these problems by developing objective factual project measures and an environment in which everyone involved takes responsibility for their results.

Looking Into a Train Wreck

Like a train wreck, the Standish Group's periodic CHAOS reports (www.standishgroup.com) about IT project success rates evoke a certain ghoulish fascination. Despite, or perhaps partly because most readers don't recognize the reports' questionable measures and analysis—including overlooking what undoubtedly is the most common real proximate cause of project failures—there's wide acceptance of the reports' basic findings that IT projects seldom are on time, on budget and what the stakeholders want.

While such an unflattering depiction of project effectiveness obviously

Robin Goldsmith is an author and testing consultant specializing in business engineering and requirements.

Photographs by Kristian Peetz

reflects a somewhat contrarian view of IT, there's plenty of room for even more contrarianism. In fact, the seeds of this admittedly contrarian article were born in a letter to the editor and subsequent correspondence with (I believe he'd accept the characterization) contrarian Bob Glass, who may have been the first to publicly question the seemingly sacrosanct inviolable CHAOS findings that the great majority of IT projects fail (see his "Loyal Opposition" article "IT Failure Rates—70% or 10-15%?" in the May-June 2005 IEEE Software).

For more than a decade, the Standish Group has published a series of CHAOS reports that describe embarrassingly low IT project success rates, starting with an abysmal 16 percent in 1994 and improving to 34 percent in 2006 (a number that looks good only in comparison to preceding years' reports—or for a baseball batter). Even though it's now well over a decade old, the 1994 report seems to continue to be the one cited (and read) most, by me and others, primarily I assume because it's the only one available in its entirety for free (find it at www.standishgroup.com/sample_research/register.php). Since subsequent reports are priced prohibitively (for me), I and presumably most people know only snippets of them reported in the trade literature, such as the current 34 percent figure (which was described in SQE's 2/22/2007 Between the Lines e-mail newsletter).

So far as I can tell, though, the Standish Group's methodology, findings and analysis have remained fairly consistent over the years, with changes essentially only in the absolute figures. Consequently, concerns raised by the 1994 report are reasonably likely to remain current.

Is IT Really That Bad?

Measurements must be both reliable and valid. The reports' year-to-year consistency indicates that the measures were made reliably. Glass questioned whether the CHAOS reports are valid and said his personal experience indicated a much lower IT project failure rate.

He may be right, both about his personal experiences and their being representative for the broader IT project population. We can't really tell either for sure, though, because the fact is that nobody else seems to have sufficient suitable objective IT project measures that could validate whether or not the CHAOS reports' main finding is accurate.

Survey responses in general tend to be shaky in both reliability and validity.

In fact, this lack of objectively measured IT project data isn't just limited to formal study reports. It's highly unlikely that (m)any of the organizations surveyed for CHAOS based their answers on reliable and valid facts, which raises serious methodological reasons to question the CHAOS figures' validity. The CHAOS data comes from surveys of IT executives' perceptions regarding project measures. I think we all recognize that survey responses in general tend to be shaky with regard to both reliability and validity.

Even when well intentioned, survey responses often are guesses or, at best, broad approximations. I don't know about you, but I'm asked to respond all the time to surveys that ask questions I don't have good answers to, either because I don't know or because none of the answers actually fits my situation. Nonetheless, it doesn't stop me from answering something, probably with vague impressions or answers that I know have no basis.

Illusion of Precision

Although I'm sure it's unconscious, statistical manipulations can both distort reality and imbue an illusory appearance of precision. For instance, "About 90 percent" sounds much less precise than "89.51 percent." Surveys often ask people to pick ranges of values; say 100-150. Having to give a single answer fails to take into account the pattern of variation in the source's actual project data, which I'm sure most of the CHAOS sources were highly unlikely to consider, let alone have quantified. Thus, the respondent may have felt the 100-150 range was most typical, even though perhaps a few instances are in the 25-50 and 400-500 ranges. It's like the old story of a person whose left foot is in a bucket of boiling water and whose right foot is in a bucket of ice. The average of the two temperatures might be a comfortable 90 degrees, but neither foot feels at all comfortable.

While CHAOS groups within project size categories, it also consolidates across categories, and it's unclear whether its calculations accurately reflect the varying sizes of reported projects. Should a one-month one-person project be given the same weight as a one-year 100-person project? Should a project's size be defined based on the original estimate or the bloated size the report tells us the project eventually reached?

Regardless, to come up with averages, it's necessary to convert each of those 100-150 range responses to an average score of 125. Although the reports do show distributions of responses for the various ranges, they focus on single average overrun percents, which take on an appearance of authority and scientific precision.

Moreover, even if the CHAOS-reported average overruns of 189 percent and 222 percent are absolutely accurate, it's unclear exactly how to interpret these reported overruns. If the budget was $100, does an overrun of 189 percent mean that the project actually cost $189 or $289?

On the other hand, despite such serious methodological issues, a lot of subjective data, including my own, does support the general tenor of the CHAOS conclusions.

For example, when I describe the 1994 CHAOS numbers in my seminars/speeches, I usually ask the participants whether the report reflects their own project experience. Over the years, thousands of attendees regularly have given what seems to be almost always unanimous affirmation. Furthermore, the reports wouldn't be so widely cited and accepted unless people do find the conclusions consistent with their own experiences.

It's All in Your Head

However, Glass also has a point. IT projects ordinarily *do* deliver systems that people can and do use. Both Glass and CHAOS are right, to an extent, and focusing on either's conclusion alone may itself obscure other significantly important issues. To understand this balance more fully, certain interesting psychological factors also need to be taken into account.

CHAOS is correct that IT projects routinely are late, over budget and not entirely what is expected. The aberration is large, and it's really not necessary to quibble over CHAOS' specific numbers. Glass is right that usually projects produce some kind of working system. Once people can start using the system, they get busy with it and tend to forget that it was late, over budget and perhaps only a portion of what they'd expected.

That's fine; life must go on—but this forgetting can involve some bigger ramifications. First, the same scenario recurs, project after project. As an industry, IT tends not to learn, or perhaps tends to learn, but not necessarily the *right* lessons, from our experience. Instead, we've learned to become quite proficient at a development process that reliably repeats failure—late, over budget and not what's expected—and then through denial essentially accepts that failure as adequate.

Second, by failing to relate our own actions to our results, we prevent the personal initiative needed to make meaningful improvement. Consider a perceptual paradox phenomenon that's evident in the worlds of politics and culture. For example, survey after survey finds low approval ratings of Congress, yet congressional re-election rates historically are close to 100 percent, sometimes not even impeded by incarceration or death. Similarly, surveys repeatedly find that people say the American education system does a terrible job, but their own local school is just fine.

People often have an understandable disconnect relating broad-scale external results to their personal sphere. That's a major reason why people seldom change their individual behaviors in response to well-known big problems. Annual reports of lung cancer death statistics tend not to cause individuals to stop smoking. The scientific community's agreement that global warming will deplete animal life in the oceans and submerge coastal cities doesn't cause individuals to cut their gas guzzling. Statistics on the damaging health impacts of obesity don't make individuals eat better or less. And CHAOS's reported 84 percent IT project failure rate doesn't cause individuals to change how they do projects.

Perhaps the problem seems overwhelming, or one's own power to affect it seems so insignificant, but ultimately it comes down to the normal psychological defense mechanisms people enlist unconsciously to protect their self-images. We've gotten so good at denying anything that reflects poorly on us, so unwilling to recognize, let alone take responsibility for our results, and so willing to shoot the messenger, that we not only fail to take appropriate corrective actions but also sometimes intentionally engage in additional self-defeating behaviors. For example, consider the dysfunctional behaviors that get acted out even more excessively every afternoon on the television scream shows.

The Real Cause of Failure

It's not only pathetic TV wanna-be celebrities who respond to dysfunction with greater dysfunction. Many, if not most IT projects are destined from the start for failure because management has arbitrarily dictated a project budget and schedule that bears no relationship to the work to be done.

But that's just the beginning of a downward spiral. When budgets and schedules are nonsense, overrunning them becomes nonsense too. So what if nonsense budgets and schedules are overrun 189 percent and 222 percent of the time? People doing the projects don't take a personal stake in the outcome because they go into the projects "knowing" the nonsense budgets and schedules are impossible to meet, which becomes a self-fulfilling prophecy, regardless of how "objectively" feasible the budget/schedule may be.

The more the worker bees grumble and miss their targets, the more the managers feel compelled to dictate yet-even-more nonsensical budgets and schedules, thereby ensuring failure and confirming to them that they were right in having to hold the

troops' feet to the fire. Dysfunction begets more dysfunction.

The business users don't know about these internal dynamics. They only know that as an industry, IT almost always blows project budgets and schedules. They perceive that IT doesn't know what it's doing, and thus they may not believe what IT says, which further impedes IT's ability to keep its promises.

Project managers' psychological defense mechanisms prevent them from becoming aware, let alone believing, that they may have credibility issues. So they attribute their difficulties to other, often-irrelevant factors and mistakenly divert attention to these perhaps non-issues instead of addressing their real credibility problems. This further reduces their likelihood of success and their already-diminished credibility. The business puts more pressure on IT management, which leads to even more nonsense dictates and more overruns; and the cycle perpetuates.

Real business requirements tend not to change nearly so much as people's awareness of them.

Close, But Missing Critical Distinctions

Ultimately, project budgets can't help being nonsense unless they're based on adequately defined real requirements. At first glance, this seems very much in line with the project success and failure factors that CHAOS analysis identifies.

Requirements and user involvement issues certainly dominate the tops of the factors lists. They indeed are important, but I fear the report simply parrots overly simplistic, widespread conventional beliefs that continue to miss the distinctions critical for making meaningful improvement.

The report focuses only on amount of user involvement. While a certain *quantity* of user involvement is necessary for discovering the real requirements, it's not sufficient. Rather, it's the *quality* of that involvement that really matters. Merely subjecting users to more of the same ineffective current practices won't produce better requirements.

It's understandable that the report fails to realize this distinction, because the industry is unaware that such a distinction exists. Perhaps one reason why user involvement is low is that managers may sense that just giving more time by itself often may not pay off.

Similarly, the report mirrors the industry's general lack of awareness of the important distinction between real, business requirements and product/system/software requirements.

The common use of the term *requirements* refers to the requirements of the product, system or software that is expected to be created. Said product, system or software actually is the high-level design of one of the possible ways how to accomplish the presumed real, business-requirements deliverable *whats* that provide value when delivered/accomplished/met/satisfied.

The CHAOS report identified incomplete and changing requirements as two separate issues. In fact, the main reason that product/system/software requirements change is because they aren't defined accurately and completely enough in the first place, which in turn is mainly due to the failure to adequately discover the *real* business requirements. Designs (including their product, system and/or software requirements) can change frequently and rapidly. Real business requirements tend not to change nearly so much as people's awareness of them.

Missing Resources And Unrealistic Expectations

The main remaining failure factors identified in the report include lack of resources, unrealistic time frames, unrealistic expectations, and lack of executive support. The first three are all largely attributable to not adequately defining the real business requirements and the product/system/software requirements to satisfy them, which of course contributes to management's propensity for arbitrarily establishing budgets and schedules.

Once a project is set to fail, albeit by management's actions or absence thereof, managers quickly distance themselves from the project.

One More Little Oversight

Roughly half the time spent in IT projects is taken up by testing, yet the CHAOS report's failure factors don't mention quality or testing.

Again, this probably reflects a lack of awareness among the surveyed industry executives, which in turn translates into the project failures described. Inadequate quality clearly causes users not to receive what they expect, and the unplanned time needed to fix defects is a major contributor to project budget and schedule overruns.

Without awareness of—and informed attention to—quality and testing, quality and testing activities are too little and too late. Too many defects escape to be found by the users. Those defects that are found during development tend to be discovered so late that they're very expensive to fix. Proactive testing can turn this situation around, actually helping projects deliver quicker and cheaper by catching and preventing more of the errors earlier, when they're easiest to fix.

All in all, in spite of significant questions about their measures and analysis, the Standish Group's CHAOS reports seem generally accurate in finding that most IT projects are late, over budget and not entirely what is expected.

To a considerable extent, such outcomes are the inevitable consequence of arbitrarily mandated project budgets and schedules, inadequately defined business requirements and too-little/too-late reactive attention to quality and testing.

Such problems persist in part because few organizations have suitable or sufficient objective factual measures of their projects and because those responsible for IT haven't created an environment in which they and other involved individuals take enough personal responsibility for their results.

The solution? Know, document and adhere to the *real* requirements.