A quick introduction: This chapter comes from *Business Driven Information Technology: Answers to 100 Critical Questions for Every Manager* by David Laube (ed.) and Ray Zammuto (ed.). Published September 2003 by Stanford Press. In 2010 the author of this chapter took the liberty of updating a few passages to maintain relevancy to his readers.

Each chapter in the book is a response to a question. This chapter answers question #92, "***Why is it important to explicitly state the intended business result of an IT project? How should this be done?***" Please note – what is expressed in this chapter is true not only of IT projects, but also of strategic initiatives.

This chapter is about a 6- or 7-minute read.

Question 92: Why is it important to explicitly state the intended business result of an IT project? How should this be done?

William W. Casey, Ph.D.

The Customer Service Representative Improvement Project

In the massive IT shop of a Fortune 50 company, five million dollars had been set aside for the Customer Service Representative Improvement project. Dozens of cubicle-ensconced developers had labored to help automate the work of hundreds of telephone answerers. After months of muddled effort, the budget was exhausted and there was little to show for the effort. The three client-side vice presidents met to sort out the mess and decide exactly how the IT department should be punished for the failure. However, as they talked, the three executives learned that each had been paying for (and orchestrating) a different version of "automation and improvement." For one, the point had been to reduce headcount. For another, the goal had been to reduce training time for new employees. The third had been hoping to increase sales. It turned out that their mushy business goal, "automate and improve," had led them in circles, and myriad functional and technical requirements had done nothing to resolve and communicate the point of the project. The project was, thus, pointless. On all projects, there is a moment of truth, a time to discuss the project's intended purpose or result. The question is, will it come before or after the dollars are spent?

Why We Do IT Projects

IT projects are not conducted because of their functional or technical requirements. They are not conducted because of their scopes. Nor, are they conducted because of their budgets or schedules. The point of any IT project is the *result* that it will produce for the enterprise. One way to express this result is the Whole Goal™, the general idea of which is borrowed from "systems thinking" gurus.[i]

A project's Whole Goal expresses the measurable *result* of the project. The Whole Goal brings clarity to project definition and helps focus a team and stakeholders. Whole Goals become a basis for declining the countless ad hoc pork barrel requests to which large budgets are usually subject.

Anatomy of a Whole Goal

Whole Goals consist of two parts. The first part is the intended result (synonyms: outcome, end-state, accomplishment, effect, achievement). Some examples of results are:

- Reduce time to access database to two seconds or less.
- Increase market share to 12% or more.
- Reduce abandoned calls rate to 8% or less. (Abandoned calls are times when telephone callers hang up rather than stay on hold.)

For anyone who has written goals before, none of these examples will seem remarkable. However, a couple of features are noteworthy. First, each example is about the result of an effort and does not describe the effort itself. For example, it is not stated whether the project manager will reduce abandoned call rate by adding more phones, adding more employees, installing better equipment, implementing voice menus, or by getting rid of customers. There is not a word about *how*. Second, each one states a

measurable result. Using the same example, we did not state “improve the customer experience,” or “deliver world-class customer service,” or any other inspiring statements of general direction. Results – as they are used in Whole Goals – are boringly objective.

Restrictions (also called restraints) are the second element of Whole Goals. In contrast to the goal, which states the result to be produced, restrictions state the results not to be produced. Stating restrictions helps prevent unintended consequences and other results of tunnel vision or local optimization.

Incidentally, let’s clear up a common confusion here. “Restrictions” are altogether different from “constraints.” For example, if you have only three Java programmers available for a big project, that is a constraint; you have no choice. If you commit to limiting their individual overtime to under 20 hours on any given week, then that is a restriction; you have a choice to work them harder than that, but you are choosing in advance not to.

A properly written Whole Goal essentially takes this form: “Please deliver this outcome, but don’t do these unpleasant things while you’re at it.” or, “Accomplish this: [result], subject to these [restrictions].” Drawing from earlier examples, here are two complete Whole Goals (result + restriction(s)):

Result

- Reduce time to access database to two seconds or less.

Restrictions

- No decrease in the amount of data available.
- No decrease in number of work stations running at one time.

Result

- Reduce abandoned calls rate to 8% or less.

Restrictions

- No increase in number of customers calling back because they have been cut off or because their problem was not satisfactorily resolved.

Good Whole Goals Enable Strategic Tradeoffs

No doubt, the omission of *schedule* and *budget* from these examples will discomfit some readers. However, the omission of these two critical restrictions from Whole Goals is quite deliberate. Here is why: executives must continually make tradeoffs between the project's result (Whole Goal), its schedule, and its budget. (That is to say that they must continually decide between good, fast, and cheap!) Consequently, it's unwise to insert budget or schedule *into* the Whole Goal, as it is difficult to balance a thing against itself. It is better to keep each of these three factors separate. In every project environment, one of these factors will tend to trump the other two. However, *all three must be clear for the tradeoffs to be clear.*

Of course, executives are not the only ones making tradeoffs. At a less strategic level, everyone on a project makes tradeoffs throughout the course of the project. Whole Goals provide the context in which to make those decisions. Even software developers contributing relatively small portions of a project have reported that Whole Goals help immensely, providing a perspective that scope statements and litanies of requirements do not.

Technology Whole Goals are Necessary But Not Sufficient

The Whole Goal of a major program is usually built on the Whole Goals of subordinate projects; Whole Goals are hierarchical. Frequently, for example, the Whole

Goal of an internal client's program will be fed by an IT project Whole Goal—but it will also be fed by Whole Goals from other disciplines such as training and process design. For example, the IT department of a bank may install a customer information system to give tellers the information they need to suggest appropriate products to individual customers. However, the intent of the system cannot be achieved without tellers learning how to sell and how to use the system. The tellers' compensation system also may need tweaking and staffing levels may need to be adjusted as tellers spend more time per customer.

When clients understand that the achievement of their Whole Goal is based on the achievement of *several* Whole Goals, not just the most expensive and obvious one (IT's), their own success is likelier and scapegoating IT is less likely.

Whole Goals Aren't Always Easy or Popular

There is an unpopular downside to Whole Goals: they are damned hard to write. They require a clarity of thinking that is nothing short of *very hard work*. Like a Japanese haiku, which packs volumes into seventeen syllables on three lines, the Whole Goal must do heavy lifting in a few words. One must *really* understand the point of an endeavor to do that successfully.

Worse still, Whole Goals surface differences of opinion. Whole Goals force answers to questions such as: what problem are we actually trying to solve, how good is good enough, and what boundaries must we not cross in the process? Such precision provokes arguments and is not for the fainthearted.

And, for better or worse, Whole Goals derail "free rides." One IT project manager who had received her Whole Goal midway into her project discovered that better than three quarters of the project's alleged requirements were obviated by understanding the

point of the project. That did not please all the employees who had been attaching requirements to her project.

Summary

Any IT project is conducted in the service of producing a result. The more clearly executives understand and communicate the project's intended purpose the more likely they are to succeed. Although precision at the detailed level of scope and requirements is common, precision at the strategic level of purpose is not. One way to achieve that is through Whole Goals, which specify the result the project is intended to produce and the results it is proscribed from producing along the way.

---

[i] For a typical discussion of the characteristics of a system see: Jeffrey A. Hoffer, Joey F. George, and Joseph S. Valacich, Modern Systems Analysis and Design, 3rd ed. (Upper Saddle River: Prentice Hall, 2002) 41. Two of those characteristics are employed in the Whole Goal definition offered here.