

## Not Sure? Try It Out

*Preston G. Smith*

December 1, 2008

As our world is apparently becoming more chaotic, we need better ways of managing projects in a volatile environment. In part, the literature has come to the rescue. Also, the entire field of agile software development is aimed at managing better under chaos.

Specific aids--such as Doug DeCarlo's book *eXtreme Project Management*--are helpful in specific circumstances. But a broader approach is to build a modus operandi that is especially effective when change is the norm. An important part of such a modus operandi for turbulent environments is experimentation.

### Why Experiment?

There are many definitions of experiment, but they all have two elements in common: you do something, and then you observe what happens. For instance, you mix two chemicals together and notice how much their temperature rises. Or you show a prototype to a customer and see if they can figure out which button to press first.

Experimentation is valuable in a world of change because it allows you to preview the change, seeing what it would be like and what might be effective in handling it if it happens. Experimentation expands your world and allows you to see things from a different perspective. In this way, you can often steer change in a direction that is better for you. At a minimum, you can keep yourself out of dead-ends. And you can prepare for possible changes, obtaining the resources in advance that might be needed if the change were to occur.

### Experiment Broadly

I purposely mentioned a very classical kind of experiment above: mixing two chemicals. This is fine if you happen to be a chemist, but if you are a manager of a project buffeted by the winds of change, you must apply experimentation more broadly. Try out different design alternatives. Get in touch with customers in different ways--and remember to sample your non-customers, too; there is a reason why they don't buy from you. Experiment with various materials or manufacturing methods. Consider alternative suppliers. Mock up various advertising possibilities.

In general, keep asking yourself, how can I prepare for what might happen tomorrow by trying it out today?

### Experiment Early

One kind of experiment is a test. But testing often leads us into a trap. We complete the project and then we test the outcome to see if it meets requirements. This type of validation testing is only one application of testing. It is excellent for what it does, but it is worthless for dealing with change. To manage change, you must test (more broadly, experiment) early. Experimentation is only useful for managing change if it occurs before the change happens.

Testing can be useful for managing change if it is done early. Test a half-done product to see how it works (it may, in fact, be good enough already!). Test various alternatives early to see which is the most robust to change.

Practitioners of the extreme programming technique in software development have pushed their testing up very early: they write the test before they write the code. Then they design the code to pass the test. This is like being handed the final exam for a class when you walk into its first session. The final exam is a map of the future.

### Learn from Failure

When experimenting, we have a tendency to run experiments that will turn out as we expect--that is, they will be successful. Always keep in mind that the kind of explorative experiments that help you deal with change are aimed at learning, not at success. When you run an experiment that you think will turn out in a certain way and it does turn out that way, you haven't learned much. Instead, design your experiments so that you really aren't sure what the outcome will be. Then you learn the most, and as you do several rounds of experiments, your rate of learning is greatest.

This sounds very logical. The real difficulty is in applying it in the real world, where success is valued more highly than failure. What will your boss think if you announce that your last five experiments failed? Your boss--and all the bosses up the line--will

need to be re-educated.

### **Exploit Technology**

In almost any field you might consider today, there is some kind of computer technology that can help you experiment far more effectively or economically than just a few years ago. There are computerized technologies for building physical parts without having drawings of them, for testing a molecule without actually creating the molecule and for trying out a circuit even before any wires are soldered.

A simple example that we are all familiar with is a word processor. Before we had these, manuscript changes were difficult on a typewriter. Consequently, it behooved us to plan our writing thoroughly and type carefully. With word processing technology, we can just “knock it out” and then move paragraphs around and let the spell checker find the typos.

Look for analogous situations in your projects where you can exploit computer technology to experiment more prolifically. And *prolifically* is the operative term here; don't just use the technology to cut costs and pocket the savings. This may cut your budget, but it won't improve your ability to manage change.

*Preston Smith is a product development consultant and trainer based in Portland, Oregon. He is co-author of the time-to-market classic Developing Products in Half the Time, and more recently author of Flexible Product Development. You can reach him at preston@NewProductDynamics.com or visit his website at FlexibleDevelopment.com.*

Copyright © 2008 gantthead.com All rights reserved.

The URL for this article is:

<http://www.gantthead.com/article.cfm?ID=246087>