

Making the Strategic Case for Process Improvement

Watts S. Humphrey



In my previous column, I wrote about management support for process improvement and how your approach should change depending on the manager you are dealing with. The questions left open were how to make the strategic case for improvement, how to make the tactical case for improvement, and how to move from a tactically based to a strategically based improvement program.

In this issue, I describe how to make the strategic case for process improvement. I start on the assumption that you can get the ear of a senior manager. You may work directly for this manager, or you may have obtained an appointment to make a presentation on the subject. In any event, you now have an appointment to see a senior manager. How do you prepare and what do you say?

The General Improvement Case

The approach to follow for almost any type of improvement effort would be much the same:

- Clearly define what you propose.
- Understand today's business environment.
- Identify the executive's current hot buttons.
- Make an initial sanity check.
- Start the plan with two or three prototypes.
- Estimate the one-time introduction costs.
- Determine the likely continuing costs.
- Document the available experience data.
- Estimate the expected savings.
- Decide how to measure the actual benefits.
- Determine the improvement's likely impact on the executive's current key concerns.

- Identify any other ways that the proposed improvement could benefit the business.
- Produce a presentation to give this story clearly and concisely.

Defining the Proposal

Before you do anything, define exactly what you want the executive to do. The best guide that I have found is to actually prepare an implementation letter for the executive's signature. Then in the meeting, if he or she says, "Great, let's do it," pull out the letter and hand it over as a proposed implementation instruction. While this reaction is not likely, the exercise will help you to produce a clear statement of what you intend to propose. Also, if you are several management levels removed from this executive, you should describe the letter as a proposed draft instruction that you have not yet reviewed with your immediate management. Better yet, show the draft letter to your manager first and get his or her suggestions on improving it.

Understand Today's Business Environment

In preparing for the presentation, remember that there is no magic formula for convincing senior managers. Every case is different. The approach must vary depending on the situation and the executive's current priorities. If, for example, this executive has just cut resources to meet a profit goal or the organization has just lost a major contract, this might not be a good time to propose an additional expense. So, plan your improvement strategy with a clear appreciation of what is happening right now in the business.

Identify the Hot Buttons

Next, find out what this executive is most concerned about. Since most executives give lots of talks and issue many statements and announcements, this is generally fairly easy to do. With few exceptions, executives use every available occasion to plug the topics they feel are highest priority. So get copies of some of this executive's recent announcements and presentations, and look at the common themes. You will usually see a fairly consistent message. The manager may frequently mention profitability, or market growth, or development cycle time. Because executives are concerned with many things, he or she will almost certainly make many points. But if there is an overriding concern, much like a television commercial, this topic will pop up every time there is an opportunity. Once you know the executive's current hot button, figure out how the process improvement you propose would address that concern, then make sure the improvement justification addresses this topic.

Make an Improvement Sanity Check

In preparing an executive proposal, the first step is to gather the known facts about the costs and benefits of the proposed improvement program. As soon as you have the data, make an initial sanity check: Does the proposed process improvement directly address the executive's key concerns? If not, are the cost savings significant enough to justify the executive's listening to the proposal? If the improvement directly addresses something the executive has been pushing for, then cost will not be a key concern. If cost savings are important, however, are the proposed savings large enough to be convincing?

Most executives know that improvements are rarely as effective as first proposed and that there are always hidden costs. A good rule of thumb is that improvements with savings of 2 or more times are usually impressive while numbers below 25% are likely to be ignored or subjected to very close scrutiny. If cost is important and you are not proposing a significant cost saving, consider putting off the presentation until you can make a stronger case.

Prototype Introduction

If the proposed improvement passes this sanity check, the next step is to analyze the costs of introduction. It is almost always a good idea to start an improvement program with one or more prototype tests. This not only reduces the initial introduction costs, it also maximizes your chances of success. Just about any change will affect both engineer and management behavior, and these changes are rarely natural or easy. Thus, many people will likely have initial problems following the new methods. To be successful, you must identify and resolve these problems at the very beginning. The longer it takes people to properly use the new methods, the more the introduction will cost and the longer it will take to show benefits. The principal advantages of starting with a prototype program are that the initial costs are lower and it is easier to watch a few limited pilot programs to make sure they are getting the needed support and assistance.

One major risk in any improvement program is that the prototype project could be cancelled or redirected. To protect your project from this risk, try to get two or three trial projects underway. That way, if one is killed or redirected, you will still have the others to fall back on.

Introduction Costs

While you will almost certainly follow a gradual introduction strategy, it is a good idea to show both the prototype and the total introduction costs. The reason is that the introduction strategy will probably change several times before you are done and you don't want to keep changing the cost-benefit story. Emphasize that you are presenting the total introduction costs for the entire organization, but that the initial costs for the prototype program will be much lower.

In any significant improvement, there will be initial introduction costs as well as continuing costs for sustaining the improvement. Since any process-improvement introduction will require some executive and management time, you need to make an appropriate allowance. Generally, however, the major costs will be the time to train and support the engineers. Even the introduction of a new tool takes training and support, so don't gloss over the introduction costs; they can amount to very big bucks.

For example, with a new programming language, a minimum of two weeks of intensive training is usually required, often followed by a period of close consultation during initial use. Similarly, a new tool will require an initial training session of several days plus guided practice sessions and continuing professional support for at least a few weeks.

In estimating these costs, remember one key guideline: Your story will be judged by its weakest point. If someone finds an error or a serious underestimate anywhere in the story, the assumption will be that similar errors infect the entire story. So be careful about making low estimates or assuming that some costs are insignificant. If you don't know the facts, find someone who does. Above all, don't make unsupported assumptions; your entire presentation could be discredited.

In addition to executive, manager, and engineering time, trainers and expert assistance will almost invariably be needed. This can add a significant cost, particularly if you plan to use outside assistance. On the other hand, the costs of building internal experts and trainers can be very large, and few executives will want to make such a significant commitment, at least until the proposed improvement has been proven with early tests.

The Continuing Costs

After the improvement has been introduced, there will be ongoing support costs. You may need continuing training to cover engineering turnover or staff growth. Expert assistance and support may also be needed. These costs can be substantial, so it is important to identify them. Describe them clearly up front and then justify them. If you don't give a complete cost story, management will sense that there are hidden costs and likely assume that these costs are much greater than they actually are.

The Process-Improvement Benefits

Next, we turn to the benefits. Here, you must address two points: first, how long will it take for the improvement program to recover the introduction costs, and second, how will the improvement address the executive's principal concerns? If you can show that the improvement will pay for itself, then the other benefits would be pure gravy. So start by making the cost case.

The way to make the cost case is to first gather the available facts on improvement benefits. Here, you are at a disadvantage. Costs are always easier to prove than savings. Executives know this, however, probably better than you do. After all, they spend much of their time justifying changes. So don't worry about proving an ironclad case; executives will rarely demand it. But they will want a logical story that hangs together and looks complete and realistic.

Improvement Experience

So, first, what are the available facts? Unfortunately, there are few statistically sound improvement studies, even for accepted process-improvement methods. While there may be some available analyses, you will probably have to rely on anecdotal evidence. This may not be as precise as a comprehensive statistical study, but such evidence can be even more convincing. The best case would be one in which someone in your industry has implemented the same improvement and described its benefits in a talk or a paper. If you can find a suitable example, summarize the general findings in the executive presentation, but then emphasize the results reported by your competition.

Calculating the Savings

There are many ways to save money. In the final analysis, however, most software cost savings result from personnel reductions. For example

- By introducing a design inspection program, you can eliminate defects early in the process and save considerable rework.
- A measured quality program can reduce the numbers of defects found in test and shorten testing time.
- A configuration-management system can save development time by ensuring that correct program versions are always available.
- A change-control system can reduce the number of uncontrolled changes and save development time.

While these savings are all real, they all have the disadvantage of being very hard to prove, either in advance or after the fact. As a result, the most convincing argument is generally that the XYZ Corporation cut their test time by x%, or that the ABC Company reduced customer-reported defects by y%. Starting from these numbers, you can then generally show the amount of money you would save if your organization had similar results.

Measuring the Benefits

In concluding the presentation, discuss how the prototypes will be designed to measure the improvement benefits. For example, if the proposed improvement is intended to reduce development cycle time, discuss how to demonstrate that it does. A common problem, however, is that few organizations have data on their current operations. Thus, even if you conduct a highly successful prototype experiment, you may have no way to show that it was successful. That is, you will have lots of “after” data but no “before” data with which to compare it. As part of the proposal, raise this issue and suggest ways to handle it.

Even when organizations have little or no data on their current operations, there are usually a few things that you can measure. For example, data are often available on the length of time by which projects have missed their planned delivery dates. There are also often records of the numbers of defects found in system test or reported by customers. Similarly, data can generally be found to calculate the percentage of the development schedule that is spent in integration and system test. Another good measure is the total development hours divided by the total lines of delivered source code. While no single measure can characterize the quality of an organization’s processes, there are many possible measures that can be obtained from most accounting and project-reporting systems.

Because you need to apply these measures to the existing projects, it is important to start looking around for available data even before you make the proposal. Then you can use these data in justifying the proposed improvement. Also, you can be reasonably sure that there will be a way to measure the benefits when you are done.

Other Benefits

While cost savings are important, not all improvements can or should be cost justified. For example, if you can show that the change will improve schedule accuracy and predictability, reduce cycle time, or make your organization more competitive, management will often approve the proposal, even if it does not clearly save money. The key is to convince management that the improvement is good for the business and then, if possible, show that it will also pay for itself. If you cannot prove the savings story, however, don’t give up. If the other benefits are compelling, management may be willing to proceed anyway.

Stay Tuned

In the next issue, I will use an example to show how to structure and give an executive presentation on process improvement. Following that, subsequent columns will deal with how to make the tactical case for improvements and then how to move from a tactically to a strategically based improvement program.

Acknowledgements

In writing papers and columns, I make a practice of asking associates to review early drafts. For this column, I particularly appreciate the helpful comments and suggestions of Dan Burton, Marsha Pomeroy-Huff, Jim McHale, Mark Paulk, and Bill Peterson.

In Closing, an Invitation to Readers

In these columns, I write about software issues and the impact of quality and process on engineers and their organizations. I am, however, most interested in addressing issues that you feel are important. So please drop me a note with your comments, questions, or suggestions. I will read your notes and consider them when I plan future columns.

Thanks for your attention, and please stay tuned.

About the Author

Watts S. Humphrey founded the Software Process Program at the SEI. He is a fellow of the institute and is a research scientist on its staff. From 1959 to 1986, he was associated with IBM Corporation, where he was director of programming quality and process. His publications include many technical papers and six books. His most recent books are *Managing the Software Process* (1989), *A Discipline for Software Engineering* (1995), *Managing Technical People* (1996), and *Introduction to the Personal Software ProcessSM* (1997). He holds five U.S. patents. He is a member of the Association for Computing Machinery, a fellow of the Institute for Electrical and Electronics Engineers, and a past member of the Malcolm Baldrige National Quality Award Board of Examiners. He holds a BS in physics from the University of Chicago, an MS in physics from the Illinois Institute of Technology, and an MBA from the University of Chicago.

The views expressed in this article are the author's only and do not represent directly or imply any official position or view of the Software Engineering Institute or Carnegie Mellon University. This article is intended to stimulate further discussion about this topic.

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University.

® Capability Maturity Model, Capability Maturity Modeling, CERT, CERT Coordination Center, and CMM are registered in the U.S. Patent and Trademark Office.

SM Architecture Tradeoff Analysis Method; ATAM; CMM Integration; CMMI; CURE; IDEAL; Interim Profile; OCTAVE; Operationally Critical Threat, Asset, and Vulnerability Evaluation; Personal Software Process; PSP; SCE; Team Software Process; and TSP are service marks of Carnegie Mellon University.

TM Simplex is a trademark of Carnegie Mellon University.