

Making the Tactical Case for Process Improvement

Watts S. Humphrey

In the December and March columns, I addressed how to make the strategic case for process improvement. It is easier to justify a process improvement program when you deal at a strategic level. Process improvement is a long-term strategic investment, and it is often hard to justify to managers whose principal focus is short term. This column discusses how to handle the shorter term tactical improvement case. We start on the assumption that you are in an organization where the management is focused on tactical issues and where nobody, at least nobody at a senior level, is willing to look beyond the current month or quarter. These managers are generally so consumed by current problems that they cannot consider anything that will pay off in a year or more. While the likelihood of success for a short-term improvement program is low, the situation is not entirely hopeless.

In this column, I talk about tactically based improvement programs and some of the strategies that you can use to bring them off. While there is no guarantee that your efforts will work, there generally are ways to achieve useful results in a relatively brief period. Although you may not succeed, it is better to try something than do nothing. So give it a shot. You might make some useful improvements, and, even better, your success might convince some managers to think and act strategically. This column recommends a series of short-term tactical steps that can lead to a long-term strategic improvement program

The Tactical Situation

In the typical organization, management claims to be thinking strategically. However, imagine that you work for a company in a highly competitive industry that is struggling to improve its quarterly earnings. While process improvement is accepted as a great idea, nobody will invest any significant money in it. What is more, your improvement proposal must be cost justified, and you must show that the costs can be recovered in less than a year without seriously disrupting any project.

Management claims that its goal is to be the industry leader, but the first question they are likely to ask is, “Who else is doing this and what good has it done them?” While you might be tempted to suggest that they sound like followers instead of leaders, restrain yourself and try to think about how to justify this proposal in a way that management will buy. This is the situation. What can you do?

Possible Approaches

While management knows all of the buzzwords, would like to act strategically, and talks about being the industry leader, managers are hypnotized by their short-term problems. Either business realities will not permit a long-term view, or some higher level manager is under severe pressure to show improved quarterly financial results. Under these conditions, you must ignore all the high-sounding phrases about leadership, quality, and improvement, and focus instead on a few pragmatic steps that will fit the current realities.

The only way to break through management's resistance is to somehow demonstrate that the organization's current short-term problems cannot be fixed with short-term Band-Aids. You must take a strategic view. The two principal approaches you can take are to: (1) make the strategic improvement activities tactically attractive, or (2) start a small tactical effort and gradually build it into a strategic improvement program.

Making a Strategic Improvement Program Tactically Attractive

The U.S. Air Force's decision to evaluate bidders based on their process maturity made the Capability Maturity Model[®] (CMM[®]) process-improvement program attractive to many managers. The Air Force evaluations forced many tactically focused managers to invest in process improvement to avoid losing business. This illustrates the advantage of having a customer demand quality improvement: it makes strategic improvement programs tactically attractive.

This strategy will generally work when you have a customer that is interested enough in quality to require that its suppliers commit to improvement programs. If you have such a customer, you can often connect your process-improvement proposal to that customer's demands. If you can get the support of your marketing department, your chances of success are pretty good. While you must keep the scale of your improvement program realistically related to the size of the likely new business, if an important customer insists on a quality program, you probably have a sound basis for a process improvement proposal.

Another approach that is almost as effective is to connect your improvement efforts to an already approved corporate improvement effort. Examples would be obtaining ISO 9000 certification or initiating a 6-sigma software quality improvement effort. Again, if you can show that the improvements you espouse will assist the organization in meeting already established goals, you have a reasonable chance of getting the improvement effort approved.

Build a Small Effort into a Strategic Program

If neither of these strategies work, you probably will not be able to make a strategic program tactically attractive, at least not in the short term. Under these conditions, you must focus on justifying a series of small improvement steps. The suggested approach is to identify one or two narrowly focused efforts that can be completed rather quickly and that don't cost a great deal of money. Then put them into place and use the resulting benefits to help justify taking the next step.

If you are careful about what improvements you pick, and if you build support for each step as you take it, over time, you can probably keep the improvement program moving forward. Then you can gradually convert your short-term tactical activities into a longer term strategic effort.

The critical issue in this situation is getting approval for the initial effort, and then getting the engineers and project managers to support each step as you take it. As long as you can demonstrate that the program is not too expensive and is producing results, and as long as you have the support of the project managers and working engineers, you can probably keep the program going. Then, given a little time, you should be able to show that you are saving the company money and improving project performance. This should allow you to gradually increase the size of the improvement program.

Suggested Tactical Improvement Priorities

In picking improvement efforts, concentrate on activities that are low cost, can be implemented by one or two projects, will produce immediate measurable results, and will attract strong project support. While there are several candidate improvement activities that meet these criteria, the ones that are probably the best bets for organizations at CMM levels 1 or 2 are code inspections, design courses, and the Personal Software Process/Team Software ProcessSM (PSP/TSP)SM. These can all be focused on individual projects and they all support various aspects of a CMM-based process improvement strategy.

The approach is to pick efforts that can be implemented without a broad organization-wide effort that requires senior management approval. Then, if the involved project leaders agree to support the proposal, management will generally go along. Because these improvement efforts can be implemented without requiring changes in the entire organization, they are good candidates for quickly demonstrating the benefits of process improvement programs.

Code Inspections

Code inspections can be put into place quickly, and they pay enormous dividends [Fagan 1976, Gilb, Humphrey 1989]. While design inspections could also be helpful, they take more time and money and don't have nearly as high a payoff—unless you have an effective design process in place. Therefore, it is usually best to defer initiating design inspections.

To start a code inspection program, first find a project leader who agrees to implement the initial trial program, and then train all of the engineers who will do the inspections. In addition, get some member of your staff qualified to moderate the inspections and to help the engineers to do the inspections properly. It would also be advisable to hire an expert to teach the initial courses. In starting an inspection program, a number of available references can be helpful, including Appendix C of my book *Introduction to the Team Software Process*, which describes the inspection process [Fagan 1976, Fagan 1986, Gilb, Humphrey 1989, Humphrey 2000a].

After you complete the first code inspections, you can usually use the engineers who did them as references to help convince the leaders of the other projects. While it will take time to put a complete code inspection program in place, it will provide substantial benefits, and it should give you a firm foundation for further improvements.

Design Courses

Until code quality is reasonably good, design improvements generally will not improve test time or product quality substantially. The reason is that poor quality code will result in so many test defects that the design problems will be lost in the noise. However, once you have code inspections in place, you get substantially improved code quality and reduced test time. That is when design courses would make sense.

While it is almost impossible to cost-justify a design course, you probably will not need to do so. Most people intuitively understand that design is important, and engineers generally will be interested in taking a design course. Start by identifying a project leader who is willing to sponsor the initial test, then find a qualified instructor and get some design courses taught. Assuming that the course is properly done, other projects will want to join in, and demand for the course will grow quite quickly.

The only additional requirement is that you have one or two qualified people available to consult with the engineers, and to advise them on how to use the design methods when they start applying them on the job. Then, after the design methods are in place, the engineers will have the criteria to judge the quality of the designs that they inspect. That is the time to introduce design inspections.

The PSP and TSP

After successfully introducing the inspection program and the design courses, you will have a substantial level of credibility and a modest staff. Then, you can think about tackling a more challenging improvement effort. This would be a good time to get yourself or one of your people trained as a PSP instructor and TSP launch coach [Humphrey 1995]. While this will take a few months and cost a little money, the training is readily available and will enable you to introduce the PSP and TSP into one or two projects in your organization [SEI].

After getting one or more staff members qualified as PSP instructors, look for a trial project. Training volunteers is the easy-sounding approach, but to successfully introduce the PSP and TSP, you must focus on entire teams, including the managers. Once you identify a team and have a qualified PSP instructor available, you can introduce the PSP and TSP in only three or four months. Even though it will generally be several months before you have data on completed projects, TSP's planning and tracking benefits will be apparent very quickly.

To spread the PSP and TSP to other projects, first convince the managers. The material in my March column should be helpful [Humphrey 2000b]. To convince the engineers to participate, get the first TSP team to meet with the potential new team, and have all the managers leave the room. The engineers from the first project will be most effective at convincing the second team to use the PSP and TSP. Once you have a few TSP teams in place, you will have the data, experience, and support to launch a broader based improvement program.

The Commitment System

Assuming that your tactically focused improvement efforts have so far been successful, you can start to move toward a broader based CMM improvement effort. Be cautious about moving too fast and keep your proposals modest until you are reasonably certain that they will be accepted. The suggested next step is to fix the organization's commitment system.

The commitment process defines the way organizations commit project costs and schedules to customers. Improvements in the commitment system generally produce significant benefits very quickly. The basic principles of the software commitment system are well known [Humphrey 1989, Humphrey 2000a, Paulk]. Improving the commitment system is an important early step in a CMM-based improvement program.

Changing the commitment system is much more difficult than anything you have attempted so far. For some reason, managers who make plans before they commit to constructing a building, starting a manufacturing program, or even taking a trip, do not

appreciate the need for planning software projects. Changing the commitment process requires that the managers change their behavior. If you thought changing engineering behavior was difficult, wait until you try to change management behavior. This is why a full-scale CMM-based process improvement program can be difficult to launch.

While the commitment system is probably the most important area to improve, unlike inspections or the PSP/TSP, it cannot be done for only one or two projects. If you can change the commitment system, however, you should be able to launch a full-scale, strategic-based process improvement program. This should include a full CMM-based effort, as well as an expansion of the PSP and TSP to cover all of the development and maintenance projects in the organization.

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 Anita Carleton, Sholom Cohen, Jim McHale, Julia Mullaney, Mark Paulk, Bill Peterson, and Marsha Pomeroy-Huff.

A Note to My Readers

After publishing the March column, I found that I had made a mistake in calculating the maintenance savings in the example. The maintenance numbers were about twice what they should have been. The March column has now been corrected. I hope this mistake has not caused you any inconvenience or embarrassment.

In Closing, an Invitation to Readers

In these columns, I discuss software issues and the impact of quality and process on engineers and their organizations. However, I am 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 in planning future columns.

Thanks for your attention and please stay tuned in.

Watts S. Humphrey

watts@sei.cmu.edu

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.

References

- [Fagan 1976] Michael Fagan, "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal*, vol. 15, no. 3, 1976.
- [Fagan 1986] Michael Fagan, "Advances in software inspections," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 7, July 1986.
- [Gilb] Tom Gilb, Dorothy Graham, *Software Inspection*, Edited by Susannah Finzi, Reading, MA: Addison-Wesley, 1993.
- [Humphrey 1989] W. S. Humphrey, *Managing the Software Process*. Reading, MA: Addison-Wesley, 1989.
- [Humphrey 1995] W. S. Humphrey, *A Discipline for Software Engineering*. Reading, MA: Addison-Wesley, 1995.
- [Humphrey 2000a] W. S. Humphrey, *Introduction to the Team Software Process*, Addison-Wesley, Reading, MA 2000.
- [Humphrey 2000b] W. S. Humphrey, "Justifying a Process Improvement Proposal, *SEI Interactive*, March, 2000.
- [Paulk] Mark C. Paulk and others, *The Capability Maturity Model: Guidelines for Improving the Software Process*. (Reading, MA: Addison Wesley, 1995)

[SEI]

Information on PSP and TSP courses is available on the SEI web site at <http://www.sei.cmu.edu/tsp>.

Copyright © 2000 Carnegie Mellon University

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; Simplex; Team Software Process; and TSP are service marks of Carnegie Mellon University.

TM Simplex is a trademark of Carnegie Mellon University.