

Justifying a Process Improvement Proposal

Watts S. Humphrey



My December 1999 column described how to make the strategic case for process improvement. In this column I provide an example of how to do this. This column thus assumes that you have the ear of a manager or executive who thinks strategically and will consider investments that will likely take a few years to pay off. In the next column, I will talk about dealing with tactically focused managers.

The Financial Justification Process

The financial justification process has five phases:

- Phase 1. Decide what to do.
- Phase 2. Estimate the likely costs.
- Phase 3. Estimate the likely improvement benefits.
- Phase 4. Produce the improvement proposal.
- Phase 5. Close the deal.

The December 1999 column generally discussed these steps. This column walks you through a hypothetical case study where Tom Jones develops a proposal to introduce the Team Software Process (TSP)SM.

Phase 1: Decide What to Do

Tom reviewed the situation in his organization and found that management's top priority was to reduce development cycle time. He decided to do this by introducing the TSP. He also talked to experts about the TSP introduction strategy and found that it had the following 7 steps:

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

- Step 1. Hold an executive seminar for selected top managers and executives from the division or laboratory. Tom estimated that there would be 20 attendees.
- Step 2. Give a one-half day planning session to determine the improvement plan. Tom assumed that 10 of the first-day attendees would participate.
- Decision 1. Tom assumed that these first two phases would be successful but decided to include a decision step to reduce the required initial commitment.
- Step 3. Train the involved managers in the Personal Software Process (PSP)SM and TSP management methods. Tom planned to include the team leaders, the managers of the team leaders, and several other managers below the executive level. He assumed there would be 10 managers in this course.
- Step 4. Provide PSP training to all the engineers who will be on the teams. Tom assumed there would be two teams with 8 engineers per team, or 16 engineers.
- Step 5. Provide general PSP training to any team members who are not programmers. This would be systems, hardware, or test personnel, for example. He assumed that the first two TSP teams would be software only and that there would be 4 system requirements team members in this category.
- Step 6. Tom planned to train two engineers to be PSP instructors so they could support the two TSP teams, handle training any team turnover replacements, and support further TSP introduction. These two instructor candidates would also attend PSP training in step 4, bringing that total up to 18.
- Step 7. The final introduction step Tom planned was to launch the two TSP teams.
- Decision 2. Assuming that these initial team launches were successful, Tom planned to ask management to proceed with broader TSP introduction.

SM Personal Software Process and PSP are service marks of Carnegie Mellon University.

Phase 2: Estimate the Likely Costs

Once Tom defined the proposed introduction program, he estimated its costs in four parts:

- Labor costs
- Internal support costs
- Consulting, training, and external support costs
- Lost opportunity costs

Estimating the Labor Costs

For the labor costs, Tom estimated the number of people to be involved in each introduction steps as shown in Tables 1 and 2. Since the TSP launches in step 7 would be part of the project, however, he did not count them as training time.

Table 1. PSP and TSP Introduction Program Training

| Step | Item | People | Prep. Time | Class Days | Engineer Days | Manager Days |
|------|---------------------------------|--------|------------|------------|---------------|--------------|
| 1 | <i>Executive Seminar</i> | 20 | | 1.0 | | 20 |
| 2 | <i>Planning Session</i> | 10 | | 0.5 | | 5 |
| 3 | <i>Manager PSP Training</i> | 10 | 0.3 | 4.0 | | 43 |
| 4 | <i>PSP Course I & II</i> | 18 | 2.5 | 7.0+7.0 | 297 | |
| 5 | <i>General PSP Course</i> | 4 | 0.5 | 3.0 | 14 | |
| 6 | <i>Instructor/Coach Courses</i> | 2 | | 10.0 | 20 | |
| | <i>Totals</i> | | | | 331 | 68 |

Next, Tom checked with the financial people and found that the cost for a day of engineering time was about \$1,000 and that a manager or executive day cost about \$2,000. While these rates seemed high to him, finance explained that they included all the costs for overhead, support, vacation, medical benefits, sick-time, workmen's compensation, insurance, retirement, Social Security, and taxes. Using these numbers, Tom calculated that the labor costs to train the two TSP teams and their managers would be $\$331,000 + \$136,000 = \$467,000$.

Table 2. Total TSP 5-year Costs

| Cost Item | Cost calculation | Total cost |
|-----------------------------|---|----------------------------|
| Internal introduction costs | \$1,000*331 engineering days \$2,000*68 manager days | \$467,000 |
| TSP coaching costs | 2 engineers*40 days*1 year | \$80,000 |
| External introduction costs | 25% to 75% of internal costs | \$136,750 to \$410,250 |
| Total 1-year costs | | \$683,750 to \$957,250 |
| TSP coaching costs | 2 engineers*40 days*4 years | \$320,000 |
| Turnover training | 3 engineers*14 days*4 years | \$168,000 |
| Total 5-year costs | | \$1,171,750 to \$1,445,250 |

Support Costs

Tom found that each TSP team would need coaching support of about 20% of the time of a TSP instructor/coach. These costs would add about 80 engineer days per year or \$80,000 a year.

Finally, any new or replacement engineers on the teams would have to be trained. Assuming an annual turnover of 20 %, that would be about 3 engineers to train per year at 14 days of training each, or another 42 days of training and \$42,000 per year.

External Costs

The costs for the external instructors and consultants for any improvement program are typically fairly large, but they are generally much smaller than the labor costs. These external costs would include delivering the courses listed in Table 1, launching and supporting the initial TSP teams, several team relaunches, and occasional consultation and assistance.

Without going out for external quotes, Tom assumed that the external support costs for this initial effort would add between 25% to 75% to the first-year labor costs, with the level of introduction cost determined by how rapidly management wanted to introduce the TSP.

Lost Opportunity Costs

Tom realized that while the engineers and managers were being trained they would not be developing or supporting products. To account for these costs, he would reduce the anticipated first-year cycle-time improvement by the 3-week engineer training time.

Phase 3. Estimate the Likely Improvement Benefits

In estimating the improvement benefits, Tom found data on the benefits other organizations had obtained with the TSP, and he also obtained data on the current performance of his organization. Finally, he used these data to estimate the likely improvement benefits for his organization.

Identify Available Improvement Data

Tom learned that Hill AFB, on their first TSP project, increased productivity by 123% over the same team's prior project [Webb]. They also cut test time from 22% to 2.7% of development time, or by 88%.

He also found a presentation that Teradyne had made on their TSP results [Musson]. Through the use of the TSP, Teradyne cut final test and field defects from a rate of 20 defects/KLOC to 1 defect/KLOC. Historically, final test and field defects had cost Teradyne an average of 12 engineering hours each to find and fix.

By using the TSP, Tom also learned that Teradyne had reduced engineering and customer acceptance test time from 9 months for an earlier project to 5 weeks for the TSP. Engineering and acceptance test defects were cut from 5 defects/KLOC to 0.02 defects/KLOC, with no further defects found by the customers.

In addition, he found that a Boeing TSP project had cut test defects by 75% from the prior project average and reduced system test time by 96% [Vu].

Organizational Performance Data

To calculate the TSP benefits for his organization, Tom next needed data on how much time these development groups spent in test, defect levels in the various test phases, and the cost of diagnosing and fixing each defect. With these data, he could then estimate the likely savings from introducing the TSP.

Tom assumed that the 16 engineers on the two trial projects would develop a total of about 80,000 LOC in 12 months. He also found that the time typically spent in integration and system test was currently about 40% of the development schedule and that the defect levels in test and field use were much like those at Teradyne, or 20 defects/KLOC. He assumed that these 20 defects/KLOC were split with 10 in integration test, 5 in system test, and 5 after product shipment. He also assumed that the engineering cost to diagnose and fix these defects was about 1.5 days each.

Estimate the Likely Cost Savings

As shown in Table 3, Tom now estimated the likely savings. First, for the reduction in test defects, he projected that the integration and system test defect/KLOC would be reduced from 15 to 1, for a saving of 14 defect/KLOC. For the total 80 KLOC planned for these two projects, he calculated that this would save $14 \times 80 = 1,120$ test defects. At a cost of 1.5 engineering days for each defect, this would be a reduction of 1,680 engineering days or \$1,680,000.

Table 3. Estimated Savings

| # | Item | Change | Days | Savings |
|----|---|-----------------|-----------------------------------|-------------|
| #1 | <i>Integration/system test defects</i> | - 1,120 defects | $1,120 \times 1.5 = 1,680$ | \$1,680,000 |
| #2 | Test time | - 4.1 months | $4.1 \times 21 \times 16 = 1,378$ | \$1,378,000 |
| #3 | First-year savings (#1 or #2) | | | \$1,378,000 |
| #4 | First-year costs (Table 2) | | | \$957,250 |
| #5 | First-year ROI ($100 \times \#4 / \#5$) | | | 144% |
| #6 | 5-year development ($\#3 \times 5$) | | | \$6,890,000 |

| | | | | |
|-----|------------------------------------|--------------|-------------------|-------------|
| #7 | Project maintenance savings | -398 defects | $398 * 1.5 = 597$ | \$597,000 |
| #8 | 5-year maintenance ($2.5 * \#7$) | | | \$1,492,500 |
| #9 | 5-year savings ($\#6 + \#8$) | | | \$8,382,500 |
| #10 | 5-year costs (Table 2) | | | \$1,445,250 |
| #11 | 5-year ROI ($100 * \#9 / \#10$) | | | 580% |

To check these savings, Tom also looked at test time reductions. The Boeing results showed a test-time reduction of 96% and Hill AFB reported an 88% reduction. Tom assumed that the TSP would reduce his organization's integration and system test time by 85%, or from 4.8 months to 0.7 months for a 12-month project. This 4.1-month saving, at 21 working days per month and 16 engineers, came to a savings of 1,378 engineering days. At the typical engineering day rate of \$1,000, the test time savings were then \$1,378,000. Since this estimate was a little lower than the \$1,680,000 saving based on defect reduction, he decided to use the lower number in the justification calculations. He then compared the \$1,378,000 one-year savings to the maximum one-time improvement cost of \$957,250 to give a one-year return on investment of 144%.

In addition, Tom also felt there would be a maintenance cost reduction. For field and customer defects, the Teradyne data showed a reduction from 5 defects/KLOC to 0.02 defects/KLOC. For the 80 KLOC planned in his organization, he estimated a reduction of 398 customer-reported defects. At 1.5 engineering days each, this would be a maintenance-cost saving of \$597,000 for each year of development work. Since the maintenance savings would not accrue until after product shipment and would be spread over several years, Tom assumed that one fourth of these savings would accrue every year for the four years after each product was developed.

Finally, Tom assumed that the initial two TSP teams would continue to use the TSP on subsequent projects. For the 5-year period, the development savings would be \$6,890,000. In addition, there would be maintenance savings, but since these savings would be spread over 4 years, the first project would accrue \$597,000 savings, the second year project would only save 3/4ths of that, and so forth. Thus, over 5 years, the maintenance savings would be 2.5 times the single-project maintenance savings, or \$1,492,500. The total 5-year savings would be \$8,382,500. Since, as shown in Table 2, the total 5-year improvement cost would be \$1,445,250, this gives a return on investment of 580%.

Cycle-time Benefits

Next, Tom used the reduction in test time to estimate cycle time improvement. With the assumed 12-month project schedule and 40% of the schedule spent in test, normal testing time would be 4.8 months. By assuming an 85% reduction in test time, he estimated that the 4.8 months would be reduced to 0.7 months. Thus, the typical 12-month schedule would be cut to 8 months, or about a 32% cycle-time reduction. For the first year, he also reduced this 4.1-month cycle time improvement by the 3-week initial team training time, for a net of 3.35 months schedule savings.

Phase 4. Produce the Improvement Proposal

At this point, Tom had completed the estimating work and needed to produce a brief management presentation. He decided on the following 7-part outline:

| | |
|------------------------|---|
| Presentation Part 1 | Opening summary. On one chart, he briefly summarized the problem, how he proposed to address it, and the likely benefits. |
| | The problem To improve cycle time and not increase costs |
| | The solution Introduce the Team Software Process (TSP) |
| | Likely benefits A 32% reduction in cycle time |
| | A \$1.4 million one-year savings, \$7,500,000 in 5 years |
| | A \$950,000 one-time introduction cost |
| | About \$500,000 subsequent 4-year support costs |
| | A 1-year ROI of 144% |
| | A 5-year ROI of 580% |
| Presentation Part 2 | The proposal. He briefly described the proposal. |
| | He first asked for a minimum commitment to steps 1 and 2 of the introduction program. |
| | Second, assuming that the first two steps were successful, he proposed to proceed with the 2-team TSP pilot program. |
| | Third, after the two teams were underway and the early experience was |

satisfactory, he planned to proceed with broader TSP introduction.

| | |
|---------------------|---|
| Presentation Part 3 | Description of the TSP. As backup, he prepared a brief description of the TSP and its objectives. |
| Presentation Part 4 | Summary of the TSP benefits obtained by other organizations. Also as backup, he prepared a summary of the available data. |
| Presentation Part 5 | Summary of the introduction plan. As backup, he made a list of the principal phases and decision points in the introduction plan. |
| Presentation Part 6 | Summary of the estimated cost savings. As backup, he made a summary of the cost savings calculations with the key assumptions and supporting data. He also reviewed these figures with finance before the presentation. |
| Presentation Part 7 | Summary of the estimated cycle-time reduction. As backup, he included a summary of the cycle-time improvement calculations with the key assumptions and supporting data. |

Phase 5. Close the Deal

Tom's final step was to make the presentation and get the order.

Closing Comments

While this example is for a specific process improvement method, the principles are quite general. To relate this example to the discussion in the December column, the 5 phases in this example relate to the topics in the December column as follows:

Phase 1. Decide what to do:

Clearly define what you propose.

Understand today's business environment.

Identify the executive's current hot buttons.

Make an initial sanity check.

Phase 2. Estimate the likely costs:

Start the plan with two or three prototypes.

Estimate the one-time introduction costs.

Determine the likely continuing costs.

Phase 3. Estimate the likely improvement benefits:

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.

Phase 4. Produce the improvement proposal:

Produce a presentation to clearly and concisely give this story.

Phase 5. Close the deal.

Hopefully, this example will help you make your own business case. If you have questions, I suggest you look at the December 1999 column which contains a more generic discussion of these same topics.

Stay tuned in

In the next issue, we will discuss the issues of convincing tactically-focused managers and executives to start a process improvement program. Following that, a subsequent column will deal with 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 Sholom Cohen, Frank Gmaldi, Julia Mulaney, Jim Over, Mark Paulk, and Bill Peterson.

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. I am, however, most interested in addressing issues 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 in.

Watts S. Humphrey
watts@sei.cmu.edu

References

- [Musson] Robert Musson, presentation at the 1999 Software Engineering Institute Symposium, Pittsburgh, PA, August 30 to September 2, 1999.
- [Vu] John Vu, presentation to the U.S. Department of Defense on the Boeing process-improvement program.
- [Webb] Dave Webb and W. S. Humphrey, "Using the TSP on the TaskView Project," *Crosstalk*, vol. 12, no. 2, February, 1999, pp. 3 - 10.

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 Process*SM (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.

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.