

Configuration Management Culture as the Kernel to Success in Software Process Improvement Efforts

Thomas C. Green and Kenneth M. Anderson

University of Colorado, Boulder {tomg,kena}@cs.colorado.edu

Abstract. For a Software Process Improvement (SPI) effort to succeed, its participants must have a sense of ownership. One practical technique for achieving that sense of ownership is to apply a meta-process based on the principals of configuration management (CM) to the SPI effort. This paper provides insight into issues of ownership surrounding actual SPI efforts and describes the use of a CM-based meta-process that successfully supported one of these efforts.

1 Introduction

Many software organizations have attempted Software Process Improvement (SPI) efforts in order to achieve higher quality products and more efficient development. Few succeed, although some achieve a measurable level of success. Of those that succeed, few are able to hold onto that success. Why do software organizations fail to make beneficial improvements? Why are some organizations unable to hold onto these improvements?

Under the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) there are five levels of maturity [Hu1]. Studies have shown that the median time to move from CMM Level 1 (Initial) to Level 2 (Repeatable) is 25 months of improvement effort. However, it is not unheard of for this transition to take over six years. Another 22 months is the median for movement from Level 2 to Level 3 (Defined), with some organizations requiring over four years. For software groups where data is available, 18.8% remain at the same level or slip back a level at their subsequent assessment [SE1]. The percentage that slip back from Level 2 to Level 1 reported in the study is likely smaller than actual. Data is available only from groups that voluntarily report their data to the SEI. If an organization slips, they may not be inclined to report their results.

Some organizations, however, are able to improve their processes faster than the norm. Furthermore, some organizations never slip back to their old ways. Why is it that some groups are so successful, while others fail?

One key factor is management commitment. If an organization's managers are not committed and supportive of the required software process changes, then the improvements are doomed to failure. Our purpose in writing this paper is to suggest a deeper reason as to why management commitment helps make SPI efforts succeed. That reason is ownership of the SPI effort by all stakeholders:

management and individual contributors (ICs). The ownership by management bolsters their commitment to make the SPI effort successful. The ownership by ICs allows them to accept the changes to the ways they must perform their work. In this paper, we present insight into actual SPI efforts and draw from this experience a process that can help SPI participants gain a sense of ownership and control over the effort, and help increase their chances of success.

2 Background

The first author was personally involved in SEI/CMM-based SPI efforts in several organizations. He experienced these efforts from a range of possible stakeholders: as an individual contributor, a software group manager, a project manager, and as an SPI consultant. Each organization started their SPI effort with the goal of achieving CMM Level 2. As such, they each began by creating documents to support Requirements Management (RM), Software Project Planning (SPP), Software Project Tracking and Oversight (PTO), Software Configuration Management (SCM), Software Quality Assurance (SQA), and Software Subcontract Management (SM).

These processes are supposed to be documented “as is” (i.e., the current methods being used to develop software at the organization are captured). Unfortunately, what is typically created is something beyond “as is”. The process documenters start out intending to document the “as is” process. Yet, as they see the inherent problems, they add text describing what *should* be done. These additions to the process, however, have not been proposed or approved by the rest of the organization. The stakeholders who need to have ownership of these types of changes are excluded, however unintentionally, from the documentation process.

3 Meta-process

In these situations, a “process for the process”, or meta-process, is needed. In such a meta-process, a procedure is defined for the constant improvement of the processes of the organization (See Fig. 1). The organization needs to recognize this meta-process to achieve success. As an organization matures, it recognizes the importance of controlling work products via configuration management (CM). Requirements, design, code, and test documents all need to be under CM control. The earlier the organization understands the SPI effort also needs to be under CM control, the faster and more effective the SPI effort will be.

Applying software development techniques (in this case, CM) to software processes is not a novel idea given Osterweil’s assertion that “Software Processes are Software Too. . .” [Os1], and Sutton’s suggestion to use CM to support software process [Su2]. However, it is by no means obvious how configuration management techniques can be applied to SPI efforts in a practical fashion. Our experience provides insight into how any organization can incorporate CM techniques into

their SPI efforts and how these techniques can instill a sense of ownership in an SPI effort by its stakeholders.

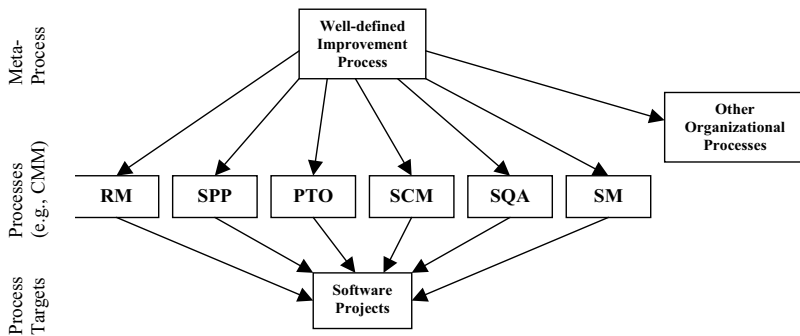


Fig. 1. Relationship of meta-process to regular processes

Processes are not static. Nonetheless, many early SPI efforts start by documenting their processes as if they are. The mere intention of the group to start an SPI effort implies their desire to change. Yet they start the SPI effort without any methods in place to allow change to the processes they are creating.

The first time a change is made to a process document in these organizations, problems start appearing. Not coincidentally, these problems are similar to the difficulties encountered by organizations that do not use CM controls on their software work products! For example, who is authorized to suggest changes to the process documents? Who is authorized to approve a change? Who can decide what the actual changes will be? What are the risks involved in making the change? Once a change is approved, who is authorized to make the change? Where are the new process documents stored? Where are previous versions kept? Who needs to be notified of the change? Will training be involved as a result of the change? When are ICs expected to implement the change?

These are all questions that need to be answered for software documents as part of a mature CM process and, we argue, they also need to be answered for SPI documents. At one of the first author's organizations, process documents were not kept under CM control. Typically the original author of a document made changes at his or her discretion, and then notified *some* people of the changes. Soon, different people in the organization had different versions of the process documents. Since various members of the organization were following different processes, the organization found itself back in the chaos of CMM Level 1!

4 Implementing the Meta-process

In another of the first author's organizations, a change management system for all process documents was instituted. All documents, including checklists, procedure

manuals, and the like, were placed into PVCS, a commercially available source management system. With this system in place, all stakeholders who needed to refer to a process document retrieved the latest version of the document from the process document repository. In addition, the latest version of each document was made available on the organizational intranet. This action alone solved the problem of ensuring that all stakeholders had access to the correct version of a process document.

The next step involved ensuring changes to a process document in the repository could not occur without prior authorization. This step was straightforward since PVCS already contained this feature.

Next, the organization needed a method to make change requests (CRs) on the process documents. To handle this task, the organization adapted their existing defect management system. Process documents were treated like any other software artifact, and any stakeholder could enter CRs against any process document. It was also possible to enter a more general request, for example, for a new type of process document.

This organization also implemented a defect review process via a Configuration Control Board (CCB) for their software documents. This system was applied to process documents. Weekly meetings were instituted to review process CRs. In this organization a single meeting was possible for all process documents. Depending on group size and expertise, several different review meetings might be held in other organizations.

The Process Configuration Control Board (PCCB) then reviewed each CR. If additional information was required it was assigned to an accountable person to research and/or retrieve for a future PCCB meeting. Each CR was given a disposition, similar to software document defect dispositions (e.g., rejected, approved, waiting for information, on hold) Representative process users needed to participate on the PCCB to ensure users accepted the change(s).

If the process document was approved for change, someone was assigned to make the change. The change was then inspected, via a peer review process, and given approval for check-in to the process document repository. At this point, if stakeholders always referred to the repository for their tasks, the changes would readily be incorporated into the organization's day-to-day work. Unfortunately, this ideal situation does not occur in practice. Thus, the final step was to notify stakeholders affected by the change, providing details and/or training.

The above meta-process can be easy to implement for any organization already using good CM procedures. It requires the management of an additional set of documents, using existing CM processes, for process changes.

For those organizations not currently following good CM procedures, it is recommended that the CR submission process closely follow the culturally accepted methods for making changes. This makes the new CR procedure more palatable, and it can be improved, over time, with the meta-process.

5 How to Start a Process Improvement Effort

It is our belief that this meta-process is an excellent candidate for being the first process adopted by an organization when starting an SPI effort. In summary, start by creating a source repository for process documents. Place whatever process documents the organization has, if any, into the repository. Then define a meta-process to guide the addition and/or improvement of the process documents, as discussed in the previous section. This improvement process should be monitored with a peer review process (which is defined in a process document, stored in the repository, and improved by the meta-process). It is our contention, that with this simple kernel in place, any software organization can achieve process improvement over time. It is our further contention that this improvement can be accomplished under any improvement model desired (e.g., SEI/CMM, SPICE) and even with no model at all.

Organizational change is not easy. To change, the culture has to change. In large organizations this is difficult, since culture change requires changes by many different people. By making the meta-process the start of a process improvement effort, the cultural change can be smaller. The organization begins by creating a culture that supports change. Then the SPI effort can slowly introduce the changes that are needed to improve the quality of the organization's software development processes. Once an organization has the meta-process in place, then anyone in the organization can make suggestions for improvement. Furthermore, these change requests will have a place to go, will be reviewed, and if changes are made, affected personnel will be notified.

6 Achieving Early Ownership

The first author has developed informal steps for beginning an SPI effort. In particular, each stakeholder is asked to submit at least ten CRs about how the organization's process should change. These CRs should contain a recommended solution, if possible, to help speed the review process. As a manager, the first author found most stakeholders have *at least* ten issues about how poorly their organization is run!

In applying this method, an interesting result occurs. Management is able to say things like "Don't complain, submit a change request!" This helps instill ownership of the process across all stakeholders right from the start. Everyone must take on the ownership and responsibility of how the organization operates and solves its process problems.

Having ten CRs submitted from every member of the organization, managers and individual contributors included, causes an initial avalanche of suggestions. This flood of activity jump-starts the improvement process and sets the stage for a culture change. It also places responsibility on management to take CR submissions seriously. Management must allow for the effort to digest the suggestions as soon as possible or the SPI movement will die. Stakeholders must see changes occurring and suggestions implemented, or their requests will cease.

Once the meta-process is visibly operating, stakeholders submit CRs whenever they notice something did not execute optimally. Managers submit CRs when major defects appear. ICs submit CRs when they do not get the work products they need to perform their jobs. This meta-process, once started and ownership by all stakeholders has been established, is self-correcting. For instance, the meta-process itself receives CRs and is improved. Even with organizations that do not follow any particular process improvement model, the changes implemented over time will move them to higher levels of effectiveness.

A useful meta-process metric is to note the number of CRs submitted per unit time (e.g., week or month). A healthy meta-process will continue to have CRs submitted as the organization matures. If the CRs per month drops significantly, it indicates a drop in ownership.

Another metric to check is the number of CRs implemented to the number of CRs submitted. A low ratio indicates inflexibility or a slow review process.

If an organization attempts to follow a particular process improvement model, the meta-process can help accelerate improvements. With a meta-process established, it is possible to submit CRs like “We need a Software Quality Assurance Procedure (SQAP)”, or “We need a template for software requirements documents, draft attached”. Following a model like the SEI/CMM, one can submit CRs that will help the organization follow the next step in the model. For SEI/CMM Level 2 that would include requests for documents and procedures that implement RM, SPP, PTO, SQA, SCM, and SM.

7 Conclusion

In this paper, we describe the experience, and lessons learned, of the first author in participating in actual SPI efforts. This experience has provided insights into how to start a practical SPI effort, based on configuration management techniques, that instills a sense of ownership to the overall process. We have argued that this sense of ownership is critical to the success of the effort. We intend to expand on this work in the future by developing a detailed set of heuristics, with accompanying tool support, to study and make software process improvement techniques more practical and easily applied by software organizations.

References

- Hu1. Humphrey, W.S. (1989). *Managing the Software Process*. Addison-Wesley, Reading, MA, USA.
- Os1. Osterweil, Leon J. (1997). Software Processes Are Software Too, Revisited. In *Proceedings of the Nineteenth International Conference on Software Engineering (ICSE 1997)*, pp. 540-548. May 17-23, 1997. Boston, MA, USA.
- SE1. Software Engineering Measurement and Analysis Team. (2000). *Process Maturity Profile of the Software Community, 2000 Mid-Year Update*. Software Engineering Institute.
- Su2. Sutton, Stanley M., Jr. (2000). The Role of Process in A Software Start-up. *IEEE Software July/August 2000* pp. 33-39.