Patterns for Conducting Process Improvement

by **Brad Appleton** *<bradapp@enteract.com>*
http://www.enteract.com/~bradapp/

Copyright © 1997 by Brad Appleton.
Permission is granted to copy for the PLoP '97 conference.

Abstract: Process improvement shares many common characteristics with product development. Recognizing these similarities is important, but so is recognizing some of the crucial differences. It is vital to the success of improvement efforts to realize that process change entails cultural change. Numerous social and technical barriers must be overcome to effect lasting improvement. Ten patterns of software process improvement are described which illustrate some important similarities and differences between process improvement and product development.

Keywords: *Patterns, Process Improvement, Software Process*

Introduction

This paper attempts to capture some of the successful best practices observed by the author during his experience with several process improvement initiatives (both as a change agent and as a change target). Instances of these practices have been observed to recur throughout the published literature on software process improvement. They are described here as "*patterns*": named nuggets of insight conveying battle-proven solutions to recurring problems, each of which balances a set of competing concerns. These patterns are part of a growing collection of software process improvement patterns named "***I-SPI***" (pronounced "I Spy"), an acronym for **I**nitiating **S**oftware **P**rocess **I**mprovement.

Definitions and Acronyms

Several acronyms are used throughout the process improvement literature. The acronyms used here are some of the more frequently recurring terms which are *not* specific to a particular process improvement framework (like the SEI CMM or ISO 9000). These acronyms are:

SPI Software Process Improvement

PIT Process Improvement Team -- other acronyms used are PWG (Process Working Group), and sometimes even SEPG (Software Engineering Process Group), though this latter acronym is more often used in place of PEG (see below).

PEG Process Engineering Group -- the SEI Software CMM uses the acronym SEPG (Software Engineering Process Group).

IAT Improvement Action Team

The Problem of Process Improvement

Process improvement affects more than just the processes used by practitioners to perform their work. Process improvement efforts disrupt delicate ecosystems deeply rooted within the community. Process change means *culture change*, replete with all the difficulties inherent in changing the perceptions, values, and normative behaviors of a community. Some of the forces that make such improvement efforts difficult are:

- Resistance to change (often due to a perceived threat of losing power, control, familiarity, or social status)
- The speed at which groups and individuals are capable of assimilating change without suffering from dysfunction (see *[Conner]*)
- The existing tolerance and readiness for change present within the current organizational climate
- Process change imposes a learning curve which typically makes things appear to get worse before they get better.
- Improvement efforts consume time and resources which many would prefer to spend on their particular development projects.

The patterns presented below address some of these issues and may be categorized into patterns of organizational structure, and patterns of process and communication as follows:

Organization Patterns	**Process and Communication Patterns**
Local Heroes	Process is Product *(process)*
Center PEG	Virtual Forum *(communication)*
PIT also Practices	Process follows Practice *(process)*
Dedicated Improvement Processors	Improvement follows Process *(process)*
Improvement Action Teams	Improvement follows Spiral *(process)*

These patterns are by no means a complete set of solutions for conducting process improvement initiatives. Their repeated success, however, has been documented throughout the published literature on process improvement. Many of the issues left unresolved or unaddressed by these patterns are discussed within their resulting contexts.

Applicability

These patterns are applicable for use by an organization or department in which management has already made a genuine commitment to sponsor and support process improvement efforts. It should be noted that for many groups attempting process improvement, obtaining senior management commitment is a herculean task. However, this issue is not addressed here. Furthermore, it is assumed that the process goals or assessment criteria have already been determined. Among the more popular criteria used are: ISO 9000, the SEI Software CMM, and SPICE. The SEI Software CMM and ISO 9000 were used most often within the authors personal experience, and among the published software process improvement literature.

The size of the groups which the author directly observed using these patterns ranged from 7 people to about 70 people, and encompassed anywhere from 1 to 10 software project teams within the group. The size of the individual project teams were between 2 and 12 people. Such groups and projects would typically be considered in the small to medium-small range. However, published literature would suggest that the observed patterns scale quite well to larger groups (perhaps with some slight variations).

Pattern ***Process is Product***

Context Senior management has committed to sponsor and support improvement efforts. You are responsible for mobilizing people and resources to try and make it happen, but process improvement is a new endeavor for your group and you're unsure how to get started and get organized.

Problem How should a process improvement initiative be organized and managed?

Forces It would be preferable to use a project management infrastructure which is well established and familiar. Yet it seems that most of your SPI efforts will be spent trying to change the way people perform activities, not engineering a software system. It is uncertain that a development project management infrastructure is appropriate for a process improvement initiative. On the other hand, if the project is not treated as projects, it may not receive the necessary consideration (and respect) from practitioners and from upper management (particularly at status and budget meetings).

Solution *Treat it like a development project!* Recruit a corresponding project team (often called a Process Improvement Team or PIT), select a project leader, and establish a repository to store process documentation and other process artifacts. Use appropriate planning, tracking, configuration management, and other methods and tools, just as they should be used for any other development project. Ensure that the visibility of the project to upper management and the rest of the organization is comparable to that of other important projects.

Resulting Context The project management infrastructure for process improvement is like that of development projects. It uses of the same kinds of methods and tools for planning, monitoring, and maintenance, and has appropriate visibility to garner the necessary resources and support from management. This not only lends uniformity to the way various kinds of projects are managed, but also helps legitimize improvement efforts so they are taken a bit more seriously by the entire organization. Whether or not the existing project management infrastructure is capable of handling most improvement-related project management issues is an open question. Important differences between the two kinds of projects haven't yet been addressed. The pattern *Process follows Practice* addresses this by recognizing the kind of development project that resembles a process improvement project.

Rationale Use of this pattern signifies the important realization of the congruence between process development and product development. A process *is* a product! Its end-users are the practitioners who will be required to perform the process. The documentation of procedures, policies, and standards are part of the finished product. Tools and training material are also part of the end result. So are any internally developed applications to automate procedures and integrate various development tools. Thus, an infrastructure similar to that used by software projects is in order.

Related Patterns By equating the contexts of process and product development, it becomes valid to consider numerous project management patterns in the context of conducting process improvement. Patterns from sources such as *[PLoPD1]* and *[PLoPD2]*, may now be applicable.

Other process improvement patterns which follow from this one:

- A *Virtual Forum* may be created during the course of setting up the process development repository and other improvement project resources.
- *Local Heroes* will help you staff the PIT appropriately.
- *Process follows Practice* recommends the initial set of activities the PIT should undertake.
- *Improvement follows Process* and *Improvement follows Spiral* discuss the process of conducting process improvement efforts.

Known Uses *[Osterweil1]* and *[Osterweil2]* describe how "*Software Processes are Software Too*." *[Wiegers]* remarks that treating SPI efforts like a project with mini-projects for each SPI increment was key to the success of Kodak's SPI efforts. *[Grady]* expresses the same sentiment of SPI efforts at Hewlett-Packard. *[Austin,Paulish]* and *[Curtis]* point to multiple SPI case studies which reached the same conclusions. *[Wakulczyk]* explains how treating improvement efforts at NORAD with the same careful management and visibility as major projects helped such efforts gain the respect they deserved. *[Larner]* states one lesson learned in SPI efforts at Lloyds Bank was to set up and run each process improvement project like a development project using existing project management standards. Another SPI case study by *[Radice]* states that treating improvement initiatives in this manner was not only a key characteristic of success, but that neglecting to conduct SPI projects this way was a recurring characteristic of failed efforts.

Pattern ***Virtual Forum***

Context You are setting up a process improvement project infrastructure using *Process is Product*. The PIT requires a method of regularly communicating with process stakeholders to communicate project status and to solicit feedback and participation.

Problem How do you periodically discuss improvement efforts with practitioners without scheduling numerous group-wide meetings that regularly compete with (and often interrupt) their product development tasks?

Forces You want to keep the entire practitioner community informed of the status of improvement efforts. It is also desirable to solicit input and feedback on process improvement issues from the practitioner community. Normally this would require regular meetings, but practitioners typically dislike attending meetings (especially ones that don't appear to directly contribute to completing their development tasks). Furthermore, coordinating schedules to accommodate everyone can be an intractable problem. Memos and notices can easily be widely disseminated, but are sometimes too formal and don't encourage reciprocal dialogues (which are needed to foster practitioner participation in improvement efforts).

Solution Create a group-wide discussion forum using a two-way communications medium that is already in wide use (local newsgroups, intranet, notes, etc.). Make sure messages on the forum are automatically archived and backed-up. Announce its availability to (and for) the entire practitioner community and encourage its use for asking questions, contributing feedback, voicing concerns, and issuing complaints. It is essential that practitioners feel they may communicate freely without fear of punishment or retaliation. But electronically disseminated text is an imperfect means of communication: it doesn't accurately convey tone, facial expression, or body language. This can be amplified by the fact that participants are somewhat removed from each other (instead of being face-to-face), resulting in occasional misinterpretations and even heated disagreements ("flame wars"). A set of guidelines (or a policy) for avoiding and resolving such predicaments should be clearly established up front.

The intrusiveness of the implementation used must be carefully considered. Virtual forums which are implemented as a selectable "channel" (like an intranet chat room, bulletin board, local newsgroup or voluntary email list) are less intrusive, but won't necessarily reach practitioner's who have chosen not to "tune in" to the channel. A virtual forum implemented as a group-wide email alias doesn't require such voluntary "channel selection", but the fact that all such email goes to the same incoming mailbox as each recipient's other email messages (unless they use customized email filters) may be regarded as a nuisance to many who simply don't wish to participate.

Use a medium already in widespread use, but do *not* force participation down people's throats. A less medium involving voluntary selection or subscription might not be seen by everyone, but those who choose to tune it out are likely to the same people who would be greatly annoyed by a less voluntary medium (and hence unlikely to participate either way).

Resulting Context Improvement efforts can be presented and discussed without having to coordinate schedules for numerous group-wide meetings. Face-to-face group meetings are still needed for important occasions, but their frequency is greatly reduced. Issues can be disseminated to the entire community at once while being as informal as desired, permitting high-frequency two-way communication between the PIT and its customers. Human contact is not replaced with technology, but is instead augmented by it to conveniently increase communication (especially across remote sites). In addition, the forum's electronic archive serves as a kind of improvement project "memory", preserving important historical comments that may be easily recalled.

Although messages or "postings" to the virtual forum may reach the entire practitioner community, there is no guarantee that each individual will take the time to read the messages, much less respond to them. Recipients are not forced to involve themselves in improvement activities, but it is very convenient for them to do so. More importantly, practitioners are provided an opportunity to stay abreast of improvement efforts and to actively voice their feelings and concerns throughout the entire improvement process.

One risk in using a virtual forum is that it may replace other modes of informal communications. Do *not* let the virtual forum completely replace meetings at the group and individual levels. The main purpose of the forum is to address the lack of attendance that would result if frequent meetings were to be scheduled. The electronic forum is not capable of conveying the same richness of expression as a face-to-face dialogue (there is no satisfactory replacement for direct human interaction).

Related Patterns *Involve Everyone [Delano,Rising]*, *Informal Networks [Dikel,Kane]*, and *Engage Customers [Cope]* are all similar to this pattern. *Brown Bag [Delano,Rising]* suggests another way of solving this problem that may be applied in concert with *Virtual Forum*.

The virtual forum may be used to facilitate and coordinate communication between the *Improvement Action Teams*, the PIT(s) and the *Center PEG*. It may also be used to solicit participation and feedback during *Process follows Practice*.

Known Uses The use of an electronic forum to facilitate dialogues for garnering SPI input and feedback are described in *[Austin,Paulish]*, *[Baumert]*, and *[McLane]*.

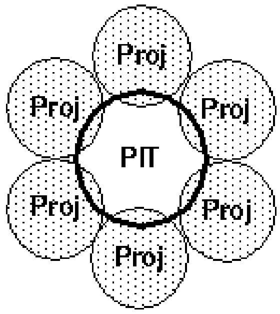
Pattern ***Local Heroes***

Context You must assemble the process improvement team (a.k.a the PIT -- see *Process is Product*). The pool of candidates from which to select team members contains people internal to the organization as well as those external to the organization (experienced consultants and prospective new-hires).

Problem How do you staff the PIT with members who can effectively lead the practitioner community in accepting and adopting process changes?

Forces Process experts are often perceived by practitioners as being steeped too much in theory and not enough in practice. It is desirable to have people who are not only knowledgeable about the software processes, but also about the reality that practitioners face in the trenches on a daily basis. Many competent practitioners may not have the requisite software process quality experience. Experts external to the organization may possess the requisite knowledge and experience, but may be viewed as outsiders by key practitioners (whose trust and respect are essential to gain inroads into the development community).

Solution Compose the PIT primarily of venerated veteran practitioners throughout the development community. These people should be "all-stars in the family": respected members of the organization with proven track records as developers or managers (cf. *Domain Expertise in Roles [Cope]*). Ideally, these talented individuals have leadership potential: when they talk, their peers tend to listen and are more likely to follow their example.



If possible, try to recruit members from all development project teams so that each project has at least one voice to represent its interests. However, don’t sacrifice experience and respect at the expense of equitable representation. When push comes to shove, it is more important to have people who are highly regarded by their coworkers than to have at least one member from each project team. If you have to make a compromise, go with the more influential individuals. This advice is borne out both by the author’s personal experience as well as in published case studies of process improvement efforts.

Some project managers will resist having their best and brightest take time away from their project commitments in order to participate in improvement efforts. The manager propose send someone else instead; someone who they feel is more "expendable" and who typically has much less influence in the development community. These less influential individuals may be highly competent, but they do not yet command the degree of respect needed to sway their coworkers to giving various improvement ideas a fair chance to succeed. Emphasize the importance of having the "local hero" be part of the PIT and try to hold out for the "real thing" if you can afford it (this is another one of those times when senior and middle management support may be needed).

Resulting Context The PIT is both socially and technically aligned with the practitioner community. PIT members have intimate knowledge of development issues, and their deeds and words are respected within the development culture. These people are capable of leading the way because the practitioners know they have "been there, done that," and done it well.

It is assumed that a sufficient number of such people exist within the organization. If they do not, or if their numbers are scarce, then it may not be possible to employ this pattern. Other patterns are needed to fill in this gap. One alternative would be to use as many people as possible from within the organization and use external experts to fill the remaining slots.

Motivating individuals to participate in the PIT is not addressed by this pattern (*Unity Of Purpose [Harrison]* may prove useful here). Patterns for establishing suitable rewards and reinforcement to encourage such participation are also needed (see *Compensate Success [Cope]* for one example). Congruent action (see *[WeinbergV3]*) must somehow accompany the creation of such rewards to ensure that practitioner's commitments are adjusted to allow appropriate time and training to participate in improvement efforts. This requires committed support throughout the management hierarchy to enlist the cooperation of product-line managers and middle-managers to lighten workloads. Gaining such cooperation from management is another issue not directly addressed here (other patterns are needed for this as well).

Related Patterns This pattern shares many elements with *Domain Expertise in Roles [Cope]*, *Case Team [Beedle]*, as well as *Grass Roots* and *Local Leader [Delano,Rising]*.

Size the Project [Cope] may help to determine how many people should serve on the PIT. *Center PEG* describes how to "Scope the PIT" for SPI initiatives that encompass a very large group of people.

PIT also Practices and *Dedicated Improvement Processors* discuss how much time should be devoted by members of the PIT to SPI efforts.

Known Uses *[Curtis]* cites SPI case studies which demonstrate the importance of staffing SPI initiatives with respected leaders (developers and managers) who are among the best and brightest in the organization (as opposed to using merely competent people with less influence in the community). *[Donaldsen,Siegel]* support these findings, as do *[Fowler,Rifkin]*. *[Wakulczyk]* describes how having an SEPG filled with peers greatly facilitated their ability to gain trust from, and stay attuned with, the developers at NORAD.

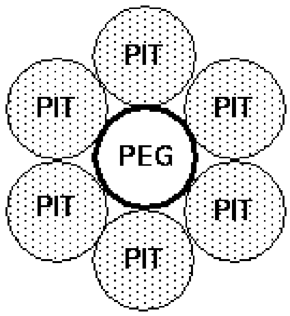
Pattern ***Center PEG***

Context The group undergoing process improvement is large enough that one PIT simply can't serve it effectively (or else a PIT of extremely large and unwieldy size would be required).

Problem How do you manage multiple PITs for a large organization?

Forces A single guiding coalition *[Kotter]* is desirable to maintain conceptual integrity and consistency of the improvement project. However, forming a single PIT using *Local Heroes* to obtain equitable representation from the various projects and departments would vastly exceed a manageable team size (such as that recommended by *Size the Project [Cope]*). Forming several smaller teams of more manageable size is possible, but could give rise to a combinatorial explosion in efforts between PITs trying to coordinate all their efforts. Furthermore, if multiple PITs are created, the question arises as to which PITs have authority over the others.

Solution Create a special PIT (often called a Process Engineering Group or PEG) to be as a center of guidance and support for the other PITs. Each PEG member will typically work full-time on process engineering and improvement (see *Dedicated Improvement Processors*). The PEG is the primary authoritative body for conducting and organizing improvement efforts in the organization.



A common implementation of this pattern is to have local PITs address the entire software process in their group or department (*PIT per subgroup*). An alternate implementation that is also commonly encountered uses local experts across the organization to assemble several process working groups, each of which focuses on a particular core competency or key process area (*PIT per core competency*).

Another variant of this pattern used by extremely large organizations is to add another level: a dedicated business unit serves as the PEG for the entire corporation, while each sector or division assembles its own sub-PEG to coordinates multiple PITs.

Resulting Context The PEG becomes a central hub of planning, communication and expertise for coordinating process improvement efforts among multiple teams throughout the organization. Both the PEG and the PITs are typically used throughout the entire lifespan of the improvement initiative.

Related Patterns This pattern bears a passing resemblance to the *[GoF]* design patterns *Mediator* and *Observer*. However, it has much more in common with *Hub, Spoke and Rim* from *[Cope]*, and *Sub-Architect [Beedle]*.

Local Heroes should be used when forming the PEG. *PIT also Practices* and *Dedicated Improvement Processors* suggest strategies to use for the other subordinate PITs (the latter is usually employed for the PEG as well).

Known Uses *[Haley]* writes of an executive SEPG steering committee used to coordinate four different process working groups at Raytheon. *[Dorsey,McDonald]* describe GTE's improvement successes using a central SEPG to coordinate 3-5 person process action teams, each devoted to a specific key process area. *[Austin,Paulish]* mention ISO 9000 improvement efforts at DuPont using a central process steering committee to coordinate multiple process subteams. *[Donaldsen,Siegel]* also describe the use of a central PEG to lead and coordinate improvement efforts of part-time PITs.

Pattern ***PIT also Practices***

Context You have identified a sufficiently sized pool of individuals from which to form a PIT using *Local Heroes*. Now you need to estimate and request resources, many of which hinge upon the amount of time you will be asking each PIT member to regularly commit to SPI activities. The amount of time you request of each person may also determine whether or not that person will be able to participate in the PIT.

Problem How much time should PIT members devote to improvement efforts to make reasonable progress without becoming completely detached from rest of the practitioner community?

Forces Part-time members might be unable to contribute the necessary time and resources to get the job done. It would be preferable to have people who can dedicate the majority of their time to SPI efforts, but it may not be feasible to pull some of the best practitioners off their current projects to work on SPI. It may be that you have a small group of people who don't really have the resources to devote any one person to work full-time on SPI. Or it may be the case that many leading practitioners are simply too important to their development projects to risk being removed from them. Those who currently participate in product development are desirable because they are intimately aware of the existing corporate culture and the development community.

Solution Have PIT members spend 10%-20% of their time on process improvement while still working on their current development projects. Make sure their schedules are adjusted to permit time for improvement-related activities. This requires cooperation from corresponding project managers, which in turn may require the support of middle and senior management. If possible, try to have at one or two PIT members devote 50%-100% of their time to handle the managerial and administrative overhead for coordinating improvement efforts.

The question still stands as to whether or not it is realistic to expect to accomplish SPI with a part-time team. Four or five hours per week to devote to improvement efforts isn't very much, particularly if PIT meetings are held on a periodic basis. Reserving eight to ten hours per week for PIT members to spend on SPI is more realistic, provided their workloads can be adjusted to accommodate PIT activities.

Resulting Context The PIT is socially connected with the practitioner community. PIT members have an intimate and ongoing awareness of the development culture because they are still a part of it. The other project teams have a representative voice in the PIT and vice versa. PIT members may not have as much time to spend on SPI as one might like, but the active connection to the practitioner community increases the likelihood of process changes being adopted and accepted (since PIT members are still considered by them to be a respected part of the family). The inherent risk is that without adequate management sponsorship and support, the time devoted by part-time members to SPI efforts may disappear whenever a crisis arises. This would jeopardize the continuity and conceptual integrity of improvement efforts. This risk is partially addressed by devoting 1-2 people half-time or full-time to SPI (but the threat may still exist). Patterns which further mitigate this concern are sorely needed.

Rationale There are those who warn against committing people only part-time to improvement efforts (*[Grady]* is one of them). Certainly there is some amount of truth to the "no pain, no gain" attitude that taking a "hit" early on will pay off in the long run. But the fact of the matter is that many groups (particularly the smaller groups attempting improvement efforts) simply can't afford the kind of up front investment required to devote most PIT members full-time to improvement efforts. Either the resources simply are not available to begin with, or else the initial damage would be too great to permit the group to sustain itself.

In these cases, a compromise is required. There is nothing wrong with taking "baby steps" if that is all you presently afford (in fact, *[Wiegers]* recommends it). Improvements may take longer to accomplish, and one still needs to worry about dwindling time commitments. But it is far better to progress slowly and to reach the goal than it is to overcommit and to fail. The resulting failure will make the next attempt doubly difficult if people have been soured against the idea from a previous bad experience.

Related Patterns In some respects, this pattern is the process development equivalent of *Architect also Implements* from *[Cope]*. It also has characteristics in common with *Grass Roots [Delano,Rising]*.

Dedicated Improvement Processors describes the other extreme to solving this problem. It may be combined with this pattern to obtain a mix of full-time and part-time PIT members.

Known Uses *[Wakulczyk]* describes success with a software engineering process group (SEPG) composed of two full-time members and thirty-two part-time members. *[Wiegers]* mentions the use of ongoing "mini-project" teams (PITs coordinated by a central PEG) consisting predominantly of part-time members to successfully implement process improvements at Kodak. *[Dorsey,McDonald]* describe their SPI success at GTE using a central PEG to coordinate Process Action Teams (PATs) composed of 3-5 part-time members. *[McLane]* mentions DEC's use of an extended SEPG (or ESEPG) composed of part-time members and states how they were vital to piloting and implementing changes within their projects.

Pattern ***Dedicated Improvement Processors***

Context See *PIT also Practices*.

Problem See *PIT also Practices*.

Forces See *PIT also Practices*.

Solution Require PIT members to dedicate their efforts full-time to the improvement process. Make sure they regularly spend time helping development projects with process-related and quality-related aspects of their work, such as: preparing and implementing plans for project management, configuration management, requirements management, etc., as well as other documents or reports. Thus, in addition to conducting process definition and improvement efforts, PIT members serve as hands-on mentors to assist development projects in successfully performing the process and tailoring it to their needs.

Full-time PITs are often referred to in the literature as Process Engineering Groups or PEGs. The term software engineering process group or SEPG is commonly used in CMM-based improvement initiatives (the SEI Software CMM states the SEPG should be devoted full-time to SPI efforts, but in practice, this is frequently not the case, particularly for smaller groups).

Resulting Context The process improvement project need not progress at a snail's pace. PIT members have ample time and resources to devote to the requisite care and feeding of the improvement project. The conceptual integrity and continuity of improvement efforts is less at risk with full-time personnel.

This pattern opts for the opposite extreme from *PIT also Practices*. It takes the risk of having members who are more isolated from the project development teams in order to have greater resources available to effect process improvement. It tries to manage this risk by having PIT members make a deliberate effort to regularly interact with the development project teams.

It may not be feasible to apply this pattern in small groups, or groups that are in the midst of avoiding a crisis: the immediate cost to the business of sparing your best and brightest practitioners may be too great to enable the organization to sustain itself, much less return to a healthy state once the benefits of SPI start to appear. *PIT also Practices* may be more suitable for such groups, even if it means progressing more slowly.

Related Patterns This pattern is similar to *Dedicated Champion [Delano,Rising]*.

PIT also Practices may be combined with this pattern to achieve a mix of full-time and part-time PIT members. In fact, *[Herbsleb,Carleton]* mentions successful CMM-based SPI efforts at Tinker Air Force Base OC-ALC which made extensive use of both full-time *and* part-time SEPG members. The OC-ALC group concluded:

> Having SEPG membership of both full-time personnel and part-time personnel is very important. The full-time members provide continuity for the process improvement efforts, while the part-time members act as advisors, advocates, change agents, and communications liaisons.

Known Uses *[Herbsleb,Carleton]* describes successful CMM-based SPI efforts at Bull HN using a full-time SEPG. *[Fowler,Rifkin]* cite multiple successful SPI initiatives carried out by full-time SEPGs. *[Donaldsen,Siegel]* also cite successes of SPI efforts using full-time PIT members.

Pattern ***Process follows Practice***

Context *Process is Product* has been applied and a PIT has been assembled. Now the PIT needs to commence with the business of changing or adapting the process to meet the improvement goals and document the result to the extent required.

Problem How do you change the process to meet the required improvement goals while at the same time ensuring that the documented process accurately depicts the reality that is being practiced by developers?

Forces The desire to begin making process changes is often very strong. So is the need to demonstrate visible progress as soon as possible in order to give credibility and confidence to SPI efforts. This flies directly in the face of forces already mentioned: resistance to change, speed and size of change (evolution versus revolution), and tolerance for change. You want to change the process documentation to address the requisite assessment criteria, but you also want the documented process to be used and followed by the practitioners (as opposed to shelfware that simply stays on the shelf, never to be used except when a process auditor happens to be visiting the premises).

Solution Start by discovering and understanding current practice throughout the group. Find existing process documentation and talk to practitioners to understand how tasks are performed. Reconcile any differences between actual and espoused processes. Document and review the newly characterized process. Then iteratively and incrementally improve the process and ensure that the documentation is updated appropriately. This defines the following major activities (each of which needs several smaller grained patterns for further elaboration):

Cherchéz les Documentation!
: Excavate the written process: Find and read all existing formal and informal process documentation. This may require assistance from the practitioners to point you in the right direction.

Know Thyself!
: Excavate the current practice: talk to practitioners to understand how they do their work. The purpose of this is *comprehension*, not criticism or appraisal. The developers must trust the person asking the questions and be able to give honest answers without fear of being punished or judged. Have practitioners help you reconcile any differences between actual practice and the espoused process and use appropriate text and diagrams to capture your findings (it is important to track down and understand unspoken behaviors as well as spoken ones). Be sure to make note of the things your group is doing right (best practices) in addition to things that appear to be going wrong (lessons learned). These will be important in determining process areas need of improvement, as well as process areas requiring little or no change.

Documentation Redux!
: Consolidate the results from capturing existing practice. Further "excavation" may be needed to clarify various issues. Be as clear and concise as possible and try to

distill the documentation down to the minimal amount of text and pictures that is necessary and sufficient to describe the existing process. Have all participants review the resulting documentation for consistency and correctness. Then baseline the results.

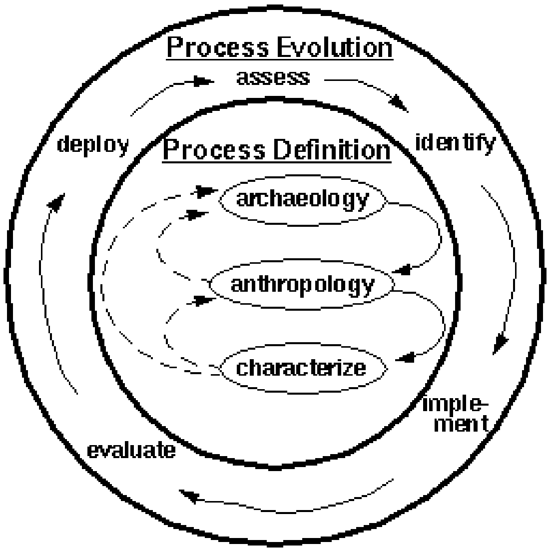
Piecemeal Growth!

Use the baselined process as a benchmark for identifying which areas to concentrate upon for the next iteration of improvement efforts. Improve the process in an incremental and evolutionary fashion. Engage practitioners in proposing, implementing, and evaluating improvements (see *Virtual Forum* and *Improvement Action Teams*). Incrementally deploy and institutionalize each improvement throughout the group. Ensure that formal process documentation is updated to keep "in sync" with process changes being newly practiced.

The above activities will *not* necessarily progress in strict sequential order. In particular, a great deal of iteration will probably be required between the first three activities before proceeding onto the fourth.

Resulting Context

This pattern defines the major phases of an early cycle of the improvement process: First employ some archaeology to uncover existing process artifacts. Second, employ some anthropology to better understand process stakeholders, and discover existing practices. Third, characterize the process, bringing together all the artifacts and practices while reconciling any differences between them (at the end of this step, review and then baseline the results). Finally, improve the system incrementally. The first three activities form a lifecycle model for process definition, while the last activity outlines the basic structure of an iterative lifecycle for process evolution.



Evolutionary and incremental change is used as a means of balancing the forces of resistance to change, speed and size of change (evolution versus revolution), and tolerance for change. Improvement progress is suspended somewhat during the "archaeology" phase, but this is deemed necessary in order to analyze the process stakeholders and understand their existing cultural behavior and values. With this knowledge in mind, improvement plans that carefully take into account the existing process and culture have a higher likelihood of being successful (or as *[Humphrey]* writes: "If you don’t know where you are, a map won’t help!").

It should be noted that the level of detail to which the process should be characterized is not clearly stated by this pattern. Does it make a difference if you are conducting improvement efforts for ISO 9000 versus the SEI CMM or SPICE? What if the existing process is already fairly repeatable and successful? What if the existing process is chaotic, unrepeatable, and unreliable? These are all valid questions in need of other patterns to provide their answers.

Rationale *Process follows Practice* builds on *Process is Product* by pointing out that process improvement is akin to a certain kind of development project: namely a legacy systems reengineering project (involving significant interface changes) where it is deemed infeasible or impractical to simply throw out the existing system and start over from scratch. Rather, the existing system must first be carefully understood and then incrementally modified so as to not adversely inconvenience the existing customer base.

Disregarding existing practice, by suddenly proposing an entirely new and completely redesigned process, sends a negative message to practitioners: everything they have been doing up until now is wrong. This is seldom true. While there may certainly be areas in need of substantial improvement, it is likely that practitioners are doing many things correctly. *Process follows Practice* emphasizes these things and makes it clear that not everything has to change. This increases familiarity and self-esteem while decreasing the size and speed of changes to be made.

Related Patterns *Core Processes* and *Strategic Direction Initiative [Beedle]* have many similarities to *Process follows Practice*.

Virtual Forum may be used throughout the application of this pattern for some of the communication that occurs between the PIT and the practitioners. *Mercenary Analyst*, from *[Cope]*, may be useful for composing and revising process documentation.

Improvement follows Spiral builds upon this pattern by suggesting a framework upon which to impose the proposed lifecycle phases. *Improvement follows Process* suggests the process to use for the various activities within the improvement lifecycle.

Known Uses Examples of first characterizing and assessing the existing process, and then using that as a benchmark for effecting continuous incremental process improvements are either cited or described in *[Krasner]*, *[Austin,Paulish]*, *[Fowler,Rifkin]*, and *[WeinbergV4]*.

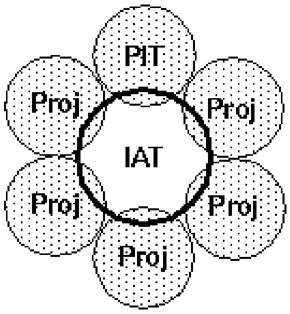
Pattern ***Improvement Action Teams***

Context A specific process area been selected for improvement. Some preliminary planning and discussion has already been conducted to flesh out the essential requirements for the improvement, and to propose some feasible alternatives for implementing it (perhaps *Virtual Forum* was used for much of this).

Problem To facilitate its acceptance while making effective use of time and effort, who should implement and deploy a given improvement idea?

Forces Although the PIT (or PEG) is primarily responsible for leading process improvement efforts, process changes are most likely to be accepted when developed in participation with the individuals who are the targets of the change. The PIT has been granted the time and resources to make such changes happen, but such time might be better spent delegating all or part of the implementation and deployment to some subset of individuals. Unfortunately, the rest of the practitioner community can't necessarily afford the same amount of time and resources as the PIT to participate in SPI efforts.

Solution Form a small Improvement Action Team (IAT) from the pool of PIT members and practitioners who championed or supported the improvement idea. The IAT should be a small team that is tightly focused on the single improvement. Non-PIT members of the IAT should devote 10%-20% of their time to implementing the improvement (which requires management cooperation to adjust their commitments). Have the IAT regularly communicate status to the PIT. After the IAT has fulfilled its charter, disband the team (except perhaps for some periodic maintenance which requires less time and effort).



Resulting Context Temporally recurring process "SWAT teams" enlist practitioners in the implementation and deployment of incremental process improvements. Most of the PIT can spend more of its time dealing with other aspects of the SPI project while still keeping tabs on the IAT and providing assistance when necessary. The IAT can be singularly focused on the specific improvement (perhaps much more so than the PIT, which is probably busy with other SPI issues as well). In addition, the development projects from which IAT members are drawn are desirable candidates for subsequently pilot-testing the improvement (provided the development project isn't too high-risk or high-profile).

Once again, as with *Local Heroes*, patterns for establishing appropriate rewards and reinforcement to encourage participation and cooperation are needed to help this pattern succeed (see *Compensate Success [Cope]*, and *Compensate Results [Beedle]*).

Related Patterns This pattern is reminiscent of *Engage Customers [Cope]*, *Grass Roots [Delano,Rising]*, *Process Owners* and *Business Architecture Team [Beedle].*

The *Virtual Forum* may be used by the IAT to communicate some of their efforts to the PIT and the practitioners. *Improvement follows Process* and *Improvement follows Spiral* should be used to conduct the efforts of the IAT.

Known *[Haley]* describes the use of ad hoc task teams (which included line-engineers) to

Uses implement individual process improvements at Raytheon. *[Fowler,Rifkin]* cite multiple SPI case studies where using such "working groups, task forces, networks, or tiger teams" met with success. *[Herbsleb,Carleton]* writes of SPI efforts at Tinker Air Force Base OC-ALC using technical working groups (TWGs) composed primarily of practitioners and coordinated by one or more SEPG members to implement specific improvements (or a set of improvements) in a given process area. The TWGs were disbanded once their charter was fulfilled.

Pattern ***Improvement follows Process***

Context *Process follows Practice* has been applied and the PIT or an IAT is ready to commence planning, designing, implementing, and deploying process changes.

Problem What process should be used for improving the process itself?

Forces Ideally, the process should be capable of encompassing self-improvement. However, if this were the case, the process wouldn't need certain improvements in the first place. Using process methods and techniques that differ from those you have recommended to practitioners damages your credibility within the development community. It also demonstrates the inability of your current process to accommodate the existing range of project domains. However, numerous activities and concerns to be considered for SPI efforts are markedly different from those of product development.

Solution Whenever plausible, use the same process that you are trying to impose upon the other practitioners, or that you have already imposed upon them. Newly proposed improvements should try to accommodate how they might be practiced for process development as well as product development.

Sometimes there will specific things you do for product development that simply don't make sense for process development (perhaps certain elements of release engineering). In this situation, the process improvement project may end up doing things differently from the product development projects (or may omit something altogether).

In either case, try to find any common elements or principles between the two approaches (or reasonable justification for why they do things differently) that can be abstracted out. The common parts should be documented, and then policies or guidelines and rationales for the alternatives can either be recorded in the process documentation, or in project documentation which tailors how the process will be applied to the specific project. This makes the overall process more flexible for future projects and is still a reasonably honest attempt by PIT members to follow their own rules.

Resulting Context *Improvement follows Process* results in congruence between the words of the PITs and IATs with their own actions, and with the desired actions of the rest of the development community. Practitioners see the PIT "practicing what it preaches", which lends credibility to their efforts. In addition the process becomes adaptable enough to accommodate many of the varying needs of both product development, and process development.

Related Patterns The recommendation of a process that is "self-aware" for the sake of its own improvement suggests an interesting similarity to *Reflection* from *[POSA]*.

Improvement follows Spiral suggests a framework for planning and organizing improvement activities within the confines of the improvement process.

Known Uses *[Curtis]*, *[Fowler,Rifkin]*, *[Donaldsen,Siegel]* all observe the importance of PIT members leading by example and bolstering credibility of their efforts by following the same orderly processes which they expect others to adopt. *[McCarthy]* and *[Cusumano,Selby]* note that Microsoft refers to this as "dogfooding", because it brings to mind the sentiment that pet product vendors should be required to "eat their own dogfood" before asking consumers to feed it to their own beloved pets.

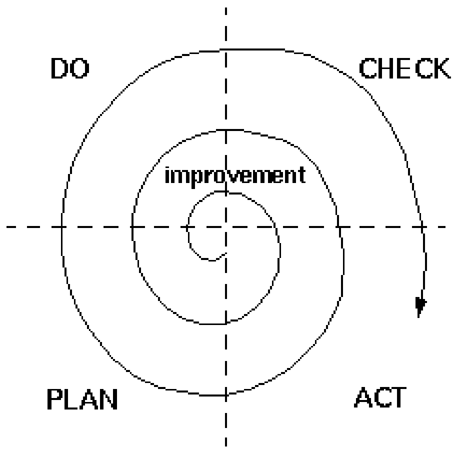
Pattern ***Improvement follows Spiral***

Context You need to come up with an overall battle-plan to structure the set of activities for carrying out process improvement. This may apply to improvement efforts in general (which are planned by the PIT or by the PIT leader/manager), or for a specific improvement has been proposed and is now ready to be implemented by an IAT.

Problem What framework should be used to structure the various activities of planning, implementation, assessment, and deployment for process improvement initiatives?

Forces Group-wide process improvement efforts need to be carefully planned if they are to succeed. Numerous risks must be identified, evaluated, and approriately resolved or addressed. Omitting an important step or overlooking a key risk can result in total project failure. Too much emphasis on planning and analysis may slow progress to a virtual standstill, or prevent progress from ever taking place. Too much emphasis on action and not enough on assessment may result in sloppy and ineffective efforts that eventually fail. Even if a suitable balance of action and reflection is found, the order and frequency in which these activities are are performed can make or break a process improvement initiative.

Solution Impose a spiral model upon the process improvement lifecycle and base it off some variant of the Shewhart cycle of *Plan-Do-Check-Act* (espoused by Deming and in TQM circles). Current incarnations of the spiral model include the original model proposed in *[Boehm88]*, the "Theory W" spiral model of *[Pressman]* and an updated model in *[Boehm96]* called the Win-Win spiral model.



Resulting Context A spiral framework for iteratively incorporating planning, assessment, and risk management activities throughout the process of implementing and deploying improvements. The spiral model is used in a manner similar to that which its proponents recommend for software development. The Shewhart cycle tailors the spiral model for use with process improvement efforts (*[Grady]* reaches this conclusion as well).

Related Patterns At first glance, *Improvement follows Spiral* might seem to contradict *Improvement follows Process* and *Process follows Practice*. However, it is in fact complementary to both of them. When combined with *Improvement follows Process*, it suggests that both the improvement process and the development process should incorporate the spiral model in an appropriate manner. The process definition lifecycle suggested by *Process follows Practice* fits into one of the earliest iteration cycles to perform, while the suggested process evolution outline maps very closely to most variants of the Shewhart cycle commonly employed in SPI initiatives.

Known Uses *[Grady]* writes at length about Hewlett-Packard's success in using the classic Shewhart cycle together with the spiral model for planning SPI efforts. *[Wiegers]* describes SPI efforts at Kodak which applied a spiral cycle of *Plan-Do-Assess-Verify* for evolutionary improvement. *[Wakulczyk]* remarks upon NORAD's success applying a spiral cycle of *Analyze-Plan-Do-Check-Act* for SPI. *[Jones,Kasunic]* and *[Radice]* both describe success with extensions to the SEI IDEAL model, iterating the cycle of *Initiate-Diagnose-Enact-Assess-Leverage* in a spiral fashion for SPI planning and implementation. *[SPC]* and *[Kellner]* also recommend the use of an evolutionary spiral process model for incremental process improvement.

Conclusions

This paper began by claiming that important social and cultural issues make process improvement projects markedly different from most product development projects. Then numerous patterns were presented, many of which seemed to contradict this claim by emphasizing how process improvement projects should be treated much the same as product development projects. Both the similarities as well as the differences between these two kinds of projects are vitally important. The process patterns seem to extoll the similarities while the organization and communication patterns focus on the important social and cultural differences. The procedures used to conduct improvement efforts need to closely conform to those used for product development, but the social organization and communication strategies employed need to accommodate the fact that the customer actually lives under the same roof as the organization itself, co-habiting with all of its members. As with any kind of software project, engaging the customer early and often in dialogues which regularly communicate status and feedback is a crucial element of success (and its absence is often a leading cause of project failures). These important issues are profoundly amplified in process improvement projects because the customers are members of the same cultural community as the improvement project managers and architects. As a result, the organization's internal ecosystem is more sensitive to the impact of improvement efforts since they effect changes in that very same culture.

The patterns presented form a mildly cohesive system for conducting improvement efforts. Many more are needed to form a comprehensive pattern language for initiating and sustaining process improvement. The important problems of how to successfully obtain senior management "buy in", setup rewards and incentives, solicit practitioner enrollment, create a shared mental model of the desired state, establish process ownership, and conduct training and education, are not addressed by most of these patterns. Precious little is mentioned regarding many of the varying needs of different organizational cultures. What needs to be done differently for calendar-driven or documentation-driven organizations? What about groups in constant crisis or crisis-aversion mode who appear unwilling or unable to spare even a modicum of resources on any but the most immediately pressing of matters? What about groups who are repeatably delivering products of satisfactory quality to their customers within acceptable schedule constraints? Many such groups might not pass a formal ISO 9000 assessment or SEI CMM level 2 or 3 audit. How (and perhaps more importantly, *why*) should they go about changing their seemingly successful processes? Patterns which provide solutions to problems such as these are still needed before the patterns in this paper can be confidently applied in all of these contexts.

Acknowledgements

The author would like to thank the following individuals for their significant contributions to this work:

- Robert Martin for his patience and guidance as my shepherd for PLoP'97
- The participants of the PLoP'97 writers workshop track on People and Process: Michael Beedle, Steve Berczuk, Charles Crowley, Joe Davison, David Delano, David Dikel, David Kane, Don Olson, William Opdyke, and Linda Rising.
- Ralph Johnson, Tim Ottinger, and Don Dwiggins for their words of encouragement
- Professor William Junk from the University of Idaho
- Karl Wiegers, for providing comments as well as sharing his insights on SPI
- Contributors to the organization patterns mailing list, including: Michael Beedle, Alistair Cockburn, Martine Devos, Greg Gibson, Beki Grinter, Neil Harrison, Bob Haugen, Luke Hohmann, David Kane, Jens Kruger, Tim Ottinger, Scott Preece, Shalom Reich, Linda Rising, Clenio Salviano, and many others

References

[Conner] Daryl Conner, **Managing at the Speed of Change**, Villard Books, 1993

[PLoPD1] James O. Coplien, Douglas C. Schmidt (Ed.), **Pattern Languages of Program Design**, Addison-Wesley, 1995, pp. 178-237

[PLoPD2] John M. Vlissides, James O. Coplien, Norman L. Kerth (Ed.), **Pattern Languages of Program Design 2**, Addison-Wesley, 1996, pp. 345-352

[Osterweil1] Leon J. Osterweil, "*Software Processes are Software Too*", **Proceedings of the 9th International Conference on Software Engineering**, ACM Press, March 1987, pp. 2-13

[Osterweil2] Leon J. Osterweil, "*Software Processes are Software Too, Revisited*", **Proceedings of the 19th International Conference on Software Engineering**, ACM Press, May 1997, pp. 540-548

[Wiegers] Karl E. Wiegers, **Creating a Software Engineering Culture**, Dorset House, 1996

[Grady] Robert B. Grady, **Successful Software Process Improvement**, Prentice-Hall, 1997

[Austin, Paulish] Robert D. Austin, Daniel J. Paulish, *A Survey of Commonly Applied Methods for Software Process Improvement*, Carnegie Mellon University SEI Technical Report CMU/SEI-93-TR-27, February 1993

[Curtis] Bill Curtis, "*Software Process Improvement: Methods and Lessons Learned*", **Proceedings of the 19th International Conference on Software Engineering**, ACM Press, May 1997, pp. 624-625

[Wakulczyk] Marek Wakulczyk, "*The Successful SEPG: S-CMM Level 2 in 2 $^{1}/_{2}$ Years*", 9th Annual Software Technology Conference, Salt Lake City, Utah, April 1997

[Larner] Chris Larner, "*More Practical Experiences and Lessons Gained by the SEPG of a Major European Bank*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ

[Radice] Ronald A. Radice, "*Extensions to IDEAL Approach to Software Process Improvement*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ

[Delano, Rising] David Delano, Linda Rising, "*Introducing Technology into the Workplace*", **PLoP/Allerton Park 1997 Proceedings**, Washington University Technical Report #wucs-97-34

[Dikel, Kane] David Kane, Christy Hermansen, David Dikel, and Ralph Malveaux, "*Organizational Patterns for Software Architecture*", **PLoP/Allerton Park 1997 Proceedings**, Washington University Technical Report #wucs-97-34

[Cope] James O. Coplien, "*A Generative Development Process Pattern Language*", **Pattern Languages of Program Design**, James O. Coplien, Douglas C. Schmidt (Ed.), Addison-Wesley, 1995, pp. 178-237

[Baumert] John H. Baumert, "*Experiences Developing and Deploying a Corporate-Wide Process Asset Library*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ

[McLane] Marie McLane, "*The First Step is the Hardest: A Systemic Approach to Continuous Process Improvement*", **Proceedings of the 1994 6th SEPG Conference**, April 1996, Dallas, TX

[Harrison] Neil Harrison, "*Organizational Patterns for Teams*", **Pattern Languages of Program Design 2**, John M. Vlissides, James O. Coplien, Norman L. Kerth (Ed.), Addison-Wesley, 1996, pp. 345-352

[WeinbergV3] Gerald Weinberg, **Quality Software Management Volume 3: Congruent Action**, Dorset House, 1994

[Beedle] Michael A. Beedle, "*cOOherentBPR - A pattern language to build agile organizations*", **PLoP/Allerton Park 1997 Proceedings**, Washington University Technical Report #wucs-97-34

[Donaldsen, Siegel] Scott E. Donaldsen, Stanley G. Siegel, **Cultivating Successful Software Development: A Practitioner's View**, Prentice-Hall PTR, 1997

[Fowler, Rifkin] Priscilla Fowler, Stan Rifkin, *Software Engineering Process Group Guide*, Carnegie Mellon University SEI Technical Report CMU/SEI-90-TR-024, September 1990

[Kotter] John Kotter, **Leading Change**, Harvard Business School Press, 1996

[GoF] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, **Design Patterns: Elements of Reusable Object-Oriented Software**, Addison-Wesley, 1995

[Haley] Thomas J. Haley, "*Software Process Improvement at Raytheon*", IEEE Software, November 1996, Vol. 13 No. 6

[Dorsey, McDonald] Terri Dorsey, Diane McDonald, "*Structured for Success: a Software Engineering Process Group (SEPG) Model*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ

[Herbsleb, Carleton]	James Herbsleb, Anita Carleton, James Rozum, Jane Siegel, David Zubrow, *Benefits of CMM-Based Software Process Improvement: Initial Results*, Carnegie Mellon University SEI Technical Report CMU/SEI-94-TR-013, August 1994
[Humphrey]	Watts Humphrey, **Managing the Software Process**, Addison-Wesley, 1989
[Krasner]	Herb Krasner, "*Software Cycle-Time Reduction (SCTR): Boldly Going Beyond Level 3*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ
[WeinbergV4]	Gerald Weinberg, **Quality Software Management Volume 4: Anticipating Change**, Dorset House, 1997
[POSA]	Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerald, Michael Stal, **Pattern-Oriented Software Architecture: A System Of Patterns**, John Wiley & Sons, 1996
[McCarthy]	Jim McCarthy, **Dynamics of Software Development**, Microsoft Press, 1995
[Cusumano, Selby]	Michael A. Cusumano, Richard W. Selby **Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People**, The Free Press, October 1995
[Boehm88]	Barry Boehm, "*A Spiral Model of Software Development and Enhancement*", IEEE Computer, May 1988
[Pressman]	Roger S. Pressman, **Software Engineering: A Practitioner's Approach**, 4th ed., McGraw-Hill, 1996
[Boehm96]	Barry Boehm, "*Anchoring the Software Process*", IEEE Software, July 1996, Volume 13 Number 4
[Jones, Kasunic]	Lawrence G. Jones, Mark Kasunic, Mike Ginn, "*Implementing SEI's IDEAL Model in a Less Than Ideal World*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ
[SPC]	Software Productivity Consortium, **Improving the Software Process Through Process Definition and Modeling**, International Thomson Computer Press, 1996
[Kellner]	Marc I. Kellner, "*A Method for Designing, Defining, and Evolving Software Processes*", **Proceedings of the 1996 8th SEPG Conference**, May 1996, Atlantic City, NJ