SOFTWARE PROCESS—Improvement and Practice
Softw. Process Improve. Pract. **4**, 187–198 (1998)

Addressing the People Issues of Process Improvement Activities at Oerlikon Aerospace

Research Section

Claude Y. Laporte[1]* and Sylvie Trudel[2]
[1]*Yortar Technologies, 481 Bissett, Saint-Jean-sur-Richelieu, Québec, J3A 1W6, Canada*
[2]*Oerlikon Aerospace, 225 Seminaire, Saint-Jean-sur-Richelieu, Québec, J3B 8E9, Canada*

This paper describes the approach used by a defense contractor, since 1992, to address the issues raised when defining and implementing engineering and management processes. First are described the steps taken to define the software engineering process, the systems engineering process and the project management process. Then issues raised during the integration of the processes are described. Finally, the steps to address the management of change are discussed and lessons learned are presented. Copyright © 1998 John Wiley & Sons Ltd

KEY WORDS: software process improvement; lessons learned; CMM; process integration; systems engineering; project management

1. BACKGROUND

The defense contractor is the integrator of a complex laser-guided missile air defense system. The system consists of five technology/product families: processing and display; platform system; sensors and effectors; command, control, communication and intelligence; and readiness system (e.g. training, simulators and test). Over 120 software and systems engineers are involved in the development and maintenance of the system.

In the organization, the approach to the conduct of process improvement activities was fourfold: first define a process and bring it under management control; secondly, support the process with methods; thirdly, support the process and methods with appropriate tools; and fourth, train all personnel in the utilization of processes, methods and tools.

2. DEVELOPMENT OF SOFTWARE ENGINEERING PROCESS

Essentially, the software process improvement initiative followed the five phases of the IDEAL[SM†] model (McFeeley 1996). The five phases of the model are (Figure 1): Initiating the improvement program; Diagnosing the current state of practice; Establishing the plans for the improvement; Acting on the plans and recommended improvements, and Leveraging the lessons learned and the business results of the improvement effort.

* Correspondence to: Claude Y. Laporte, Yortar Technologies, 481 Bissett; Saint-Jean-sur-Richelieu, Québec, J3A 1W6, Canada. E-mail: claporte@yortar.com

CCC 1077-4866/98/040187-12$17.50

†IDEAL is a service mark of Carnegie Mellon University

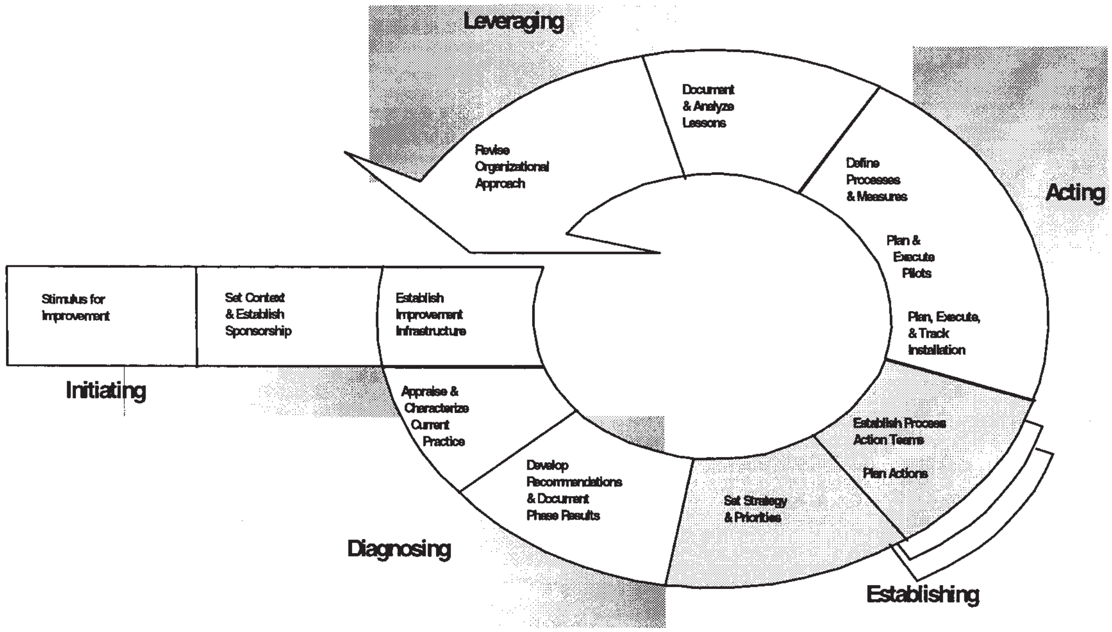

Figure 1. The IDEAL improvement cycle

During the initiating phase (fall of 1992), a business case was prepared and presented to the president. Recognizing that software engineering was a core competence of the organization, the president approved the establishment of a Software Engineering Process Group (SEPG) (Fowler and Rifkin 1990). A budget was also approved for the conduct of a software process assessment (SPA) and the development of an action plan. An SPA is an assessment method developed by the Software Engineering Institute of Carnegie Mellon University. Briefing sessions were held to inform the organization about the software process improvement effort.

During the diagnosing phase (spring 1993), an SPA was performed jointly by the SEPG and by independent assessors certified by the Software Engineering Institute (SEI). Strengths and weaknesses were identified and priorities for improvements were recommended. An action plan skeleton was presented to the president, identifying the resources required for its implementation.

During the establishment phase (summer–fall 1993), a detailed action plan was prepared by the SEPG. During a three-day workshop, assessment findings and recommendations were reviewed and a strategy was developed. It was decided that working groups would be established to define individual processes under the close coordination of the SEPG. For each process, a process owner, i.e. a person responsible for the implementation and improvement of a process, was identified. Working groups of four to six members would be staffed with representatives of software engineering, systems and sub-systems engineering, quality assurance and configuration management. Each member of the working groups would spend up to 8 hours per week on process related activities. In each working group, a member of the SEPG would act as a facilitator. At regular intervals, SEPG members would meet to resolve issues raised within their groups and pass along lessons learned within their own working groups. For each working group, a mini action plan was prepared by the SEPG. The action plan listed the following elements: goals of the working group; identification of the owner of the process; identification of the part-time participants; implementation steps; risk issues; timetable; level of effort planned, and reference documents. We have decided to use a modified version of the ETVX (Radice 1985) notation for the description of the processes. Essentially, the notation describes, for each step of the process, the inputs required to perform the activities, the outputs produced, entry criteria that will allow activities to be performed, exit criteria that will allow artifacts to exit the current step, and measures that will be captured when executing the activities.

To help define the processes, the working groups also used extensively a document produced by the SEI (Olson *et al* 1993), that describes each key process area (KPA) of the Capability Maturity Model^{SM} for software (CMM®) (Paulk *et al.* 1993) using the modified ETVX notation.

During the acting phase, initiated in winter 1994, working groups started their activities. Working groups were kicked off in one to two month intervals. This way, problems inherent to the dynamics of teams were solved, and lessons learned were captured before starting another group. Once the processes were defined, pilot projects were identified for a trial period. Each process is described at three levels of details. The top-level view is a black box approach describing the major steps required satisfying the goals of the KPAs. A second level of details describes each black-box with the following information: the objective of the activities to be performed; inputs required to perform the activities; a list of activities; outputs produced; entry and exit criteria controlling the initiation and completion of each process step, measurements (e.g. size, effort, quality), and persons responsible for performing and supporting each process step. At the third level of details, methods are described in process guides (e.g. size estimation, risk assessment). Each person who has to use the processes receives his own copy of the software engineering guidebook, which contains processes, methods and guides. Each person is also trained on the utilization of the processes, methods and guides.

The following software processes were developed, tested in pilot projects and implemented: software development; software maintenance; reverse engineering; project planning and tracking; software quality assurance; software configuration management; software subcontractor management; documentation management, and document inspection (Gilb and Graham 1993).

Self-assessments and process audits were also performed on all projects. The objective of the self-assessments was not to 'fix the people' but to bring to the surface barriers to the institutionalization of the processes. The focus on the process rather than on the people is critical for company-wide acceptance of the new process. Each project team was interviewed separately and composite results of the self-assessment were presented to management and project teams. A questionnaire was used to probe projects. The questionnaire used scoring guidelines developed by Motorola (Daskalantonakis 1994). Motorola developed a ten-level scoring scale, which allows a finer evaluation of the institutionalization of each key process area. The scoring guidelines measure the attainment of the following three elements: first the approach, i.e. criteria that show the organization's commitment to and management's support for a practice; second the deployment, i.e. the breadth and consistency of practice implementation; and third the results, i.e. the breadth and consistency of positive results over time. With such a scale it is easier to measure the progress made by each team from one audit to another. After a self-assessment, an action plan was developed to address the findings and implement corrective actions.

Finally, during the leveraging phase, lessons learned from projects and processes were collected, analyzed and implemented. The software planning and tracking process was defined such that it is the first process to be initiated in any project and also the last process to be called at the completion of a project. In the planning phase, the project leader has to plan and estimate the effort required to conduct lessons learned sessions. During the tracking phase, lessons learned reviews are performed. Usually, in a lessons learned meeting, the members of the project are present, and someone from the SEPG facilitates the meeting. In order to make sure that the lessons are not lost the SEPG analyzes each lesson in order to identify if a process step could be improved (Basili and Green 1994). If this is the case, modifications to the process, methods or guides are made.

As the processes are being used in projects, artifacts are collected and stored in a process asset library (PAL). Presently, the PAL contains mostly paper documents. As the organization is moving toward an environment where each practitioner will have access electronically to documents, the PAL will contain electronic copies of documents produced. The PAL librarian has read and writes privileges while practitioners have only read privileges. The librarian will also perform configuration management functions on the artifacts of the PAL. Table 1 lists the artifacts that will be stored in the PAL as projects are producing documents.

The second formal assessment (i.e. a CMM-based appraisal for internal process improvement, CBA IPI) (Dunaway and Master 1996) by certified SEI assessor from the Applied Software Engineering

Table 1. Content of the software process asset library

Software engineering policy	Lessons learned
Process descriptions	List of process owners
Forms and templates	Process improvement suggestions
Examples of documents produced	Training material
Business case examples	Quality assurance reports (e.g. reports from audits)
Proposal examples	Quality data (e.g. results from inspections)
Software development plans	List of software tools under configuration
Tailored processes	Historical data (e.g. project estimates)
Tailoring guidelines	Software methods documentation
Process definition process	Charter of software engineering process group

Centre (ASEC) was conducted in February 1997. The organization achieved SEI level 2 certification. In addition, it had also satisfied eight of the 17 level 3 goals. Two level 3 key process areas were fully satisfied: software product engineering and peer review (i.e. document inspection process).

3. DEVELOPMENT OF A SYSTEMS ENGINEERING PROCESS

In 1995, a mini assessment of the systems engineering practices was conducted using the Systems Engineering Capability Maturity Model[SM‡] (SE-CMM) (Bate 1995). The objective of the mini assessment was to help identify priorities for improvement within the 18 process areas of the SE-CMM. After analysis of the results of the mini assessment, management decided to put a higher priority on the engineering process areas as defined in the SE-CMM. Literature was reviewed and a decision was made to use, as frameworks, the SE-CMM and the Generic Systems Engineering Process ©[§] (GSEP) developed by the Software Productivity Consortium (SPC 1995). The GSEP has been developed to incorporate most of the practices of the SE-CMM. A working group, composed of systems engineers, software engineers and a representative from quality assurance, was established to define and facilitate the implementation of a systems engineering process. Another objective of the working group was to integrate the software engineering process to the systems engineering process.

In addition to defining the process, each member of the working group had a secondary duty: as each step of the beta version of the process was defined, members of the working group were tasked to collect information as listed in Table 2. Also, since the organization had been certified as an ISO 9001 organization in 1993, one representative from the quality assurance department monitored progress in order to make sure that the process being defined was compliant to ISO requirements.

4. DEVELOPMENT OF A PROJECT MANAGEMENT PROCESS

After a few years of software process development, the organization felt that it would benefit from a standardized project management process. It was also felt that an organizational project management process would help the organization achieving a level 3 of the software CMM, e.g. Intergroup Coordination and Integrated Software Management key process areas. In fall 1996 a mandate was given to a working group to develop and implement a Project Management Process (PMP). The working group selected the Guide to the Project Management Body of Knowledge ©[¶], developed by the Project Management Institute (PMI 1996), as the framework for the organizational process. The working group was composed of project managers

Table 2. Secondary tasks of members of working group

Update process descriptions
Monitor compliance with the SE-CMM
Monitor the interfaces with the software engineering process
Identify process and product measurements
Identify roles and responsibilities
Define glossary
Identify methods, best practices, artifacts, CASE tools, lifecycle representations, project templates, estimation guidelines, course material, training resources and lessons learned
Establish the systems engineering process asset library

‡ SE-CMM is a service mark of Carnegie Mellon University.
§ Copyright by the Software Productivity Consortium.
¶ Copyright by the Project Management Institute.

and representatives from engineering disciplines, and practioners from quality assurance, production, configuration management and logistic support. The software engineering assessment that had been performed in February 1997 also identified weaknesses in project management practices. The assessment team made a few recommendations to address these weaknesses.

5. INTEGRATION OF ENGINEERING PROCESSES

As mentioned earlier, throughout the development of the systems engineering process, the working group kept on the agenda the integration between this process and the software process. It was decided to adopt, as a framework for the integration, the Integrated Systems and Software Engineering Process ©† (ISSEP) from the SPC (SPC 1996). Since many problems, when developing complex computer-based systems, are discovered at integration time, the solution is to use a process that will decompose the systems in parts that can be developed independently and easily integrated together at the system level. It was also noted that because of digitization of electromechanical systems, the apparent space of software was increasing on projects; from nominally 30% in the mid-70s, software has now reached 60% to 70% of the non-recurring activities in system development. Furthermore, the expansion of integrated case technologies, which crossed departmental barriers, through common process framework, reinforced the desire of the organization to integrate both software and systems engineering process and to focus the organization into an integrated project team approach. In other words, software and systems engineering are beginning together at the inception of a project.

The ISSEP model defines a decomposition strategy for system development as well as a set of management and technical activities and interfaces between processes. ISSEP describes activities at three levels: the system level; the configuration item (CI) level, and the component level. It is at the component level that software and hardware are developed. Figure 2 illustrates the integration between processes. The 'manage development effort' and the 'define system increment' boxes are described in detail in the systems engineering process (Laporte and Papiccio 1997, Laporte *et al.* 1998). The 'develop software configuration item' box is essentially the actual organizational software engineering process, while the 'develop hardware configuration item' box, i.e. a design engineering process, is a process that should be documented in the future.

6. ORGANIZATIONAL PROCESS COORDINATION

In early 1997, it was felt that the implementation of these processes would need organizational coordination and direction. It was decided to establish a steering committee called the Process Action and Coordination Team (PACT). The PACT is composed of three vice-presidents, the manager responsible for quality assurance activities and the coordinator for process performance improvement. The functions of the PACT are:

- Establish time-to-market, quality, costs and product performance objectives to be supported by organizational processes.
- Set priority in accordance with company vision and yearly objectives.
- Liaise with executive committee.
- Establish consensus among different groups.
- Provide support for process performance improvement:
  - (i) review results of assessments and audits;
  - (ii) charter technical area working groups;
  - (iii) budget for resources for process groups;
  - (iv) monitor process performance.

Process owners, i.e. managers responsible for the effectiveness and the efficiency of process, methods and tools, develop process improvement plans (PIP). For the software process, the person responsible for the SEPG reports to the software process owner. Process owners have also been delegated the responsibility to review the tailoring of the process before a development or maintenance project is approved. Knowing that project managers and the process owner may, occasionally, have conflicting views about the tailoring of the process, the policy was written to handle such conflicts. In

† Copyright by the Software Productivity Consortium.

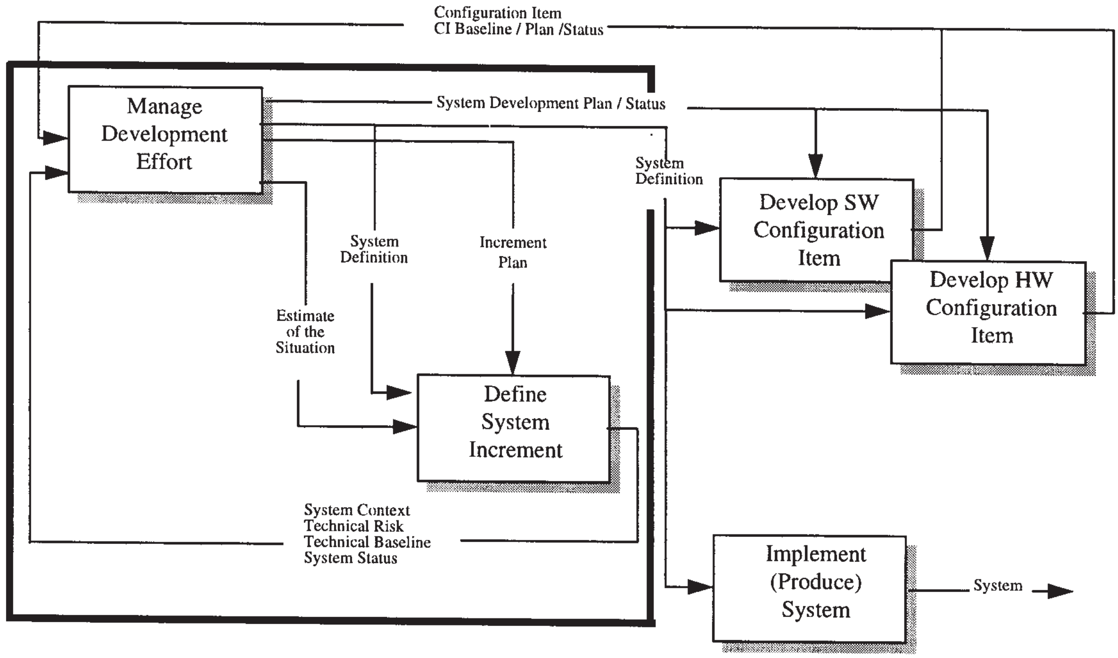

Figure 2. Integration of engineering processes

the event of a deadlock between a project manager and the process owner, both would present a risk analysis to a vice-president for the final approval of the tailored process and consequences. Since 1997, process owners report the progress toward their respective process improvement plan to the PACT.

7. THE MANAGEMENT OF CHANGE

Since the management of change is a key element of a successful process improvement program, a series of actions were planned, using the IDEAL model as a guide, in order to facilitate the development, the implementation and the adoption of the processes, methods and tools (Laporte 1994). As an example, to build the sponsorship level, the President of the organization attended an executive seminar on process improvement at the SEI; two directors attended a three-day seminar discussing the CMM, process, process assessment and improvement. Also, one member of the SEPG attended two courses at the SEI: managing technological change and consulting skills. Briefing sessions were held and articles were written in each company's newsletter to explain the why, what and how of process assessment and improvement and describing the progress made. Finally, surveys were conducted in order to assess the organization's readiness for such a change in practices. The surveys identified strengths of the organization and potential barriers to the planned improvement program. Also, in order to get support and commitment for the future implementation of processes, working groups were staffed with representatives from many departments.

In order to facilitate the conduct of working group activities, a certain number of meeting guidelines (Siddall 1996) were proposed, by the facilitators, to the members of working groups during the kick-off meeting of their group. The proposed guidelines are listed in Table 3. The facilitator read each proposed rule and asked participants if they agreed with the rules. Once the discussion was over, the facilitator reminded the participants that he would be facilitating each meeting using the set of rules selected. Also, after a few meetings, the facilitator invited participants to become secondary facilitators, i.e. when a participant observed a behavior which violated one of the meeting guidelines, he was authorized to raise the issue with the offender. Eventually, the group can manage the 'soft issues' by itself without the help of an outside facilitator.

It was also decided that consensus decision

Table 3. Proposed meeting guidelines

- One conversation at a time.
- In the meeting or out, but not both.
- Be as open as possible.
- Each of us is responsible for the effectiveness of the team.
- Silence is consent.
- We will know how a decision will be made before it is made (e.g. by consensus, majority or minority rule, autocracy, unanimity).
- We expect the discussion leader to control.
- Unacceptable for members to be negative.
- Few recreational stories.
- Differences will be respected.
- Everyone has the right to express their feelings and opinions.
- Avoid blaming individuals.
- We will not dump on each other.
- We will give supportive feedback openly and directly.
- We will listen intently, and with respect, to others and will not interrupt them.
- Everyone is expected to contribute.
- We will focus on our goals.
- We will adhere strictly to all time frames.
- We will come to the meeting prepared for the agenda.
- We will meet our commitments to the team.

making was the preferred decision-making option. We defined consensus according with the definition found in the Team Handbook (Scholtes 1996): consensus is not unanimity, consensus is based on the assumption that solutions are more likely to succeed if all of the key participants are 'comfortable enough' with the outcome to move forward. Occasionally, during meetings we use 'thumb voting' procedure (Popick and Shear 1996) to make decision by consensus. Thumb voting allows the following three alternatives: first, if the proposition is favored, the thumb is up; second, if someone can live with the decision, the thumb is to the side; third, if someone cannot live with the decision, the thumb is down. In the latter case, the members of the working group have to take time to understand the issues at stake and propose an alternative that everyone can live with.

Each working group was managed like a project; it had a charter, a budget and a schedule. The process owner would focus on the content of a specific software process while the facilitator would focus on the process of developing a specific software engineering process.

Occasionally, members of the working groups had to evaluate the effectiveness of their group. A survey (Alexander 1991) was distributed at the end of a meeting. Individually, members complete the survey and send it to the facilitator of their group. The survey addresses the following issues: goals and objectives; utilization of resources; trust and conflict resolution; leadership; control and procedures; interpersonal communications; problem solving; experimentation, and creativity. Issues that were raised by members were discussed in order to generate suggestions for improvement.

8. LESSONS LEARNED

These years of process improvement have enabled us to learn certain lessons likely to be used by other organizations in the future.

8.1. Lesson 1: Set Realistic Expectations for Senior Management

Appropriate expectations must be set prior to embarking on a process improvement journey. The trap, especially for CMM level 1 organizations, consists in communicating to management the idea that the initiative will be easy, fast and inexpensive, has to be avoided at all costs.

A typical scenario looks like this: senior management realizes the benefit that attaining a maturity level can represent for his organization's competitiveness. As second step, a project manager or an external consultant states, in order not to upset the top management, that this objective is easily attainable. As a third step, top management gives managers the mandate to attain this objective in a very short time. During the assessment, the managers face a string of countless findings, findings that had been known by developers for a long time, but remained ignored due to the mode of management that consists in dealing continuously with the problems created (i.e. fighting fires), in a clumsy way at times, by managers. Top management, that had already announced its objective to its peers from other organizations, realizes suddenly that this objective will take a lot more time and resources than had been estimated. At that time, three reactions are possible. Top management may accept the findings and confirm that it will continue to support the objectives announced. It may announce discreetly that it will be lowering its objectives. Finally, it can deny everything and renounce to implement an action plan to correct the deficiencies highlighted by the assessment. This decision could have a destructive effect on developers, since they know for a fact that the deficiencies they had been deploring for a long time are now known by everybody and will remain ignored for a long time.

The lesson to be remembered is to prepare a first action plan – some sort of a brief appraisal of the situation status – preferably by someone who is not involved in the sector targeted and to assess the time and resources necessary to assessing, writing and implementing the action plan. One has to remember top management does not like bad surprises. Moreover, it is better not to proceed to an assessment if it is not intended to deal with the findings. As a matter of fact, once the problems are identified and publicized within the organization, if the management decides not to act, it then sends a very bad message to practitioners.

8.2. Lesson 2: Secure Management Support

A second lesson for CMM level 1 organizations consists in realizing that most of the assessment findings target the deficiencies of project management processes. It is necessary to create an environment where the management is ready to invest in the implementation of processes rather than blame its managers; in other words 'where the management is ready to fix the process, not the people'. This is one of the reasons why it is necessary to also keep informed senior management representatives so that they can show understanding and full commitment when these findings are publicized within the organization.

Beside senior management buy-in, it is essential that middle management and first line managers become strong supporters of the process improvement program. The strongest signal sent by managers is their day-to-day activities, because 'what a manager does talks louder than what a manager says'. The developers must receive very clear signals announcing that the changes advertised will be implemented and that they themselves will have to adopt new practices.

8.3. Lesson 3: Identify Management Needs, Expectations and Understanding of the Problem

The involvement of process owners or managers is largely related to their understanding of the current situation (i.e. strengths and weaknesses). Once convinced that the current situation is undesirable, they will provide the leadership (e.g. direction and momentum) to implement solutions. They can also keep a working group focused on solving the right problems, since it is very easy, after a few meetings, for members of a working group to start solving what they perceive to be the problems.

8.4. Lesson 4: Establish a Software Process Engineering Group

The software Capability Maturity Model suggests the formation of a formal Software Engineering Process Group (SEPG) for any organization heading toward level 3. Even for a level 1 organization, it would be better that a small number of persons became active in process activities a few months before the on-site assessment. The SEPG should take this time to familiarize itself with the Capability Maturity Model and associated process improvement methods and tools. Ideally, in a large organization, there should be one full-time person on the SEPG while the other members could be assigned on a part-time basis. Beside their technical competencies, the members of the SEPG should be selected based on their enthusiasm for improvement and the respect they have within the organization.

8.5. Lesson 5: Start Improvement Activities soon after an Assessment

With regards to the development of the action plan, the organization should capitalize on the momentum gained during the assessment period. The organization does not have to wait for a completed action plan to start process improvement activities. Some improvement activities can begin soon after the completion of the on-site assessment. The implementation of certain improvements is an important motivation factor for all members of the organization.

During the assessment, it is recommended to collect both quantitative and qualitative data (i.e. indicators) which will be used later to measure the progression realized. One could obtain data on slipped budgets and schedules, or measure the degree of satisfaction of the customers regarding product quality level. Since senior management will have made investments, it is very appropriate to be able to demonstrate that these investments have been profitable.

8.6. Lesson 6: Train all Users of the Processes Methods and Tools

Once the processes are defined, it is essential to train all users. Otherwise, all related documents will end up getting dusty on shelves. It is illusory to think that developers will study, by themselves, new processes in addition to their workload. Training sessions also serve as a message that the organization is going ahead and will require that its developers use these practices. During the training sessions, it is necessary to indicate that, despite everybody's good will, errors are bound to happen while using new practices. This will help reducing developers' level of anxiety in their using these new practices. It is a good thing if a resource-person is available to help developers (i.e. hotline) when the latter face obstacles while implementing new practices.

8.7. Lesson 7: Manage the Human Dimension of the Process Improvement Effort

The author also wishes to make the reader aware of the importance of the human dimension in a process improvement program. The people responsible for these changes are often extremely talented software engineering practitioners, but they may not be too well equipped in change management skills. The reason for this is simple: during their training, they focused on the technical dimension and not on the human aspect. However, the major difficulty in the whole improvement program is precisely the human dimension. Also while preparing the technical part of the action plan, the change management elements have to be planned (Laporte 1994). This implies, among other things, a knowledge of (1) the organization's history with regards to any similar efforts, successful or not, made formerly; (2) the company's culture; (3) the motivation factors; (4) the degree of emergency perceived and communicated by (a) the management, (b) the organization's vision, and (c) the management's real support. The authors are convinced that the success or the failure of an improvement program has more to do with managing the human aspect than managing the technical aspect.

8.8. Lesson 8: Process Improvement Requires Additional 'People Skills'

In an organization that truly wants to make substantial gain in productivity and quality, a major cultural shift will have to be managed. Such a cultural shift requires a special set of 'people' skills. The profile of the ideal software process facilitator is someone with a major in social work and a minor in software engineering. The implementation of processes implies that both management and employees will have to change their behaviors. With the implementation of processes, management will need to change from a 'command and control' mode to a more 'hands-off' or participative mode. As an example, if the organization truly wants to improve its processes, a prime source of ideas should come from those who are working, on a daily basis, with the processes, i.e. the employees. This implies that management will need to encourage and listen to new ideas. This also implies that the decision-making process may have to change from the autocratic style, e.g. 'do what you are told' to a participative style, e.g. 'let us talk about this idea'. Such a change requires support and coaching from someone outside the functional authority of the manager who has to change its behavior. Similarly, employees' behavior should change from being the technical 'heroes' who can solve any bug, from being passive and unheard in management issues, to work in teams and generate and listen to others' ideas to make improvement.

Also, in the first few months of the introduction of a new process, a new practice or a new tool, both management and employees must acknowledge that mistakes will be made. Unless a clear signal has been sent by management and a 'safety net' has been deployed to recognize this situation, employees will 'hide' their mistakes. The result is that not only the organization will not learn from them but other employees will make the same mistakes again. As an example, the main objective of the inspection process is to detect and correct errors as soon as possible in the software process. Management has to accept that in order to increase the errors detection rate, results from individual inspections will not be made public, only composite results from many inspections (e.g. at least ten inspections) will be made public. When this rule is accepted by management, employees will feel safe to identify mistakes in front of their peers instead of hiding them. The added benefit to correcting errors early in the process is that those who participated in an inspection will learn how to avoid these errors in their own work.

Facilitating such a change in behaviors requires skills that are not taught in technical courses. It is highly recommended that the people responsible for facilitating change be given appropriate training. The author recommends a course given by the SEI, the title of which is 'Managing technological change'. If there is a lack of such a course, the authors recommend two books that may facilitate the management of change: the first one (Block 1981) gives advice to anybody acting as internal consultant; the second one (Bridges 1991) gives the steps to be followed for writing and implementing a change management plan.

8.9. Lesson 9: Carefully Select Pilot Projects

It is also very important to carefully select pilot projects and participants to the pilots, since these projects will foster adoption of new practices throughout the organization. Also, first time users of a new process will make mistakes. It is therefore mandatory to properly coach the participants and provide them with a 'safety net'. If participants sense that mistakes will be used to learn and make improvements to the process instead of 'pointing fingers', the level of anxiety will be reduced and they will bring forward suggestions instead of 'hiding' mistakes.

Managing the human dimension of the process engineering initiative is the component which not only fosters the adoption of change but also creates an environment where changes could be introduced at an increasingly greater rate. Members of the engineering organization now realize that managing the 'soft stuff' is as important as managing the 'hard stuff'.

The utilization of models such as the CMM for software and systems engineering is slowly changing the culture of the organization from the 'not invented here' to the 'not reinvented here' mindset. Although a few practitioners still believe that they are different, most see the benefits of reusing someone else's work. They also see that the organization encourages them to look for solutions instead of constantly reinventing the wheel. Engineers are now intensively using the Internet to look for practices developed by other organizations and adapting these practices to the environment of the organization. Practitioners attend conferences sponsored by organizations such as the SEI and INCOSE to identify best practices for their utilization in day-to-day activities.

8.10. Lesson 10: Conduct Process Audits

Process audits should be conducted on a regular basis for two main reasons: first, to ensure that practioners are using the process, and second, to discover errors, omissions, or misunderstandings in the application of the process. Process audits help to assess the degree of utilization and understanding of the practioners. As an example, a documentation management process was released and practioners were asked to produce and update documents using this new process. It is widely known that engineers are not prone to documenting their work. We launched an audit and measured process compliance. As expected, see Table 4, results were not exhilarating. The engineering manager kindly reminded engineers, in writing, to use the process. He also informed them that a second audit would be performed in the future. As shown in Table 4, the results of the second audit are substantially better than the first audit. Also, the auditor gathered feedback from engineers; such information will be used by the process owner to improve the process.

Table 4. Results of audits performed on the documentation management process

Activity	Results from first audit	Results from second audit
Comments made by reviewers	38%	78%
Approval matrix completed	24%	67%
Effort log completed	18%	33%
Review checklist completed	5%	44%
Configuration management checklist completed	5%	27%
Distribution list completed	38%	39%
Document formally approved	100%	100%

8.11. Lesson 11: Conduct Team Effectiveness Surveys

We have found that such a tool promotes open discussion with members of a group besides improving the performance. Usually, people are not very prone to raise 'soft' issues. We have found that this tool provides the facilitators with information that help him probe delicate issues. As an example, if the majority of a working group reports that interpersonal communications are weak, the facilitator can probe the members and invite them to propose solutions. After a few meetings, the results of a new survey will show if the solutions really helped the team improve their communications.

8.12. Lesson 12: Start a Process Initiative from the Top Level Process

Our process improvement initiative was a bottom-up exercise, i.e. first software process was developed, then systems engineering, then project management, where each additional process 'sits' on top of the other. Historically, this was the selected strategy because, in 1992, only the software CMM was available, then came the systems CMM and after, the Body of Knowledge in project management. If an organization had to start a process initiative today it would be easier and more efficient to start from the top, by developing the project management process, then the systems and finally the software process. It would also be possible to develop these process in parallel once the requirements for the top level process were well stabilized.

8.13. Lesson 13: Get Support from Organizational Change Experts

As mentioned above, surveys were conducted in order to 'measure' issues such as culture, implementation history and team effectiveness. Once the surveys were compiled, we had some indications of organizational strengths and weaknesses. The difficult part was to decide what to do next. As an example, one issue on the survey is risk taking. If the survey showed that people resent taking risks, one possible cause for such behavior was that people did not want to be blamed for an error. Having found this cause was not too helpful since we would have to find the cause for this behavior, and so on. It would have been very helpful to have access to someone with expertise in organizational change. This would have saved a lot of long discussion and wrong answers.

8.14. Lesson 14: Tie Process Improvement activities to Business Objectives

It was observed that software and systems engineering process improvement really picked up momentum when a common focal point was created between management, engineers and customers. Understanding that the real benefit of process improvement lies in improving product quality, reducing time-to-market and cost, consequently improving the ability of the organization to better compete. Additionally, a multi-year process improvement plan (PIP) is a very important tool to illustrate the links between business objectives, project requirements and process development or improvement. Essentially the PIP illustrates that the engineering of processes is not a paper exercise but an important infrastructure for the successful accomplishment of projects. Being a multi-year plan, the PIP also shows to practitioners the long-term commitment of management to process improvement activities.

8.15. Lesson 15: Adopt a common vocabulary

To succeed in any project endeavor, a common vocabulary is a basic requirement. As we developed the processes, we realized that different players had different meaning for the same word, or the same word had different meanings, and some words were not well known to some individuals. We therefore mandated one team member as the 'glossary keeper'. His role was to collect a vocabulary, propose some 'clean-up' in the terminology, and to gradually build a common glossary for all processes that have to be disseminated across the process users.

9. CONCLUSION

We have shown that the development and deployment of engineering and management processes entails technical and management competencies. Five elements are necessary for a successful implementation of organizational changes. First, management sets a direction and process objectives are linked to business objectives. Without a clear direction, confusion may mislead people from reaching the desired change. Second, people are trained to perform new tasks. Without the proper training, anxiety among the organization's staff is likely to slow down the occurrence of change. Third, incentives are provided to facilitate the adoption of changes. Fourth, resources are estimated and provided, otherwise frustration may put an end to the organization's willingness to change. Fifth, an action plan is developed and implemented to avoid false starts (SEI 1993). Also, these years of process improvement activities have demonstrated that constant attention to the 'people issues' is critical to the success of technological changes. We suggest managing those 'people issues' as risk items and to track them all along the improvement effort.

REFERENCES

Alexander, M. 1991. The Encyclopedia of Team-Development Activities, edited by J. William Pfeiffer. University Associates, San Diego, California.

Basili, V. and S. Green. 1994. Software process evolution at the SEL. *IEEE Software*, July.

Bate, R. 1995. A Systems Engineering Capability Maturity Model, *version 1.1*. Software Engineering Institute, CMU/SEL-95–01, November.

Block, P. 1981. *Flawless Consulting*. Pfeiffer & Company.

Bridges, W. 1991. Managing Transitions. Addison Wesley.

Daskalantonakis, M. K. 1994. Achieving higher SEI levels. *IEEE Software*, July.

Dunaway, D. and S. Master. 1996. CMM-based appraisal for internal process improvement (CBA-IPI): method description. CMU/SEI-96-TR-007.

Fowler, P. and S. Rifkin. 1990. Software Engineering Process Group Guide. Software Engineering Institute, Report CMU-SEI-TR-24, September.

Gilb, T. and D. Graham. 1993. Software Inspection. Addison Wesley.

Laporte, C.Y. 1994. Process improvement and the management of change. Proceedings: 4th IEEE Computer Society Workshop on Software Engineering Technology Transfer, Dallas, April 28–29.

Laporte, C. and N. Papiccio. 1997. Development and integration of engineering processes at Oerlikon Aerospace. Proceedings of the Seventh International Symposium of the International Council on Systems Engineering, August 3–7, Los Angeles, California.

Laporte, C., A. Guay and J. Tousignant. 1998. The application of a systems engineering process to the reengineering of an air defense system. Proceedings of the 8th International Symposium of the International Council on Systems Engineering, July 26–30, Vancouver, Canada.

McFeeley, B. 1996. IDEAL: a user's guide for software process improvement. Software Engineering Institute, CMU/SEI-96-HB-001, Feb.

Olson, J.G., L.P. Gates, J.L. Mullaney, J.W. Over, N.R. Reizer, M.I. Kellner and R.W. Phillips. 1994. Software Process Framework for the SEI Capability Maturity Model: Repeatable Level. Software Engineering Institute, CMU/SEI-93-SR-07.

Paulk, M., B. Curtis, M.B. Chrissis and C.V. Weber. 1993. Capability Maturity Model for Software. Software Engineering Institute, SEI/CMU-93-TR-24.

Popick, P.R. and S.A. Shear. 1996. Ten lessons learned from implementing integrated product teams. Proceedings, International Council on Systems Engineering 6th Annual International Symposium, Boston, July 7–11.

PMI. 1996. *A Guide to the Project Management Body of Knowledge*. Project Management Institute.

Radice, R. 1985. A programming process architecture. *IBM Systems Journal*, **24**, no. 2.

Scholtes, P. 1996. The Team Handbook. 2nd edn, Joiner Associates Inc.

SEI. 1993. Managing Technological Change, *Course notes*. Software Engineering Institute.

Siddall, D.R. 1996. The Team Notebook. Publication in process.

SPC. 1995. *A Tailorable Process for Systems Engineering*. Software Productivity Consortium, SPC-94095-CMC, January.

SPC. 1996. *Integrated Systems and Software Engineering Process*. Software Productivity Consortium, SPC-96001-CMC, May.