Software and Systems Engineering Process Improvement at Oerlikon Aerospace

Claude Y. Laporte, Nicola R. Papiccio
Oerlikon Aerospace

Abstract

In order to reduce cycle time, increase customer satisfaction and lower costs, Oerlikon Aerospace has initiated, three years ago, a project to define and implement software and systems engineering processes. The initiative started by performing a formal assessment of current software engineering practices. An action plan was developed and multi-functional working groups were tasked to define and facilitate the implementation of software processes. A second initiative was started a year ago with the objective of defining and implementing a systems engineering process and, integrating to the systems engineering process the software engineering process already in use.

Background

Oerlikon Aerospace is a systems 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 60 software and systems engineers are involved in the development and maintenance of the system. The software is divided in four domains: weapon software, command, control and communication software, simulation software, and instrumentation software. The softwares were written in a variety of languages ranging from assembler to Ada. The softwares have been documented using military standards such as 1679, 2167A and 498. Over 20 software engineers maintain the software assets.

Development of a Software Engineering Process

At Oerlikon Aerospace, the approach, to process engineering 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.

Essentially, the software process improvement initiative followed the five phases of the IDEAL model (McFeeley 1996). The five phases of the model are: 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.

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 Oerlikon Aerospace, the president approved the establishment of a Software Engineering Process Group (SEPG)(Fowler 1990). A budget was also approved for the conduct of a Software Process Assessment (SPA) and the development of an action plan. Briefing sessions were held to inform the organization about the software process improvement effort.

During the Diagnosing phase (spring 1993), a 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 (see figure 1). 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. Because of its simplicity, the ETVX (Radice 1985) notation was selected for the description of the processes. To help define the processes, the working groups also used extensively a document produced by the SEI (Olson 1994), that describes each Key Process Area (KPA) of the Capability Maturity Model for Software (CMM) (Paulk 1993) using the ETVX notation.

- Review the findings of the assessment
- Introduction to the capability maturity model (cmm)
- Preparation of a plan by the working group
- Brainstorm on strenghts and weaknesses of current process
- Understand the current process
- Compare the current process with the cmm
- Describe first level (i.e. overall view) process steps
- Describe second level of the process using the selected notation
- Describe/update, if necessary, third level components:
    - Procedures
    - Users' guides
    - Checklists
- Review process steps
- Select a pilot project
- Brief participants
- Monitor the pilot
- Institutionalize the process:
    - Modify, if necessary, policies and procedures
    - Develop the training material
    - Train all users (technical and non-technical) of the process
    - Monitor the utilization of the process
    - Measure the process and products
    - Improve the process

Figure 1: Process Definition Steps

During the Acting phase, initiated in winter 1994, working groups started their activities. Working groups were kicked of in one to two months 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 level of details: the top-level view is a black box approach describing the major steps required to satisfy 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 processes were developed, tested in pilot projects and implemented: software development, software maintenance, software project planning and tracking, software quality assurance, software configuration management, software subcontractor management, documentation management and document inspection (Gilb 1993).

To illustrate the work performed, the planning and tracking process is described. At the higher level of details, there are three phases (see figure 2): the planning activities during a proposal phase, the project planning phase after contract award, and the project tracking phase. The proposal phase either takes the original vision of a potential product and transforms it into a business case or, for a contractual development, the requirements of the request for proposal are analyzed: size, cost and schedule estimates are performed, and a risk analysis is done. For both cases the main outcome of this phase is a go no-go decision. Since, during the contract negotiation phase, it is possible that some requirements (i.e. schedule, software requirements) have been modified, the planning phase after contract award is a requirred to finalize the plans prepared during the proposal phase. During the third phase, project data are collected, analyzed and adjustments to the initial plans are made.

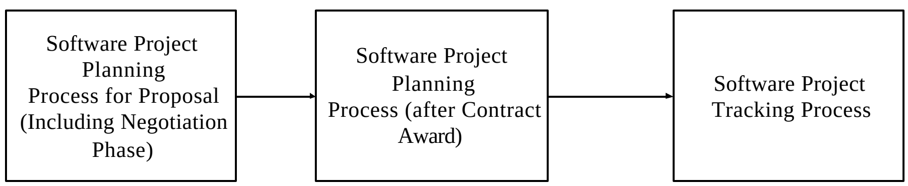
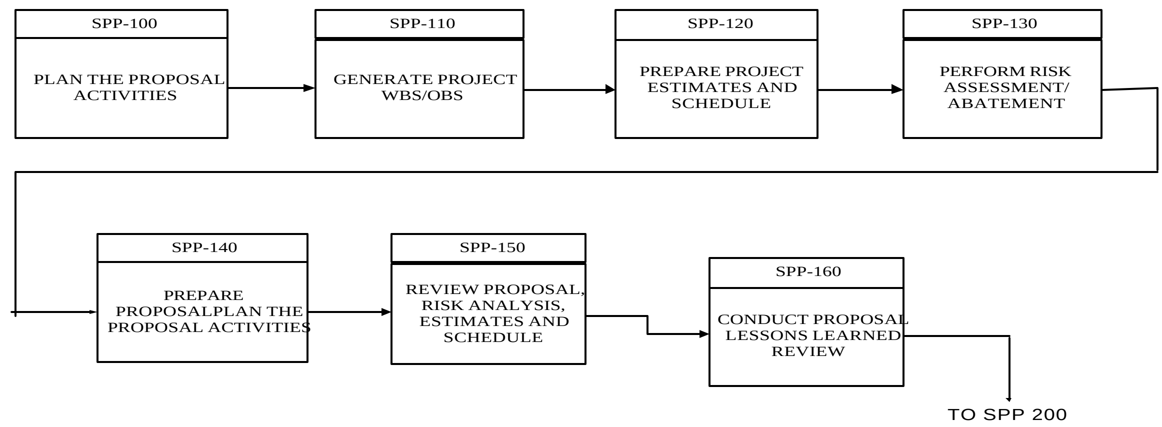


Figure 2: Three Phases of the Project Planning and Tracking Process

The second level of details of the planning and tracking activities during the proposal phase is illustrated in figure 3. As shown, each step of the process is numbered; also, each step is defined with a verb and a noun. The steps could be used as building blocks and could be linked together according to the needs of the project. It is the responsibility of the project manager to tailor the building blocks. Even though the steps are illustrated as a linear set of steps, feedback to previous steps are allowed. Feedback loops have not been illustrated in order not to clutter the diagrams.



WBS = Work Breakdown Structure
OBS = Organization Breakdown Structure

Figure 3: Software Planning Process for Proposal

Figure 4 illustrates the third level of details. The figure shows the ETVX diagram of step SPP-120. Since the diagram cannot contain all the information for a particular step, diagrams are complemented by a textual representation where all elements of the steps are listed. In the process engineering guidebook, each step is illustrated using two notations: the ETVX diagram and the textual description. In the guidebook binder, the diagrams are on the left side and the textual information are on the right side, i.e. facing the ETVX diagrams.

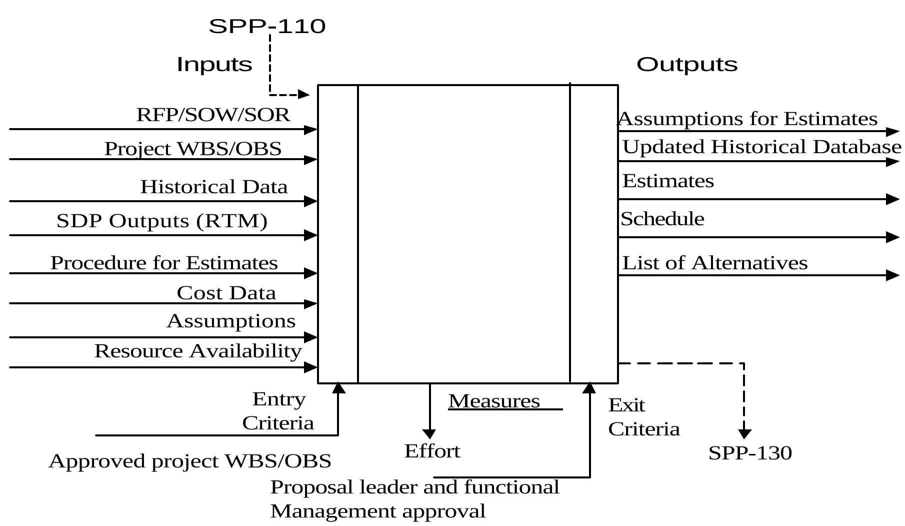


Figure 4: ETVX Diagram of Step SPP-120

A reverse engineering process is presently being defined. It will draw on the experiences based on the process developed under the STARS program (Software Technology for Adaptable, Reliable Systems) (STARS 1995). The reverse engineering process will have the following three major steps: first, a define project step which will include 1) define objectives, 2) identify baseline, 3) define reengineering project plan; a second major step to reverse engineer the software system, and a third major step to “forward” engineer the software.

In order to constantly improve the process, all users have been invited to propose corrections, modifications or improvements to the process. A process improvement form has been distributed to all users of the process. The SEPG collects, analyzes and proposes improvements to appropriate process owners. Once the modification to the process is completed, a new version is distributed to all users.

Audits were also performed on all projects. The objective of the audits 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 audits 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 has 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 each audit, a mini action plan was developed to address the findings and implement corrective actions.

Another feature was built in the process in order to capture the lessons learned. In our organization, we have defined the software planning and tracking process 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. During the planning phase, the project has to estimate the effort required to conduct lessons learned reviews. During the tracking phase, lessons learned reviews are performed in each project. In order to make sure that the lessons are learned by the organization, each lesson is analyzed in order to identify if a process step could be improved (Basili 1994). If this is the case, modifications to the process, methods or guides are made before the project is allowed to exit from the last step of the tracking process.

As the processes are being used in current 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 practioner will have access electronically to documents, the PAL will contain electronic copies of documents produced. The PAL librarian has read and write privileges while practioners have only read privileges. The librarian will also perform configuration management functions on the artifacts of the PAL. Table 1 lists the artifacts that are stored in the PAL as projects are producing documents.

• Software Engineering Policy	• Software Version History
• 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 (SDP)	• 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
• Lessons Learned	

Table 1: Content of the Process Asset Library

Finally during the Leveraging phase, lessons learned from projects and processes are collected, analyzed and implemented. These lessons will be used to prepare the next improvement cycle which is planned to start in the fall of 1996 by a re-assessment of the software engineering process (i.e. a CMM-Based Appraisal for Internal Process Improvement, CBA IPI) (Dunaway 1996) by certified SEI assessor from the Applied Software Engineering Centre (ASEC).

Development of a Systems Engineering Process

Although the organization had in use ISO-9001 compliant procedures describing the work that systems engineers have to perform, it was decided that a systems engineering process had to be defined in order to integrate, seamlessly, disciplines associated with systems engineering. In 1995, we conducted an internal assessment of our systems engineering practices using the Systems Engineering Capability Maturity Model (SE-CMM) (Bate 1995) and the SE-CMM Appraisal Method (SAM). The objective was to help identify priorities for improvement within the 18 process areas of the SE-CMM. Three systems engineers and two management staffs answered the SAM questionnaire. Results from the questionnaire were compiled and a maturity level for each process area was computed. After analysis of the results management decided to put a higher priority on the engineering process areas as defined in the SE-CMM. Managers reviewed the current literature 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 11 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 is to integrate the current software engineering processes to the systems engineering process. This objective is part of the progress that has to be made to work at SEI level 3 of the CMM for software.

The GSEP document describes, using the IDEF notation (USAF 1981), management and technical activities and also the artifacts produced by each activity. The major management activities, as illustrated in figure 5, are: understand context, analyze risk, plan increment development, track increment development and develop system. The major technical activities, as illustrated in figure 6, are: analyze needs, define requirements, define functional architecture, synthesize allocated architecture, evaluate alternatives, validate and verify solution and control technical baseline. Each major activity is broken down in a certain number of smaller activities which are described, individually using the ETVX notation. Our strategy is to define a beta version of the technical activities, then of the management activities, use the beta version on pilot projects and make corrections to both management and technical activities of the process before full deployment.

In addition to defining the process, each member of the working group has a secondary duty, as each step of the beta version of the process is defined, members of the working group were tasked to collect the following information: updates to process descriptions, monitor compliance with the SE-CMM, monitor the interfaces with the software engineering processes, 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, lessons-learned, and establish the systems engineering process asset library. Finally, since Oerlikon Aerospace has been certified as an ISO 9001 organization, in 1993, one representative from the quality assurance department monitors our progress in order to make sure that the process being defined is compliant to ISO requirements. Oerlikon Aerospace is planning to perform an independent systems engineering assessment, by the end of 1997, to measure the progress made and plan a second phase of systems engineering process improvements.

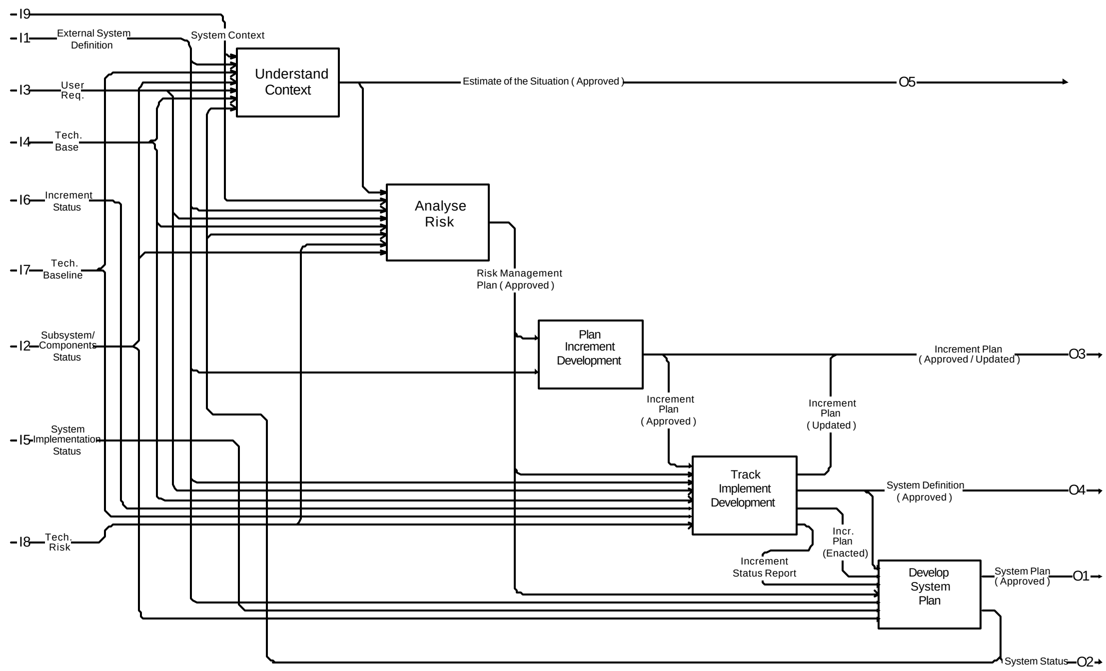


Figure 5: Management Activities of the Systems Engineering Process

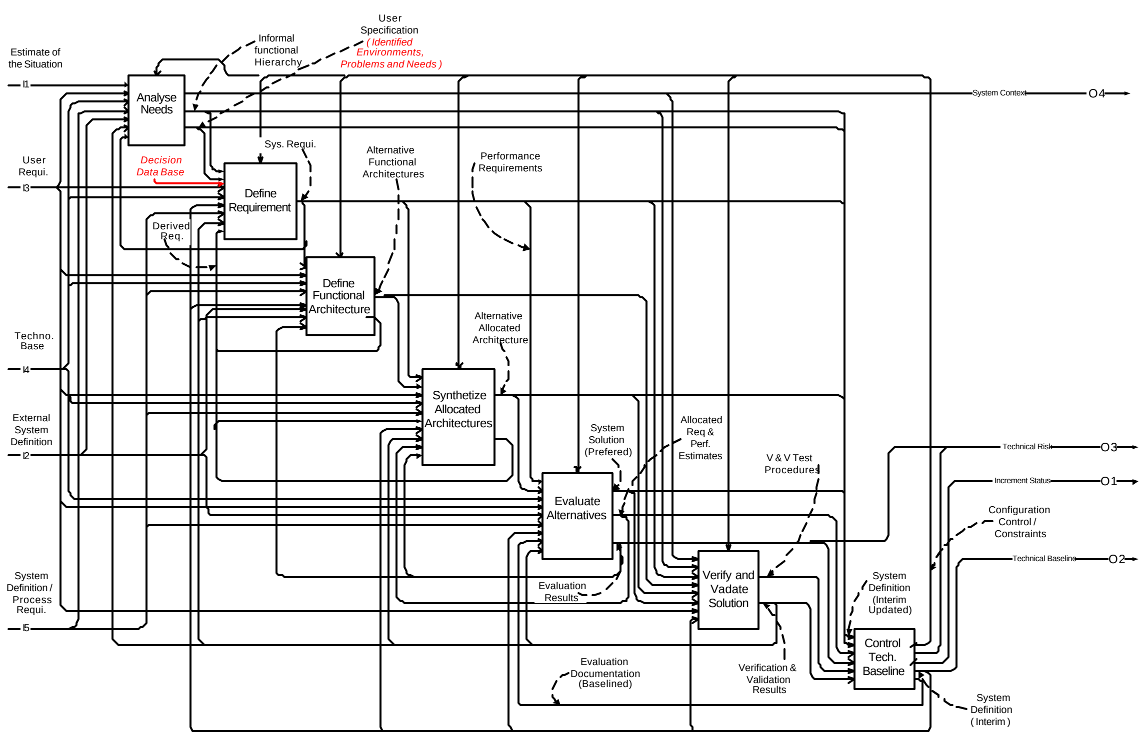


Figure 6: Technical Activities of the Systems Engineering Process

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 in order to facilitate the development, the implementation and the adoption of the processes, methods and tools (Laporte, 1993). As an example, to build the sponsorship level, the president attended a one-day 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 to 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: software engineering, systems engineering, sub-systems engineering, quality assurance, contract management, and configuration management. Each working group was managed like a project. It had a charter, a budget and a schedule. A process owner, i.e. a manager responsible for the definition, implementation and improvement of each process was part of a working group. A member of the SEPG acted as a facilitator in each working group. Therefore, 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.

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-of meeting of their group. The proposed guidelines are listed in table 2. It was decided that consensus decision making was the preferred decision-making option. We defined consensus, according with the definition found in the Team Handbook (Scholtes 1988): 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. During meetings we use "thumb voting" procedure (Popick 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 later 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.

- One conversation at a time.
- In the meeting or out, but not both (i.e. participants should make a commitment to participate to the meeting for the full duration).
- 100-mile rule (i.e. no interruptions, e.g. telephone messages, allowed unless urgent)
- How decision will be made (e.g. by consensus, majority or minority rule, autocracy, unanimity).
- Once a decision is made, participants support it inside and outside the meeting.
- Be as open as possible.
- We listen, with respect, to others and do not interrupt them.
- Silence is consent.
- Few recreational stories.
- Differences are respected.
- Avoid blaming individuals.
- We come prepared to meetings.
- We publish minutes and action items at each meeting.

Table 2: Proposed meeting guidelines

Finally, on a periodic basis, members of the working groups have to evaluate the effectiveness of their group. A survey (Alexander 1991) is distributed at the end of a meeting, members are asked to complete the survey and to 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. At the following meeting, issues that were surfaced by members are discussed in order to generate suggestions for improvement.

Lessons Learned

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 project requirements and process development. 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 practioners the long-term commitment of management to process improvement activities.

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 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" (NIH) to the "Not Reinvented Here" (NRH) mindset. Practioners see the benefits of reusing someone else's work. They also see that the organization encourage 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. Practioners attend conferences sponsored by organizations such as the SEI and INCOSE to identify best practices for their utilization in day-to-day activities.

Next Steps

A training program will be defined. For software engineers, we have identified a career development program developed by the British Computer Society (BCS) (Taylor 1991). This program is currently used by employers, since 1985, mainly in United Kingdom and in other countries. This program is becoming available in North America through DPMA (Data Processing Management Association). The key features of the program are: cyclic and pre-planned and documented programs of training and experience worked out between employer and employee; industry-wide performance standards; evaluation of the completion of these program by independent experienced professionals; registration of completed programs in a standardized Log-Book owned by the employee. The performance standards are based on the BCS's Industry Structure Model (ISM). The ISM defines over eighty detailed job descriptions and up to 10 competence levels, for each job description, ranging from an unskilled entry level to a senior manager or director. Each competence level describes the recommended academic background, the experience and level of skill at entry, tasks and attributes, and training and development required. In addition to the BCS program, the practices described in the CMM level 3 training KPA (Paulk 1993) and in the People CMM (Curtis 1995) will also be used to define the training program (Carpenter 1995) for software engineers. A similar approach will be used for the other engineering disciplines.

Presently most of our process assets are paper documents. As we progress, these documents will be made available on the company local area network. Practioners will have read only access privileges. Only process owners and the PAL librarian will have all read and write privileges.

As we are making progress in institutionalizing systems and software engineering processes and methods, we will be using more CASE tools. Since CASE tools are quite expensive both in acquisition costs and maintenance costs, we cannot afford to make mistakes. But as the organization matures, our requirements for CASE tools will be better defined and the tools selected will better support the execution of the systems and software engineering processes and methods.

As the engineering division moves toward concurrent engineering and integrated product development, the structure of the organization as well as the performance management process will need some adjustments in order to capture the full benefits of these new work practices.

Conclusion

Our organization has made substantial investments toward the definition and implementation of engineering processes, methods and tools. Improvements require significant investments but, both the technical and management activities will allow complex projects to be developed in a disciplined environment. Engineers and managers will be able to perform their activities more effectively and efficiently. The engineering division is slowly moving from the “not invented here” to the “not reinvent here” culture.

Biographies.

Claude Y. Laporte obtained in 1973 a Bachelor in Science from le Collège Militaire Royal de Saint-jean. In 1980, he obtained a MS in physics at Université de Montréal, and in 1986, a MS in Applied Sciences from the Department of Electrical and Computer Engineering at École Polytechnique de Montréal. He was an officer within the Canadian Armed Forces during 25 years and a professor for over 10 years. From 1988 to 1992, he was involved in the implementation of the Applied Software Engineering Centre. He left the Canadian Forces in 1992 at the rank of major. Since then, he has joined Oerlikon Aerospace where he coordinates the development and implementation of software and systems engineering processes, methods and tools. He also chairs the Montréal Software Process Improvement Network (Montréal SPIN).

Nick Papiccio graduated with a Bachelor degree in Administration Sciences in 1980 from le Collège Militaire Royal de Saint-Jean. He has also completed all the courses for the Master in Project Management with the Université du Québec à Montréal. Since 1982, he has been involved in systems and software engineering especially with the Canadian Patrol Frigate Program. He retired from the Canadian navy as a lieutenant-commander specialized in software engineering. During his tenure in the navy, he was trained by Ed Yourdon in software design and development (1984). He took many courses in quality engineering, configuration management, systems engineering, logistic engineering and instructional system development. Since 1995, he is the manager of software engineering at Oerlikon Aerospace. He has introduced the first I-CASE environment in Canada, integrating both software and systems engineering in a common process framework. He is currently involved with the creation of a Center of Excellence in Software and Systems Engineering for Canada.

References

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

Basili, V., Green, S., “ Software Process Evolution at the SEL”, IEEE Software, July 1994.

Bate, R., “ A Systems Engineering Capability Maturity Model”, version 1.1, Software Engineering Institute, CMU/SEI-95-01, November 1995.

Carpenter, M.B., Hallman, H.,K., “Training Guidelnes: Creating a Training Plan for a Software Organization”, Software Engineering Institute, CMU/SEI-95-TR-007, September 1995.

Curtis, B., et al, “ People Capability Maturity Model”, Software Engineering Institute, CMU/SEI-95-MM-02, September 1995.

Daskalantonakis, M.K., “Achieving Higher SEI Levels”, IEEE Software, July 1994.

Dunaway, D., Master, S., “CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description”, Software Engineering Institute, CMU/SEI-96-TR-007., April 1996.

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

Gilb,T., Graham, D., “Software Inspection”, Addison Wesley, 1993.

Laporte, C.Y., “Process Improvement and the Management of Change”, Proceedings: 4th IEEE Computer Society Workshop on Software Engineering Technology Transfer, Dallas, April 28-29 1994.

McFeeley, B., “IDEAL: A User’s Guide for Software Process Improvement”, Software Engineering Institute, CMU/SEI-96-HB-001, Feb. 1996.

Olson, T.G. et al., “A Software Process Framework for the Software Engineering Institute’s Capability Maturity Model”, Software Engineering Institute, CMU/SEI-94-HB-01, Sep. 1994.

Paulk, M. et al, “Capability Maturity Model for Software”, Software Engineering Institute, SEI/CMU-93-TR-24, 1993.

Popick, P.R., Shear, S.A., “Ten lessons Learned from Implementing Integrated Product Teams”, Proceedings, International Council on Systems Engineering 6th Annual International Symposium, Boston, July 7-11, 1996.

Radice, R., “A Programming Process Architecture”, IBM Systems Journal, vol. 24, no. 2, 1985.

Scholtes, P. “ The Team Handbook”, Joiner Associates Inc., 1988.

Siddall, Donald, R., “The Team Notebook”, Publication in process, 1996.

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

SEI, “Relationships Between the Systems Engineering Maturity Model and Other Products, Version 1.0”, Software Engineering Institute, CMU/SEI-94-TR-26, Nov. 1995.

STARS, “Army STARS Demonstration Project Experience Report”, United States Air Force, Air Force Material Command, Feb. 1995.

Taylor, J.A., “ Training, Career Development and Registration for Safety Critical Software Systems Specialists”, IEEE AES Systems Magazine, September 1991.

USAF, “Integrated Computer-Aided Manufacturing Architecture”, Function Modeling Manuel (IDEF0), United States Air Force, AFWAL-TR-81-4023, 1981.