

SOFTWARE AND SYSTEMS ENGINEERING PROCESS DEVELOPMENT AND INTEGRATION AT OERLIKON AEROSPACE

Claude Y. Laporte, Nicola R. Papiccio

Oerlikon Aerospace

This paper describes the approach used by Oerlikon Aerospace since 1993 to define and implement software and systems engineering processes:

- First, the steps taken to assess and define a software process are described using the Software Engineering Institute's Capability Maturity Model (SEI CMM) framework.
- The steps to develop the systems engineering process using the SEI CMM framework and two processes from the Software Productivity Consortium are then described.
- Process integration is discussed.
- Since the human dimension of the implementation of new technologies is critical to the success of our effort, a few human issues are discussed.
- Finally, lessons learned and the next steps are described.

Background

Oerlikon Aerospace is an integrator of a complex laser-guided missile air defence system. The system consists of five technology/product families:

1. processing and display
2. platform system
3. sensors and effectors
4. command, control, communication and intelligence
5. readiness system (for example, training, simulators, and test).

Over 80 software and systems engineers are involved in the development and maintenance of the system. The software is divided into five domains:

- weapon software
- command, control, communication, and intelligence software
- simulation software
- instrumentation software
- training and test software.

The different software domains were written in a variety of languages, ranging from assembler to Ada. All software has been documented using military standards such as 1679, 2167, and 498. Over 40 software engineers maintain the software assets.

Development of a Software Engineering Process

At Oerlikon Aerospace, the approach to process engineering was fourfold:

1. Define a process and bring it under management control.
2. Support the process with methods.
3. Support the process and methods with appropriate tools.
4. Train all personnel in the utilisation of processes, methods, and tools.

Essentially, the software process improvement initiative followed the five phases of the IDEAL model [11]. Each of the five phases of the IDEAL model (see Figure 1) is described in more detail in the following paragraphs:

1. Initiating the improvement program

During the Initiating phase (Fall 1992), a business case was prepared and presented to the President. Recognising that software engineering is a core competence of Oerlikon Aerospace, the President approved the establishment of a Software Engineering Process Group (SEPG) [8]. 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 organisation about the software process improvement effort.

2. Diagnosing the current state of practice

During the Diagnosing phase (Spring 1993), an SPA was performed jointly by the SEPG and independent assessors

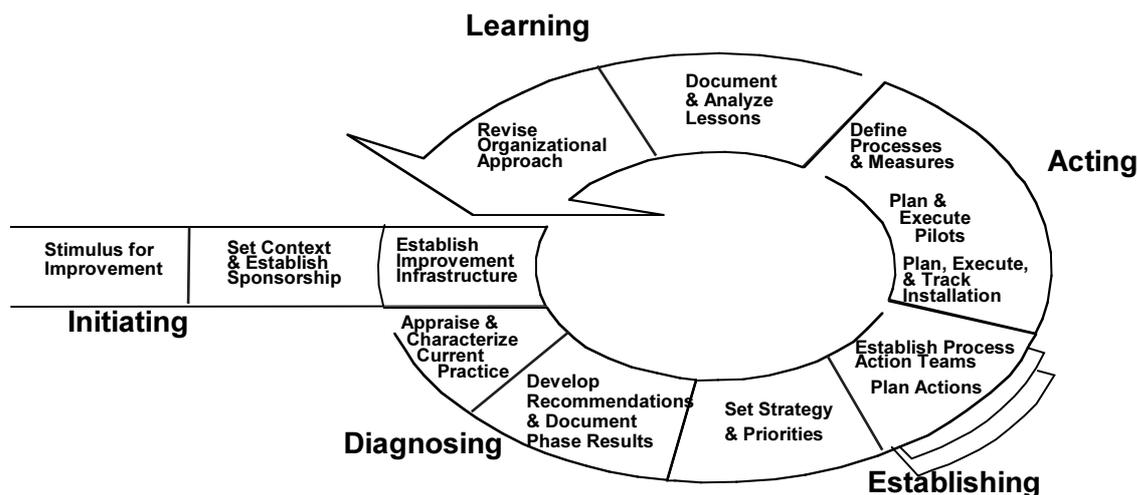


Figure 1: The IDEAL Improvement Cycle

trained 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.

3. Establishing the plans for the improvement

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—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 subsystems engineering, quality assurance, and configuration management. Each member of the various working groups would spend up to eight 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
- reference documents.

We decided to use a modified version of the ETVX [16] notation for the description of the processes. The notation describes, for each step of the process, the inputs required to perform the activities, the outputs produced, the entry criteria that allow activities to be performed, the exit criteria that allow artifacts to exit the current step, and the measures that would be captured when executing the activities. To help define the processes, the working groups also extensively used a document produced by the SEI [12] that describes each Key Process Area (KPA) of the CMM [13] using the ETVX notation.

4. Acting on the plans and recommended improvements

During the Acting phase (Winter 1994), working groups started their activities, each being kicked off in one to two month intervals. In this way, problems inherent to the dynamics of the 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 was described in three levels of detail:

- a. The top-level view is a black-box approach, describing the required major steps satisfying the goals of the KPAs.
- b. A second level of detail describes each black-box with the following information:
 - the objective of the activities to be performed
 - the inputs required to perform the activities
 - a list of activities
 - the outputs produced
 - the entry and exit criteria controlling the initiation and completion of each process step
 - the measurements (such as size, effort, quality)

- the persons responsible for performing and supporting each process step.
- c. At the third level of detail, methods are described in process guides (for example, size estimation, risk assessment). Each person who had to use the processes received and was trained in the utilisation of his/her own copy of the software engineering guidebook containing 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
- document inspection [9].

A reverse engineering process is being defined presently. It will draw upon the experiences of the process developed under the STARS program (Software Technology for Adaptable, Reliable Systems) [22]. The reverse engineering process will have the following three major steps:

- a. a define project step which will include these activities:
 - define objectives
 - identify baseline
 - define reengineering project plan.
- b. a major step to reverse engineer the software system
- c. a major step to "forward" engineer the software.

To illustrate the work performed, the planning and tracking process was described. At the higher level of detail, there were three phases (see Figure 2):

- a. the planning activities during a proposal phase
- b. the project planning phase after contract award
- c. the software 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, analyses the requirements of the request for proposal: size, cost, and schedule estimates are performed as well as a risk analysis. In both cases, the main outcome of this phase is a go/no-go decision. Since during the contract negotiation phase some requirements (such as schedule or software requirements) may have been modified, the planning phase is required after the contract award to finalise the plans prepared during the proposal phase. During



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

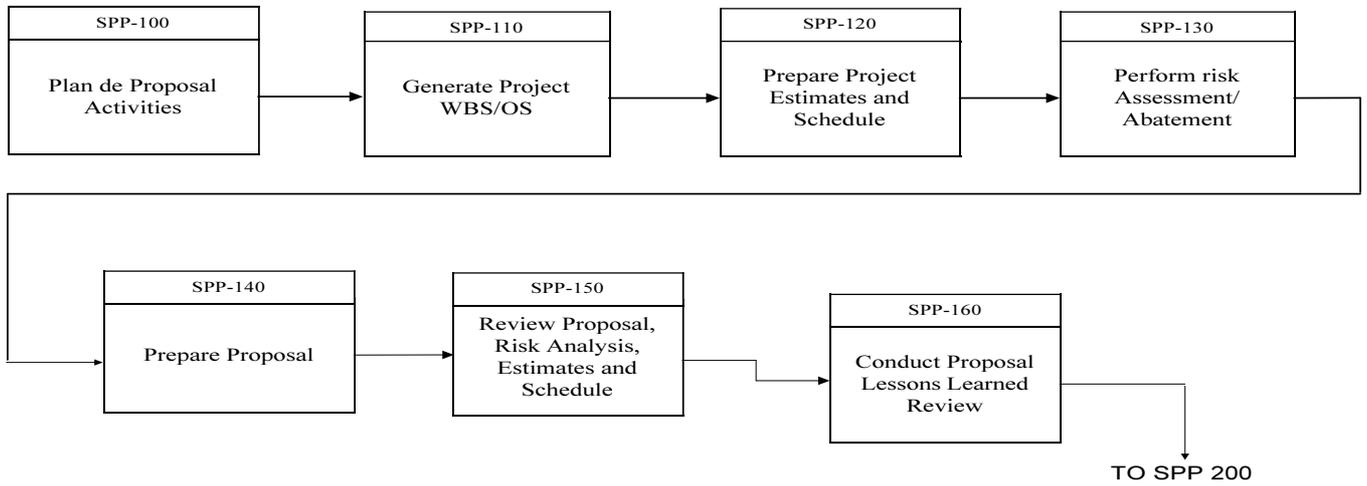


Figure 3: Software Planning Process for Proposal

WBS= Work Breakdown Structure

OBS= Organisation Breakdown Structure

the third phase, project data are collected and analysed, and adjustments to the initial plans are made.

The second level of detail of the planning and tracking activities during the proposal phase is illustrated in Figure 3. As shown, each step of the process is numbered. In addition, 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. Although the steps are illustrated as a linear set of steps, feedback to previous steps is allowed. (Feedback loops were not illustrated so as not to clutter the diagrams.)

Some activities can also be done concurrently. Figure 4 illustrates the third level of detail, showing the ETVX diagram of step SPP-120. Since they cannot contain all the information for a particular step, the diagrams are complemented by a textual representation in which all elements of the steps are listed (for example, step description, activities, and references). In the process engineering guidebook, each step is illustrated using two notations: the ETVX diagram and the textual description. In the guidebook, the diagrams are on the left side and the textual information is on the right, facing the ETVX diagrams. To constantly improve the process, all users were invited to propose corrections, modifications, or improvements to the

process. A process improvement form was distributed to all users of the process. The process owner collected, analysed, and implemented the improvements to process. Once the modification to the process was completed, a new version was distributed to all users. Self-assessment was also performed on all projects.

The objective of the self-assessment was not to "fix the people" but to bring to the surface any barriers to the institutionalisation of the processes. The focus on process rather than on 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. It used scoring guidelines developed by Motorola [6] incorporating a ten-level scoring scale that allows for a finer evaluation of the institutionalisation of each KPA. The scoring guidelines measure the attainment of the following three elements:

- the approach: criteria that show the organisation's commitment to and management's support for a practice
- the deployment: the breadth and consistency of practice implementation

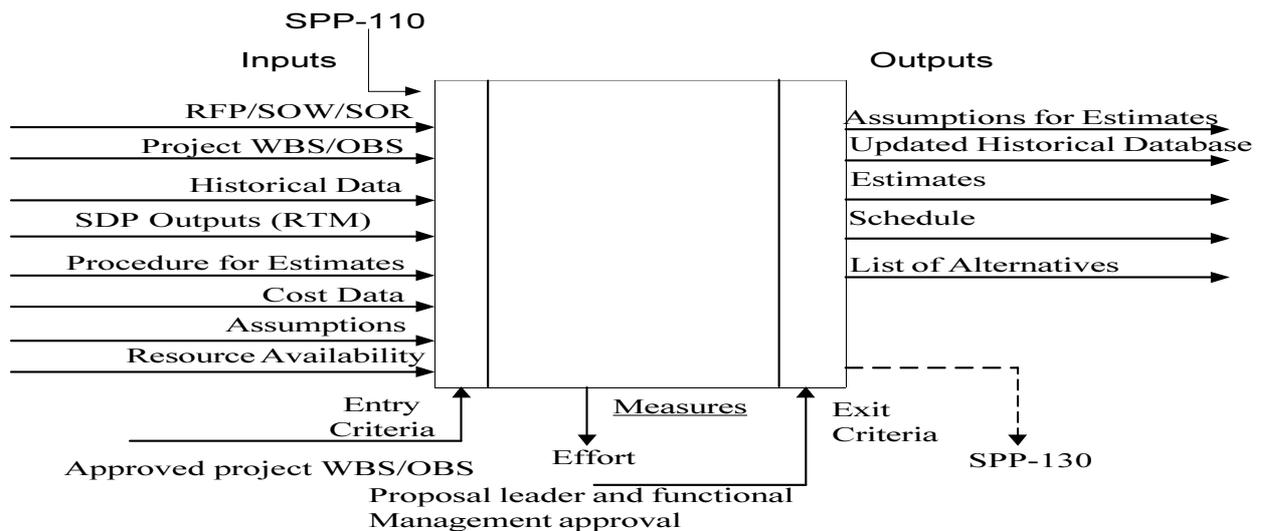


Figure 4: ETVX Diagram of Step SPP-120

c. the results: the breadth and consistency of positive results over time.

By using such a scale, it became easier to measure the progress made by each team from one audit to another. After the self-assessment, an action plan was developed to address the findings and implement corrective actions.

Another feature was built into the process to capture the lessons learned. In our organisation, we defined the software planning and tracking process such that it is the first process to be initiated in any project and 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. Usually, in a lessons-learned meeting, the members of the project are present, and someone from the SEPG facilitates the meeting. To make sure the lessons really are learned by the organisation, the process owner analyses each lesson with the SEPG to identify whether or not a process step could be improved [2]. If an improvement is identified, modifications to the process, methods, or guides are made under the supervision of the process owner and the SEPG.

As the processes are being used in current projects, artifacts are collected and stored in a process asset library (PAL). The PAL currently contains mostly paper documents. Since the organisation is moving toward an environment in which each practitioner will have electronic access to documents, the PAL will contain electronic copies of documents produced. The PAL librarian has read/write privileges, while practitioners have read-only privileges. The librarian will also perform configuration

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 of 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

Table 1: Contents of the Software Process Asset Library (PAL)

management functions on the artifacts of the PAL. Table 1 lists the artifacts stored in the PAL as projects produce their documents.

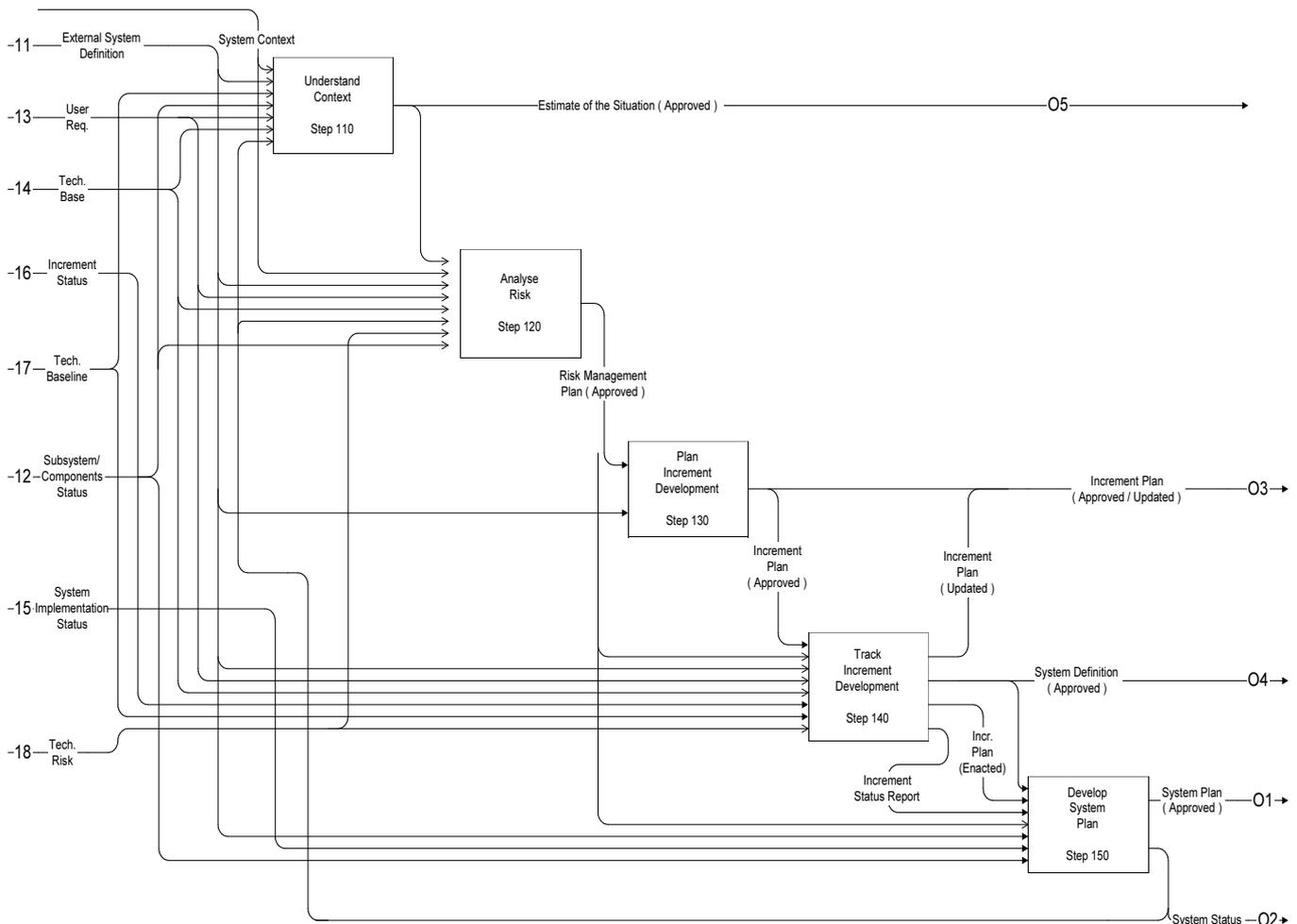


Figure 5: Management Activities of the Systems Engineering Process.

5. Learning the lessons learned and the business results of the improvement effort

Finally during the Learning phase, lessons learned from projects and processes are collected, analysed, and implemented. These lessons are used to prepare the next improvement cycle, by performing a reassessment of the software engineering process.

The second formal assessment (a CMM-Based Appraisal for Internal Process Improvement, CBA IPI) [7] performed by an SEI assessor from the Applied Software Engineering Centre (ASEC) was conducted in February 1997. Oerlikon Aerospace achieved a strong Level 2 rating since it also satisfied eight of the 17 goals for Level 3 certification. Two Level 3 KPAs were fully satisfied: Software Product Engineering and Peer Review (for the document inspection process).

Development of a Systems Engineering Process

Although the organisation had been registered to ISO 9001, it was decided that the systems engineering process had to be defined to integrate seamlessly the disciplines associated with systems engineering. In 1995, we conducted a self-assessment of our systems engineering practices using the Systems Engineering Capability Maturity Model (SE-CMM) [3] and the SE-CMM Appraisal Method (SAM).

The objective of the self-assessment was to help identify priorities for improvement within the 18 process areas of the SE-CMM. Three systems engineers and two management staff members answered the SAM questionnaire. Results from the questionnaire were compiled and a maturity level for each process area was computed. After completing the analysis of the results, 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 the SE-CMM and the Generic Systems Engineering Process (GSEP) developed by the Software Productivity Consortium [19] as frameworks. The GSEP has been developed to incorporate most of the practices of the SE-CMM. A working group, comprised 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 was to integrate the software engineering process into the systems engineering process.

The GSEP document describes, using the IDEF notation [24], management and technical activities and the artifacts produced by each activity. The major management activities, as illustrated in Figure 5 (page 13), are to:

- understand the context
- analyse risk

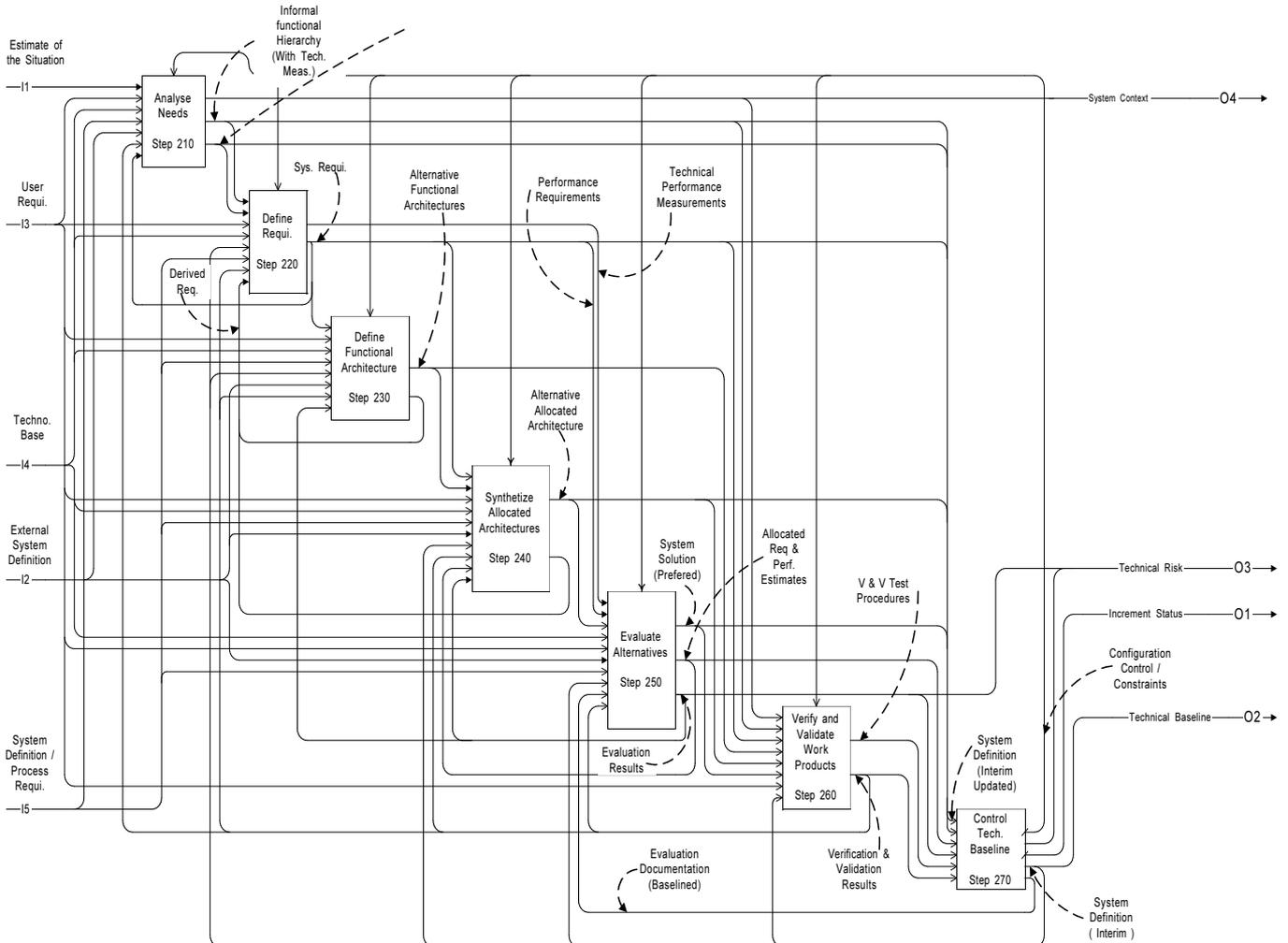


Figure 6: Technical Activities of the Systems Engineering Process.

- plan incremental development
- track incremental development
- develop the system.

The major technical activities, as illustrated in Figure 6 (previous page), are to:

- analyse needs
- define requirements
- define functional architecture
- synthesise allocated architecture
- evaluate the alternatives
- validate and verify the solution
- control the technical baseline.

Each major activity is broken down into a number of smaller activities that are described individually using the ETVX notation. Our strategy was to define a beta version of the technical and 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 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 the following information:

- 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 a glossary
- identify methods, best practices, artifacts, CASE tools, life cycle representations, project templates, estimation guidelines, course material, training resources, and lessons learned
- establish the systems engineering process asset library.

Finally, since Oerlikon Aerospace had been registered as an ISO 9001 organisation in 1993, one representative from the quality assurance department monitors our progress to make sure that the process being defined is compliant to ISO 9001 requirements. Oerlikon Aerospace is planning to perform an

independent systems engineering assessment in 1998 to measure the progress made and to plan a second phase of systems engineering process improvements.

Integration of Processes

As mentioned earlier, throughout the development of the systems engineering process, the working group kept the integration between this process and the software process on its agenda. It decided to adopt a document produced by the SPC [20] entitled Integrated Systems and Software Engineering Process (ISSEP) as a framework for the integration. Since many problems are discovered at integration time when developing complex computer-based systems, the solution is to develop a process that will decompose the systems into parts that can be developed independently and that can be integrated together easily at the system level.

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 7 illustrates the integration between processes. Figures 5 and 6 illustrate the activities in the boxes Manage Development Effort and Define System Increment. The Develop SW Configuration Item box is essentially the software engineering process, while the Develop HW Configuration Item box—the design engineering process—represents a process presently being documented.

As we integrated the processes, two types of issues surfaced. First, some activities had been documented in all processes. As an example, risk management activities had been defined in the software engineering process because it was felt that risk management was important. Since risk management is mandated in the SE-CMM, it was also documented in the systems engineering process. Also, a project management process was being developed with some risk management activities. To prevent duplication of these activities, the issue was resolved by assigning the primary responsibility for risk management to the project management process and leaving the dedicated risk checklists in the engineering processes.

Another type of issue surfaced because all the frameworks mandated some of the activities used. For example, the management of subcontractors is mandated by the SW-CMM, the SE-CMM, and the project management process of the Project Management Institute [15]. Since this process had already been defined during the software initiative, the scope of this process was broadened to include hardware acquisitions.

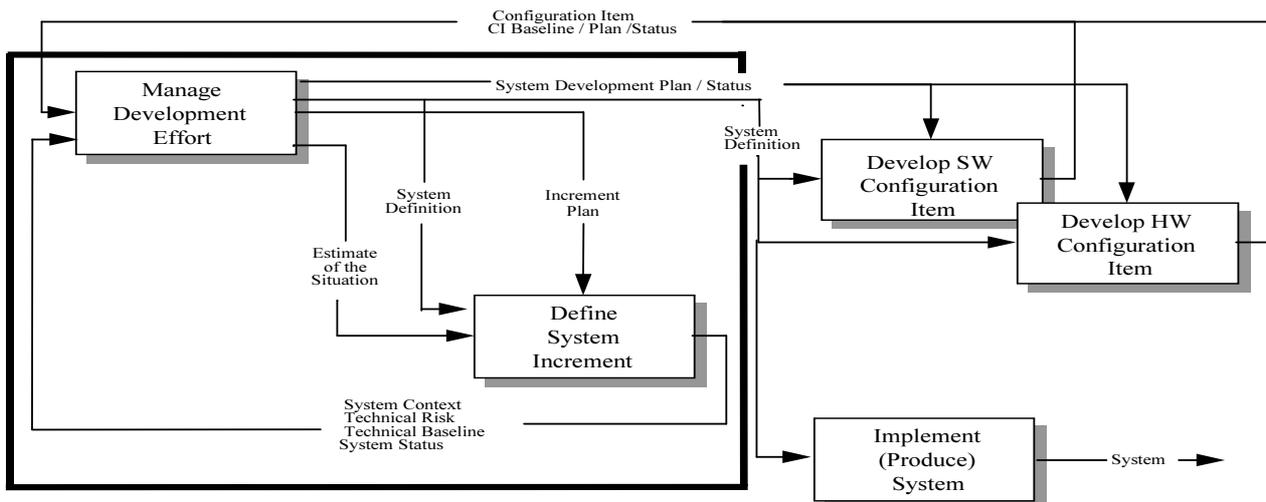


Figure 7: Integration of Processes

It was decided that the process ownership would remain with the department responsible for acquisitions. The relationship between the project management process and the subcontractor management process is viewed as a client-server relationship, in which a project manager issues his requirements for a particular acquisition, the requirements are then transferred to the subcontractor process, and once the goods are delivered to the satisfaction of the project, the subcontractor process is stopped.

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 to facilitate the development, implementation, and adoption of the processes, methods, and tools [10]. For example, to build the sponsorship level, the President of Oerlikon 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 on 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 to describe the progress made. Finally, surveys were conducted to assess the organisation's readiness to such a change in practices. The surveys identified strengths of the organisation and the potential barriers to the planned improvement program.

To obtain support and commitment for the future implementation of processes, working groups were staffed with representatives from many departments, including software engineering, systems engineering, subsystems 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—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. The process owner, therefore, 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.

To facilitate the conduct of working group activities, a number of meeting guidelines [18] 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 2.

<ul style="list-style-type: none"> • Allow only one conversation at a time. • Participate in the meeting or out, but not both (i.e. participants should make a commitment to participate to the meeting for the full duration). • Apply the 100-mile rule (i.e. no interruptions: e.g. telephone messages are not allowed unless urgent). • Establish how decisions 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. • Listen with respect to others and do not interrupt them. • Silence is consent. • Share few recreational stories. • Respect differences. • Avoid blaming individuals. • Come to meetings prepared. • Publish minutes and action items at each meeting.
--

Table 2: Proposed Meeting Guidelines

Consensus decision making was the preferred decision-making option. We defined consensus according to the definition found in [17]:

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 a "thumb voting" procedure [14] to make decisions by consensus. Thumb voting allows the following three alternatives:

1. If the proposition is favoured, the thumb is up.
2. If someone can live with the decision, the thumb is to the side.
3. 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 with which everyone can live. Finally, members of the working groups have to evaluate the effectiveness of their group.

A survey [1] was distributed at the end of a meeting, and members were asked to complete the survey and send it to the facilitator of their group. The survey addressed the following issues:

- goals and objectives
- utilisation of resources
- trust and conflict resolution
- leadership, control, and procedures
- interpersonal communications
- problem solving
- experimentation
- creativity.

At the meeting that followed, issues that were surfaced by members were discussed 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. The real benefit of process improvement lies in improving product quality and reducing time-to-market and cost. Consequently, it improves the ability of the organisation 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 practitioners the long-term commitment of management to process improvement activities.

It is also very important to carefully select pilot projects and participants in the pilots since these projects will foster the adoption of new practices throughout the organisation. 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 to make improvements to the process instead of pointing fingers, their levels of anxiety will be reduced and they will bring forward suggestions instead of hiding their mistakes.

Managing the human dimension of the process engineering initiative is the component that not only fosters the adoption of change but also creates an environment in which changes

could be introduced at an increasingly greater rate. Members of the engineering organisation now realise that managing the "soft stuff" is as important as managing the "hard stuff".

The use of models such as the CMM for software and systems engineering is slowly changing the culture of the organisation from the "Not Invented Here" (NIH) to the "Not Reinvented Here" (NRH) mindset. Practitioners see the benefits of reusing someone else's work. They also see that the organisation encourages them to look for solutions instead of constantly reinventing the wheel. Engineers now are using the Internet extensively to look for practices developed by other organisations and for adapting these practices to the organisation's environment. Practitioners attend conferences sponsored by organisations such as the SEI and INCOSE to identify best practices for their utilisation 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) [23]. Employers have used this program since 1985, mainly in United Kingdom and in other countries. This program is available in North America through DPMA (Data Processing Management Association). The key features of the program are:

- cyclic, preplanned, 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 standardised Log Book owned by the employee.

The performance standards are based on the BCS's Industry Structure Model (ISM). The ISM defines over 80 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 [13] and in the People CMM [5] also will be used to define the training program [4] for software engineers. A similar approach will be used for the other engineering disciplines.

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

As we are making progress in institutionalising 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 organisation 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 organisation as well as the performance management process will need some adjustments to capture the full benefits of these new work practices.

Conclusions

Our organisation has made substantial investments toward the definition, implementation, and integration 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 Reinvented Here" culture in which practitioners are constantly looking for practices to be pilot tested and integrated into the process asset library of the Oerlikon Aerospace.

References

- [1] M. Alexander, *The Encyclopedia of Team-Development Activities*, edited by J. William Pfeiffer, University Associates, San Diego, California, 1991.
- [2] V. Basili, S. Green, "Software Process Evolution at the SEL", *IEEE Software*, July 1994.
- [3] R. Bate, *A Systems Engineering Capability Maturity Model, Version 1.1*, Software Engineering Institute, CMU/SEI-95-01, November 1995.
- [4] M. B. Carpenter, H. K. Hallman, *Training Guidelines: Creating a Training Plan for a Software Organisation*, Software Engineering Institute, CMU/SEI-95-TR-007, September 1995.
- [5] B. Curtis, et al, *People Capability Maturity Model*, Software Engineering Institute, CMU/SEI-95-MM-02, September 1995.
- [6] M. K. Daskalantonakis, "Achieving Higher SEI Levels", *IEEE Software*, July 1994.
- [7] D. Dunaway, S. Master, *CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description*, Software Engineering Institute, CMU/SEI-96-TR-007, April 1996.
- [8] P. Fowler, S. Rifkin, "Software Engineering Process Group Guide", Software Engineering Institute, Report CMU-SEI-TR-24, September 1990.
- [9] T. Gilb, D. Graham, *Software Inspection*, Addison Wesley, 1993.
- [10] C. Y. Laporte, "Process Improvement and the Management of Change", *Proceedings: 4th IEEE Computer Society Workshop on Software Engineering Technology Transfer*, Dallas, April 28-29 1994.
- [11] B. McFeeley, *IDEAL: A User's Guide for Software Process Improvement*, Software Engineering Institute, CMU/SEI-96-HB-001, Feb. 1996.
- [12] T. G. Olson, et al., *A Software Process Framework for the Software Engineering Institute's Capability Maturity Model*, Software Engineering Institute, CMU/SEI-94-HB-01, Sept. 1994.
- [13] M. Paulk, et al, *Capability Maturity Model for Software*, Software Engineering Institute, SEI/CMU-93-TR-24, 1993.
- [14] P. R. Popick, S. A. Shear, "Ten lessons Learned from Implementing Integrated Product Teams", *Proceedings, International Council on Systems Engineering 6th Annual International Symposium*, Boston, July 7-11, 1996.
- [15] PMI, *A Guide to the Project Management Body of Knowledge*, Project Management Institute, 1996.
- [16] R. Radice, "A Programming Process Architecture", *IBM Systems Journal*, vol. 24, no. 2, 1985.
- [17] P. Scholtes, "The Team Handbook", Joiner Associates Inc., 1988.
- [18] D. R. Siddall, *The Team Notebook*, Publication in process, 1996.
- [19] SPC, *A Tailorable Process for Systems Engineering*, Software Productivity Consortium, SPC-94095-CMC, January 1995.
- [20] SPC, *Integrated Systems and Software Engineering Process*, Software Productivity Consortium, SPC-96001-CMC, May 1996.
- [21] SEI, "Relationships Between the Systems Engineering Maturity Model and Other Products, Version 1.0", Software Engineering Institute, CMU/SEI-94-TR-26, Nov. 1995.
- [22] STARS, "Army STARS Demonstration Project Experience Report", United States Air Force, Air Force Material Command, Feb. 1995.
- [23] J. A. Taylor, "Training, Career Development and Registration for Safety Critical Software Systems Specialists", *IEEE AES Systems Magazine*, September 1991.
- [24] USAF, "Integrated Computer-Aided Manufacturing Architecture", *Function Modeling Manual (IDEF0)*, United States Air Force, AFWAL-TR-81-4023, 1981.

Claude Laporte and Nicola Papiccio: Oerlikon Aerospace Inc., 225 boul. du Seminaire Sud, Saint-Jean-sur-Richelieu, Quebec, Canada, J3B 8E9. E-mail: c.laporte@oerlikon.ca.