The Application of a Systems Engineering Process to the Re-Engineering of an Air Defense System

Claude Y. Laporte, Alain Guay, Jacques Tousignant
Oerlikon Aerospace Inc.
225, boul. du Séminaire Sud
Saint-Jean-sur-Richelieu (Québec)
Canada J3B 8E9

ABSTRACT

During the last five years, our organization has developed and implemented software and systems engineering processes. In this paper, we briefly describe the systems engineering process development, its application to the re-engineering of two major components of an air defense missile system, and some lessons learned from the two projects.

PROCESS DEVELOPMENT BACKGROUND

Oerlikon Aerospace (OA) is the systems integrator of an air defense missile system. The missile system consists of a missile launcher mounted on a tracked vehicle, together with radar and optical sensors, electronic control systems and communication equipment. Over 100 systems and software engineers are involved in the development and maintenance of the system.

In fall 1992, recognizing that software engineering was a core competence, the president approved the establishment of a Software Engineering Process Group (SEPG) and the budget for a software capability assessment, as well as for the preparation of a Process Improvement Plan (PIP). In spring 1993, a formal software assessment was performed jointly by the SEPG and by independent assessors certified by the Software Engineering Institute (SEI). Strengths and weaknesses were identified and used as a basis for PIP development. During a second formal software process assessment conducted in February 1997, OA achieved a strong SEI level 2 certification, and even satisfied 8 of 17 goals for SEI level 3 certification.

Although the organization had been ISO 9001 certified since 1993, it was decided that a formal systems engineering process also had to be developed and implemented in order to seamlessly integrate disciplines associated with systems engineering, including software engineering. In 1995, an internal assessment of our systems engineering practices was performed, a decision was made to use, as frameworks, the Systems Engineering Capability Maturity Model (SE-CMM) and the Generic Systems Engineering Process (GSEP) developed by the Software Productivity Consortium (SPC 1995).

Our adaptation of GSEP (Laporte 1997) describes management and technical activities, roles and responsibilities, metrics and artifacts produced by each activity. OA Systems Engineering Process (SEP) major steps, sub-steps and activities are summarized in Table 1, for both systems engineering management activities (steps 110 through 150) and technical activities (steps 210 through 270). OA plans to perform an independent systems engineering assessment in 1998, to measure the progress made and plan for a second phase of systems engineering process improvement.

THE RE-ENGINEERING OF THE AIR DEFENSE SYSTEM

The re-engineering initiative targeted the two following subsystems: the launcher control electronics and the operator consoles. The launcher control subsystem is composed of a main data processor which coordinates the operation of the sensors and the launch and guidance of the missiles, a missile tracker processor, a target tracker processor, and a servo control processor. The operator consoles consist in a radar console, which allows controlling the radar and communication subsystems, and an electro-optical console, which allows controlling optical sensors and missile launcher.

Before using the SEP, both projects had to be divided in increments: a system definition phase of the subsystem in its new configuration, and a detailed hardware/software development phase. The identification of each increment is based on the nature

of the deliverable product at the end of the increment. In both cases, the first increment deliverable would be a system requirement specification, and the second increment deliverable would be a set of design and equipment specifications, plus a qualified working pre-production prototype. At this point in time, OA has completed the first increment on the console project, while the first increment on the launcher control project is still on-going. The results obtained so far are discussed below.

THE INSTANTIATION OF THE SEP MANAGEMENT ACTIVITIES

Step 110 Understand Context

In step 110, Understand Context, participants identify and review all relevant information that may influence the system definition. This step consists of three major sub-steps: 111 Define the approach, 112 Estimate the situation and 113 Review the context.

The 111 Define the Approach sub-step consists in four activities (Refer to Table 1). The first activity is Define the Increment Objectives. The objectives were defined as follows, for the launcher control project:

- Improve subsystem growth potential to allow for system performance improvements and addition of new requirements from current and potential customers.
- Define a new computer architecture that would reduce both production and life-cycle cost, minimize impacts on other subsystems, be ready for production at a pre-determined date, resolve obsolescence issues, allow graceful degradation, enhance technical performance parameters, and use maximum of standardized "commercial-off-the-shelf" hardware and software.

The first objective reflected the major problem of the current system, and the second objective was a design target, more oriented on how to perform the work. In the Consoles project, the focus was placed on acquisition cost reduction, since the current equipment was entirely subcontracted. The equipment was also affected by part obsolescence and lack of growth potential, as was the launcher control. It is important to define the main objective of the increment in very concrete terms, so the project team members can focus in the same direction when technology or performance trade-off analyses are performed.

The second activity of step 111 is Identify Project Constraints. A major constraint of the launcher control project was that the computer architecture of the new subsystem had to interface with existing components such as the missile, the electro-optical module and the power distribution unit. A major constraint of the console project was that the new human-computer interface (HCI) had to meet conflicting requirements and needs from different potential customers, and trade-off analyses would have to be performed to eliminate conflicts and still be appealing to users. In fact, the HCI is always the most visible part of a system to the customer, and can improve the "marketability" of a product. Both projects anticipated minimum resources, because funding external to the company was not yet identified.

The third activity, of step 111, is Identify Project Stakeholders, those people having a "vested" interest in the success of the project. They may be found both inside and outside the organization. Apart of the project team members, the identified stakeholders were the same for both projects:

- the current customer representatives: Project Management Office,
- the current end users: military unit, and in-house instructors,
- the marketing and business development people, and
- senior management.

It is important to properly identify stakeholders, so they can provide tangible support to project management and meaningful input during technical reviews. In both projects, different stakeholders were identified for management and technical aspects.

The fourth and last activity, of step 111, is Develop Project Alternatives. Depending on the nature of the development, the type of life-cycle model may be a critical choice. (e.g. waterfall, evolutionary or iterative). For instance, if both product function/performance and product technology are new to the organization, the development risk is high, and the life-cycle should be chosen to mitigate that risk: an iterative model would be used. In the launcher control project, it was decided to adopt the basic waterfall V-model (Forsberg 1996), since the current functionality would be kept, and only rehosted on new hardware (of known technology). It was also decided that a pilot project would be very useful to test the new SEP and to validate the engineering level of effort estimates. The team selected the missile tracker processor rehosting for the pilot project, because of its relative small size and simple interface, even though it is still critical to mission success. Also, alternative technologies had

to be subjected to trade-off analysis: COTS versus custom electronic components, VME versus other types of communication bus, and others. In the console project, the first alternative was to rehost only the current functionality with new hardware, and implement new requirements from potential customers in a later phase. This approach reduces project risks (cost and schedule), but augments technological risk from potential redesign of the architecture coming from new requirements in the next phase. The second alternative was to go ahead with a larger set of requirements at the start, including both the current and new functionality, to minimize the risk of major rework and insure growth potential. The second alternative was selected.

The second major sub-step of Understand Context is 112 Estimate of the situation. It consists in documenting all the data generated in Understand Context, assumptions, decisions and their rationale, to put together the current project knowledge.

The third major sub-step of Understand Context is 113 Review Context. It consists in reviewing the estimate of the situation with the stakeholders and obtaining their commitment on the adequacy of proposed assumptions and decisions. This sub-step is critical since it can be considered as a "Go-No go" decision point in the project. There is not yet a lot of resource spent, but there is enough information for the stakeholders to judge the pertinence of the project, agree on the strategy taken, and commit on resource allocation.

Step 120 Analyze Risk

During step 120 Analyze Risks, risks are analyzed, risk mitigation strategies are developed, and stakeholders commitment is made on mitigation strategy (Sub-steps 121 and 122). In both the launcher control and console projects, a Risk Management Plan (RMP) was developed. The RMP had two sections, "Risk descriptions and Impacts" and "Mitigation Strategies and Associated Risks". The first section identified and categorized risks: project risks such as budget overrun, schedule delays mostly due to lack of dedicated resources, and technical risks such as the lack of experienced personnel in using a new SEP and a new CASE tool (CORE®). Also, since the two projects were performed concurrently, it was necessary to closely monitor integration, validation and verification activities, and interfaces definition with the rest of the missile system. Finally specific risks like availability of COTS hardware, mastering of new technologies such as VME, development of new custom circuit card assembly (CCA), and development of new communication bus (e.g. Mil-Std-1553). The risks impacts were represented by a weighted probability of occurrence and consequence index. This risk matrix was stored in a database and was continuously updated during the two projects. The second section of the RMP associated and developed a mitigation strategy for each risk (Sub-step 123). The strategy included description, monitoring approach, schedule and cost impact, and required resources of the mitigation activity. In some cases, the same mitigation strategy addressed several risks. Mitigation strategies included activities such as pilot projects, engineering models and mock-ups, additional analyses, and components and subsystem modeling. Specific participant training was also planned in some areas. Finally, a formal review with stakeholders helped to identify other risks, gather mitigation suggestions, and obtain final commitment (Sub-step 124).

Step 130 Plan Increment Development

In step 130 Plan Increment Development, the Systems Engineering Management Plan (SEMP) (Sub-step 133) was developed, describing how the project would be performed in terms of process framework, methodologies and tools. It also provided the Organizational Breakdown Structure (OBS), the Work Breakdown Structure (WBS) and a detailed schedule. The SEMP detailed the activities for the first increment (i.e. the system definition) and provided an overview of the second increment (detailed hardware and software development, and manufacturing of the prototype). Sub-step 131, Execute risk aversion, was performed on a continuous on-going basis and was not a pre-requisite to develop the SEMP. Sub-step 132, Review Development Alternatives, was actually performed while documenting the SEMP itself. Developing the SEMP actually required a deep insight in both managerial and technical issues and details, since the development approach is closely affected by the nature of the product to be developed, the maturity of existing technologies, the experience and expertise of developers, and the dynamics of effective development teams.

The launcher control project SEMP divided Increment 1 in two main activities: reverse and forward engineering. For each current CCA, the reverse engineering activities documented low-level and high-level functional and interface models and requirements. The goal of reverse engineering was to avoid misconceptions, gray areas, and erroneous or missing information when migrating from the old to

the new functional architecture. Since the new configuration included additional requirements, such as new system threats, forward systems engineering activities were to be performed, in accordance with SEP steps 210, 220 and 230. With all functions properly defined (legacy system and new requirements), functional allocation could be performed to new hardware (Step 240). Alternative technologies would have to be studied, leading to trade-off analyses defining the proposed final launcher control system definition (step 250). Increment 2 would iterate with the same process through the definition of subsystems and main components. (Increment 1 was more a waterfall process)

The console project SEMP also divided Increment 1 in two main and parallel activities: definition of the requirements and technology search. The main function of the console being the operator interface with the system, the operational scenarios derived from the mission were the most direct source for requirement definition (for both tactical and non-tactical deployment). Technology search on displays, pointing devices, processors and accessories for HCI did not require complete system definition to be started. Increment 1 would be completed by a preliminary physical architecture definition, with alternative interfaces using the available technologies. Increment 2 would include as part of the detailed design construction of functional models to compare viable MMI alternatives with operators.

In both projects, monitoring of Technical Performance Measures (TPMs), requirement management approach, integration of specialty engineering disciplines, training plan, configuration management, quality assurance, progress and technical reviews were also relevant sections of the SEMP. As usual, a review with stakeholders was performed to give a go-ahead to the plan.

Steps 140 Track Increment Development and 150 Perform Increment Closure

The last two steps in Manage Development Effort, 140 Track Increment Development and 150 Perform Increment Closure, were performed differently in the two projects. Budgets and resources for the launcher control development were severely reduced (sub-step 134), so that the scope would be restricted to the rehosting of the missile tracker processor. This project is currently at the beginning of its system definition phase. A go-ahead was given to the console development project (sub-step 134), mainly because the cost benefit of the project was found so important that the payback would be considerable, even with the worst business option considered.

THE INSTANTIATION OF THE SEP TECHNICAL ACTIVITIES

Step 210, Analyze needs, was found very important for the console project. The legacy system was known from everybody within the company and by the current Customer, the Canadian Forces, so that comments, suggestions for improvement and deficiencies were identified and provided from multiple sources. In addition, potential customers' requirements were also considered. There was a need to compile all that information, classify all items as essential, highly desirable or desirable, eliminate redundancies, resolve conflicting statements, in view of a formal review with identified stakeholders. That review was held together with the risk management plan review (sub-step 124), so all technical and managerial aspects could be considered together.

The "Problems, Needs and Constraints" document (sub-step 212) resulting from that review was then used, together with the existing system specifications, to generate the first set of requirements for the new console project (sub-step 221). Some difficulties were experienced in defining the best way to document the requirements for the new system. It was found after a few trial and error iterations that the requirements could be better organized after the concept of the new system is developed, at least at a preliminary stage. It meant that steps 220 Define Requirements, 230 Define Functional Architecture and 240 Synthesize Allocated Architecture were actually performed more or less in parallel, a little of each at the time, each progressing in turn from the progress made in the other. (Iterative life cycle model). For that purpose, an engineering model of the HCI was developed on Visual Basics ® (following a fast prototyping approach), allowing early review of the expected behavior for the new system by all stakeholders. It was also beneficial, to define system requirements, to have the project participants working as a team, each expertise present on a continuous basis. The operational, hardware and software aspects of the system were then equally considered in the formulation of the requirements, taking advantage of the data gathered in parallel by the technology search team. The work was performed by reformulating the actual text of the requirements, until the group agreed on each one. The resulting requirements were then documented in a CORE® database, together with verification requirements (to

insure that each defined requirement is verifiable) (Step 260) and requirements issues (to insure a follow-up on missing information from other projects or subsystems). This effort led to a preliminary definition of the system behavior, its operational concept, preliminary physical configuration and software architecture, all required to properly estimate Increment 2 of the project, Detailed hardware and software development.

LESSONS LEARNED

It was very important to carefully select pilot projects and participants to the pilots since these projects would foster adoption of new practices throughout the organization. Also, first time users of a new process would make mistakes; it was therefore mandatory to properly coach the participants. If participants sensed that mistakes would be used to learn and make improvements to the process instead of "pointing fingers", the level of anxiety would be reduced and they would bring forward suggestions instead of "hiding" mistakes. Most of the participants for both projects were therefore selected within the SEP development working group, and the other participants were given a two day training session on the draft SEP.

Both projects were planned using the SEP steps as WBS elements. It was found that for some areas of the process, specific deliverables are difficult to determine precisely, because the end product of the complete SEP iteratively grows as steps are performed. It is therefore difficult to closely monitor the progress of the activities and report progress to management. Another dimension of this situation is the definition itself of the increment on which the SEP applies. The project increments must be carefully defined so that they are not too big, and their activities too long to be properly tracked, not too small, so their activities require micro-management to be tracked. Project manager experience was found a critical asset for project and increment definition. A manageable increment size is also critical for the proper performance of design reviews, in that participants to the review keep focus on the increment scope. It was found that SEP Series 100 steps were mostly performed in sequence, as per a waterfall model, while SEP Series 200 steps were mostly performed through multiple iterations, until the overall result is satisfactory.

Finally, even with a formal development process, managing the human dimension of the process engineering initiative was found the component which not only fosters the adoption of change but also creates an environment where changes can 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".

CONCLUSION

A new Systems Engineering Process was developed at OA and implemented on two pilot projects, the rehosting of a missile system launcher control subsystem, and the redesign of a missile system operator console. The process was found very useful to plan activities and collect technical and managerial information more formally in the course of the projects. The formal process did not eliminate the need for experienced managers and competent personnel with the proper expertise, but it did help to manage and improve the dynamic human dimension of the development projects.

REFERENCES

Forsberg, K., Mooz, H.., "Application of the 'Vee' to Incremental and Evolutionary Development", Proceedings of the Symposium of the International Council on Systems Engineering, St.Louis, MO, July 1995.

Laporte, C.Y., Papiccio, N.R., "Development and Integration of Engineering Processes at Oerlikon Aerospace", Proceedings of the Seventh International Symposium of the INCOSE, August 3-7, 1997, Los Angeles, California.

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

BIOGRAPHIES

Claude Y. Laporte obtained in 1973 a Bachelor in Science from le Collège Militaire Royal de Saint-Jean. In 1980, he obtained an MS in physics at Université de Montréal, and in 1986, an 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. 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 processes, methods and tools.
Alain Guay obtained a Bachelor in Mechanical Engineering and a Certificate in Administration from Université Laval respectively in 1986 and 1987. He is now completing a M.B.A. at Université de Montréal. He has been working for Oerlikon Aerospace for the last 10 years. He was first employed as a design engineer. During the last 6 years he lead the design and integration group. He is now working as a project engineer and is involved in the development and implementation of system engineering process, methods and tools.

Jacques Tousignant is a system engineer specialized in ergonomics. He obtained his physical engineering degree from Laval University in Quebec, in 1977, and a diploma in Ergonomics at École Polytechnique de Montréal in 1994. He worked for nine years with Bombardier Inc., in the design, development and test of mass transit vehicles, and for the last eleven years with Oerlikon Aerospace Inc. in the design, planning and performance of engineering tasks on air defense systems. He was involved in managing a human engineering and safety program, in the design and development of training simulators, in the management of systems engineering tasks related to product support, and particularly in the last year in the development of the new operator console for the AD system. He is also involved in the development of engineering procedures required for accreditation to ISO 9001 requirements, and the improvement of systems and software engineering processes.

Table 1 The Systems Engineering Process at OA

Major Steps	Sub-Steps	Activities
110 Understand Context	111 Define Approach	Define key objectives
		Identify constraints
		Identify stakeholders
		Develop alternatives
	112 Estimate of Situation	Create/update Estimate of the Situation (EoS)
	113 Review Context	Review and validate the EoS with stakeholders
		Obtain stakeholders commitment
120 Analyze Risk	121 Perform Risk Analysis	Identify potential risks
		Identify potential loss and consequences
		Analyze risks dependencies
		Identify risks probability of occurrence
		Prioritize risks
		Identify risk aversion strategies for each risk
	122 Review Risk Analysis	Review risk analysis
		Identify risks to be part of the Risk Management Plan (RMP)
	123 Plan Risk Aversion	Define a risk monitoring approach
		Estimate risk aversion strategy cost and schedule
		Recommend risk aversion strategies
	124 Commit to Strategy	Obtain stakeholders commitment
130 Plan Increment Development	131 Execute Risk Aversion	Execute risk aversion strategies
		Evaluate the impacts and results
		Update RMP accordingly
	132 Review Development Alternatives	Review risk aversion results with stakeholders
		Select preferred development strategy
		Obtain stakeholders commitment
	133 Plan Increment Development	Create the Systems Engineering Management Plan (SEMP)
	134 Commit to Plan	Review the SEMP with stakeholders
		Update SEMP as required
		Obtain stakeholders commitment
		Brief senior management on the SEMP

Major Steps	Sub-Steps	Activities
		Obtain and assign work packages to start activities
140 Track Increment Development	141 Monitor and Review Increment Development	Capture and analyze increment status
		Produce management metrics
		Perform regular project reviews with stakeholders
	142 Update Increment Plan	Update SEMP as required
	143 Review Technical Product	Perform design reviews
		Document reviews with minutes of meeting and action item lists
150 Perform Increment Closure	151 Baseline System Definition	Baseline work products configuration
		Track changes to the work products
		Store work products
	152 Assess Increment Closure	Evaluate technical success of the increment
		Evaluate project success of the increment
		Evaluate increment against external system plan
		Compile lessons learned
		Log all data in Configuration Management (CM) database
	153 Update External System Plan	Review metrics and lessons learned and feed external system plan
		Feed metrics and lessons learned to SEP process owner
		File lessons learned in the process asset library
	154 Commit to Proceed	Review the external system plan with the stakeholders
		Update external system plan as required
		Obtain stakeholders commitment
		Document commitments
210 Analyze Needs	211 Determine Stakeholders	Identify stakeholders
		Identify all participants
		Develop strategy to obtain stakeholders expectations
	212 Define Problem Domain	Identify and classify user and customer concerns: problems and needs
		Identify customer/developer system constraints
		Analyze system operational scenarios
		Define system Technical Performance Measures (TPMs)
		Determine system environment
	213 Develop Informal Functionality	Decompose user needs into informal high-level functions in engineering terms
		Develop TPMs
220 Define Requirements	221 Determine Behavioral Requirements	Generate behavioral requirements (system functions)
	222 Determine Performance Requirements	Determine quantitative testable performance characteristics to meet user needs
		Refine and augment TPMs
	223 Map Behavior to Performance	Map behavior/functions to performance (how well functions are performed)
	224 Refine Requirements	Identify derived requirements
		Make sure all requirements are verifiable
		Refine derived requirements
		Assess requirements against system TPMs
230 Define Functional Architecture	231 Partition Requirements into Functions	Partition horizontally the system requirements into a set of complementary functions
		Identify further derived requirements
	232 Define Lower Level	Decompose top level functions into lower level functions

Major Steps	Sub-Steps	Activities
	Functions	Define the functional architecture - Functional Flow Block Diagram (FFBD)
		Identify further derived requirements
	233 Define Functional Interfaces	Define information between the system and the environment (context diagram)
		Define input and output information to each function
		Specify internal and external interfaces behavior
240 Synthesize Allocated Architecture	241 Allocate Functions to Alternative Solutions	Identify candidate mechanisms to carry functions
		Define alternate physical architectures
		Partition the mechanisms to define their interactions
	242 Define Physical Parameters	Define algorithms and parameters
		Define specific characteristics of physical entities
	243 Define Physical Interfaces	Identify physical interfaces
		Define interactions between mechanisms
	244 Integrate Design	Integrate system design from lower levels up
		Analyze impact at top level
		Feed back impact to lower level designs
	245 Refine Physical Architecture	Refine each solution characteristics
		Identify derived functions, like failure mechanisms
250 Evaluate Alternatives	251 Assess System	Build system models
		Evaluate system by engineering specialties
	252 Perform Sensitivity Analysis	Analyze system sensitivity to varying parameters and environments
	253 Allocate Performance to Technical Parameters	Define technical parameters for subsystems and components
		Optimize alternate architectures
		Document trade-off analyses
	254 Assess Technical Risks and Problems	Identify risks and problems in each alternative
		Evaluate risks impacts
	255 Identify and Perform Trade-offs	Identify viable system alternatives
		Assess against risks, problems and TPMs
	256 Select best System Solution	Evaluate system trade-off results
		Select the preferred solution
260 Verify and Validate Work Products	261 Define V&V Procedures	Develop clear V&V approach and procedures
		Verify procedures against quality measures
	262 Verify System	Verify all characteristics are verified
		Document variations with expectations
		Analyze variations for causes
	263 Validate System	Verify work products address needs
		Document variations with expectations
		Analyze variations for causes
270 Release System Definition	271 Control Technical Decision Data	Maintain and control critical information pertaining to system definition decisions
	272 Control System Configuration	Review and evaluate system changes
		Document changes from previous releases
		Maintain configuration control on released system definitions