Risk Management Applied to the Reengineering of a Weapon System

Claude Y. Laporte
École de Technologie Supérieure

Guy Boucher
Oerlikon Contraves Inc.

In this article, a systems engineering process is briefly described followed by a discussion of the application of risk management practices to the reengineering of operator console stations of a missile weapon system. Lastly, 12 lessons learned are presented.

Oerlikon Contraves Inc. is a systems integrator specializing in the design, assembly, integration, testing, and delivery of complete systems solutions, including an air defense missile system. The system consists of a missile launcher mounted on a tracked vehicle or a fixed platform, together with radar and optical sensors, electronic control systems, and communication equipment.

The organization has been ISO 9001 certified since 1993. In 1997, it was also assessed at the Software Engineering Institute's (SEI) Capability Maturity Model® (CMM®) Level 2 [1] by independent assessors certified by the SEI. In addition to satisfying Level 2 goals, the organization also met eight of the 17 Level 3 goals.

In 1995, it was decided that a formal, systems engineering process had to be developed and implemented in order to seamlessly integrate disciplines associated with systems engineering. The development effort was initiated by performing an internal assessment of the organization's systems engineering practices. A decision was made to use as frameworks the CMM for Systems Engineering and the Generic Systems Engineering Process developed by the Software Productivity Consortium [2].

The systems engineering process (SEP) [3] describes management and technical activities, roles and responsibilities, and metrics and artifacts produced by each activity. The management activities of the SEP's major steps are summarized in Table 1 while Table 2 illustrates the technical activities (steps 210 through 270). The process had been applied to the reengineering of two subsystems: the launcher control electronics and the radar and electro-optical operator consoles [4].

"Dealing with formal risk management represented a mentality change not only for the project team but also for the entire organization."

Table 1: *The Management Activities of the Systems Engineering Process*

Major Steps	Substeps
110 Understand Context	111 Define Approach
	112 Estimate Situation
	113 Review Context
120 Analyze Risk	121 Perform Risk Analysis
	122 Review Risk Analysis
	123 Plan Risk Aversion
	124 Commit to Strategy
130 Plan Increment Development	131 Execute Risk Aversion
	132 Review Development Alternatives
	133 Plan Increment Development
	134 Commit to Plan
140 Track Increment Development	141 Monitor and Review Increment Development
	142 Update Increment Plan
	143 Review Technical Product
150 Perform Increment Closure	151 Baseline System Definition
	152 Assess Increment Closure
	153 Update External System Plan
	154 Commit to Proceed

The launcher control subsystem is composed of a main data processor that 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 of a radar and communication subsystems, and of an electro-optical console to control both the optical sensors and the missile launcher.

The Reengineering of Operator Consoles

The reengineering of the consoles was divided into two major increments: a system definition increment of the subsystem in its new configuration and a detailed hardware/software development increment, which was further broken down into several sub-increments. The identification of each increment was based on the nature of the deliverable products at the end of the increment. In both cases, the first increment deliverable would be a system requirement specification, and the second increment deliverables would be a set of design and equipment specifications plus a qualified working preproduction prototype.

The following paragraphs describe what was accomplished during increment one as well as what is being planned and performed for increment two. The emphasis of this article will be put on the risk activities that have been performed.

Overview of Increment One

Requirements Management

The system engineering CASE Tool CORE has been used to develop the console requirements. The database included the following types of information:

- Originating requirements (behavioral and nonbehavioral).
- Interface requirements.
- Verification requirements.
- Physical architectures.
- System diagrams.

The CORE database was baselined

after the completion of increment one.

Developing an Engineering Model

An engineering model was developed during increment one. The model ran on a standard PC, and its purpose was to show the new concept of operation and the proposed man-machine interface (MMI). The model was formally shown to stakeholders. Comments were collected and analyzed to modify and improve the system requirements in a second iteration.

Technology Search

A series of technologies related to either hardware or software has been researched and trade-off analyses have subsequently been documented. Many potential suppliers were met and a few employees attended real-time embedded conferences as well as virtual machine environment (VME) and high tech shows.

Training

Beside the training provided on the new systems engineering process, the only formal training provided had been on tools: the graphical user interface (MMI) CASE tool, and CORE, the system definition CASE tool. Training was also later performed on VxWorks operating system, Rhapsody software development CASE tool, and unified modeling language software development methodology.

Overview of Increment Two

The plan for increment two consisted of proceeding with both the hardware and software detail design based on the interim system definition and the engineering model generated during increment one. The detailed development will include the construction of an engineering unit to support the hardware and software development and the construction of a preproduction unit that will support system integration and qualification activities. In addition, simulators will be built in parallel to support development, integration, and validation efforts.

The plan for increment two also included other nonrecurring activities such as the production jigs, tooling and logistic activities, technical publications, and training.

The Application of Risk Management Activities

SEP Step 120: Analyze Risk

In SEP step 120, risks were analyzed, risk mitigation strategies were developed, and stakeholders' commitment was made on mitigation strategies (Substeps 121 and 122). The high-level process, as illustrated in Figure 1 and Table 3 (see page 26), describes what risk management activities should be performed, but it does not prescribe any particular method. The members of the project were aware of the method used by software engineers since a method was described in the project planning and tracking activities of the company software engineering process. After a brief discussion, the team decided to use the method proposed by the U.S. Air Force [5]. At the beginning of the project, it was felt that this step looked like a paper exercise and was not very useful. However, it was the first development project to proceed with a formal method to handle risks.

Major Steps	Substeps
210 Analyze Needs	211 Determine Stakeholders
	212 Define Problem Domain Assess Problem Needs and Constraints
	213 Define Environment
	214 Develop Informal Functionality
220 Define Requirements	221 Determine Behavioral Requirements
	222 Determine Performance Requirements
	223 Map Behavior to Performance
	224 Refine Requirements
230 Define Functional Architecture	231 Partition Requirements into Functions
	232 Define Lower Level Functions
	233 Define Functional Interfaces
240 Synthesize Allocated Architecture	241 Allocate Functions to Alternative Solutions
	242 Define Physical Parameters
	243 Define Physical Interfaces
	244 Integrate Design
	245 Refine Physical Architecture
250 Evaluate Alternatives	251 Assess System
	252 Perform Sensitivity Analysis
	253 Allocate Performance to Technical Parameters
	254 Assess Technical Risks and Problems
	255 Identify and Perform Trade-off
	256 Select Best System Solution
260 Verify and Validate Work Products	261 Define Verification and Validation Procedures
	262 Validate System
	263 Verify System
270 Release System Definition	271 Control Technical Decision Data
	272 Control System Configuration

Table 2: *The Technical Activities of the Systems Engineering Process*

A risk management plan (RMP) was developed containing two main sections. The first section described the program overview and defined terms based on the following:

- Type of risk (cost, program, schedule, technical, and supportability).
- Assessment of risk impact (catastrophic, critical, marginal, and negligible).
- Overall categorization of risk (high, moderate, and low).

The RMP specified who was responsible for the risk management and how the risks were to be managed during the increment. This section was quite generic; it could be reused by other projects.

The second section of the RMP was specific to the project. It was mainly composed of a single matrix that listed all of the identified risks. The risk identification process was performed through brainstorming sessions with both the development team members and stakeholders. Along with the list of risks in the same matrix was the following information:

- Type of risk (cost, program, schedule, or technical).
- Probability of occurrence (very low, low, medium, high, or very high).
- Impact (catastrophic, critical, marginal, negligible, and cost).
- Overall risk (high, medium, low, and cost).
- Identification of impact on other projects.
- Brief resolution plan.
- Drop-dead date.
- Person(s) responsible (member of the project team, functional manager, project manager, or engineering director).
- Hours or resources required performing the project.
- Resolution status (open or close).

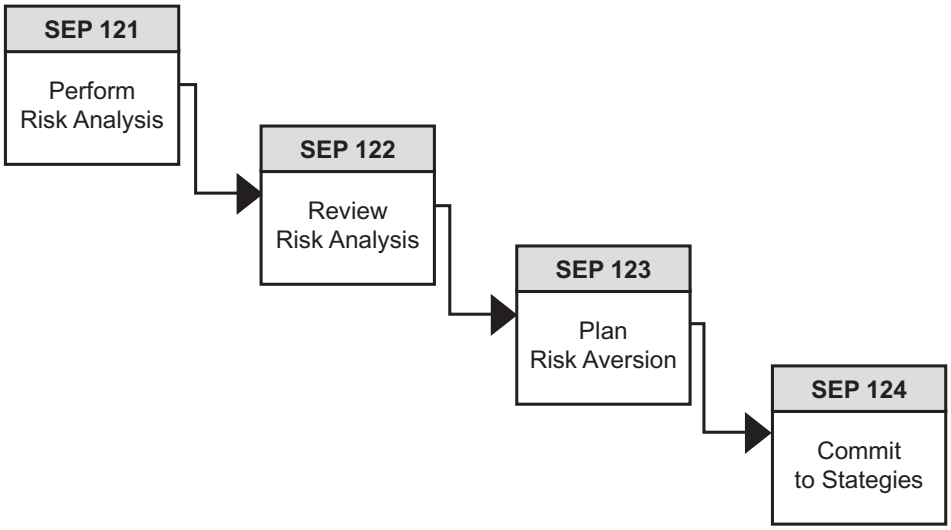

Figure 1: *Risk Management Activities*

The Implementation of the Risk Management Plan

The actions and status of the risks were then reviewed on a weekly basis during project reviews. When a mitigation plan was required, e.g., special resources and a considerable amount of hours, then specific risk activities were directly integrated in the detailed work breakdown structure and scheduled like any other major development items.

Some of the risks identified were as follows: 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 process, and a new CASE tool (CORE). Also, since this project was performed concurrently with another project, 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 commercial off-the-shelf hardware, mastering of new technologies such as VME, development of a new custom circuit card assembly, and development of new communication bus were also identified.

Risk 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 increments.

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 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 (substep 124).

Lessons Learned

Quantification of Risks Issues

During increment one, risk only had a *qualitative* score, i.e., high, medium, or low. We found that this had two major drawbacks compared to quantitative evaluations:

- It did not have the same weight or necessary attention from management.
- No money/resources were set aside should the risk issue have occurred. This could lead to budget overruns.

Evaluation of Risks in a Systemic Perspective

For increment two, we quantified and costed all risks, even the ones that the team had no control over such as hiring or allocating budgets and expenses. The development team would ultimately be impacted should a risk occur. As a result, the company decided to put money aside for risks in the budget for increment two.

Risk Management Is Not Free, But It Is a Wise Investment

We quickly found out that some risks required a lot of effort to mitigate. One example was the activity related to the engineering model in increment one. It was decided to proceed with an engineering model to mitigate a risk previously identified that related to the fact that we had no customer requirement. The fear was then that we would proceed with a design that would not meet any potential customer wishes.

Approximately 800 hours were spent to model a new concept of operation and an MMI. This included activities such as model design and, even more important, validation of the concepts with a selected group of operators from inside and outside the company.

This model allowed us to develop and refine the system requirements as well as define software use cases with a very high confidence level that they would remain stable throughout the entire design chain. Although it was difficult to precisely assess the amount of time/money that has been and will ultimately be saved, one can imagine what would be the cost of delivering a product that would not meet customer expectations.

Another example is a pilot project performed in increment two. This pilot project came as a result of a risk identified that expressed the concerns that we would enter the software design phase with a new methodology, new CASE tools (design and GUI), and a new development environment. About 1,000 hours were spent on a mini-project that had the main objectives to verify the capabilities of the tools, to verify the integration of the tools, and to propose a design method.

The results and conclusions obtained through the development of this pilot project were crucial to generating a proper software development plan that needs

Table 3: *The Risk Activities of the Systems Engineering Process*

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

to clearly show, organize, and plan the work of a group of more than 20 persons for a 24-month time frame. The pilot project represented about 1.5 percent of the total software design effort, but it was sure worthwhile since it ensured that the remaining 98.5 percent of the project would be done properly and correctly.

Pilot Projects As a Risk Mitigation Strategy

It was very important to carefully select pilot projects and their participants 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 was reduced. This also led individuals to bring forward suggestions instead of *hiding* mistakes. Most of the participants for both projects were therefore selected within the working group who developed the SEP. Other participants were given a two-day training session on the SEP.

Management's Response to Risks

Dealing with formal risk management represented a mentality change not only for the project team but also for the entire organization. Yet, when risk management activities were done properly by the development team, management was more prone to agree and support the risk activities that resulted from the risk analysis.

Risk Mitigation Leads to Design Decisions, Development Strategies

The results of the risk mitigation activities related to technical risks will necessarily lead to, or as a minimum be an input to, design decisions and will provide direction for follow-on activities. In fact, whether a mitigation plan arose from generating an analysis, conducting a test, or constructing a physical or behavioral model, the result will be the confirmation of a hypothesis or the identification of the best design alternative. Ultimately, this leads to design decisions and subsequent development strategies.

Training as a Risk Management Issue

One important aspect of risk management was training. Previously, most plans showed a nice flow-down of activities with associated efforts, as it should be. However, these plans also reflected the fact that they were all conducted by highly skilled personnel that knew exactly what to do at all times. This obviously did not represent reality. Therefore, appropriate training became mandatory to manage the risks, and training activities were built into the project plan.

Dividing a Project Into Increments As a Risk Management Strategy

Project increments must be carefully defined so that they remain manageable. Their associated activities were not too long to be properly tracked, and on the other end were not too small so that their activities required micro-management. Project manager experience was found to be a critical asset for project and increment definition. A manageable increment size was also critical for the proper performance of design reviews; in those reviews, participants kept their focus on the increment scope.

"It was very important to carefully select pilot projects and their participants since these projects would foster adoption of new practices throughout the organization."

New Process Implementation Risks

It was found that for some areas of the SEP, specific deliverables were difficult to determine precisely. This situation happened because the end products (i.e., project documents) grew iteratively as process steps were performed. It was therefore difficult to closely measure the progress of the activities and report progress to management. As a result, *lessons learned* were generated, and this led to the development of a specific set of methods/instructions to support the project manager and his team by providing better definitions and tracking/reporting methodologies. The lessons learned and the various instructions have been distributed and electronic copies are available on the company intranet.

Risk Associated With People Issues

Managing the human dimension of the project was found to be an element that not only fostered the adoption of the new process, but also created an environment where changes were introduced at an increasingly greater rate. Members of the engineering organization realized that managing the *soft stuff* was as important as managing the *hard stuff*. Additional information about managing people issues can be found in a previous CrossTalk [6].

Risk Management Activities Are Planned and Included in the Project Plan

Since a substantial amount of energy was expended in *risk management* activities, those activities were identified, estimated, and incorporated in the project plan. It is important to note that risk management is part of the standard company work breakdown structure and a level of effort is estimated and planned accordingly. In addition, as the major risk mitigation strategies become part of the system plan to be approved by the organization, commitment is established. The costs associated to that risk effort and associated mitigation strategies are then tracked as any other work breakdown structure activity.

Appointing a Project Risk Officer

When a project is composed of many projects similar to the one described in this article, all risk activities may represent a substantial effort. Also, the risks have to be analyzed at the project level since risks in one subproject may create risks at the project level. Risks from different subprojects may be analyzed and mitigated at the project level instead of being mitigated individually. It was found that all project risk activities were better managed by one individual. A project role called risk officer had been established. The risk officer hat was allocated, as a secondary duty, to a member of the team who was interested by this role and had a lower load in the project.

Conclusion

A new SEP involving managing risks had been deployed and used in the re-design of a missile system operator console. The risk management activities were found to be very useful to plan activities and collect technical and managerial information more formally in the course of the projects. It also helped manage and improve the dynamic human dimension of the development project.◆

References

1. Paulk, M. et al. Capability Maturity Model for Software. Pittsburgh: Software Engineering Institute, 1993.
2. Software Productivity Consortium. A Tailorable Process for Systems Engi-

neering. Software Productivity Consortium, Jan. 1995.
3. Laporte, C. Y., and N. R. Papiccio. Development and Integration of Engineering Processes at Oerlikon Aerospace. Proc. of the Seventh International Symposium of the INCOSE. Los Angeles, CA, 1993.
4. Laporte, C. Y., A. Guay, and J. Tousignant. The Application of a Systems Engineering Process to the Reengineering of an Air Defense System. Proc. of the Eighth Annual International Symposium of the INCOSE. Vancouver, British Columbia, Canada, 26-30 July 1998.
5. U.S. Air Force. USAF's Software Risk Abatement Handbook. AFSC/AFLC Pamphlet 800-45, 30 Sept. 1988.
6. Laporte, C. Y., and S. Trudel. "Addressing the People Issues when Developing and Implementing Engineering Processes." CrossTalk Nov. 1999.

Additional Reading

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

About the Authors

Claude Y. Laporte is a software engineering professor at the École de Technologie Supérieure in Montreal. He was an officer in the Canadian Forces and retired at the rank of major. He joined Oerlikon Contraves Inc. in 1992, then called Oerlikon Aerospace, where he coordinated the development and implementation of engineering and management processes. Laporte has a bachelor's degree in science from the Canadian Military College of Saint-Jean, a master's of science degree in physics from the Université de Montreal, and a master's degree in applied sciences from the Department of Electrical and Computer Engineering at École Polytechnique de Montreal.

École de Technologie Supérieure
1100 Notre-Dame Ouest
Montreal, Quebec
Canada, H3C 1K3
E-mail: claporte@ele.etsmtl.ca

Guy Boucher is a project manager on various projects at Oerlikon Contraves Inc. He was formerly radar principal engineer at the company. Boucher served the Canadian Forces for a period of five years as a radar engineer on a long-range radar station and as a Canadian representative on a joint development program of the U.S. Over-the-Horizon-Backscatter Radar in Bangor, Maine. He left the Forces in 1987 at the rank of captain. Boucher has a bachelor's degree in electrical engineering from the Royal Military College of Canada.

Oerlikon Contraves Inc.
225, boul. du Séminaire Sud
Saint-Jean-sur-Richelieu, Quebec
Canada, J3B 8E9
E-mail: gboucher@oerlikon.ca

WEB SITES

Software Technology Support Center

www.stsc.hill.af.mil

The Software Technology Support Center is an Air Force organization established to help other U.S. government organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and to accurately predict the cost and schedule of their delivery.

Risk Management

www.acq.osd.mil/io/se/risk_ management/ index.htm

This is the Department of Defense (DoD) risk management Web site, a working group composed of representatives from the services and other DoD agencies involved in systems acquisition to assist in the evaluation of the department's approach to risk management. The working group will continue to provide a forum for sharing experiences and knowledge in order to provide program managers with the latest tools and advice on managing risk.

Software Program Managers Network

www.spmn.com

The Software Program Managers Network (SPMN) is sponsored by the deputy under secretary of defense for Science and Technology, Software Intensive Systems Directorate. It seeks out proven industry and government software best practices and conveys them to managers of large-scale DoD software-intensive acquisition programs. SPMN provides consulting, on-site program assessments, project risk assessments, software tools, guidebooks, and specialized hands-on training.

Software Engineering Institute

www.sei.cmu.edu

The Software Engineering Institute (SEI) features information on "Building High Performance Teams Using Team Software Process℠ (TSP℠) and Personal Software Process℠ (PSP℠)." The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense The SEI's core purpose is to help others make measured improvements in their software engineering capabilities.

The Software Productivity Consortium

www.software.org

The Software Productivity Consortium is a nonprofit partnership of industry, government, and academia. It develops processes, methods, tools, and supporting services to help members and affiliates build high-quality, component-based systems, and continuously advance their systems and software engineering maturity pursuant to the guidelines of all of the major process and quality frameworks. Based on the members' collective needs, its Technical Program builds on current best practices and information technologies to create project-ready processes, methods, training, tools, and supporting services for systems and software development.