Risk Assessment

Risk Assessment Template for Software Development or Acquisition Projects

The role of Risk Assessment and Risk Management is to continuously Identify, Analyze, Plan, Track, Control, and Communicate the risks associated with a project.

The Webster's definition of risk is *the possibility of suffering a loss*. Risk in itself is not bad. Risk is essential to progress and failure is often a key part of learning. Managing risk is a key part of success.

This document describes the foundations for conducting a *risk assessment* of a large-scale system development project. Such a project will likely include the procurement of Commercial Off The Shelf (COTS) products as well as their integration with legacy systems.

Niwot Ridge Consulting
4347 Pebble Beach Drive
Niwot Colorado 80503
www.niwotridge.com

Authors, Approvers, and Reviewers

Revision A: 17/March/1998 .....GB Alleman
- ✓ Original Publication

Revision B: 28 April, 1998.....GB Alleman
- ✓ Final release

Revision C: 5 May, 1998 .....GB Alleman
- ✓ The real Final Release

Revision D: 20 February, 2001 .....GB Alleman
- ✓ Thoroughly updated with new format, expunged content names, and bibliography

Distribution List

Confidentiality Notice

This document contains information provided under the terms described in the agreement between Niwot Ridge Consutling and *Company Name*. In addition, this information is provided to *the Company* for their sole use and is not to be shared with any vendor mentioned in this report, consulting firm or other outside party, without the expressed written permission of Niwot Ridge Consulting, 4347 Pebble Beach Drive, Suite 104, Niwot Colorado 80503.

The beneficial result that I can hope for as a consequence of this work is that more attention will be paid to the precise statement of the alternatives involved in the questions being asked.

– *Theory of Probability*, H. Jeffery, Cambridge University Press, 1939, p. vi.

Table of Contents

Table of Figures

Overview

Running away from risk results in a no–win strategy for all participants. The Business Unit organization cannot avoid the risks associated with the design, deployment and operation of a major software development or deployment project. Moving aggressively after a business opportunity means running *toward* risk, rather than *away* from risk.

However, running successfully toward risk requires more than just a competent process and an ability to think on your feet. The management of risk requires the deployment of the discipline of *Risk Management*.

> *I am used to thinking three or four months in advance, about what I must do, and I calculate on the worst. If I take so many precautions, it is because it is my custom to leave nothing to chance.*
>
> *– Napoleon I, in a conversation with Marshall Muart, March 14, 1808.*

The Concept of Risk

Current definitions of risk, as a noun, include:

- The possibility of suffering, harm or loss – danger
- A factor, element, or course involving uncertain danger – hazard
- The danger or probability of loss to an insurer
- The amount an insurance a company stands to lose
- A person or thing considered with respect to the possibility of loss to an insurer – a poor risk

In operations research, Risk is a more general term. The concept of decision under risk describes a situation where there is probability associated with an outcome or choice, regardless of the nature of outcome. For the most part the term is used as reflected in the following:

- The possibility of loss, injury, disadvantage, or destruction.
- Someone or something that creates or suggests a hazard or adverse chance
- The chance of loss or the perils to the subject matter of insurance covered by a contract.

In the context of software engineering and development, risk can be defined as the possibility of suffering a diminished level of success (loss) within a software–dependent development program. This prospect of loss is such that the application of the selected theories, principles, or techniques may fail to yield the right software products. [1]

The potential loss to the software program and specifically the association of risk with the program involves a value judgment on the potential impact of risks to the successful outcome. The term loss, danger, hazard, and harm, all of which reflect a negative perception, involve at least a relative assessment of value. [2]

[1] "The SEI Approach to Managing Software Technical Risks," *Bridge*, October 1992, pp. 19–21.
[2] *Anatomy of Risk*, W. D. Rowe, Roger E. Krieger, Malabar, FL, 1988.

Many attributes of a program can be used to characterize value in the context of software–dependent development programs.

Some examples are:

- Customer satisfaction
- Software execution speed
- Software code size
- Data of delivery
- Number of software defects
- User friendliness

It is clear from these definitions of risk that uncertainty expressed as possibility of probability is involved with risk. Uncertainty involves both descriptive and measurement uncertainties. [2] In addition, the nonlinear, nondeterministic character of the dynamics of the environment also contributes to uncertainty. [3] Uncertainty also arises from the inability to measure or describe exactly the circumstances associated with risk, but collectively from the kinematic and dynamic characteristics of the environment as it evolves with time.

The interrelationship of uncertainty and time is evidenced in the uncertainty associated with risk, in that this uncertainty reflects the uncertainty regarding future events. [4]

Managing Risk as a Team

Team Risk management defines the organizational structure and operational activities for collectively managing risks throughout the enterprise. [5] The Team Risk Management approach is built on the principles described in Figure 1.

Principle	Effective risk management requires
Shared product vision	A shared vision for success based on commonality of purpose, shared ownership, and collective commitment.
Forward–looking search for uncertainties.	Thinking toward tomorrow, anticipating potential outcomes, identifying uncertainties, and managing program resources and activities while recognizing these uncertainties.
Open communications	A free flow of information between all program levels through formal, informal, and impromptu communication and consensus–based processes.
Value of individual perception	The individual voice which can bring unique knowledge and insight to the identification and management of risk.

[3] *Application Strategies for Risk Analysis*, R. N. Charette, McGraw–Hill, 1990.
[4] *Third Wave Project Management*, R. Thomsett, Yourdon Press, 1993.
[5] "An Introduction to Team Risk Management," R. P. Higuera, et al, *CMU/SEI–94–SR–1*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

Systems perspective	That software development and integration be viewed within the larger systems–level definition, design and deployment.
Integration into program management	That risk management be an integral and vital part of program management.
Proactive strategies	Proactive strategies that involve planning and executing program activities based upon anticipating future events.
Systematic and adaptable methodologies	A systematic approach that is adaptable to the program's infrastructure and culture.
Routine and continuous processes	A continuous vigilance characterized by routine risk identification and management activities throughout all phases of the life cycle of the program.

Figure 1 – Principles of Team Risk Management

Risk Categories

The following risk categories are used to focus the reader on separating the risk of successful software *deployment* from the risk of *deploying* the software successfully.

This may seem like a trick of the phrase, however there are several subtitles here:

- Having the software system operate in a successful manner does not imply that the system itself is successful. Since the users of the system assume that the deployed software will somehow aid in their work day, the system must not only work, it must *add value* to the user's environment.

- Having met the user's needs while deploying the successful software system is not sufficient. The Company business operations must also benefit in tangible and measurable ways.

The job of risk management is to identify, address, and eliminate sources of risk before they become threats to the success of the project. Risks can be addressed at several levels. [6]

- Crisis management – fire fighting, address risks only after they have become problems.

- Fix on failure – detect and react to risks quickly, but only after they have occurred.

- Risk mitigation – plan ahead of time to provide resources to cover risks if they occur, but do nothing to eliminate them in the first place.

- Prevention – implement and execute a plan as part of the project to identify risks and prevent them from becoming problems.

- Elimination of root causes – identify and eliminate factors that make it possible for risks to exist at all.

[6] *A Managers Guide to Software Engineering*, R. S. Pressman, McGraw-Hill, 1993.

Risk Definitions

- Cost Risk – the degree of uncertainty associated with system acquisition life cycle budgets and outlays that may negatively impact the program.
- Performance Risk – the degree of uncertainty in the development and deployment process that may keep the system from meeting its technical specifications or that may result in the system being unsuitable for its intended use.
- Risk – the condition of having outcomes with known probabilities of occurrence, not certainty of occurrence.
- Risk Abatement – the process of reducing the amount of risk to a system.
- Risk Analysis – examining the change of outcomes with the modification of the risk drivers. This examination is more involved than risk assessment and should result in the identification of the most crucial variables with insights into desired options of risk handling.
- Risk Assessment – the process of examining a program and identifying areas of potential risk.
- Risk Drivers – those variables that cause probabilities of cost, schedule, performance, or support risk to fluctuate significantly.
- Risk Handling – the identification of options available to reduce or control selected risk drivers.
- Schedule Risk – the degree of uncertainty associated with the ability of a program to achieve desired milestones (outcomes) on time.
- Support Risk – the degree of uncertainty associated with the ability of the support organization to maintain, change, or enhance software of the fielded system within the planned support concepts and resources.

Software Risks

- Software Project Risk – defines operational, organizational and contractual software development parameters. Project risk is primarily a management responsibility. Project risk includes constraints, external interfaces, supplier relationships, or contract restrictions. Other examples are unresponsive vendors and lack of organizational support. Perceived lack of control over the projects external dependencies makes project risk difficult to manage. Funding is the most significant risk in most risk assessments.
- Software Process Risk – includes both management and technical work procedures. In the management procedures, there is risk in activities such as planning, staffing, tracking, quality assurance, and configuration management. In technical procedures, risk is found in engineering activities, design, programming, and testing. Planning is the management process most often found in risk assessments.
- Software Product Risk – contains intermediate and final work product characteristics. Product risk is primarily a technical responsibility. Risk will be found in the stability of the requirements, design performance, software complexity, and test specifications. Because the system requirements are often perceived as flexible, product risk is difficult to manage.

Structure of Risk Analysis

There is a hierarchy of risk analysis associated with the deployment of software based systems. [7] The importance of Figure 2 is that risks can be classified into categories to better isolate the mitigation of each risk component. Each software system is unique with its own particular set of risks. The risks can be partitioned as:

- Potential Cost
- Schedule
- Technical / Business Consequences

In order to be successful, the software–based system must meet its technical and business requirements within cost and schedule constraints.

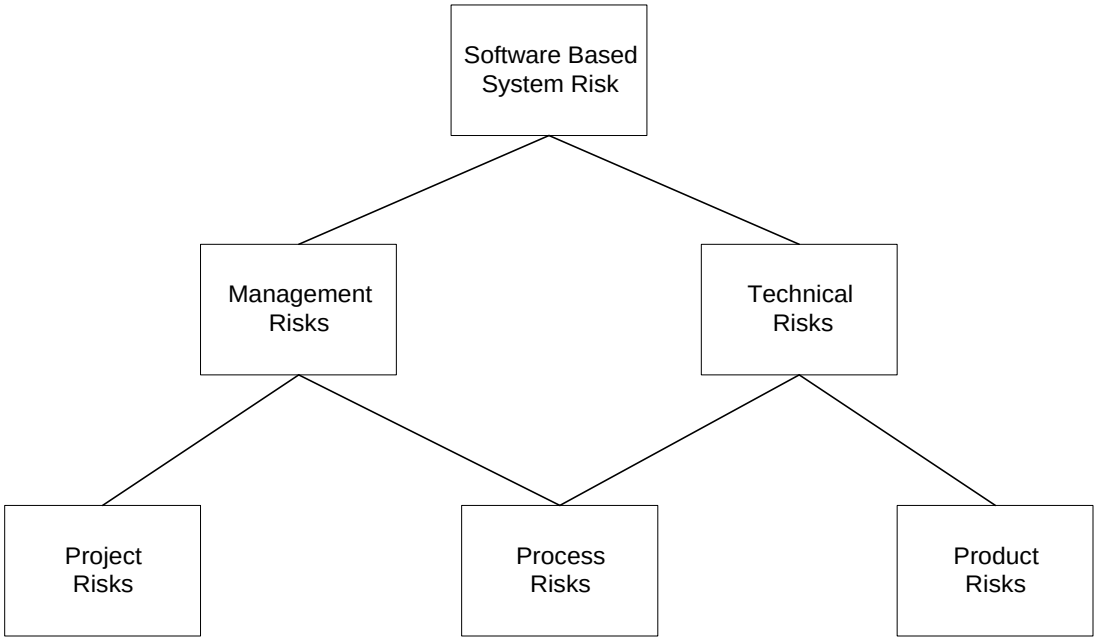

Figure 2 – Software Project Risk Hierarchy

Software Based Systems Risk Management

Risk management is the practice of assessing and controlling risk that affects the software project, process or product. The basic concepts of software risk management are:

- Goal – risk is managed in relation to a specific goal and can affect only the work that remains to achieve the goal. What is the risk in the plan? What is the risk in the remaining work? A clearly defined goal with measurable success criteria bounds the acceptable risk.

[7] *Taxonomy Based Risk Identification*, M. Carr, S. Konda, I. Monarch, F. Ulrich, C. Walker, Technical Report CMU/SEI–93–TR–6, Software Engineering Institute, Carnegie Mellon University, 1993.

- Uncertainty – is that which we do not know. It is inherent in all of the assumptions and the future itself. There is always a degree of uncertainty in risk occurrence. The probability of risk occurrence is always greater than zero and always less than 100 percent.

- Loss – unless there is a potential for loss, there is no risk. The loss can be either an undesirable outcome or a lost opportunity.

- Time – is needed to anticipate and prevent problems. Time is the great equalizer, since every day that is made available to the project is an additional day to deal with the consequences of risk. By managing risk, time can be used to an advantage, rather than being wasted.

- Choice – unless there is a choice, there is no risk management.

- Intelligent Decisions – are made on awareness, insight and understanding of the risks associated with the choices available. Risk management provides a process to communicate risk information and provide visibility into the risks at a project level.

- Resolving Risk – is done by developing and executing a risk action plan to resolve the risks. The key to resolving risk is finding the risk elements when there is time to take action and knowing when to accept a risk.

- Preventing Problems – the resolution of risk prevents problems and surprises. Risk management is a proactive strategy to reduce the problem of costly rework.

Risk Evaluations

The next sections provide the risk evaluation criteria, their measures within an organization and the steps to be taken to mitigate the risks by the risk management staff.

> *I keep six honest–serving men*
> *(They taught me all I knew);*
> *Their names are What and Why and When*
> *And How and Where and Who*
>
> *– Rudyard Kipling, Elephant's Child*

Risk and uncertainty are not the same. Risk involves knowing the range of outcomes. Uncertainty involves not knowing the range of outcomes. Risk evaluation is important to the effective management of this project. However, risk evaluation is not difficult, it is a matter of asking many questions.

The following questions and range of outcomes are *representative* of a typical IT development process performed at *The Company*. These questions are not meant to be exhaustive – since the construction of the question list is a continuous improvement process.

Both the *current situation* and the *risk mitigation* columns have prototype answers to guide the reader through the thought processes.

Predevelopment Risks

Predevelopment risk items are associated with the activities prior to the development and deployment of a major data processing system. These risks drive the size of the project. If the size becomes unmanageable or becomes larger than planned, then there is a risk that the project will be delivered late and / or over budget.

In addition, size directly relates to performance risk. Without knowledgeable estimates of the projects size, predictions of the systems performance are difficult.

They consist of:

- Size factors related to the size and complexity of the proposed project.
- Structural factors related to the organizational complexity of the project.
- Technology factors related to the technology components of the system.

Predevelopment Size Drivers

Pre Deployment Size Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)			
Number of Departments (other than IS) involved with the system?	1	2	5			The number of departments will eventually include all the Business Unit organizations. For the initial deployments, smaller groups should be targeted.
Total development manhours for the system?	5 Man Years	10 Man Years	20 Man Years	12 Man Years		The majority of the actual development will be performed using system integrators and vendors of COTS products. This estimate is for the Company supplied personnel. Although these appear to be high, the project management, data mapping, and overall contribution for all personnel is a serious commitment for the Company.
What is estimated project implementation time	12 months or less	13 months to 24 months	24 months or more	24 months		The current schedule for the project includes the deployment of a enterprise system, capturing of the design data and creation of the foundation for the future of data management. Breaking up the project into smaller deliverables reduces the risk of a total failure. At the end of each smaller deliverable, a working solution needs to be available. At all points in the project, deliverables needed to be made available, so that if the project runs into problems, or the project is halted for any reason, there is a useable component. Defining the schedule around these usable components is a critical success factor.

Pre Deployment Size Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Data processing breadth – expressed number of programs, size of programs, number of transactions	10's	50's	100's	50 individual programs. This number is derived from the current environment and includes all the identifiable pieces of code that are needed to integrate the system, not just the ones visible to the end users.		The primary role of the Executive Steering Committee (ESC) will be to focus the efforts of the Data Management Project Team on components of the project that have the highest payback with an acceptable level of risk. This will require a full understanding of the alternative deployment strategies, their cost and benefits. This will also require the full cooperation of the ESC as well as the managers affected by the project roll out. Without this cooperation, the risk becomes one of deploying too many components to have a reliable integration result.
Expected frequency of change – the number and/or size of changes that will be made to the initial needs statement	10%	20%	50%	30%, since the project has started there have been several changes in major scope. As the project progresses it is expected that the rate of change will increase, until it becomes stable at some point.		By freezing the requirements and building the system capabilities around a COTS product, the risk of changing requirements can be minimized. By developing a business reengineering process around the requirements, the underlying business activities can be adapted to the capabilities of a COTS system.
Number of unique logical business inputs that the system will process – expressed in number of business transactions processed in the course of a day. This is commonly referred to as the *number of object points*.	100	200	500	300, since the current environment is function rich.		This risk may be unavoidable, since the number of functions the system needs to perform is driven by the Business Unit environment as well as standard manufacturing practice. Some business process reengineering can be done to reduce the number of functions, but the intent of any modern system is to provide flexibility to changing business needs, including the addition of new capabilities. The challenge will be to add this new functionality within the architecture of the system.

Pre Deployment Size Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Number of unique logical business outputs generated by the system – number of business transactions or reports or messages produced per day by the system.	100	200	500	300, this is a similar number as above.		This risk mitigation strategy is similar to above for the inputs.
Number of logical files (views) that the system will access – the number of individual views or database subschemas that will be accessed by the system during the totality of system processing.	20	50	100	15, this is an arbitrary number at this point. The actual number will be dependent on the specific vendor configuration. However, the logical number of files is different form the physical number.		This risk factor is more appropriate on the integration side of the system, since the connections between the various systems will be impacted by the number of files within a specific system.
Number of major types of on–line inquiries expected – the number of requests that will be made by users other than the normal business outputs generated by the system.	50	100	200	50, this number should be a design parameter.		The current environment provides the ability to create *private* queries for each user community. By designing a *well–formed* user interface, the number of uncontrolled queries can be limited.
Telecommunications – the use of communication facilities in conjunction with automated systems operation. Risk associated with the number of connected users, the amount of hard–copy documents produced and the sophistication of the processing.	100 Users	300 Users	500 Users	400, this is a derived number since the range of users can be very large at any specific time in the systems usage. The primary issue is to confirm that the system scales appropriately, with the number is users. This always means that the system *DOES NOT* scale linearly but scales in some Log Normal form, with additional resources providing significant increases in capacity. If the system scales linearly, then it will fail to meet the performance requirements in a short time.		The number of distinct users is not an actual risk factor, the risk comes from having multiple user communities, that are deploying the system in different manners. This risk can be addressed by limiting the number of different deployment environments. Unifying the span of the deployment, so that the support issues are minimized. The scalability of the system will be addressed in the performance section.

Figure 3 – Size Drivers

Predevelopment Structure Drivers

These risk factors influence the structural aspects of the system. These structural factors address the changes that must be made to deploy the system into the work environment. If these risks are not addressed, the effectiveness of the system will be less than planned.

Pre Deployment Structure Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)			
If replacement system is proposed, what percentage of existing functions are replaced on a one–to–one basis	0% to 25%	25% to 50%	50% to 100%	40% of the functions. This is an estimate at best. Since the current system consists of an assemblage of products and integration's, it is assumed that much of this code will be replaced. However, the *functionality* of the new system will model the current system, since the manufacturing of products remains the same.		Identifying the *must have* functions in the current system, the number of new functions can be limited. When a new function (or a function that is deemed *must have*) is suggested, some form of analysis should be done to determine the impact of this request. The business benefits are the first place to perform this analysis. This should be done by the Best Practices Team (BPT) and the Architecture Decision Team (ADT). These two teams will determine the impact on the architecture as well as the benefits to Business Unit.
What is severity of procedural changes is user department caused by proposed system?	Low	Medium	High	Medium – since the current departments operate as separate units, unifying them under a larger organization and deploying this unified procedural environment within the new system will require careful work.		The BPT and the ADT will facilitate the effort. By focusing on the business benefits, the managers will be able to determine what procedures need to be changed, and unified to benefit overall organization. The question always to be asked is *when will this change be paid back*?
Does user organization have to change structurally to meet requirements of new system?	Minimal	Somewhat	Major	Somewhat to Major, since the current organization is not built around the current system (expect to facilitate the movement of products).		This is opportunity for the BPT to unify the processes within all layers of the organization.

Pre Deployment Structure Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
What is the general attitude of the user?	Good – understands value of the proposed solution	Fair – some reluctance	Poor – opposed to the proposed system solution	Good to Fair – the understanding for the need is present, but the understanding of the effort is not. The organization is not well versed in changing itself.		This is an opportunity of the BPT. By focusing on the business benefits, the individual managers can see the future in a positive light. Only by developing the business case, can the focus be taken away from *fear* of loss of their bonus to expectation that bonus will be maintained if not improved.
How committed is upper–management to this system?	Extremely enthusiastic	Adequate	Somewhat reluctant or unknown	Adequate – the current understanding of the system is not fully developed. The risk here is that as the system details become more developed there will be less interest. The total cost and the resource commitments are not well developed at this point. What is developed is the desire to move from Copics to the next generation. Another risk is that the project focus will be bogged down in the details of the implementation and the strategic activities will become lost.		The focus of the project, at the executive level, should always be strategic. The tactical details should be pushed to lower levels of the project. Once the business objectives have been defined, the scheduled develop and the activities budgeted, the executive levels of management should focus on progress to plan.
Has a joint data processing / user team been established?	Full–time user representative appointed	Part–time user representative appointed	No	Part time – the current team dynamics are still developing. This risk item should change over time, with the focus on developing a data and process focus. Without such a focus the team will always be considered temporary.		The commitments that have been made so far need to be expanded and made permanent. This can be done through the organizational changes suggested in the IT Strategy. The permanent members of the team, will then be able to move within the Business Unit organization without concern for their previous positions.

Pre Deployment Structure Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Technology Experience. Does the team have direct experience with the proposed technologies?	In use today	Technology understood, but not fully deployed	Not in use today	The issue here is what technology. The state of the art is an Object Broker system with CORBA compliant components. This may be beyond the state of the art for the current environment.		The risk here is that the deployment of a technology that can be supported by the current Business Unit environment *may* an instant legacy system. Some means must be taken to determine what is desirable and what is possible. The vendor's ability to meet the state of the art is also a issue, since many vendor claim to be CORBA compliant, but few are actually deploying such systems.
Technology Availability. Is the proposed technology available in a form that is sufficient to the task. This includes the ability to deploy the technology in the Business Unit environment.	Available today. This technology is proven and deployed in the industry	Emerging today. The technology is emerging as the basis for solving problems in the industry.	Emerging in the future.	The selection of the technology has not been made.		This risk item can be addressed through the system architecture.
Technology Maturity. Is the proposed technology mature to the point it can be deployed in an industrial production environment?	Mature	Developing	Coming	The current technology approach is not determined. The ideal technology would be a full CORBA implementation, using wrappers for the existing IMS database components until they are migrated to the final system.		The selection of the technology will be determined by the selection of the final ERP system. One advantage of Avalon, is its CORBA base. Using Rational Rose and the object tools from IBM, the Copics database elements could be integrated.
Cost Models. Are there cost models available for the deployment of the system and the supporting technology?	Available	Understood but not available	Not understood	The current state of cost modeling within the Business Unit is immature. Costs are more than the price paid for the software and the expenses needed to make it function. Costs are also associated with the consequential effects of deploying the software into the production environment. What effects does the software have on the organization?		Developing the understanding that the cost of the system is more than software costs. This understanding can be developed through the process modeling that should take place over the project lifecycle. This modeling will capture the costs as well as the impacts on the process of deploying alternative systems.

Pre Deployment Structure Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Configuration Management. Is the a formal configuration management process in place for the software components being deployed?	Yes, this process is well proven.	Maybe, but the process is new to the environment	No, there is no formal process to control the configuration of the software components	No, there is no process in place. The current software deployment environment does not qualify as sufficient for the upcoming tasks.		The SEI System Integration Capability Maturity Model needs to be deployed within the Business Unit. This model does not create a solution, but is a guideline for improving the processes associated with deploying an integrated system.
Organizational Breadth – the number of diverse organizational units involved in the application system and/or the number of users organizations that must sign off on the requirements definition	Small	Medium	Large	Medium. The current organization is evolving to a flatter organization. The risk factor in place today will be changing.		The continued flatting of the organization will help reduce this risk.
Political implications of implementing the system – the level of agreement among all units in the organization as to the need for the system and the approach being used to accomplish the system adjectives	High agreement	Moderate agreement	Low agreement	Moderate agreement. However, the current level of agreement is taking place in the absence of any real deployment conflict. The real test will come when one of the managers must forego some important capability for the betterment of the overall organization.		This is an area where professional facilitation and architectural focus can be used. The *big picture* must be taken, complete with cost benefit analysis of each functional system.

Pre Deployment Structure Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)			
Specificity of user requirements – the level of detail in which the requirements are specified. Measures the amount of additional detail and/or decisions that need to be made before system can be developed or deployed.	Low requirements details	Moderate requirement details	Highly specified requirements	The current requirement details are encapsulated with Copics. The environment has created a situation where *breaking out* of the Copics world is seen as a *migration* process. The risk will be that the migration to a new system results in endless new requirements, since the previous system has been in place for so long. The risk here is that the new system will open up new requirements which must be managed within the context of COTS deployments		By adopting an existing system, the requirements generation can be minimized. The selection of COTS products can further reduce the risk, if the underlying business processes are adapted to the product capabilities.
Availability of backup hard–copy documents – the number of original source documents and hard–copy format that will be produced and retained during the system processing	Documents are readily available	Documents are available but are not current or accurate	Documents are not available	The issue here is how the requirements are to be determined form the existing documentation as well as how what documents the system produces during the normal processing cycles.		The System Requirements phase of the Data Management project will determine the actual risk here. It is not clear that there are consistent processes (paper based) throughout the enterprise. Alternatively, if these processes can be determined in a timely manner.
Level of user management agreement on system objectives – the agreement within the user(s) department on the stated objectives for the system.	High levels of agreement	Moderate agreement	Low levels of agreement	Moderate – although there has been much discussion at the executive level, the discussion at the level needed to actually deploy the system has not taken place. The risk here is that the users of the system have not been included in the initial architecture or requirements phases of the project.		The actual users of the system have been represented by the team members. However, at some point a broader user community must be allowed to have input to the system. This can be done through a Critical Design Review process or by syndicating the requirements specification through the organization using the current team members.

Pre Deployment Structure Drivers	Probability of Adverse Effects: Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Percentage of the proposed system that is already performed by the user – measures the newness of the system tasks to the user area. Differentiates between existing tasks being automated and new tasks (new meaning a new method for processing information.	Large percentage	Moderate Percentage	Low percentage	Moderate – the risk here is that any new system will replace a legacy user interface and database system. This replacement system will have very few behaviors like the previous system. Since the business processes have adapted to the current system, along with all of its undesirable features, replacing the current system with a *well functioning* system will create a disconnect in the users mind.		There is no mechanism to deal with this risk, except massive amounts of training. The training should be used as part of the vendor selection process. This will allow the users to determine if the product can meet their needs before the final selection takes place.
Importance / criticality of the business system to the user – measures the importance of the specific system to the user as it relates to the user completing the mission of the user function.	Low importance, the system provides support functions and these functions can be performed in its absence	Moderate importance, the system is part of the daily operation.	High importance, the daily operations depend on the system functioning properly.	This risk depends on the specific components being discussed. It is assumed that the future systems will become a critical component to the daily business operations. Without the system the business could not work.		Mitigating this risk is a *reverse* risk avoidance operation. The success of the Data Management project and the related applications will make the Company dependent on these systems. Which now creates a risk that if they are not available then there is a risk associated with the business operations.

Pre Deployment Structure Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Low (0.0 < P <0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)			
Project management approach and structure – the organization of the project in relationship to the size of the project and the technology being utilized. Includes such consideration as division of duties with the project, relationship between the user and IT personnel, as well as the management and status reporting methods.	Well structured project management activities	Moderate structure to the project management activities	Low structure of the project management activities.	Moderate to Low – since this is the first large project that the Business Unit has undertaken, there is a moderate risk that the project will be impacted by the skills and experience of the team members. The members of the team are mature managers, so the business aspects of the project will not be new. It is the technical and project activities themselves that create the risk.		Develop good project management skills and provide outside help in the technical and project activities. Having *been there* is the key to success in the management of software projects. Extensive reading and course work will also help. There are many outside resources to help the project manager and the team. Continuous education is a must.

Figure 4 – Structure Drivers

Predevelopment Technology Drivers

Pre Deployment Technology Drivers	Probability of Adverse Effects					
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Distributed processing – is the proposed technology based on well–established technologies with verifiable architecture components?	Yes	Maybe	No	The proposed system will follow the current distributed processing guidelines. The risk here is that the vendors of subsystems, will have significant influence on the system design and create an environment that does not meet the needs of the Company.		The IT Strategy must be used at all times when evaluating vendors. The current corporate guidelines are not sufficient to control the introduction of technology. A test and certification environment should be build (at least for the Business Unit users) to verify that all new system are interoperable with the existing environment. Vendors should be engaged early on, at a very detailed level, to determine exactly how their systems work and the impacts on the existing systems.
New domains of technology – are new domains of technology being deployed with the project? Are these domains subject to verification within the scope of the project plan?	No	Somewhat	Yes	The intention of the system architecture is to install well developed technologies. The vendors in this market place are not driven by the latest technology. In fact, just the opposite may be true, which is this risk in reverse.		The system architecture will be used to control the technology domain. A risk analysis should be performed for each system component.
Human machine performance – are the technologies to be used for the human interface components of the system well established? Are these components part of an accepted standard in the industry?	Well accepted standards	Somewhat accepted standards	Unique standards	The user interface must follow the Company standards, however, direct control over the vendor's implementation of these standards is out of the control of the project. There is risk that the standards will some how be counter to the needs of the Company.		The definition of the standards must go beyond the simple statement of Windows compliance. Specific environment descriptions, with protocol stacks, runtime specifications and resource usage requirements. The deployment of a Common Operating Environment (COE) for the desktop should be planned. This specification can then be used to validate any vendor's offering. In addition, the verification of the workstation environment can take place in a testing environment.

Pre Deployment Technology Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Algorithm speed and accuracy	Low requirements, simple processing with known performance requirements	Moderate requirements, which push at the boundaries of the environment at times.	High requirements that push at the boundaries of the environment the majority of times	The current environment is provided by the mainframe processing systems. Other than the parts management system (which already has performance problems), the next generation system will be driven by the ERP scheduling and materials planning algorithms.		A testing and verification environment should be deployed. In this environment, the performance the system can be verified before it is deployed. In the Logia example, the performance of the system does not appear to scale properly. All client / sever applications should scale in some logarithmic form. Linear scaling will result in system failure early in the deployment cycle.
Security	Low levels of security. The system is essentially open to all users, with few exceptions	Moderate levels of security. The system provides named user security and restricts access to major functional components.	High levels of security. The system requires security for each activity as well as the data is uses	Low to Moderate – the concept of an open system is not yet developed within the Company. For many of the objects managed by the system, full access can be provided. For others there should be restricted access. The risk here is that a security model has not yet been developed for the data and processes. In the absence of this model, the security will become a *patchwork* of processes.		Develop a full security model for the system and the data it manages. There will many data components that are open to all users. Other information, like design models, is private. The secondary level of security is to provide a *profile* approach in which Access Control Lists (ACL) are used to define the security capabilities. This approach should be specified in the general system requirements.
High reliability and fault tolerance	Low need for fault tolerance	Moderate need for fault tolerance	High need for fault tolerance	Moderate need, since the system will be targeted to the production environment. The risk here is that the underlying system architecture is not adequate to meet the fault tolerance requirements.		Define the reliability and availability requirements in terms of system parameters. The term fault tolerance has many meanings, define ones that can be delivered by the vendors as well as meet the needs of the Company. This area requires careful consideration, since the requirements for fault tolerance has direct impacts on performance and complexity. This is an architecture tradeoff issue.

Pre Deployment Technology Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Reusable software components	Low need for reusable components	Moderate need for resusable components	High need for reusable components.	Moderate, in principle the reuse of software components is a mandatory requirement for the next generation system. In fact it is the reusability of the data that is the requirements.		Define the data reusability requirements in the system architecture. The actual component reuse rate is out of scope, since they are provided by the product vendors.
Makeup of project team in relationship to technology used – the inclusion on the project team of the necessary skills to effectively utilize the system technology.	Low - Skill sets are present	Moderate - Skill sets can be acquired	High - Skill set are not available	Moderate, the current development environment is targeted toward mainframe applications. The current contractors are developing (or maintaining) C/S code without the aid of a methodology.		The introduction of modern software development techniques must be done. These can be acquired through training and recruiting. Building for the future is vital, and the state of the art is moving rapidly.
Applicability of the design methodologies and standards to the technology in use – the adaptability of the existing processing methodologies and standards to the technologies being used.	Highly applicable	Moderately applicable	Not applicable	Not applicable – there are no design methodologies in place within the Company.		Create a standard for system design for the project. This standard will be based on the architecture patterns of the various vendors products, the architectural environment in place today and the desired architectural environment of the future.
Margin of Error – the amount of time between the entry of a transaction and the response to the transaction. For example, is there a reasonable amount of time to make adjustments, corrections, or perform analyses before the transaction is completed?	High margin of error	Moderate of error	Low margin of error	Moderate margin of error – the business processing system will have external checks as well as personnel making the final decisions. The system is nit fully automated with robots and machining centers.		Install manual check points to verify the results on the applications. This can be a simple as cross checks with scheduling, pervious days runs, trend analysis for parts inventories and other historical data. The PDM and ERP systems should be specified to provide such checks and balances.

Pre Deployment Technology Drivers	Probability of Adverse Effects					
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Technical complexity of the system – the number of tasks and interrelationship between those tasks that must be accomplished to satisfy the user needs.	Low technical complexity	Moderate technical complexity	High technical complexity	Moderate technical complexity – the integration of the various system components create a risk that the complexity of data and processing steps will be beyond the ability of the Business Unit to manage.		Create a clear and concise documentation standard for all integrated components. Process and data flows, timing charts, interface specifications (IDL in the OO world), datamodels, and workflow simulations. Use the deployment technology as the documentation tools.
Adaptability to change – the ease which it is expected that changes to the system requirements can be incorporated into the system. This will be dependent upon the architecture of the system and its adaptability to the needs of the system.	Low adaptability required	Moderate adaptability required	High adaptability required	High – it is expected that the system will form the foundation of the next generation manufacturing environment. The risk here is that this environment is not yet defined. Selecting an architecture and possibly the software components will establish the system boundaries before the actual business boundaries are discovered.		The IT Steering Committee needs to define the maximum boundaries for the target system. Is flexible manufacturing a goal? Will a new plant be built with machining centers and automated material handling? Will the current semi-batch processes be eliminated in favor of flow through manufacturing? This vision needs to be articulated before the final architecture of the system can be completed.
Utilization of equipment – how much the system will push the equipment to its capacity to meet the needs of the users. For example, if a two–second response time is needed and given the complexity of the tasks and the volume of work, what is the amount of tolerance within the systems capacity to meet those processing needs?	Low utilization	Moderate utilization	High utilization	Moderate utilization – the performance of the hardware is catching up with the software requirements. The risk is that the ERP and PDM systems as *assumed* to operate in the minicomputer environment the same way they did in the mainframe environment.		Establish clear performance measurement tests for all vendors. Establish reserve capacity tests and performance acceptance testing at this boundaries of this capacity. Do not let the vendor *explain away* the performance measurement problems. Actual test results are required.

Pre Deployment Technology Drivers	Probability of Adverse Effects					
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)	Current Situation	Risk Level	Risk Mitigation Steps
Personnel – skill level, number and knowledge of user processing of the project team members including any supporting technical staff(s).	Low skill level required	Moderate skill level required	High skill level required	Moderate to high skill level is required for the migration from the mainframe and simple C/S requires not only technical skills but architectural skills. The integration risks are moderate to high, since disparate applications will be *federated* with external databases. There is no experienced staff that has performed such a task in the past.		Define a detailed design and verification plan for the system. Engage a system integrator that has delivered a system like this in a similar environment. Recruit a key person to define the architecture and manage the technical integration.
Documentation – amount, correctness, type and usability of the documents supporting the system.	Low	Moderate	High	Moderate – not because of any external requirement, because of the low experience level of the staff. By producing detailed documentation of the architecture, data and process flows, the overall system information base can be maintained.		Adoption of a design and documentation methodology. This process should be enforced for all components of the project. Resist all attempts to bypass this process.
Pioneering aspects – the newness of the technology and/or technological approaches used in this application. The newness can be within either the organization or the newness of the technology as offered by the vendor.	Low	Moderate	High	High – in the Company environment, this will be a pioneering effort. Moving the mainframe to the C/S environment and federating the data and processes is without precedent		Use the existing ERP rollout as an example for how to proceed with the conversion from Copics to Avalon. This will not help with the integration of PDM, but the skill sets needed for that project should e used to define this project.

Pre Deployment Technology Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
How knowledgeable is the user in the technology – determines whether the user personnel can understand the implications of use of the technology, and their ability to define requirements and discuss requirements in relationship to it impact on technology.	Highly knowledgeable	Moderately knowledgeable	Weakly knowledgeable	Weakly knowledgeable – the current C/S systems are simple and not demanding. The CAD experiences have no real-time synchronization requirements or production load requirements. The mainframe environment hides the database issues inside the applications.		Staffing must be added to bring the technical skill levels up. The business skills are capable of defining the boundaries to the architecture and requirements.
Processing knowledge of the user tasks – the ability of personnel to challenge the accuracy and need of user requirements in relationship to the mission and tasks.	High	Moderate	Low	High – the staff assigned to the project have a high level of understanding of the business processes		The irony here is that since the current system has many manual steps the users are very familiar with the business processes. This knowledge can not be use to automated many of the steps.
Degree of Complexity of processing logic – measures whether the logic needed to perform the user requirements will be simple, average or complex.	Low	Moderate	High	Low – the processes deployed in the shop are of low complexity when compared to other manufacturing environments. The PDM and ERP systems available on the market have been developed for aerospace and automobile manufactures.		This is both a advantage and disadvantage. Since the vendors have developed complex capabilities for the big customers, this capability may not be needed at the Company. Since the complexity is not needed there are many alternatives to the PDM, EDM and ERP system.
Need for automated error detection and correction procedures – measures the complexity of the procedures that need to be incorporated into the system to detect inaccurate or incomplete input transactions and make automatic correction to those errors.	Low	Moderate	High	Moderate, since the current environment requires manual intervention to correct errors. The next generation system must add significant value here. The rules for correcting the errors can be well defined. The deployment of the error handling processes will be complex in the heterogeneous environment.		The capability of correcting errors in the database and transaction processing is a well-understood activity. The design of the system therefore must address this issue early in the design cycle. Built in transaction editing and database business rule approaches must be considered. This will place additional performance burdens, but the relief of manual interventions will be paid back many fold.

Figure 5 – Technology Drivers

Decision Drivers

Political Drivers

- Choice of Equipment – the current standards are not in conflict with the potential vendors offerings
- Choice of integrator – there has been no integrator selected.
- Schedule and budget – the schedule and budget have not been defined in sufficient detail to determine if there is a risk. The budget numbers have been provided by Gartner Group and should be considered a broad estimate. Lower numbers should be targeted. [8]
- Allocation of responsibilities – the current Business Unit organization can form the basis of the responsibilities. The Project Manager role needs more clarification and support staff.

Marketing Drivers

- Gold Plating – there is little risk of over specifying the solution. The desire to simplify is driving many of the decisions.
- Choice of Equipment – the equipment standards are more than adequate
- Schedule and Budget – the realization that careful scheduling is required protects the desire to have the system too early.

Solution Drivers versus Problem Drivers

- In–house components – this risk is present at nearly all companies. The concept of *reusing* software systems that have already been paid for is a great incentive to save money and time. The question is *what is the risk to the project by reusing this software*?
- Product Champions – there are many internal champions for various software solutions to identified system needs. These champions have a vested interest in seeing their solution prevail in the final product mix. The risk here is that the potential solutions may not be appropriate.

Short Term versus Long Term

- Staffing – the need for short-term staff versus long term staff is a risk to the planning process. The ramping of the staff must follow the needs of the project.

[8] There is a risk associated with relying of the Gartner Group for advice on subjects they are not qualified to deliver. Gartner's main expertise is the analysis of products and market trends. They, however, are not system architects nor do they deliver working solutions to the field. This is not to mean that what they do provide does not have value – it does. Care must be taken in reading too much into the forecasts for the future. A system integrator or other resources (vendors, consulting firms, trade organization, installed sites, etc.) provide useful information. The processing of all of this information needs to take place in an environment of *informed* analysis.

- Software reuse – the reuse of existing component is both a risk and a requirement. Since the legacy systems must remain in place while newer systems are being deployed, dealing with reuse issues is a risk that must be addressed in the project plan.

- Premature Reviews – the desire to review progress and provide direction too soon in the process is a risk at most companies. The manufacturing environment creates a work environment where progress is measured on a daily basis. In the system development and architecture environment, *think time* is vital to the success of the project. This time includes just thinking about the solutions as well as studying the subject materials associated with the systems. The risk is that management does not understand this new environment, and the participants will not be allowed to use their think time to address complex problems. *The solution can not be purchased like a raw material – this is an intellectual process which takes time, and time means money.*

Post Development Risks

The following risk items are applied to the proposed software system *AFTER* it has been deployed into production. Once the system has been deployed, its continued operation and maintenance is just as important as its original deployment. Continued training is also a requirement for the successful system operation.

Post Development Cost

Software cost estimates are affected after the deployment of the system.

Cost Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Requirements						
Requirements Size – how do the requirements for this system compare with other systems that have been deployed by the same team?	Small noncomplex or easily decomposed	Medium, moderate complexity, decomposable	Large, highly complex or not decomposable	The requirements or moderate to large. The specific of COTS adds complexity in this case, since the proposed systems may not be open enough to integrate all components and meet the requirements.		Modify the business process to adapt to the capabilities of the COTS applications. Avoid at all costs the specification of custom software.
Hardware Resource Constraints	Little or no hardware imposed constraints	Some hardware imposed constraints	Significant hardware imposed constraints	Little or no constraints, using the Company standard hardware will support the majority of vendor's products.	0.2	Continue to maintain the hardware standards.
Software Resource Constraints	Little or no software imposed constraints	Some software imposed constraints	Significant software imposed constraints	Little or no constraints. The use of COTS applications implies they will run in a standard environment	0.2	
Technology	Mature existing, in house experience	Existent, some in house experience	New or new application, little experience	Some in house experience. The introduction of Unix and Oracle will add some complexity	0.4	Training in these systems will allow experience to be gained *on the job*. The vendors systems are usually well integrated with the operating systems and databases.
Requirements Stability	Little or no change to established requirements	Some change in baseline expected	Rapidly changing or nor baseline	Some change expected. Since the requirements are just beginning to be developed, there is time to impose structure.	0.4	Provide clear methodologies for managing requirements (SEI Guidelines).

Cost Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Personnel						
Personnel Availability	In place, little turnover expected	Available, some turnover expected	High turnover, not available	Available, but recruiting may be a problem. The risk here is that new technology usually is more attractive than older technology. Both are required for the system to work.	0.6	Start now with recruiting efforts. Define a set of clear job descriptions. Hiring from outside the area may be necessary. Developing a recruiting plan is as important as the technology itself.
Personnel Mix	Good mix of software disciplines	Some disciplines inappropriately represented	Some disciplines not represented	Some disciplines inappropriately represented. With the heavy emphasis on the manufacturing applications, the underlying system technology and computer science architecture is missing. Also there is no experienced software development project manager (systems integration manager) present.		These positions can be filled through internal transfers or outside recruiting. Consulting service can be used to *startup* the training cycles.
Personnel Experience	High experience ratio	Average experience ratio	Low experience ratio	Average experience ratio. Except in the areas of the mainframe, most of the experience levels are moderate. No large system integration, or modern database or development activities (Objects, C/S, multi tier)	0.6	These experiences can be learned. Training, education and external advice can bridge the startup cycle.
Personnel Management Environment	Strong personnel management approach	Good personnel management experience	Weak personnel management experience	Good to weak personnel management approach. The current approach of *private* developments (individual projects) will have difficulty scaling to the larger project environments.	0.7	Deploy a fully developed software integration and test environment. Experienced software manager, and several key integration developers, with experience in Oracle, C/S and object technology. Move away from the support paradigm into the development paradigm.

Cost Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Availability	Compatible with need dates	Delivery dates in question	Incompatible with need dates	Compatible, except for the Y2K and the Copics phase out schedules. The Y2K schedule is fixed and the Copics phase out is not firm.	0.4	Work backward for the Copics phase out.
Modifications	Little or no change	Some change	Extensive change	Little or no change.	0.2	This assumption should be turned into a requirement. Take all software as is.
Languages and API's	Compatible with system and PDSS requirements	Partial compatibility with requirements	Incompatible with requirements	Unknown at the moment. The one example is Logia, which is written in a nontraditional language (Power Builder) for the type of application.		Enforce the language requirements for all *glue* components. The vendor's language cannot be defined, but the developed code must be controlled. Install a complete development environment. Recompile all components between point releases. Adopt the Microsoft method of having a clean build at the end of the day (define day appropriately).
Rights / Licensing	Compatible with PDSS requirements	Partial compatibility with PDSS requirements	Incompatible with PDSS requirements	Compatible – the licensing standards operate in a mature business environment.	0.2	Except for any small applications, the targeted vendors have mature license experiences.
Certification	Verified performance application compatible	Some application compatible test data available	Unverified little test data available.	Unverified – this is a risk, since the vendors have very little capacity to predict the performance of their systems.		A full performance evaluation before selection is required. Continued performance tuning and test lab environment should be established.

Cost Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Tools and Environment						
Facilities	Little or no modifications	Some modifications existent	Major modifications, nonexistent	Some modifications – the current development environment within the Business Unit is *light* on tools and equipment.	0.4	Tools and equipment budget should be defined in the initial planning stages. This should also include training, benchmarking, site visits and time set aside for information gathering and research.

Figure 6 – Cost Drivers

Post Development Performance

One way asses performance risk is by identifying the factors that allow risk drivers to be identified. By identifying these factors, the appropriate tools to address the risk can be deployed.

The risk factors have been divided into essential elements that provide the greatest amount of uncertainty in achieving technical and performance objectives.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Requirements						
Complexity	Simple or easily allocatable	Moderate, can be allocated	Significant or difficult to allocate	Moderate complexity. The integration of multiple applications and the data they manipulate is *complex*. There will be temporal complexity as well, with multiple applications making synchronized access and updates to the shared database entities.		By deploying the system in stages, the complexity can be absorbed over time. The primary approach to complexity is to continuously maintain a well documented and tested baseline. Full software development behaviors must be deployed within the Business Unit. The previous arguments (in the Needs Analysis and IT Strategy) that the Company is not in the software development business is not true, the Company is just not typing on the keyboard. All other aspects of software development are present. This is an education issue, that can be addressed through effort and training.
Size	Small or easily broken down into work units	Medium, or can be broken down into work units	Large, cannot be broken down into work units	Medium, the system by definition can be broken down in smaller work units.	0.5	The system architecture must define a partitioned set of applications. These will keep the control of the system complexity through the architecture processes. Use tools to manage the complexity. CASE tools and system analysis tools can identify the complexity points and maintain the documentation required to control this area.
Stability	Little or no change to established baseline	Some change in baseline expected	Rapidly changing or no baseline	Some changes – since this project will be considered *discovery design* there will changes in the	0.7	The deployment of test bed systems will aid in the performance prediction. The vendors must be required to demonstrate performance scaling with lab numbers not marketing information.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
				performance requirements.		
Post Deployment System Support	Agreed to support concept	Roles and missions issues unresolved	No support concept or major unresolved issues	Unresolved – this is a new role for the Company	0.8	Use established standards for defining the roles and responsibilities. Vendors, textbooks and web resources all have templates for defining the roles and responsibilities.
Constraints						
Computer Resources	Mature, growth capacity within the design constraints	Available, some growth capacity	New development, no growth capacity, inflexible	Available – the hardware selection has not taken place, but the standard vendor has machines with large capacities. However, there is a tendency within the Business Unit to buy-low. This is a risk, since the cost tradeoffs for a production system create problems in the future. There is no established client hardware environment. Since many of the potential users are now on 3270's		Careful analysis of the performance requirements and upgradability of the systems is needed. A standard production offering should be deployed for both servers and clients.
Personnel	Available, in place, experienced, stable	Available, but not in place, some experience	High turnover, little or no experience, not available	Available but not in place – this is a large risk in the locales the system will be deployed.	0.8	Recruiting is needed. In addition, some creative alternatives are needed, such as an offsite research center, located where the skill sets are, complete with a lab and direct telecommunication connections. This type of facility is currently deployed in other industries, since the availability of skilled labor is tight in all technology markets.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Standards	Appropriately tailored for the application	Some tailoring, all not reviewed for applicability	No tailoring, none applied to the project	No tailoring – standards for software do not exist at the Company. There are IT standards but they are targeted at hardware and networking.	0.7	Develop standards. There are many resources from which to acquire the standards.
Equipment and test capabilities	Meets requirements	May meet requirements, uncertain availability	Incompatible with system requirements, unavailable.	May meet requirements – the test lab environment is immature	0.8	Develop a mature test and development laboratory. This would include client and server hardware software evaluation facilities test and support staff. Treat this lab just like a vendor. Problems could be identified and fixes verified before placing them in production,
Environment	Little or no impact on the system design	Some impact on the system design	Major impact on the system design	Little or no impact – the current computing environment can be handling with standard equipment. Some conditioning may be necessary for PC's on the shop floor.	0.2	Define the PC environment
Performance Envelopes	Operation well within boundaries	Occasional operation at boundaries	Continuous operation at boundaries	Occasional operation at the boundaries – since the performance boundaries are not yet defined, they will surely be reached at some point.		Define the performance models from the vendors and the integrated system. Make use of performance monitoring and modeling tools. Allocate performance analysis as part of the project plan and continuing operations budget. Do not rely on the vendor's predictions for performance numbers. Measure the actual numbers before proceeding.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Technology						
Language	Mature, approved HOL used	Approved or non–approved HOL used	Significant use of non–approved HOL	Moderate risk here since the control of languages is not a Company tradition. The best example is Logia written in Power Builder interfacing with C/C++ environments. The risk is that there is no experience in this area.	0.5	Establish language standards and well as runtime standards for all integrated components. This would include the middleware components as well as database integration software.
Hardware	Mature, available	Some new products being introduced to the project	New environment with new products and use	Mature	0.1	The standard hardware is capable of meeting the needs.
Tools	Documented, validated, in place	Available, validated, some new deployment required	New deployed for this project.	New for this project – there is no tradition of using tools for analysis, support or maintenance. A good example is the problems with Logia. There are no debugging tools being used to determine the causes of the system lockup.	0.8	A full set of diagnostic software tools should be available. The vendor must be running the same hardware and software environment as the Company. This must be verified during vendor selection as well as site visits. The vendor should be treated as another supplier to the Company, complete with TQM audits and supplier qualification visits.
Data Rights	Fully compatible with support and follow on	Minor incompatibility with support and follow on	Incompatible with support and follow on	Fully Compatible – this means that the rights to the data are not an issue. Another question may arise though with external purchasing information used in the CSM applications.	0.1	There is no data that must be acquired or produced that has right-to-access issues.
Experience	Greater than 4 years	Less than 4 years	Little or none.	Moderate risk, since this is a discovery design project.	0.7	Training and recruiting the skill sets for the system architecture and performance analysis.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Development Approach						
Prototypes and Reuse	Used, documented sufficiently for use	Some use and documentation	No use or documentation	No use – this is a replacement system. The risk here is that software will be developed for functions that are already in the legacy system. The performance impacts are small, but the cost impacts are unknown.	0.4	The mitigation has not been determined.
Documentation	Correct and available	Some deficiencies, available	Nonexistent	Little or no use of documentation – there is no traditional (outside of the mainframe environment) of providing detailed documentation of the deployed systems.	0.7	Develop and deploy documentation standards. These standards can be acquired from a variety of sources, textbooks, web resources, industry standards (IEEE, ACM), and trade organizations. This is a standard task for a contract developer, using the industry guidelines.
Environment	In place, validated, experience with use	Minor modifications, tools available	Major development effort	Minor modifications – mainly in the network area.	0.3	The current networking topology is spoke and hub, while the data and process usage is fully connected peer-to-peer.
Management approach	Existing product and process controls	Product and process controls need enhancement	Weak or nonexistent	Weak and nonexistent – the development and deployment of C/S software has not been the role of the Business Unit. The risk here is to schedule and budget. The technology risk can occur when the lack of experience in acquiring C/S applications appears.		Training, education and recruiting for the C/S environment. Professional advice should be acquired for the system architecture, vendor management and other *one time* technical issues.

Performance Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Integration	Internal and external controls in place	Internal and external controls not in place	Weak or nonexistent	Weak – there is no traditional of managing integrators.	0.7	Training and experience will address these issues. Having a clear test and acceptance plan for the vendor. Requiring industry norms for the integrator will help eliminate future problems, since vendors that understand the standards are usually capable in other areas as well.

Figure 7 – Performance Drivers

Post Development Support Drivers

Support Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Design						
Complexity	Structurally maintainable	Certain aspects difficult	Extremely difficult to maintain	Certain aspects difficult to maintain – the integrated environment in which data and processes are connected in the C/S architecture will create new problems for performance monitoring, debugging and support.	0.6	Deploy a full test environment to verify problems, test fixes and deploy new releases.
Documentation	Adequate	Some deficiencies	Inadequate	Some deficiencies – the creation of documentation of the integrated system is a risk. There is no tradition of generating internal documentation since the mainframe environment provided this with the system.	0.7	The deployment of documentation standards and personnel will be required over the life of the project.
Completeness	Extensive PDSS incorporation	Some PDSS incorporation	Little PDSS incorporation	Little PDSS incorporation – the concept of service contracts is not a tradition within Business Unit.		Develop the concept of PDSS and service contracts. Use existing materials to build this knowledge.
Configuration Management	Sufficient, in place	Some shortfalls	Insufficient	Insufficient – there are no facilities for managing configurations.	0.9	Build a configuration management tradition. The actual work will be dome by the vendors, but the management must come from the Company. Training will provide the skills for this function.

Support Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Stability	Little or no change	Moderate, controlled change	Rapid or uncontrolled change	Moderate controlled change – the skills of the mainframe environment are vital here. The risk is that these mainframe skills will be lost over time.	0.6	Training will be required for this skill set.
Responsibilities						
Software Management	Defined, assigned responsibilities	Some roles and missions issues	Undefined or unassigned	Some roles and mission issues – the Business Unit is not yet organized for the deployment of large C/S applications. Plans are being made, but the execution of those plans has not taken place.	0.7	Models of the C/S deployment and support environment are available from vendors and other resources. Training is mandatory here.
Hardware Management	Defined, assigned responsibilities	Some roles and missions issues	Undefined or unassigned	Some roles and mission issues – the hardware support issues are understood, it is not clear how the support will take place in a distributed environment.	0.4	The methods of supporting the hardware are well documented.
Configuration Management	Single point control	Defined control points	Multiple control points	Single control point – the Business Unit provides this control point today.	0.2	Continue with the single control point.
Software Identification	Consistent with support agreements	Some inconsistencies with support agreements	Inconsistent with support agreements	Some inconsistencies – the management of vendor supplied version is weak (with Logia as an example). This is a continuing problem with software vendors, since this is a non-value-added process and eats at the vendor's profit margins.		Develop a contractual set of guidelines for managing the software deliverables.

Support Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Technical Management	Consistent with operational needs	Some inconsistencies	Major inconsistencies	Some inconsistencies – the experience level is low for the size of the project.		Training, recruiting and external advice will provide *startup* support.
Change Management	Responsive to use needs	Acceptable delays	Nonresponsive to user needs	Acceptable to noresponsive – the change management tools do not exist, but the understanding of the consequence do.		Develop a formal change management process for all software components. This would a quality system as well.
Tools and Environment						
Facilities	In place, little change	In place, some modification	Nonexistent or extensive change	Non existent – the support and diagnostics tools are weak.		Develop a complete set of support and diagnostic tools for the distributed environment.
Software Tools	Delivered, certified, sufficient	Some resolvable concerns	Not resolved, certified or sufficient	Not resolved – there is no tradition of software tools.		Develop a *tools* mentality within the Business Unit. Tools form the basis of skills, just as in any other trade – especially woodworking.
Computer Hardware	Compatible with the operational system	Minot incompatibilities	Major incompatibilities	Compatible – the Company standard hardware environment is supported by all possible software vendors		The standardized hardware environment must be maintained. This implies that any vendors that do not run on the standard hardware should be considered noncompliant.
Production Hardware	Sufficient for field operations	Some capacity questions	Insufficient	Sufficient – the Company standard hardware environment is supported by all possible software vendors		The standardized hardware environment must be maintained. This implies that any vendors that do not run on the standard hardware should be considered noncompliant.
Distribution of Software	Controlled and responsive	Minor response concerns	Uncontrolled or noresponsive	Uncontrolled – the current environment does not provide for a standard software distribution environment,		Installation of a software distribution system is a requirement. This can be purchased and installed using standard products.

Support Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Supportability						
Changes	Within projections	Slight deviations	Major deviations			
Operational Interfaces	Defined, controlled	Some *hidden* linkages	Extensive linkages	Extensive linkages – the current interconnections are buried with mainframe applications. Documentation exists, but is held by a few people.		The next generation should be based on an integration specification language (IDL), and make use of metadata tools and models.
Personnel	In place, sufficient experienced	Minor discipline mix concerns	Significant discipline mix concerns	Minor discipline mix concerns – the experience base is primarily in mainframe applications.		Training and personnel additions will be needed to support the C/S environment. If CORBA is added, specific skills will be needed here as well.
Release Cycles	Responsive to user requirements	Minor incompatibilities	Nonresponsive to user needs	Nonresponsive – the concept of release control in the C/S environment is just developing. The examples in Logia will serve as a measurement.		Install release management tools.
Procedures	In place, adequate	Some concerns	Nonexistent, inadequate	Some concerns – the level of formal documentation is low		Install formal methods for documentation and test.

Figure 8 – Support drivers

Schedule Drivers

Schedule risk has a direct effect on cost risk. Like cost risk analysis, schedule risk analysis is deterministic. Both can normally be broken down into components that make up the system's overall cost or schedule.

Schedule Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Resources						
Personnel	Good disciplines mix in place	Some disciplines, not available	Questionable mix and/or availability	Some disciplines are not available – the support and management of the oracle database and the federation processes is currently unavailable	0.6	Recruiting for this position must take place as soon as the funding as secure for the Data Management project.
Facilities	Existent, little or no modifications	Existent, some modifications	Nonexistent, extensive changes	Existent – once the workstation and server test beds are installed the facilities will support the deployment and testing of the data management software	0.3	Complete the installation and setup of the test environment as described above.
Financial	Sufficient budget allocated	Some questionable allocations	Budget allocation in doubt	Some questions – as is always the case, the funding for such a major effort needs careful analysis.	0.5	Incremental planning and funding can be used to address this risk. The project plan should always have stopping points at which the system is useable and no further development is needed to put the system into production.
Need Dates						
Market Driven	Drive by reasonable market demands	Some question about validity of market demand	Unrealistic market demand	Some questions here – the demand for the new system is driven by the production needs and the plan for outside sales revenue.	0.6	Clarification on the manufacturing goals and the impacts of the outside sales goals. Also the plans for any new manufacturing facilities and the impact of those facilities on the PDM and ERP requirements.
Economic Driven	Stable commitments	Some uncertain commitments	Unstable, fluctuating commitments	Some uncertainty – this is a fact of life in the business	0.6	Incremental deployment with clear stopping points.

Schedule Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Competitive Driven	Consistent with the industry	Some pressures for the outside	Heavy pressure to *catch up* with the competition	Some pressures from the outside – the Company is not at the state of the industry. The economic impact of this situation is not clear. More analysis would be needed to determine if the IT systems are a hindrance to growth that would put the business at risk.	0.6	Benchmarking of similar industries will be required to determine the extent of the gap.
Tools Driven	In place, available	Some deliverables in question	Uncertain delivery dates	Some deliverables in question – the tools environment is less than mature.		Provide a test and development environment with software tools o support the integration of the system components
Migration Driven	Nature progression form current system to next generation	Some pressure to move to the new system	Serious gaps between the proposed system and the current system	Serious gaps – the proposed system is at least two generations away from the current system.		Closing these gaps is the purpose of the project. Identifying the gaps and the plans to close them is the purpose of the detailed project plan and the system requirements analysis phases of the project.
Y2K Driven	Y2K issues being managed within industry guidelines	Some question of the impact of Y2K on the project	Serious doubt about this project because of Y2K	Serious doubt – the impact of Y2K is not yet known, so this is a high risk		Confirm the impacts of Y2K on the project.
Technology						
Availability	In place	Some aspects still in development	Totally still in development	Some aspects still in development - the full development and test environment is not yet in place. All that is needed is to purchase the proper equipment and software		This risk can be solved with money and a little time.

Schedule Drivers	Probability of Adverse Effects			Current Situation	Risk Level	Risk Mitigation Steps
	Improbable (0.0 < P <0.4)	Probable (0.4 < P < 0.7)	Frequent (0.7 < P < 1.0)			
Maturity	Application verified	Some applications verified	No application verified	Some applications verified – the risk here is that the target applications for the future system may not support the concept of *federation*.		This verification will be part of the system requirements and vendor selection process.
Experience	Extensive application experience	Some application experience	Little or none	Some applications experience – the concepts of EDM, PDM and ERP are well known in Company. The deployment of C/S systems for these applications is not well known within the Business Unit.		There is experience levels within the Wood Group. These should be used from the beginning to define the scope and aid in the requirements definition for the ERP system.
Requirements						
Definition	Known, baselined	Baseline, some unknowns	Unknown, no baseline	Baselines, some unknowns – the COTS approach still creates risk at the integration level. The functionality can be verified during vendor selection. No vendor is likely to verify their software integrated with another package, possibly a competitor.		The integration verification becomes the responsibility of the system integrator and the system architect.
Stability	Little or no change projected	Controllable changes projected	Rapid or uncontrolled change	Controlled changes – the planning process is at risk if the scope cannot be controlled		Tight control of requirements and use of COTS is a must here.
Complexity	Compatible with existing technology	Some dependency on new technology	Incompatible with existing technology	Some dependency – the use of *federated* systems is new to the Company		Good architecture, test and verification environment and good advice will help with this risk.

Figure 9 – Schedule Drivers

Bad Excuses for Not Doing Risk Management

Failed projects abound. Analyzing them *after the fact* is quite easy. The excuses created from this analysis have been collected by the Software Program Manager Network. Some of these excuses are presented here:

- We have no risks.
- Give us an hour and we'll generate the top ten risks.
- Making the risks public will cause the project to be canceled
- The customer gets mad every time we bring up a potential problem
- We'll deal with the problems when the arise
- This is a development project – why should we worry about the supportability and maintainability risks?
- Our planning horizon is six months out
- We plan to start risk management next year, after we define the process and train everyone.
- The commercial software industry doesn't waste time on risk management
- If I gave a realistic assessment of the situation no one would listen
- That external interface is not our responsibility
- Using that tool is not a risk, the salesman said so
- That method is proven and therefore not a risk, the conference speaker said so
- People outside the projects who don't understand the context will invent a worst-case scenario
- This project is too small to do risk management
- Corporate management won't buy into this concept of risk management
- My technical people will rebel of we identify as a risk a lack of skills needed to do the development
- We have no cost or schedule risk because new technology will increase our productivity – by a factor of five or ten
- We can't identify risks based on industry metrics, because we're different
- Our methodology is the latest state of the art, so we have no schedule risks
- Our method is evolutionary so requirements volatility is not a risk

Bibliography

Managing Risk, Elaine Hall, Addison Wesley.

Software Risk Management, Barry Boehm, IEEE Computer Society.

Managing the Software Process, Watts Humphrey, Addison Wesley.

Continuous Risk Management Guidebook, Software Engineering Institute.

Assessment and Control of Software Risks, Capers Jones.

Software Engineering Risk Management: Finding Your Path Through the Jungle, Dale Karolak, IEEE Computer Society.

"Large Scale Project Management is Risk Management," Robert N. Charette, *IEEE Software*, 13(4), July 1996, pp. 110-117.

Software Development Risk: Opportunity, Not Problem, Roger L. Van Scoy, Software Engineering Institute, CMU/SEI-92-TR-30.