INFORMATION TECHNOLOGY RISK MANAGEMENT

The Concept of Risk, Its Management, and the Benefits to an IT Project

> 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 managing the development or deployment of an Information Technology (IT) system using deterministic, linear, and causal analysis contains several pitfalls. As IT systems grow in complexity, the interaction between their components becomes non–linear and indeterminate, creating many opportunities for failure.

There are two key properties of such systems:

- The presence of significant and credible risk.
- The multiplicity of legitimate perspectives on the functional requirements of the users of the system.

In these systems, risks are created through the interaction between the emerging system capabilities and the business processes it is intended to serve:

- The interaction of the stakeholders with the developing system creates changes in the core system requirements. As the stakeholders come to understand the capabilities of the new system they make new demands of the software.
- The interaction of the system with the business processes creates a new understanding of how the system impacts these business processes. As the stakeholders come to understand the system they make new demands of the business processes. These new requirements in turn impact the system requirements.

These behaviors may appear to describe an unstable set of requirements and in fact they do. They also describe a set of requirements that is driven normal market forces. Any properly run business should be able to change its priorities for a variety of reasons. Increasing understanding of the problem domain is the prime source of these changes. Business environments change. Technology changes over time. Change is a reality of the business world. Without change and the appropriate reaction to change businesses would fail.

> Change before you have to – Jack Welch

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Risk is Created by Failing to Deal with Change

In the context of software engineering, development and software project management, risk can be defined as the possibility of suffering a diminished level of success (loss) within a software–dependent development program. The 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 and the association of risk to the program involves a value judgment as to the potential impact of risk to the outcome. The term loss, danger, hazard, and harm, all of which reflect a negative perception, involve at least a relative assessment of value. [2]

Various definitions of risk state that uncertainty expressed as a probability is involved with risk. Uncertainty involves both the description and measurement of 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. Uncertainty collectively forms the kinematic and dynamic characteristics of the environment as it evolves with time.

The interrelationship of uncertainty and time is evidenced in the probability of the outcome of future events. [4] It is the management of this uncertainty through risk mitigation that is a critical success factor in an IT project.

Three Myths of IT Project Management

IT projects traditionally use formal management processes for the acquisition or development, deployment, and operation that emphasize planning in depth. This approach organizes work into phases seperated by decision points. Supporters of this approach emphasize that changes made early in the project can be less expensive than changes made late in the project.

In the past this approach has been called waterfall. [5] The waterfall approach contains several erroneous assumptions that negatively impact IT projects:

- *Planning* – It is not humanly possible to produce a plan so that its implementation is merely a matter of executing a defined set of tasks.
  - Plans for complex projects rarely turn out to be good enough for this to occur.
  - Unanticipated problems are the norm rather than the exception.
- *Change* – It is not possible to protect against late changes.
  - All businesses face late changing competitive environments.

---

[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.

[3] *Application Strategies for Risk Analysis*, R. N. Charette, McGraw–Hill, 1990.

[4] *Third Wave Project Management*, R. Thomsett, Yourdon Press, 1993.

[5] The term *waterfall* has been used many times as a *strawman* by the agile community. In fact very few pure waterfall projects exist today. This is not to say there are not abuses of the concept of waterfall – sequential development based on the simple algorithm `REPEAT [Design, Code, Test] UNTIL Money = 0`. In practice, development and deployment processes based on incremental and iterative methodologies are the norm. The literature contains numerous references and guidelines to this iterative project management approach dating back to the 1980's.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

- The window of business opportunity opens and closes at the whim of the market, not the direction of the project manager.

- *Stability* – Management usually wants a plan to which it can *commit*. By making this commitment, they give up the ability to take advantage of fortuitous developments in the business and technology environment.
  - In a financial setting this is the *option value* of the decision.
  - Deferring decisions to take advantage of new information and new opportunities is rarely taken into account on IT projects.

Risk Categories

Risk categories can be used to separate the risk of successful software deployment from the risk of deploying the software successfully. This may seem like a trick phrase, but there are several subtleties 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 business operation must also benefit in tangible and measurable ways.

The job of risk management is to identify, address, and eliminate the 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.

Software Specific 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,

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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

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 1 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.

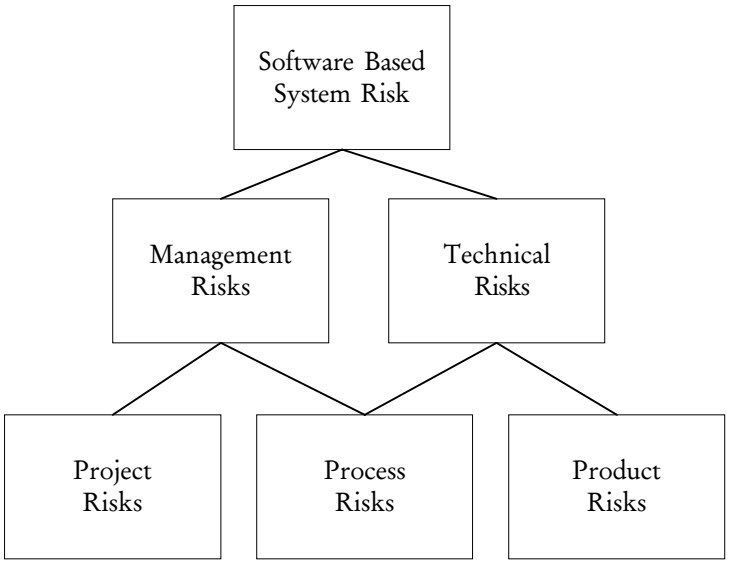

Figure 1 – Software Project Risk Hierarchy

[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.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Software Based Systems Risk Management

Risk management can be defined as 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.
- *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.

What's Next?

Next we will address the risk evaluation criteria, their measures within Information Technology organizations and the steps that can be taken to mitigate these risks.

A Thought

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.

> 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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre–Development Risks

In the development of a risk assessment and mitigation process the categorization of risks is critical. For the purposes of simplicity there are three major categories:

- Pre–development risks – that occur prior to the start of the development phase
- Decision risks – that are made during the development phase
- Post–development risks – that occur after the system has been deployed

This installment presents a set of *Pre–Development* risks that can be used to guide the risk assessment process. These risks should be considered *typical* and act as a guide to developing the actual risks in this phase. These risks will need a detailed assessment tailored to the specific project as well as a mitigation strategy for each risk

Pre–development Risks

There are many risk items associated with the activities prior to the development and deployment of the System. They include:

- Factors related to the size and complexity of the proposed project.
- Factors related to the organizational complexity of the project.
- Factors related to the technology components of the system.

These risks drive the size and complexity 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 is directly related to performance risk. Without knowledgeable estimates of the projects size, predictions of the systems performance are difficult.

Pre–development Size Drivers

Table 1 describes some typical pre–development size risk drivers. This information identifies the risks associated with the scope of the project and the resulting impacts on effort, code size, testing and other size related factors.

Pre–development Size Drivers Probability of Adverse Effects	**Low** (0.0 < P < 0.4)	**Medium** (0.4 < P < 0.7)	**High** (0.7 < P < 1.0)
Number of Departments (other than IT) involved with the system?	1	2	5
Total development manhours for the system?	5 Man Years	10 Man Years	20 Man Years
What is estimated project implementation time?	12 months or less	13 months to 24 months	24 months or more
Information processing breadth – expressed number of programs, size of programs, number of transactions.	10's	50's	100's
Expected frequency of change – the number and/or size of changes that will be made to the initial needs statement	10%	20%	50%
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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre–development Size Drivers	Low	Medium	High
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
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
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
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

Table 1 – Typical Pre–Development Risk Drivers

Pre–development Structure Drivers

Table 2 describes the risk factors that 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 and the risks associated with this deployment. If these risks are not addressed, the effectiveness of the system will be less than planned. The technology associated with the project cannot be used to mitigate these risks. Managerial processes are the mitigation to these types of risks.

Pre–development Structure Drivers Probability of Adverse Effects	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%
What is severity of procedural changes is user department caused by proposed system?	Low	Medium	High
Does user organization have to change structurally to meet requirements of new system?	Minimal	Somewhat	Major
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
How committed is upper–management to this system?	Extremely enthusiastic	Adequate	Somewhat reluctant or unknown
Has a joint information processing / user team been established?	Full–time user representative appointed	Part–time user representative appointed	No
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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre-development Structure Drivers	Low	Medium	High
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 a specific 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.
Technology Maturity. Is the proposed technology mature to the point it can be deployed in an industrial production environment?	Mature	Developing	Coming
Cost Models. Are there cost models available for the deployment of the system and the supporting technology?	Available	Understood but not available	Not understood
Configuration Management. Is the 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
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
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
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
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
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
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
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.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre-development Structure Drivers	Low	Medium	High
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.

Table 2 – Pre-Development Structure Drivers

Pre-Development Technology Drivers

Table 3 describes the risk drivers associated with the underlying technology. This is typically where the risk assessment process focuses, ignoring the items in Table 1 and Table 2. by focusing only on the technology aspects many of the risks associated with business systems and their impact on the organization are missed.

Pre-development Technology Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Distributed processing – is the proposed technology based on well-established technologies with verifiable architecture components?	Yes	Maybe	No
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
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
Algorithm speed and accuracy – how complex are the algorithms and their accuracy for this system? By algorithms this cam mean many different things, but the intent is to evaluate the core processing code and assess its complexity.	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
Security – how secure must the system be? There are many definitions of security, but some type of assessment should be done to quantify the requirements. Almost never can security be added on to the system after it has been constructed. Also the security models of COTS products need to be examined for compatibility.	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
High reliability and fault tolerance – is an elusive terms. Many requirements state high reliability, but a quantitative measure of reliability is needed.	Low need for fault tolerance	Moderate need for fault tolerance	High need for fault tolerance
Reusable software components – how reusable must be software be? Reusability is another elusive attribute.	Low need for reusable components	Moderate need for resusable components	High need for reusable components.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre-development Technology Drivers	Low	Medium	High
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
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
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
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
Adaptability to change – the ease with 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
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
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
Documentation – amount, correctness, type and usability of the documents supporting the system.	Low	Moderate	High
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
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
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
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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Pre-development Technology Drivers	Low	Medium	High
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

Table 3 – Pre–Development Technology Drivers

What's Next

Starting with the pre–development risks factors gives the project manager insight to the behavior of the client and the environment. The technology aspects of the project are but one risk domain.

The decision-making risks will be discussed next. This includes the politics as well as other intangible risk factors.

A Thought

Categorizing risk, gathering the factors for a specific project, creating a set of mitigating factors, and compiling all this into a risk management plan is hard work.

In the end it is the *process* of performing this work that is important.

> Plans are unimportant; planning is essential – D. D. Eisenhower

Decision, Development & Post–Deployment Risks

There are many decision drivers in the risk assessment and management process. These risks are usually intangible when first examined. Without consideration for their impact on the project they will become tangible very soon.

Political Drivers

- Choice of equipment – are the current standards in conflict with the potential vendors offerings? If so who will resolve this conflict?
- Choice of integrator – is there has a competent integrator selected for the selected vendor's products? Is the vendor going to provide the system integration? Is there a potential conflict of interest in this relationship? Who is ultimately responsible for the successful delivery of a working system?
- Schedule and budget – have the schedule and budget have been defined in sufficient detail to determine if there is risk? Who is responsible for defining these numbers? Have these numbers been verified against any industry benchmarks?
- Allocation of responsibilities – who forms the basis of the team? Has the Project Manager role been clarified with all the stakeholder?.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Marketing Drivers

- Gold Plating – is there risk of over specifying the solution? Is there a desire to simplify the decision processes and resulting artifacts?
- Choice of Equipment – are there equipment standards in place that will support the emerging architecture?
- Schedule and Budget – careful scheduling is required protect the risk to have the system too early. Is the Project Manager capable of addressing these schedule pressures? Is there an ultimate authority that the Project Manager can turn to for help?

Solution Drivers versus Problem Drivers

- In–house components – is there a risk of in–house components displacing commercial off the shelf software? 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 – are there internal champions for various software solutions? Do 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, rather than the preconceived needs of the staffing authorities.
- Software reuse – the reuse of existing components is both a risk and a requirement in many environments. Since the legacy systems must typically 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 can be a risk. Many work environments create situations 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 cannot be purchased like a raw material – this is an intellectual process, which takes time, and time means money.*

Development Risks

The development or deployment of software is fraught with risk. The knowledge that this is the case is well known. Addressing risk during the development or deployment phase of a project as a specific item in a project plan is actually rare.

Many of the items identified below would be traditional risk categories for a linear, deterministic, high–ceremony project management method. Agile project management methods have built in facilities to deal with many of the risk identified below.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Requirements Risks

Gathering, classifying, and prioritizing requirements are difficult tasks. Simple questions about the state of the requirements process can reveal risks.

Requirements Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Are the requirements changing or yet to be fully defined?	Requirements are stable	Requirements are changing, but the stakeholders are fully engaged with the requirements elicitation process.	Requirements are changing, but there is no formal mechanism for managing these changes.
Are the requirements clear?	There consensus between the stakeholders and the development or deployment resources on the requirements.	Consensus between the participants is still being developed.	Consensus has been established. There are still open discussions of the requirements.
Is the system envisioned by the requirements feasible?	The participants agree that the system can be constructed with the technology and resources at hand.	There are issues that remain unaddressed.	It is not yet clear how to implement the system.
Are the requirements tracked in some way? Is there an identified resource responsible for this tracking and the resolution of any conflicts or gaps?	There are formal requirements tracking facilities.	There are informal requirements tracking facilities.	There are no formal requirements tracking facilities.
Is the requirement uniquely identified?	Each requirement is identified, placed in some hierarchy, and has a priority.	The requirements are uniquely identified but their relationships are still unclear.	The requirements are still unorganized.
Do the deduced requirements have a valid source and rationale?	Each deduced requirement is traceable to its source.	The deduced requirements are identified, but their sources are not.	The deduced requirements are well defined and are not traceable to their sources.
Does the requirement have a source so it can be traced?	All requirements can be traced in some manner, either through a requirements matrix, a document, or a specific stakeholders communication.	Requirements can be traced to their source, but their interaction is not known.	The source of the requirements is not known.
Does the requirement have a type?	The requirements have unique types which are use for traceability and categorization	The requirements have broad types, but these types are not traceable to the source.	The requirements have not been *typed*.
Is the requirement ambiguous, unclear or vague?	Each requirement is clearly stated and understood by the stakeholders.	Each requirement is defined, but it's understanding requires some effort on the part of the participants.	Some of the requirements need clarification.
Does the requirement adequately address the business goals of the project? *Why is this requirement here?*	The requirement can be traced to a specific business goal.	The requirement is understood but requires some effort to be traced to a business goal	The requirement cannot be traced to a business goal.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Requirements Drivers	Low	Medium	High
Does the requirement conflict with some domain constraint, policies or regulation?	There are not conflicts in the requirements.	There are some conflicts in the requirements, but there are plans to address these conflicts	There are conflicts, but currently there are not plans on how to address them.
Is the requirement related to an organizational or political issue in opposition to the business goal of the system?	There are no requirements that are in conflict with political or organizational issues.	A small number of requirements needs are connected with political of business issues.	It is not understood how many of the requirements have been generted by political of business issues.
Does the requirement take in consideration the needs of all stakeholders?	The requirements have been verified with the stakeholders and a consensus has been reached.	The requirements have been verified with the stakeholders but no consensus has been reached.	The requirements have not been verified with the stakeholders.
Does the requirement need a scenario to be elicited?	The requirement is clearly stated and does not need further development.	The requirement needs further development through a scenario.	It is not clear which requirements need further development.

System Design Risks

The process of design has changed dramatically over the past few years. The introduce of Commercial Off The Shelf (COTS) components, the use of complex CASE tools, high level programming languages that are tightly coupled to the CASE tools, as well as the reuse of design patterns has eliminated many of the traditional design risks. New risks have been put in their place.

System Design Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Functionality – are there specific algorithms that must be defined in order to meet the functional requirements?	There are specific algorithms that satisfy the requirements.	The algorithms that satisfy the requirements are being developed.	The algorithms that satisfy the requirements have yet to be discovered.
Difficulty – does the design depend of assumptions that can be defined and managed within the skills and scope of the project? This is the age-old question: *do we have to invent new physics to solve this problem?*	The design depends on realistic and conservative assumptions	The design depends on assumptions that are still being developed.	The design depends on unrealistic and optimistic assumptions.
Interfaces – are all the interfaces, external and internal known at this time?	Internal and external interfaces are well defined and there is consensus on their acceptance criteria.	Internal and external interfaces are being defined.	Internal and external interfaces have yet to be defined.
Performance & Quality – can these attributes be defined?	There are no expected problems with performance and quality and there is consensus on their acceptance criteria.	The performance and quality problems have been defined, but there is since work to be done on the consensus for their acceptance criteria.	The performance and quality problems are currently unknown.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

System Design Drivers	Low	Medium	High
Testability – have the acceptance tests been defined? Are there clearly defined criteria for accepting the system that are agreed upon by the stakeholders?	The software is easily testable and the acceptance criteria are defined and accepted by the stakeholders.	Some effort will be needed to test the software, but the acceptance criteria are known.	The testability of the software is currently unknown. The acceptance criteria are also unknown.
Hardware constraints	There are no hardware constraints	The hardware constraints have been identified.	The hardware constraints are not yet known.
Software reuse that must be modified to meet the new requirements	There is a low level of re–use software in the system	There is a moderate level of re–use software in the system.	There is a significant level of re–use software in the system.

Integration and Test

The integration and test process creates many opportunities for risk. As the system becomes functional, the stakeholders can now put behaviors in place of specifications. This is an important motivator for incremental and iterative development processes that put the system in the hands of the stakeholders early and often.

Integration & Test Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Environment	There is sufficient hardware and resources to perform integration and testing.	The resources and hardware are still being defined.	Adequate resources and hardware have not been defined.
Product	Acceptance criteria have been defined and agreed upon.	Acceptance criteria are still being developed.	Acceptance criteria have yet to be defined.
System	Sufficient integration and test time has been allocated.	The integration and test time is still being defined.	The integration and test time has not yet been defined.
Maintainability	The product design and documentation are sufficient to maintain the system using another team.	Some original resources will be needed to maintain the system.	The original development team will be required to maintain the system.
Specifications	Test specifications are adequate to verify the system after a maintenance cycle.	Additional work will be needed to verify the system after a maintenance cycle.	The effort needed to verify the system after a maintenance cycle is currently unknown.

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.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Post–Development Requirements Drivers

Requirements or the lack of requirements create many opportunities for risk. After the system h been deployed these requirements continue to impact the risk processes. The existence of a working system does not remove the impact of requirements. Since the requirements evolve as the system is deployed, simply because the understanding of the requirements grows with time, requiremenets need to be re–verified throughout the life cycle of the system development or deployment.

This is an area where disconnects between the stakeholders and the development staff can occur. The typical complaint is *you built what I told you, but not what I actually wanted*.

Requirements Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
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
Hardware Resource Constraints imposed on the system by external forces.	Little or no hardware imposed constraints	Some hardware imposed constraints	Significant hardware imposed constraints
Software Resource Constraints imposed on the project by management, talent availability, local resources, or the business environment.	Little or no software imposed constraints	Some software imposed constraints	Significant software imposed constraints
Technology being used for the development or deployment of the system.	Mature existing, in house experience	Existent, some in house experience	New or new application, little experience
Requirements Stability – how stable are the requirements [8]	Little or no change to established requirements	Some change in baseline expected	Rapidly changing or nor baseline

Personnel Drivers

The personnel assigned to the operational aspects of the system are risk factors. These personnel have similar profiles as the developers in the previous sections. Their impact on the system must be considered in a similar way.

Personnel Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Personnel Availability	In place, little turnover expected	Available, some turnover expected	High turnover, not available
Personnel Mix	Good mix of software disciplines	Some disciplines inappropriately represented	Some disciplines not represented
Personnel Experience	High experience ratio	Average experience ratio	Low experience ratio

[8] The concept of requirements stability is an important topic. In many instances it is not desirable for the requirements to be stable. This may appear to be an *illogical* statement, since all of the traditional project management methods focus on stabilizing the requirements before any further work takes place. In the *real world* of business stable requirements are simply not possible. The quest for this situation cannot be met, since the project manager cannot control the external forces of the business domain. What is needed in a project management method that can deal with changing requirements. This is the basis of all *agile* methods.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Personnel Drivers	Low	Medium	High
Personnel Management Environment	Strong personnel management approach	Good personnel management experience	Weak personnel management experience
Availability	Compatible with need dates	Delivery dates in question	Incompatible with need dates
Modifications	Little or no change	Some change	Extensive change
Languages and API's	Compatible with system and PDSS requirements	Partial compatibility with requirements	Incompatible with requirements
Rights / Licensing	Compatible with PDSS requirements	Partial compatibility with PDSS requirements	Incompatible with PDSS requirements
Certification	Verified performance application compatible	Some application compatible test data available	Unverified little test data available.

Tools and Environment

The presence of absence of tools and facilities for the operation of the system are risk factors.

Tools and Environment Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Facilities	Little or no modifications to existing facilities	Some modifications to existing facilities	Major modifications, nonexistent

Post Development Performance Risks

One way to assess 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 Requirements Drivers

The performance assessments made prior to the development or deployment of the system will now be tested.

Performance Requirements Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Complexity	Simple or easily allocatable	Moderate, can be allocated	Significant or difficult to allocate
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
Stability	Little or no change to established baseline	Some change in baseline expected	Rapidly changing or no baseline
Post Deployment System Support	Agreed to support concept	Roles and missions issues unresolved	No support concept or major unresolved issues

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

System Constraints Drivers

Although they appear to be performance and resource management risks, constraints are usually bound be financial resources. Solving these types of problems is much more difficult than it appears, since the availability of many of these resources can not be controlled by the project manager.

System Constraints Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Computer Resources	Mature, growth capacity within the design constraints	Available, some growth capacity	New development, no growth capacity, inflexible
Personnel	Available, in place, experienced, stable	Available, but not in place, some experience	High turnover, little or no experience, not available
Standards	Appropriately tailored for the application	Some tailoring, all not reviewed for applicability	No tailoring, none applied to the project
Equipment and test capabilities	Meets requirements	May meet requirements, uncertain availability	Incompatible with system requirements, unavailable.
Environment	Little or no impact on the system design	Some impact on the system design	Major impact on the system design
Performance Envelopes	Operation well within boundaries	Occasional operation at boundaries	Continuous operation at boundaries

Technology Drivers

The underlying technology of the deployed solution contains risks that need to be addressed *before* the deployment takes place.

Technology Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Language	Mature, approved HOL used	Approved or non–approved HOL used	Significant use of non–approved HOL
Hardware	Mature, available	Some new products being introduced to the project	New environment with new products and use
Tools	Documented, validated, in place	Available, validated, some new deployment required	New deployed for this project.
Data Rights	Fully compatible with support and follow on	Minor incompatibility with support and follow on	Incompatible with support and follow on
Experience	Greater than 4 years	Less than 4 years	Little or none.

Development Approach

The artifacts of the development process will be carried into production and maintenance.

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Development Approach Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Prototypes and Reuse – are there processes in place to validate the system using some form of prototypes or the reuse of previously successful project components?	Used, documented sufficiently for use	Some use and documentation	No use or documentation
Documentation – for components and technology that will be reused or plan to be reused is there sufficient documentation to reliably deploy these components?	Correct and available documentation.	Some deficiencies, but the documentation is available	The documentation is nonexistent and the source code is not sufficient to reverse engineer the product.
Environment – is there an environment in which the products, developers, and stakeholder can validate their needs?	In place, validated, experience with use of the technology and business processes.	Minor modifications, tools available to maintain the working integrity of the system.	Major development effort will be needed to establish a working system that can be validated with the stakeholder.
Management approach – are there established management processes along with processes for controlling the integrity of the software?	Existing product and process controls.	Product and process controls need enhancement.	Weak or nonexistent management controls.
Integration – are the processes in place to manage the configuration and integration processes of the system?	Internal and external controls in place	Internal and external controls not in place	Weak or nonexistent

Post-Development Support Drivers

The continuing support and enhancement of any complex software system is a risk. The post-development risk factors can be as complex as the development stage factors. In many cases the failure of the project tasks placed early in its life cycle. In other cases it is only after the project is placed into operation that serious risks are discovered.

Post-Development Design Drivers

The design of the system, its architecture, the complexities of the purchased components, databases, and other supporting facilities all have risks associated with them. Many of these risks will not be discovered until the system is deployed.

Design Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Complexity	Structurally maintainable	Certain aspects difficult	Extremely difficult to maintain
Documentation	Adequate	Some deficiencies	Inadequate
Completeness	Extensive PDSS incorporation	Some PDSS incorporation	Little PDSS incorporation
Configuration Management	Sufficient, in place	Some shortfalls	Insufficient
Stability	Little or no change	Moderate, controlled change	Rapid or uncontrolled change

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Design Drivers	Low	Medium	High
Reliability of Commercial Off The Shelf Systems	Stable components	New behaviors discovered after deployment	Errant behaviors discovered after deployment

Post–Development Responsibilities

The business organization needed to support the deployment is a risk. An assessment and mitigation of these risks can only take place at the highest levels of the business.

Development Approach Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Software Management	Defined, assigned responsibilities	Some roles and missions issues	Undefined or unassigned
Hardware Management	Defined, assigned responsibilities	Some roles and missions issues	Undefined or unassigned
Configuration Management	Single point control	Defined control points	Multiple control points
Software Identification	Consistent with support agreements	Some inconsistencies with support agreements	Inconsistent with support agreements
Technical Management	Consistent with operational needs	Some inconsistencies	Major inconsistencies
Change Management	Responsive to use needs	Acceptable delays	Nonresponsive to user needs

Tools and Environment

The operational environment and the tools used to support this environment have risk factors.

Tools and Environment Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Facilities	In place, little change	In place, some modification	Nonexistent or extensive change
Software Tools	Delivered, certified, sufficient	Some resolvable concerns	Not resolved, certified or sufficient
Computer Hardware	Compatible with the operational system	Minot incompatibilities	Major incompatibilities
Production Hardware	Sufficient for field operations	Some capacity questions	Insufficient

Supportability

How the system will be supported and maintained needs to be addressed.

Supportability Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Operational Interfaces	Defined, controlled	Some *hidden* linkages	Extensive linkages

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Supportability Drivers	Low	Medium	High
Personnel	In place, sufficient experienced	Minor discipline mix concerns	Significant discipline mix concerns
Release Cycles	Responsive to user requirements	Minor incompatibilities	Nonresponsive to user needs
Procedures	In place, adequate	Some concerns	Nonexistent, inadequate

Schedule Drivers

The schedule for the development of project is important, but the schedule for the deployment and operation of the system is just ass important.

Resources

The resources needed to deploy and operate the system present risks.

Resource Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Personnel	Good disciplines mix in place	Some disciplines, not available	Questionable mix and/or availability
Facilities	Existent, little or no modifications	Existent, some modifications	Nonexistent, extensive changes
Financial	Sufficient budget allocated	Some questionable allocations	Budget allocation in doubt

Market or Need Dates

The business, commercial, and political pressures on the project to deliver on a specific date creates many risks.

Market Need Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Market Driven	Drive by reasonable market demands	Some question about validity of market demand	Unrealistic market demand
Economic Driven	Stable commitments	Some uncertain commitments	Unstable, fluctuating commitments
Competitive Driven	Consistent with the industry	Some pressures for the outside	Heavy pressure to *catch up* with the competition
Tools Driven	In place, available	Some deliverables in question	Uncertain delivery dates
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

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503

Technology

The underlying technology creates risk factors in the post deployment phase

Technology Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Availability	In place	Some aspects still in development	Totally still in development
Maturity	Application verified	Some applications verified	No application verified
Experience	Extensive application experience	Some application experience	Little or none

Post–Development Requirements

Again the system requirements drive schedule risks in the post–deployment phase

Requirements Drivers Probability of Adverse Effects	Low (0.0 < P < 0.4)	Medium (0.4 < P < 0.7)	High (0.7 < P < 1.0)
Definition of any COTS components. Do these components have the capability to be extended into the future requirements needs?	Known, baselined	Baseline, some unknowns	Unknown, no baseline
Stability of any COTS components, developed components, or legacy components	Little or no change projected	Controllable changes projected	Rapid or uncontrolled change
Complexity of any component.	Compatible with existing technology	Some dependency on new technology	Incompatible with existing technology

A Final Thought

All of the risks identified so far are just *guideposts* for the actual risk assessment and mitigation. This information should be added as three new columns to the tables above.

- Current business situation – a detailed narrative of the current risk situation.
- Calculated risk level for the current situation – a numeric calculation of the risk level
- Risk mitigation steps – the specific steps to be taken to mitigate the risk

> A ship on the beach is a lighthouse to the sea – Dutch Proverb

Copyright 2002, ©, Glen B. Alleman, Niwot Colorado, 80503