1-1-2009

ORE: A framework to measure organizational risk during information systems evolution

Aditya Agrawal
Bond University, aagrawal@student.bond.edu.au

Gavin Finnie
Bond University, Gavin_Finnie@bond.edu.au

Padmanabhan Krishnan
Bond University, Padmanabhan_Krishnan@bond.edu.au

ORE: A Framework to Measure Organizational Risk During Information Systems Evolution

Aditya Agrawal[1], Gavin Finnie[1] and Padmanabhan Krishnan[1]

[1]Bond University, School of Information Technology, {gfinnie,pkrishna}@bond.edu.au

Abstract. Information systems change initiatives often represent the single largest investment (and therefore risk) for large corporations yet there exist few management frameworks in the literature to help decision makers measure organizational risk in a balanced manner during this organization-wide change process. The ORE framework has been developed as a design science artifact based on the Leavitt diamond paradigm as a multi-criteria, relative risk, condition consequence, management decision framework enabling decision makers to calculate and compare risk evolution at fixed points of the change cycle and make structured and balanced risk mitigation decisions. In this paper the principles, architecture and elements of ORE are described.

1 Introduction

Information systems are a vital part of corporate operations, tactics and strategy and are often critical to business competitive advantage (Klaus, Rosemann and Gable 2000). With any change in process or systems there are always associated risks and information system evolution projects are often the largest corporate wide change projects. Hence their failure can seriously impact the continued survival of the corporation (Sumner 2000). 67% of enterprise application initiatives can be considered negative or unsuccessful (Davenport 1998). However software process change is inevitable and should ideally be based on quantitative software measurement programs (Offen and Jeffery 1997). Yet most organizations do not have formal risk assessment methodologies and metrics to help management measure the change in risk as the organization evolves (Dedolph 2003). One of the reasons for lack of management use of formal methods is that few models directly address the ultimate purpose of metrics, which is to provide managers causal support for improved decision making (Fenton, Radlinkski, Neil and Marquez 2007).

There are two approaches to addressing this shortcoming. First approach is to develop a framework where absolute risk and its impact are measured. It is easy to use a framework where absolute risk is measured qualitatively (e.g., three levels such as high, moderate and low). However such a framework has limited applicability as the qualitative risk is always high. One can also measure absolute risk and impact in financial terms but developing accurate absolute risk measures is a time consuming and difficult process. This is because one needs to estimate the events that are risky, the probability of each event actually occurring and the impact (cost) of the event when it eventuates. From a change management or system evolution point of view an early quantitative indicator of potential risk and its evolution is useful (Nogueira 2000). Nair in (Nair 2006) identifies the three main estimation methodologies as Analogy, Top Down and Bottom Up. Analogy uses historical date to estimate current measurements; Bottom Up combines individual component estimations to compute overall estimations while Top Down emphasizes overall estimations and ignores low level individual components estimations.

The Project Management Institute's (PMI) Guide to the Project Management Body of Knowledge (Project Management Institute 2000) defines risk management as the systematic process of identifying, analyzing, and responding to project risk with the aim of minimizing the probability and consequences of adverse events to project objectives." Curtis (Curtis 1989) explains how software systems needed to be analyzed in their broader organizational context. Technical metrics such as Function Points, Halstead's measures and McCabe metrics for cyclomatic complexity etc. were developed to measure and analyze different artifacts in the software development process. However risk management of individual software development/evolution projects are only a small part of the overall risk in enterprise wide systems evolution (Keene 1981). Chang (Chang 2004) use an open-ended Delphi type survey to capture two main categories of information system implementation issues as lack of incorporation of organizational context and reluctance to accept dissenting views. Enterprise wide risk evaluation therefore requires both individual project specific factors and organizational wide factors to avoid change which is a technical success but an organizational failure. The Leavitt Diamond (Leavitt 1965) provides a balanced organizing paradigm for consideration of all major organizational aspects during corporate change. It identifies the main dimensions as Task, Technology, Structure and People. For technology based organizational change Sawy (Sawy 2001) proposes a translation of these dimensions as business processes, organizational form, information technology usage and human resources.

For technical risk Higuera and Haimes (Higuera and Haimes 1996) describe a taxonomy-based approach to risk identification. Possible risks are categorized into Product Engineering, Development Environment and Program Constraints. Each category contains a list of factors and an associated questionnaire questions as guidance for risk elicitation. The taxonomy based approach provides an empirical referential structure enabling the risk assessment team to quickly and comprehensively survey and choose the risks (and weight them) most relevant to their organizational domain. The software development, evolution and deployment phenomenon itself is considered as an n-dimensional space where each dimension is a relevant risk causing element to the successful completion of the project (Gluch 1994). At any point in time the state of the system is thus expressed as a point in space with values relating to each risk dimension. This enables management to assess whether the state is desirable/undesirable and how the state might change and take corrective measures as necessary. This methodology emphasizes a relative risk, condition consequence approach to risk identification and measurement (away from root cause analysis) enabling decision making oriented measurement.

The approaches described provide important guidance in developing a risk measurement framework. The Organization Risk Evaluation (ORE) framework has been developed to provide a quantitative measure of risk and support development of a risk scale calibrated to the organizational specifics which allows management to compare risk evolution versus system evolution and make strategic risk mitigation decisions. Based on PMI guidelines ORE has been designed to support a structured approach to risk measurement that helps to manage change complexity and can be used throughout the evolution lifecycle. All three estimation methodologies have been reflected in the ORE architecture. It supports an organizational condition consequence approach by viewing the organizational system risk at a point in time due to contributions by metrics that are part of all four Leavitt organizational dimensions and by emphasizing cause and effect relationships in assessing risk evolution. Balance and measurement of all organizational dimensions are considered within the metrics and architecture of ORE, combining judgment and measurement.

Two paradigms characterize much of IS research, behavioral science and design science. In design science knowledge and understanding of the problem domain and its solution are achieved in the building and application of the designed artifact. IS research is therefore concerned with the development of behavioral theories and design artifacts to add to the knowledge base, and application studies to the environment to test and refine the knowledge base (Hevner, March, Park and Ram 2004). A new information system is a complex political and social game (Keene 1981) where the context is critical and it is difficult to isolate the impact of a small number of variables due to highly complex interactions. Hence interpretative research methods (Skok and Legge 2001) are relevant. They need to be combined with traditional software engineering risk identification and measurement techniques to enable measurement of organizational risk.

ORE has been developed as a design science model whose purpose is to enable risk measurement during information systems evolution. It is a multi-criteria, relative risk, condition consequence framework that can be applied at comparable times in various aspects of information system development, deployment and change. The time granularity of framework application is user-definable, allowing multiple applications of ORE during the same system evolutionary phase, which is flexibility missing in most risk assessment methodologies.

This paper is organized as follows. Section 2 provides an overview of the ORE architecture and its organizational decision making model. ORE is formally defined in Section 3 and the constituent elements are described in Section 4. Finally Section 5 concludes the paper and describes further avenues of work.

2 ORE Framework Overview

The ORE framework has a multi-level architectural design based on a set of core principles and a hierarchical organizational decision making model that enables effective use in management decision making. ORE supports both buy and build approaches to information systems evolution. Figure 1 illustrates the framework architecture.

The typical levels of decision making in an organization can be divided as operations, tactics and strategy (Emery and Trist 1965). ORE provides support for all three levels of decision-making. Based on the principle of divide and conquer, a corporate wide evolution project would be broken into management wrappers named technical sub-projects. Using the principle of separation of concerns (Torsten and Guido 1997) three levels of management would look after the operational, tactical and strategic dimensions of the evolution respectively.

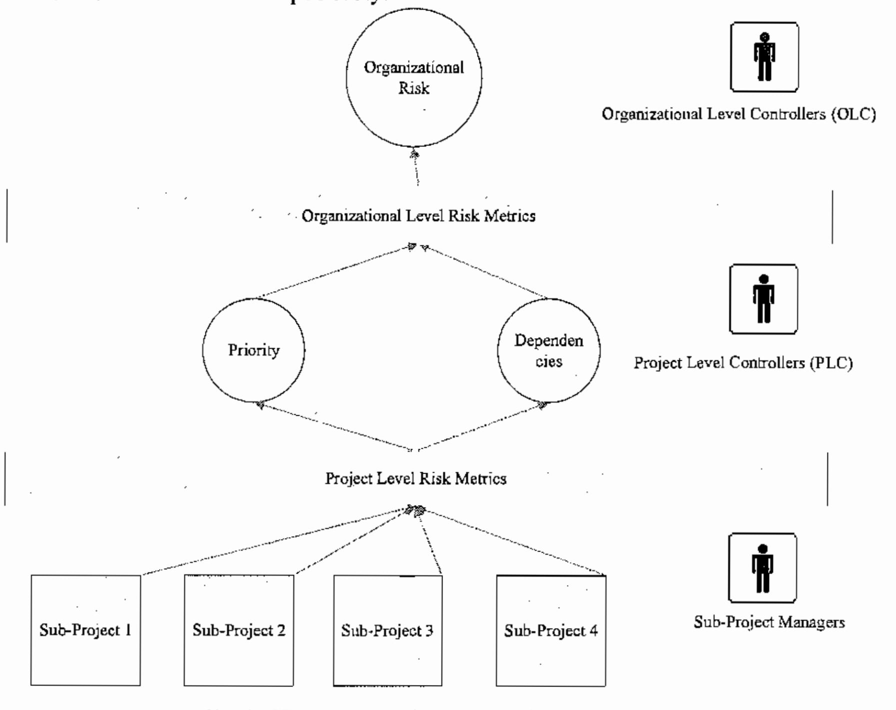

Fig. 1: ORE Framework Conceptual Design

The technical sub-project managers would use ORE sub-project factors (and associated metrics) to measure operational concerns across all organizational dimensions using a balanced Leavitt diamond based approach. At the tactical level the PLC (the Project Level Controllers) would use the risk assessments from all sub-projects to assess the progress of the corporate project. In a typical organization the PLC would be the business heads of all the functional departments affected by the change. The PLC would make tactical resource allocations decisions and smooth communication flows between and within project teams. Their judgment of progress would flow to the OLC (Organizational Level Controller) who typically would be a team of executives. Based on their understanding of business objectives the OLC would set and revise project dependencies and priorities and establish the strategic direction of the systems change.

The ORE methodology is based on the paradigm of fixed time sampling in a cyclic process as illustrated in Fig. 2. ORE is designed to support the constant change (Volker and Rohde 1995) by allowing organizational decision makers to measure risk at certain comparable points in time such as the inception phase of each change cycle (this aspect is defined formally in Section 3). More importantly, ORE allows application multiple times within the samc evolutionary phase. Hence the time granularity of ORE application is flexible. ORE timestamps each risk assessment and uses them to build a self-referential risk scale that is fully customized to the organizational specifics and can be credibly used to measure and compare organizational risk at later comparable times. The results of the analysis can be used to make risk mitigation decisions thus providing structure to an evolving and changing world.

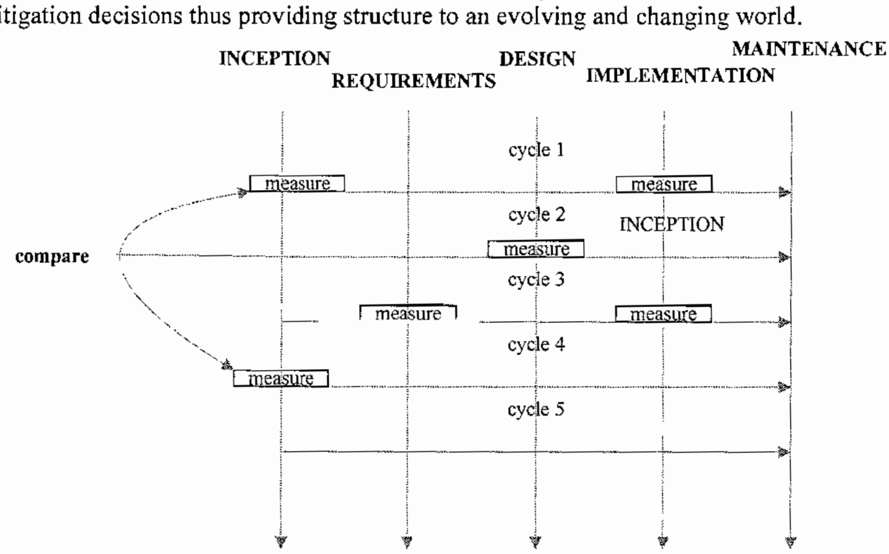

Fig. 2: ORE Allows Fixed Time Sampling in a Cyclic Process

3 Formal Definitions of ORE

The design science framework defined in (Hevner et. al. 2004) and the representational theory of measurcment (Roberts 1979) is used to define the ORE framework as a design science artifact and facilitate real world comprehension of the measures developed.

This work produces a viable model (ORE) which provides utility in understanding and measuring organizational risk during information systems evolution. The process described in the paper essentially involves building a framework (the artifact) based on past research using a measurement theoretic definition and providing heuristic guidance for its application and validation.

The ORE methodology of risk computation involves identifying a set of factors F with members $F_1.....F_n$, assuming fixed weights $\omega_1.......\omega_n$ for each factor. For each factor F_i identify a set of metrics M_i consisting of metrics $M_{i_1}....M_{i_n}$. At any given time t, only a subset of F may be relevant, say F^t. Further the contribution to a factor $F_i \in F^t$ at time t may only be a subset M_i, say M_i^t. The risk value due to factor F_i at time t can be denoted as $\sum_{M_i \in M_i^t} M_i(t)$. Thus organizational risk value at time t is $\sum_{F_i \in F^t} \omega_i * F_i(t)$. Two organizational risk measures at times t_1 and t_2 are comparable if $F^{t_1} = F^{t_2}$, and for every $F_i \in F^{t_1}$, $M_i^{t_1} = M_i^{t_2}$, that is the set of factors and metrics are the same.

Complete measurement theoretical and design science formal definitions of ORE as well as proposed framework validations are described in the ORE Technical Report which is available at http://shakti.it.bond.edu.au/~sand/csa0702.pdf. They are omitted from here due to space limitations. Design however is both a process and a product. It describes the world as acted upon (artifact) and as sensed (process). Hence we have included the formal methodology definition to enable essential comprehension of the ORE design.

4 ORE Components: Sub-Project Risk Assessment

A sub-project is a management wrapper with specific technical objectives. ORE defines a balanced set of factors to measure technical, process, organizational form and people issues during sub-project evolution which are described in the following subsections. Different subsets of these factors or metrics may be chosen for specific sub-projects based on whether a buy or build approach to information systems has been undertaken, organizational specifics etc. In the following subsections each factor is described using the notation: $\text{Factor} = \sum_{i \in I} f_i * w_i$

The contribution of each factor is calculated as a weighted sum of the various metrics for that factor indicated by an index set I. Only the elements of I, viz. the set of metrics is stated.

4.1 Technology Change

The use of new technologies in a company has been observed to be an important and recurring problem in the industry (Tatikonda and Rosenthal 2000). Measuring the effect of technology change requires consideration of the amount of technological change in the product (or service), the amount of process change this entails, and the interdependence and integration of the new technologies with existing technical infrastructures. All these factors are weighted and combined in an appropriate manner based on the domain to provide a holistic measure of the risk due to technological change. Technology change metrics are: Product Technology Novelty, Process Technology Novelty and Technology Interdependence.

4.2 Size Change

Incorrectly sized projects (with respect to implications on cost, effort and time projections) add risk to the change process (Ropponen and Lyytinen 2000). Size is measured by using the Function Points metric (Albrecht 1979) which is suitable for management information system projects (Nair 2006).

4.3 Requirements Change

Requirements change metrics are the Birth rate (BR) and the Death rate (DR). Studies have shown that the early parts of the system development cycle such as requirements and design speeifications are especially prone to error due to changing business requirements. The birth rate and death rate metrics provide a way to calculate the effect of new requirements during evolution (BR) and the changes in previous requirements (DR). A typical function can be $\mathrm{INT}(\frac{BR+DR}{10})$ (Nogueira 2000), where INT( ) takes a numeric argument and converts it to the nearest integer.

4.4 Personnel Change

Loss of key personnel during software evolution often triggers loss of key knowledge and experience (Bennett and Rajlich 2000). Staff turnover and productivity trends during a project are important indicators of change management success and indicate new staffing requirements, training and learning requirements on the new system etc. all of which importantly affect project risk. The first metric is $\frac{\mathrm{Direct}}{\mathrm{Idle}}$ which represents the productivity ratio and stands for the direct time spent on the project all the personnel involved divided by the idle time they spend. This ratio has been found useful in differentiating high productivity scenarios from lower productivity ones (Nogueira 2000). Values between 2 and 6 represent high productivity scenarios while values less than 2 represent low productivity scenarios. The second metric is $\Delta\mathrm{Personnel}$ which represents the personnel turnover during the last evolution cycle.

4.5 Parallelism

Software evolution with a high degree of parallelism might allow several tasks to be carried out concurrently. However it also represents a greater management overhead. On the other hand less parallelism leads to the weakest link issue due to sequential dependencies (Capital Broadleaf International Pty. Ltd. 1999). A balanced approach to scheduling activities based on the specifics of the environment is the middle path towards risk mitigation. Therefore parallelism is measured using the ratio $\omega_{EC} \dfrac{\text{Concurrent Phases}}{\text{SequentialPhases}}$ where ω_{EC} is the environment constant.

ω_{EC} is based on the suitability of the environment for concurrent/sequential work. The ratio will have a value of 1.0 if the environment does not impose any advantage for either scheduling mechanism. For example it will seek to increase risk if the environment is better suited for sequential work and there are more concurrent phases.

4.6 Ranked Metrics

The following project sub-metrics are designed to elicit subjective expert assessments regarding people and political based risk criteria for IS projects based on the recommendations of the Information Systems Audit and Control Association (Information Systems Audit and Control Association 2002).

Development Platform: The sub-project manager makes a judgment about the risk due to development platform ranked on a five point scale based on whether the development platform is proprietary or open source, the length and quality of the platform's track record, scalability of the development platform (Fenton 1998) and support and constant improvement provided for the platform.

This metric is expected to be of utility in build based information system evolution projects where modules or the entire system are being custom designed and coded on single or multiple development platforms.

Manpower Outsourcing: Manpower outsourcing is an important part of modern corporate strategy and this metric incorporates this business reality into the risk assessment and decision making process. It is expected to be of utility in build based information systems evolution projects. The sub-project manager makes a judgment on a five point scale based on the percentage of manpower outsourcing relative to the total manpower required for the project, overseas or local outsourcing and the number of manpower suppliers and their track record.

Project Team: The importance of team cohesion and leadership is difficult to overestimate in a project and the criteria are designed to elicit the project manager's assessment of project team leadership and inter-team communieation and cohesion. Each team is ranked on a scale of 10 based on size, organization and experience of team and the project management experience of team leader.

4.7 Default Sub-Project Factor Weights

ORE assigns a default set of weights to all sub-project factors based on empirical results collected and organized by the software risk program at Carnegie Mellon University (CMU SRP) (Higuera et. al. 1996) described in Section 1. The CMU SRP organizes risk causing dimensions into three main classes (with associated subclasses), Product Engineering, Development Environment, and Program Constraints. All the main classes have approximately equal weighting. Sub-project factors are assigned to appropriate classes. Therefore Technology Change and Size Change are assigned to Product Engineering (subclass Design). Requirements Change to Product Engineering (subclass Product Engineering). Personnel Change to Program Constraints (Resources). Parallelism has been assigned to Development Environment (Management Process). Development Platform is assigned to Development Environment (Development System), Manpower Outsourcing to Program Constraints (Resources) and Project Team to Development Environment (Work Environment).

Factors from all main classes are represented in ORE; however some subclasses are not used. Subclasses in use are scaled to ensure they equal total risk for their class while maintaining their weightings between each other within each main class. The overall sub-project risk equation may be expressed as,

ρ (subproject) = 0.1(Technology Change) + 0.1(Size Change) + 0.2(Requirements Change) + 0.37(Personnel Change) + 0.18(Parallelism) + 0.09(Development Platform) + 0.37(Manpower Outsourcing) + 0.06(Project Team) (1)

4.8 Project Priority and Dependency

Project priorities (0-10) are set by the Organizational Level Controller (OLC) based on considerations of sub-project scope, budget and political issues. Priorities are used to weight the sub-projects risk contribution to organizational risk and enable structured consideration of what-if scenarios and resource allocations.

The dependency measure helps describe cause and effect relationships between sub-projects. ORE looks at project dependency relationships as graphs and by default propagates the entire parent project risk (weighted by priority) to the children. For example if projects B, C and D are dependent on A then each of their risks equals the sum of their own risks and their dependency risks (i.e. A's risk). Once a project is complete its risk becomes zero; hence the graph self-adjusts the risk values as evolution progresses. Additionally not all projects may be related to one another in this simple way. These dependency relationships and the effect of dependency can be modified as per the organizational specifics.

4.9 Overall Organizational Risk Equation

Drawing from Eq. 1**Error! Reference source not found.** and priority and dependency models the default organizational risk equation may be mathematically expressed as

$$\rho(\text{organizational}) = \sum_{i=1}^{n} (\rho_i * \sigma_i + \delta_i) \tag{2}$$

Where ρ_i stands for sub-project risk of project *i*, σ_i stands for priority for sub-project *i*, and δ_i stands for $\sum_{j \in \{\rho_{i_1} \ldots \rho_{i_n}\}}^{n} \rho_j * \sigma_j$ where ρ_{i_x} represents the *xth* project on which ρ_i depends and σ_j represents the priority of ρ_j. There are *n* sub-projects in the organizational information systems evolution.

4.10 Tool Support and Case Study

As per design science guideline 5, Research Communication (Hevner et. al. 2004), to demonstrate framework application for a management audience we have developed a detailed hypothetical case study application of ORE. A preliminary version of the ORE framework based tool also has been developed for testing and development purposes. A copy of the case study and the tool is available at http://shakti.it.bond.edu.au/~sand/projects.htm. Due to space limitations they have not been further described in this paper. Please address any comments to {gfinnie,pkrishna}@bond.edu.au.

5 Conclusions and Directions for Further Work

The ORE framework is a design science artifact to measure organizational risk during information systems evolution. Organizational risk is considered a composite of the risk caused by different organizational elements based on the Leavitt's Diamond paradigm. The framework develops its own scale of reference by timestamping the organizational risk assessments conducted. ORE enables fixed time sampling of risk during cyclic organizational change and maintenance processes and is based on the paradigms of balance, cycle and time. Balance between prescription and generality is part of design by providing default functional configurations and weightings to all framework metrics and models with the flexibility to modify them to fit domain specifics. ORE generates sub-project risk measures for project managers, collated sub-project risk for Project Level Controllers and a composite organizational risk measure including sub-project risk, priority and dependencies for the Organizational Level Controller. ORE therefore provides a scalable and sufficiently abstract structure to measure and manage the complexity and risk of organizational change during information systems evolution. ORE also attempts to enforce decision making discipline by incorporating numbers and judgment within the methodology.

There are several avenues for further work on ORE. The framework is generic and needs refinement for different application domains. ORE is currently being applied in various corporate projects and we plan to document usage experience in forthcoming application papers. Application and documentation in a variety of business scenarios will allow the framework to mature and guidelines for the framework elements to develop aiding in its application in different domains. Communication flows are an important indicator of organizational success of the information systems change process and can be measured through artifacts such as number of emails and chat messages in a given period of time. These can be visualized as a Kiviat graph and ratio of balance can be scaled and included into the organizational risk calculation (Bruegge and Dutoit 1998). The organizational risk could be broken up into front office and back office components to allow decision making to focus on lessening front office risk at the cost of increasing back office risk and deploying resources as necessary (Evangelidis 2003). The Analytic Hierarchy process (Saaty 1980) or Multi-Attribute Utility theory (Keeney and Raiffa 1993) can be used to enhance the effectiveness and structuredness of the weighting process. The framework could be integrated into standards such as the IEEE Standard for Quality Assurance Plans (IEEE Standards Board 1998) used to help in building and assessing secure systems.

References

Albrecht, A.J. (1979) Measuring Application Development Productivity. Paper presented to Proceeding of IBM Application Development Joint SHARE/GUIDE Symposium, pp.83-92, Monterey, CA.

Bennett, K. and Rajlich, V. (2000). Software Maintenance and Evolution: A Roadmap. Paper presented to Conference on Future of Software Engineering, Association for Compnting Machinery, pp. 75-87, Limerick, Ireland.

Bruegge, B. and Dutoit, H.A. (1998). Communications Metrics for Software Development. IEEE Transactions on Software Engineering, 24(8): pp. 615-628.

Capital International Pty. Ltd. Broadleaf (2005). @Risk 4.0 Tutorial Notes: Quantitative Risk Modeling. Retrieved on 15th September 2005 from www.broadleaf.com.au/tutorials/Tut%20Starting.pdf.

Chang, S.I. (2004). ERP Lifecycle Implementation, Management and Support: Implications for Practice and Research. Paper presented to Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii, USA.

Curtis, B. (1989).Three Problems Overcome with Behavioral Models of the Software Development Process. Paper presented to Proceedings of the Eleventh International Conference on Software Engineering, IEEE, pp. 389-399, Pittsburgh, USA.

Davenport, T.H. (1998). Putting the Enterprise into the Enterprise System. Harvard Business Review, pp. 121-31.

Dedolph, F.M. (2003). The Neglected Management Activity: Software Risk Management. Bell Labs Technical Journal, 8(3), pp. 91-95.

Emery, F.E. and Trist, T.L. (1965). The Causal Texture of Organizational Elements. Human Relations, 18(1), pp. 21-32.

Evangelidis, A. (2003). FRAMES-A Risk Assessment Framework for E-Services. Electronic Journal of E-Government, retrieved on 12th October 2006 from www.ejeg.com, 2(1), pp. 21-30.

Fenton, N.E. (1998). Software Metrics, A Rigorous and Practical Approach, Revised Edition. Course Technology.

Fenton, N.E., Radlinkski, L., Neil, M. and Marquez, D. (2007). Improved Decision-Making for Software Managers Using Bayesian Networks. Submitted to The 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, Drubovnik, Croatia, September 3-7.

Gluch, D.P. (1994). A Construct for Describing Software Development Risks. Technical Report CMU/SEI-94-TR-14, Software Engineering Institute, http://www.sei.cmu.edu.au, Carnegie Mellon University.

Hevner, A.R., Mareh S.T., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), pp. 75-105.

Higuera, R.P. and Haimes, Y.Y. (1996). Software Risk Management. Technical Report CMU/SEI-96-TR-012 ESC-TR-96-0I2, Software Engineering Institute, http://www.sei.cmu.edu, Carnegie Mellon University.

IEEE Standards Board (1998). IEEE Standard for Software Quality Assurance Plans. Standard published by the Software Engineering Standards Committee of the IEEE Computer Society, USA.

Keene, P.G.W. (1981). Information Systems and Organizational Change. Communications of the ACM, 24(1), pp. 24-33.

Keeney, R. and Raiffa, H. (1993). Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Revised Edition. Cambridge University Press.

Klaus, H., Rosemann, M. and Gable, G.G. (2000). What is ERP? Information Systems Frontiers, 2(2), pp. 141-62.

Leavitt H. J. (1965). Applied Organizational Change in Industry: Structural, Technological and Humanistic Approaches. In James G. March, editor, Handbook of Organizations, pp. 1144–1170, Rand MeNally and Company, Chicago USA.

Nair, M. (2006). A Survey of Software Estimation Techniques and Project Planning Practices. Paper presented to Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE, Las Vegas USA.

Nogueira, J. (2000). A Risk Assessment Model for Evolutionary Software Projects, PhD Thesis submitted to the Naval Postgraduate School USA.

Offen, R.J. and Jeffery, R. (1997). Establishing Software Measurement Programs. IEEE Software, 14(2), pp. 45-53.

Project Management Institute (2000). A Guide to the Project Management Body of Knowledge (PMBOK Guide). Technical Report, Project Management Institute of USA Inc.

Roberts, F.S. (1979). Measurement Theory with Applications to Decision Making, Utility, and the Social Sciences. Addison Wesley.

Ropponen, J. and Lyytinen, K. (2000). Components of Software Development Risk: How to Address Them? A Project Manager Survey. IEEE Transactions on Software Engineering, 26(2), pp. 98-111.

Saaty, T.L. (1980). The Analytic Hierarchy Process. McGraw Hill, NY USA.

Sawy, O.A.E., (2001). Redesigning Enterprise Processes for E-Business. Mc-Graw-Hill Irwin, Singapore.

Skok, W. and Legge, M. (2001). Evaluating Enterprise Resource Planning Systems Using an Interpretive Approach. Paper presented to ACM SIGPCR, San Diego, CA, USA.

Information Systems Audit and Control Association (2002). IS Auditing Procedure P1: IS Risk Assessment Measurement. Standard published by the Information Systems Audit and Control Association.

Sumner, M. (2000). Risk Factors in Enterprise Wide Information Management System Projects. In Proceedings of the ACM SIGCPR conference on Computer Personnel Research, pp. 180 – 187, Chicago, USA.

Tatikonda, M. and Rosenthal, S. (2000). Technology Novelty, Project Complexity and Product Development Project Execution Success. A Deeper Look at Task Uncertainty in Product Innovation. IEEE Transactions on Engineering Management, 47(1), pp. 74-87.

Torsten P. G.R., Guido T. (1997). Separation of Powers and Public Accountability. The Quarterly Journal of Economics, 112(4), pp. 1163-202.

Volker W., Rohde M., (1995). Towards an Integrated Organization and Technology Development. Paper presented to Proceedings of the Conference on Designing Interactive Systems, Processes, Practices, Methods, and Techniques. Ann Arbor, USA.