Article

Comparative study of risk administration in centralized and distributed software development atmosphere

Riaz Shah, **F. Bahadur**, **Sheraz Ahmed**, **Faiza Kanwal**
Department of Information Technology, Hazara University, Mansehra, Pakistan
E-mail: riazshah15654@gmail.com, msosfaisal@gmail.com, sheraz.cse15669@yahoo.com, ms_cs41@yahoo.com

Received 8 August 2016; *Accepted 15 September 2016*; *Published 1 December 2016*

Abstract
Risk administration is used to increase the possibility of success of any future project by exploring its reservations. It will meet all the remedies to make the software development project successful by keeping in view all the future problems that may occur during the project process. It includes the identification of risk and their assessment in the project course and tries to make improvement to make project constructive. Risk administration goals are to overcome project task risks those are identified before starting of the project and during the implementation. This paper describes the phases in the risk administration process and provided methods to analysis and safety of administration. The paper focuses on a study risk administration in centralized and distributed software development projects. This study recognizes valuable, constant and free communication as the basics for victorious risk administration. Therefore, it registers all incoming information memorize much in the same pattern as the “black box” device during an aircraft flight. The description and evaluation tools are also included, may be used during the risk administration study in the software development atmosphere.

Keywords risk administration; software development; centralized and distributed atmosphere; risk administration; software risk; risk model.

Network Pharmacology
ISSN 2415-1084
URL: http://www.iaees.org/publications/journals/np/online-version.asp
RSS: http://www.iaees.org/publications/journals/np/rss.xml
E-mail: networkpharmacology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Risk administration is not new tool and a lot of standards and it is an essential component of good administration and decision making at all levels of an organization. At present, a further general standard on risk administration is in preparation as a common ISO/IEC standard (Sahibudin et al., 2008) describing a systemic top down as well as a functional bottom up approach. This standard is intended to support existing industry or sector specific standards.

1.1 Software risk

This development along with application linked with software exposes the community to several reservations. Initial, the breakdown of any software challenge to be a business starting brings about money along with time spend and an unseen income opportunity. Move up the risk associated with like breakdown is referred to as it challenge risk. An additional risk relates to the security from the residents and the atmosphere. Fading of any software technique can lead to a car disaster that, from the most harmful situation, could guide to loosing individual lifestyle. Here is the software security risk.

Regardless of the improvement within technologies, it jobs yet confront exactly the same troubles because 30 in years past (Kerr et al., 1995). However, certain requirements from the customers usually are not deeply realized, that brings about continual growth from the technique range or maybe within sexual rejection from the remaining technique. This contribution of people is actually nonstop incorporating the element connected with individual brain along with character on the techie problems from the jobs. Finally, it is error flat, the cooperation one of many challenge clients is repeatedly weak. Because of this, the anticipations from the buyer usually are not satisfied.

Entirely, it calls for a number of main upgrades on the software growth along with exchange process. Certainly one of like important solutions recognized by all of the software engineering along with challenge operations guidebooks (Turner and Jain, 2002) may be the hazard operations.

1.2 Risk administration

Risk administration is used to increase the possibility of success of any future project by exploring its reservations. It includes the analysis of possible risk in the project course and the alleviation of their negative potential. Boehm (Keil et al., 1998) argues in his research that by reducing risk in the project it will lead to reduce around 40% of software costs. Risk analysis is a project wise approach for the detection of software development project risk. It is commonly measured that better risk evaluation occupy good communication on risk and suitable documentation that may be collected on the foundation of experiences and project risk knowledge which help to avoid the risks.

1.3 Notational conventions

Some models included in this paper are built with the Unified Modeling Language (UML) from Object Administration Group. The requirement of UML notation and some direction on the practical use of UML may be found in Booch et al. (1997), Fensel et al. (2003), Azam et al. (2014) correspondingly.

2 Review of Literature

Quality and achievement of a research is often a indication of the time and endeavor invested in developing research ideas and concepts. The immediate ambition of a literature survey is to establish whether the idea is worth pursuing or not.

2.1 Software development project risk

A simple definition of project risk states that it is a problem that has not however occurred but which could cause loss to one’s project if it did (Wiegers, 1998). The idea of risk is linked with a number of human endeavors ranging from space investigation and company achievement to information systems improvement (Barki et al., 1993).

Experiential studies on how managers deal with risks show that the managers are not necessarily normal in reacting to risks. They seem at a risky choice as one that contains a hazard of a very poor performance (March and Shapira, 1987*)*. Also, risk is not a probability concept; it deals with the magnitude of the bad outcome. Accordingly, managers act in a loss-aversive manner rather than a rational manner as predicted by the traditional theory. The extensive literature review resulted in the identification of over 100 risk factors. The next step was to attempt to group like factors together in order to get a clear picture of the general types of software project risk factors. This resulted in the formation of 12 general types of software project risk categories.

Team related factors

- Effectiveness of task communication
- Project manager characteristics
- Organizational atmosphere and support
- External factors
- Role of the user
- Formalization of project charter
- Project assessment and scheduling
- Tools and technology
- Requirement permanence and correctness
- Effectiveness of project monitoring
- Cross cultural and sex issues

2.2 Risk administration practices

Risk administration is concerned with a phased and orderly approach to evaluate and control the risks occurring in a specific context. Software project risk administration is risk administration applied to the development and/or deployment of software exhaustive systems. A typical risk administration framework involves implementing and monitoring measures to reduce risk. Project risk administration encompasses both hard skills such as estimating and scheduling tasks, and soft skills, which include inspiring and managing team members (Kirsch et al., 2002).

In addition, risk administration approaches characteristic a catalog of risk resolution techniques. These are derivative from local causal theories on how risky incidents affect software development and how interventions affect development trajectories. A thorough review of literature on risk administration strategies for software projects, helped to categorize a range of risk resolutions techniques which are discussed below following categories:

- Headship strategies
- HR policies
- Training
- Project synchronization
- User synchronization
- Requirement administration
- Evaluation techniques
- Suitable methodology
- Project control

2.3 Check lists on software project risk and risk administration

One of the most general methods for identifying the existence of risk factors and risk administration strategies in a particular project are the checklists.

One of the revolutionary studies in this regard is the top 10 risk list of Boehm (1988). His list has been compiled by inquiring several large software projects and their general risks and is thus empirically beached. One of the most quoted worldwide studies on software project risk factors was conducted by Carstensen and Schmidt (1999). In an endeavor to reimburse for some of the previous shortcomings in checklists of risk factors, Schmidt et al. (1996) conducted a survey of project managers and built-up an general list of risk factors in software development. The particular research was conducted by three concurrent Delphi surveys in three dissimilar settings: Hong Kong, Finland and the America. In each country, a group of project managers was fashioned and a “ranking-type” Delphi review was used to solicit risk items from the group.

2.4 Review of studies on project relating risk, administration and its result

The studies referred above consider software risks along numerous dimensions and have provided some empirically founded insights of typical software risks and risk administration strategies to alleviate them. Overall, these studies provide insights into risk administration consideration, but are weak in elucidation the true impact of risk and risk administration practices on the project outcome. A few studies have gone additional to create how risk administration efforts decrease the exposure to software risk and can thereby enlarge software quality and advance software development.

A number of system recital criteria have been developed and empirically tested. Saarinen (Saarinen, 1990) planned a system achievement calculate with four dimensions: system development process, system use, system quality, and organizational impacts. Process outcome measures refer to the “successfulness” of the development process of the project. The focus is on finishing the project within budget, within schedule and the on the overall quality of the development process. Both aspects are important as the software delivered by the project may be of high quality but the project itself may have exceeded the time and cost projections. On the other hand, well managed projects which come in below cost and time budgets may convey poor products.

3 Centralized Software Development and Risk Administration

Today’s software development has inspired away from the “single team single location single administration framework” paradigm to distributed, collaborating teams with flexible administration relations. In addition, modern experience with complex projects has exposed that older development practices, with completely particular requirements and sign offs and totally prearranged interfaces between key components, have substantial problems and are particularly vulnerable both to agenda pressure and to unpredicted changes and events. Finally, financial factor have encouraged inter organizational development practices such as outsourcing and off-shoring.

For these reasons, less centralized approaches to development have been pursued.

In multi-organizational development, participating teams work for different organizations. Multi-organizational development can be either: Contractual, with one central authority (either one of the developer organizations or, less repeatedly, a customer) and other teams functioning on particular components with watchfully specified predefined inter-faces and behavior, or accommodating, with teams functioning on sub-systems or low joined components with iteratively specified interfaces and performance, often without a clear, generally accepted central authority for resolving differences and conflicts. Both distributed development (Beranek et al., 2005) and Centralized software development (Deek and McHugh, 2003) introduce a number of new risk administration concerns and change or intensify others.

Centralized software development entails a widespread change in the software engineering practices, from business case and product visualization through development processes to administration policies. Cooperation and communication concerns are considerably different, not only in level but also in kind. Software progress requires a common product visualization and architecture, widespread idea and design exchange, continuous communication, and dynamic use of session, approval, and agreement constrained only by thinker property, solitude, and safety considerations.

3.1 Principles of centralized risk administration

Victorious association requires collaboration-aware administration, intra- and inter-organizationally. This entails collaboration-aware risk administration, which is an extension of traditional risk administration as well as team based risk administration (Higuera et al., 1994a, b).

In the continuing application of the risk administration process to large software development programs, the most theatrical effect has been in opening the communication channels for dialogues within organizations relating to risk and risk administration. In addition to the common benefits of a rigorous approach to risk administration, mutual risk administration may itself be an significant early step in establishing trust and managing cultural and language problems. Cultural familiarity and trust have constantly been recognized between the top four important success factors in cooperation (Powell et al., 2004).

3.2 A framework for effective risk administration for CSD: A layered approach

An efficient risk administration sketch should be based on Centralized-risk administration principles and should give clear description of decisions, actions, and farm duties related to the risk administration functions defined in a collaboration-aware risk administration sketch must:

- Tackle customary intra-organization risk recognition and administration in collaborating agencies.
- Handle risks recognized as introduced or intensify by CSD, including risks within a single organization, ensuing from interfaces; factors have communication, and collaboration.
- Handle Centralized risks not well managed intra directorially.
- Drive incremental alteration of policies, processes, and activities as needed.
- Support negotiation to resolve conflicts and to allocate tasks for risk administration.

Three substitute strategies for collaborative risk administration include:

- Assign duty for administration of Centralized risks in customized, person organization risk administration plans.
- Grip new risks in a monumental risk administration plan.
- Follow a coated policy.

3.3 Three critical risk factors: trust, culture and communication

Booming recognition, classification, and assessment of risk factors that arise in the joint software development domain are key challenges to software projects. Even though the mainstream of the customary risk factors apply to CSD and some additional factors have been recognized in the literature, there is a further need to systematically recognize, differentiate, and classify them and to support their efficient treatment in RMMM plans for large-scale, high risk Centralized development. Differences in culture are principally a risk source (an origin for problems), whereas trust is principally a risk driver (a demonstration of an existing problem). Communication can be a source (e.g., mistranslation of requirements) or a driver (manifesting lack of administration sup-port), or both. Each of these three factors is described in some detail below.

4 Risk Administration For Distributed Atmosphere

The significance of risk administration has been well recognized by the project administration population. In risk administration is listed among nine key knowledge areas related to project administration. In relative to software project risks, much work has been done at Software Engineering Institute (SEI) (Galagher, 1999; Higuera et al., 1994b; Jones, 1994; McConnell, 1993; Miler and Górski, 2001).

Software projects are bare to different risks and risk administration in such projects is still insufficient as is shown by the proportion of failed, tardy or too expensive projects (ACT Insurance Authority, 2004; Ahmad and Ehsan, 2013; Khan et al., 2014). The aim of a project is to bring in time and within the financial plan constraints, a product that meets stakeholders' needs and hope. The critical factors of the project achievement are the quality, the time and the financial plan. Present software projects are often face growing and altering client demands and are put under schedule pressure. The systems are increasing in size and become gradually more complex. To condense the development time, the systems are built out of reused (but often not reusable) components.

The thought of having a continually open and extremely existing channel for communicating and memorizing risk-related information is shown in Fig. 1. As the project advances, risks can be recognized either during planned project actions or casually, e.g. when people talk to each other at mealtime, journey or during their free time. The thought of risk black box comes from the reality that memorizing this risk related information should be successful and as complete as possible (much like it is done during the aircraft flight). The difference to the aircraft black box is that we desire to use this information with the practical attitude, although we do not keep out its use for observation (e.g. to examine the risk history after the project success/failure).

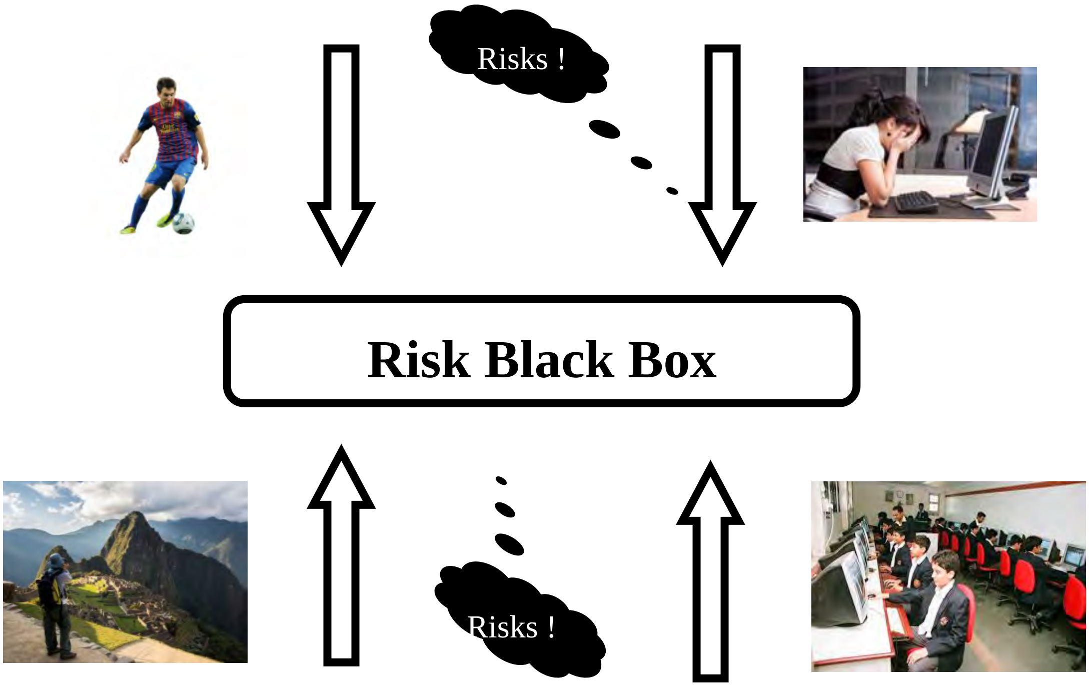

Fig. 1 Constantly open risk memorizing channel.

5 Risk Model in Centralized and Distributed Software Atmosphere

A risk circumstances in the context of a given process can be articulated in more detail by investigating the ingredient sub-processes of the appropriate process (the super-process). The domestic error circulation within a given process can be mapped to the outdoor error circulation between that process' sub-processes. An error in a sub-process is also an error in its super-process. When an error in a sub-process causes this sub-process to fail, the failure remains internal to the super-process, unless that sub-process' outdoor state is part of the super-process' outside state (i.e. the sub-process delivers part of the super-process' service). In the opposite case, the error reaches the super-process' service border and leads to the process failure. The internal structure of a risk circumstances mapped to a process' sub-processes is shown in Fig. 2.

6 Risk Analysis in Centralized and Distributed Software Development

The new techniques of risk investigation planned for the Process Model-based Risk evaluation method. The techniques cover the preliminary processing of the information on recognized risk with risk snapshots, relative grade of risks, and the investigation of the overall process risk. The planned techniques are described in detail in the following sections, preceded by a short overview in the introduction.

Risk analysis aims at provided that the judgment makers with the information on which of the recognized risks should be mitigated and which could be established as well as which risks to alleviate first. To accomplish this intention, the risk examination needs some indicators that permit differentiating the recognized risk scenarios according to the level of posed risk. Two new risk indicators are planned:

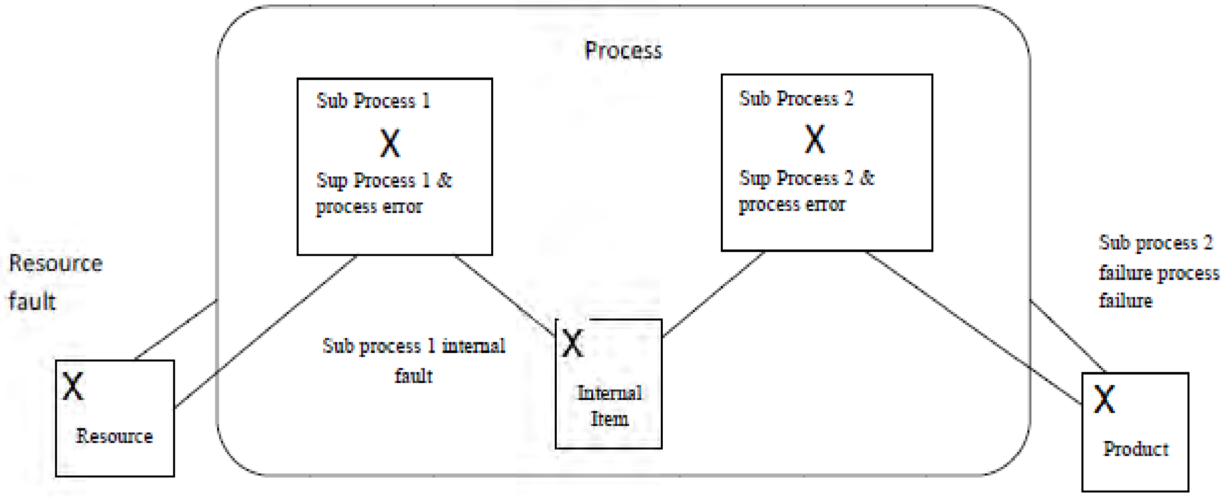
Process
Sub Process 1
X
Sup Process 1 & process error
Sub Process 2
X
Sup Process 2 & process error
Resource fault
X
Resource
Sub process 1 internal fault
X
Internal Item
Sub process 2 failure process failure
X
Product

Fig. 2 Inner structure of a risk scenario within a process with sub-processes.

Risk indication – the information on how the risk was recognized in the risk recognition phase, Risk ranking – the ranking points clearly assigned to the analyzed risk scenarios by the invited participants of a risk examination session. The next sections detail the concept of the risk snapshot, clarify risk ranking with the planned indicators as well as converse the assessment of the overall process risk.

Generally process risk is defined as the universal level of risk present in the whole process. It is very complex (if not impracticable) to approximate the overall process risk precisely. It is planned that the overall risk connected with the demanding classes of replica elements is used as an indicator towards the approximation of the overall process risk. In the subsequent sections, the overall risk metrics for the classes of replica elements and the indicator of the overall process risk are defined.

Overall risk of activities – RA. Let RA indicate the overall risk connected with the behavior in a process model. RA is estimated as the risk of action R (A) summed up for all actions of the model, as given by equation 1.

$$R\,A = \sum R\,(A),\ R\,A \in R+ \cup \{0\} \qquad (1)$$

7 Conversation and Results

The range of the planned method covers all the activities concerned in the risk evaluation:

- Risk recognition
- Risk examination
- Risk citations
- Risk communication

The risk evaluation process defined within the method follows the key ideology of risk administration indicated in the literature:

- Team contribution
- Nonstop process
- Open communication with supplies for information security
- Learning from experience

8 Conclusions and Future Work

Many approaches have been previously proposed under a common flag of the risk administration to enhance the projects' possibility of victory. However, the indication shows that there is still a large space between what we presently have in arms beside the project risk and what we would desire to have. The analysis of the methods for the risk evaluation seems mainly valuable. Application of systematic support to risk recognition and examination (through clear software process modeling and devoted techniques) with keen software tools provides for early detection of project risks and increases the efficiency of risk alleviation.

In this paper, we have provided a complete, if beginning, approach to joint risk administration. We tinted the differences between conventional and collaborative software development (CSD) that involves numerous organizational units and recognized risk administration values for CSD that enlarge conventional and team-based risk classifications. On the foundation of previous literature and our personal field study, we then present a structure for CSD risk administration and a layered approach for its execution. Practitioners can use these thoughts to build up an valuable risk administration plan for their particular kind of joint software atmosphere. Finally, an indicator of the overall process risk was planned to evaluate the shared level of risk from the process activities, artifacts and roles. This indicator further allows for the higher process analyses such as the imitation of risk declaration by process improvements or the judgment of process' risk broadmindedness. We emphasized the important role of communication in the risk administration process and planned a thought of a risk "black box" memorizing all the risk-related information arising in the project. We notable three hierarchical layers of risk evaluation and explained how they interrelate.

Finally, we presented a process of permanent risk evaluation captivating benefit from all the above ideas. The proposed method may be further enhanced and comprehensive in the areas like new risk patterns associated to other classes of risk events, new metrics of process model construction providing more information on process risk, wider range of tool support throughout further expansion of the Risk Guide tool.

References

ACT Insurance Authority. 2004. Risk Management Toolkit. ACT, USA

Ahmad S, Ehsan B. 2013. The cloud computing security secure user authentication technique (multi-level authentication). International Journal of Scientific and Engineering Research, 4(12): 2166-2171

Azam F, et al. 2014. Framework of software cost estimation by using object orientated design approach. International Journal of Scientific & Technology Research, 3(8): 97-100

Barki H, Rivard S, Talbot J. 1993. Toward an assessment of software development risk. Journal of Management Information Systems, 10(2): 203-225

Beranek PM, Broder J, Romano N, Reinig B. 2005. Management of virtual project teams: Guidelines for team leaders. Communications of the Association for Information Systems, 16(10): 247-259

Boehm BW. 1988. A spiral model of software development and enhancement. Computer, 21(5): 61-72

Booch G, Rumbaugh J, Jacobson I. 1997. The Unified Modeling Language for Object-Oriented Development, Documentation Set Version 1.0. Addison-Wesley, MA, USA

Carstensen PH, Schmidt K. 1999. Computer supported cooperative work: New challenges to systems design. In: Handbook of Human Factors (Itoh K, ed). Tokyo, Japan

Deek FP, McHugh J. 2003. Computer-supported Collaboration with Applications to Software Development. Kluwer Academic Publishers, USA

Fensel D, Motta E, van Harmelen F, et al. 2003. The unified problem-solving method development language UPML. Knowledge and Information Systems, 5(1): 83-131

Galagher BP. 1999. Software Acquisition Risk Management Key Process Area (KPA). A Guidebook Version 1.02. SEI Report CMU/SEI-99-HB-001. Carnegie Mellon University, Pittsburgh, USA

Higuera RP, Haimes YY. 1996. Software Risk Management. SEI Report CMU/SEI--96-TR-012, Carnegie Mellon University, Pittsburgh, USA

Higuera RP, Dorofee AJ, Walker JA, Williams RC. 1994a. Team Risk Management: A New Model For Customer-Supplier Relationship, Special Report CMU/SEI-94-SR-5. http://www.sei.cmu.edu/publications/documents/94.reports/94.sr.005.html

Higuera RP, Gluch DP, Dorofee AJ, Murphy RL, Walker JA, Williams RC. 1994b. An Introduction to Team Risk Management. SEI Report CMU/SEI--94-SR-01. Carnegie Mellon University, Pittsburgh, USA

Jones C. 1994. Assessment and Control of Software Risks. Prentice Hall, USA

Keil M, Cule PE, Lyytinen K, et al. 1998. A framework for identifying software project risks. Communications of the ACM, 41(11): 76-83

Kerr EA, Hays RD, Mitchinson A, et al. 1995. Managed care and capitation in California: how do physicians at financial risk control their own utilization? Annals of Internal Medicine, 123(7): 500-504

Khan K, Qadri S, Shabir Ahmad S, et al. 2014. Evaluation of PMI’s risk management framework and major causes of software development failure in software industry. International Journal of Scientific & Technology Research, 3(11): 2014

Kirsch LJ, Sambamurthy V, Dong-Gil K, et al. 2002. Controlling information systems development projects: The view from the client. Management Science, 48(4): 484-498

March JG, Shapira Z.1987. Managerial perspectives on risk and risk taking. Management Science, 33(11): 1404-1418

McConnell S. 1993. Code Complete. Microsoft Press, USA

Miler J, Górski J. 2001. Implementing risk management in software projects. In: Proceedings of the 3rd National Software Engineering Conference. Warsaw, Poland

Powell A, Piccoli G, Ives B. 2004. Virtual teams: A review of current literature and directions for future research. ACM's DATA BASE for Advances in Information Systems, 35(1): 6-36

Saarinen T. 1990. System development methodology and project success: an assessment of situational approaches. Information and Management, 19(3): 183-193

Sahibudin S, Sharifi M, Ayat M. 2008. Combining ITIL, COBIT and ISO/IEC 27002 in order to design a comprehensive IT framework in organizations. In: Modeling and Simulation 2008. AICMS 8 Second Asia International Conference. 749-753, IEEE

Turner R, Jain S. 2002. Agile meets CMMI: Culture clash or common cause? Extreme Programming and Agile Methods—XP/Agile Universe 2002. 153-165, Springer, Berlin, Heidelberg

Wiegers K. 1998. Know your enemy: software risk management. Software Development San Francisco, 6: 38-44