

CorMod: A CAUSAL MAPPING APPROACH TO IDENTIFYING PROJECT DEVELOPMENT RISK

Abdullah J. Al-Shehab, School of Computing, Mathematical and Information Sciences
University of Brighton, Watts Building, Moulsecoomb, Brighton, BN2 4GJ, UK
a.a.shehab@brighton.ac.uk

Robert T. Hughes, School of Computing, Mathematical and Information Sciences
University of Brighton, Watts Building, Moulsecoomb, Brighton, BN2 4GJ, UK
r.t.hughes@brighton.ac.uk

Graham Winstanley, School of Computing, Mathematical and Information Sciences
University of Brighton, Watts Building, Moulsecoomb, Brighton, BN2 4GJ, UK
g.winstanley@brighton.ac.uk

Abstract

Software development is widely acknowledged to be a process prone to failure. We argue that there is thus a need for improved project risk management frameworks and tools. Further, a coherent, concise and accessible risk model can be useful to project development team members in capturing disparate perceptions of project risk and providing a common basis for its analysis, prediction and mitigation. This paper proposes a core model that forms a reasoned foundation for the management process, and which can be extended on a project-to-project basis. This 'Core-Model' is based on cognitive and causal maps. The Core-Model has been applied experimentally to two case studies: a) the evaluation of the performance of a completed project; and b) an attempt at predicting the outcome of a new and ongoing project. The use of causal mapping techniques resulted in the added value of intuitive and interactive visualization of the interrelations between project components and of the propagation of risk through chains of cause and effect.

Keywords: Software Development, Cognitive and Causal Maps, Risk Management.

1 INFORMATION SYSTEM PROJECT FAILURE

In 1985, world-wide software costs were reportedly 140 billion euro, and that figure had risen to 450 billion euro in 1995 (McManus 2004). McManus has also suggested that with a growth rate of 12 per cent per year, the projected expenditure for the year 2005 could be estimated to be 1400 billion euro of which 30% would have been spent on failed projects.

A major reason for continued difficulties is the rapid introduction of new technologies and the extension of Information System/Information Technology (IS/IT) to new application areas. For example, according to Watson & Haley (1998), 50% of data warehousing projects typically fail.

This paper addresses the issues of software development risk management approaches and how we can improve the situation through a simple, coherent and high level model that has the ability to visualise clearly and at an early stage the predicted path of a software development project through the identification and analysis of project risk factors.

2 RISK AND RISK MANAGEMENT/EXISTING APPROACHES

One textbook definition of risk management is “the identification of the hazards and possible problems, the evaluation of their importance and the drawing up of plans to monitor and deal with those problems” (Hughes & Cotterell 2002, p.134). Thus good risk analysis should be able to help reduce the likelihood of project failures.

Risk identification paradigms tend to have a similar content (see, for example, IEEE 2001). They usually involve comparing the characteristics of the proposed project against a checklist of possible risks. Risk taxonomies are structured in several different ways. Some list risks according to the stage of development in which they occur, for example McManus (2004), and Dorofee et al (1996). Some people rank risk according to their importance, for example Schmidt et al (2001), Boehm (1991), Keil et al (2002). Others categorise risk according to the environment they exist in, e.g. project, organisation and market environments (Charette 1997). Euromethod (1996) divides risk into business and project domains. Other sources list risk randomly, for example Karolak (1996) and Smith (2001).

In most cases, however, dependencies among these risks or potential relationships between them are not explicit. For example, unskilled developers, incomplete software product and schedule over-run might be listed separately in taxonomy, but there could be (causal) relationships between these factors. Identifying the consequences of risk is as important as identifying the risk itself. One of the few frameworks that identified consequences was Euromethod. The Euromethod (EM) was a major initiative of the European Union, first published in 1996 (Turner and Jenkins 1996), aiming to provide an organized approach to the planning and development of information systems in the European Market for both the customer and supplier.

EM is richer than other risk taxonomies because it represents risk in term of consequences, relations and dependencies. EM representation is based on a textual description of risks and their consequences. It was however found that it was possible to represent EM as a causal map (see figure 1).

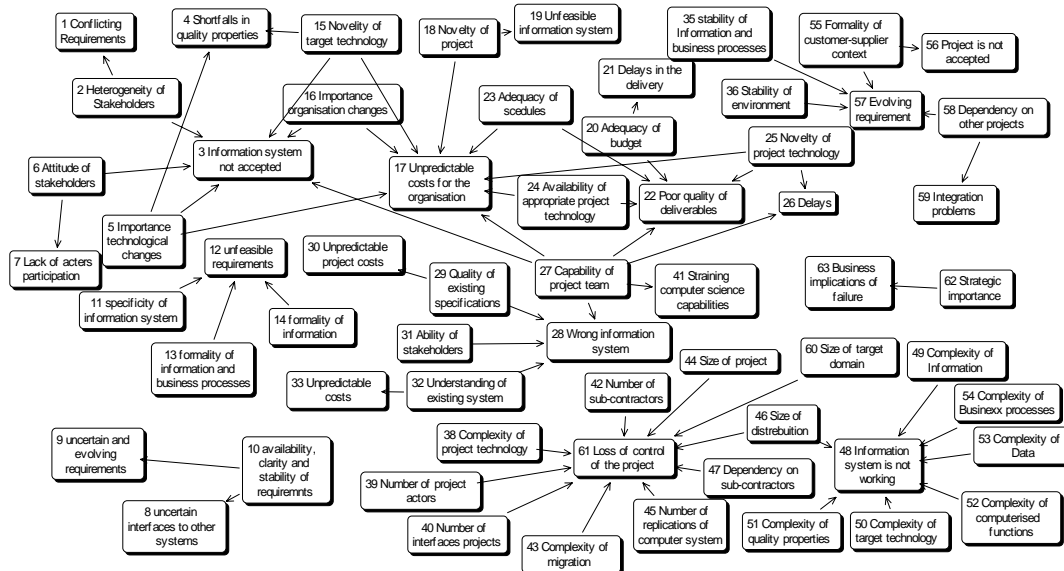


Figure 1 Euromethod diagram.

Visualizing EM in this way has several interesting emerging properties. From the diagram it was noticed that some risks are more ‘sensitive’ than others, that is, they have more possible contributing factors. For example, ‘wrong information system’ is affected by four different risks. Another property is the ‘potency’ of risk, that is the degree to which the risk influences other factors. For example,

'capability of project team' affects six other factors. These observations led us to consider visual representation techniques, in particular cognitive and causal maps that could lead to improved risk identification and analysis. Although there are several ways of representing risk (Scavarda, 2004), we have chosen Causal Maps because of their inherent ability to describe causality explicitly.

3 COGNITIVE AND CAUSAL MAPS FOR MODEL DEVELOPMENT.

Cognitive and Causal Maps (CCM) have been applied in many disciplines, but in IS/IT they are in their early stages (Narayanan and Armstrong 2005). A causal map, of which Figure 1 is an example, is a network diagram representing causes and effects (Bryson et al 2004). The diagram contains two basic elements: concepts which are the nodes in the network and causal relationships which are represented by the arcs between the nodes. Concepts are considered as the variables of the system and the causal relations (or 'links' or 'edges'), carry either a positive or negative sign, implying the type of causal relationship and effect (Tsadiras and Margaritis 1997). A causal map can represent how a system works, and in the context of project risk identification and analysis, a development project can be seen as a particular type of system. Causal maps can also act as a teaching tool (Bryson et al 2004 and Scavarda et al 2004). Huff (1990) suggests that an advantage of causal maps is that they can portray information about a system more succinctly than a corresponding textual description. A cognitive map is a causal map that represents peoples' beliefs about a situation.

Using CCM in risk identification appeared to have some potential advantages, including facilitating group participation, enhancing communication between project stakeholders, providing a clear 'picture' of the current state of the project and the immediate impact of interrelationships and dependencies through visualization. The following sections describe how the practicality and effectiveness of applying CCM techniques to project risk analysis were explored.

4 EXPERIMENTAL PROCEDURES

After an initial exploratory pilot study using a group of post graduate students knowledgeable in IS/IT project management to refine the experimental procedures (Al-Shehab et al 2005a), research focussed on a major project based in Kuwait that was considered a failure. This problematic project, started in 1998-1999, had already raised many failure issues at the beginning of 2000, and had suffered from various setbacks. At one point the project was stopped for a period of time, and many stakeholders thought that it had failed and been abandoned. However it was reinitiated and went through much revision of the project design and management approaches.

For experimental purposes, the project team members were divided into two groups; managers (5 participants) and technical staff (8 participants). One of the objectives at this stage was to produce two combined maps, one for managers and one for staff. It was considered that having both groups in one group session was undesirable, as the managers might have an undue influence on staff opinion in an open session. The final two combined maps represented the perceptions of how the project failed. Initially, CCMs were drawn individually by project team members, guided by a facilitator. Individual maps were then combined to produce a single map for each of the two groups using the following protocol:

- Similar factors identified by different participants were merged.
- No other factors were deleted and all factors were transcribed to a single map
- All the causal links between factors were transcribed to the single map
- The combined map was then reviewed and modified by all the individuals meeting together.

A large amount of data was collected, and was analysed using Decision Explorer™ (<http://www.banxia.com/demain.html>) and have been reported in Al-Shehab et al. (2005b).

Observations included:

- Visualisation of the factors influencing the outcome of the project as a whole was facilitated

- Identification of interrelations among the project concepts was readily achieved
- It was possible to focus on a simple diagram for documenting a project

CCMs appeared to be very effective in documenting past software projects clearly and with visual richness. Bartlett (2002) has argued that one of the best ways of presenting risk information is visual 'risk modelling', i.e. presenting a graphical interpretation of risk. This suggested the possibility of using CCMs to predict IS/IT development project behaviour at an early stage of the project

5 DEVELOPING RISK MANAGEMENT BASE TOOL

A survey was conducted which identified several tools that help project managers to some extent in risk management. Some of these tools relate to estimating time and budget uncertainty issues. An example of this is @RISK® (Davey 2000), which uses Monte Carlo Simulation. Other tools relate to managing, tracking, prioritizing and reporting risks during the execution of a project. For example, Risk Radar® which also has the ability of depositing risk information into a repository for future projects (see www.spmn.com). Others are simply drawing tools that provide the ability to create diagrams, for example Decision Explorer™. Our intention was to develop a tool that could be a 'starting point' for project managers and team members in risk identification and analysis.

The results of work so far, focusing on the post-evaluation or post-mortem approach, gave us the idea of developing a base project model, represented in the form of a causal map, and describing the initial state of a new software project. It seemed likely that the analysis of individual past projects using CCMs could produce individual causal maps which differed considerably, and would make the creation of new predictive maps difficult. A possible solution could be the identification of a subset of fundamental factors that would be applicable to a range of projects. It was recognised that this 'core model' might need to be supplemented by further factors that would reflect the particular environment in which a specific project was to be conducted.

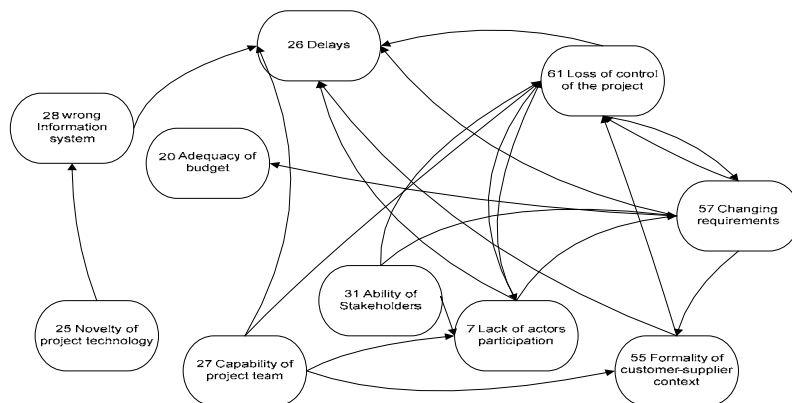


Figure 2. Managers and Staff combined similar concepts.

The starting point for creating the causal core model (CorMod) was the identification of similar concepts existing in the two combined maps. This led to contention where similar but arguably distinct factors could be grouped under some collective title, for example, 'changing requirements' and 'new requirements'. This led in turn to a clarification of concept titles, e.g. 'wrong information system' was changed to 'quality of delivered system'. The preliminary diagram of the combined managers group and staff group is shown in figure 2.

One problem with merging maps related to differences in the terminology used by different individuals and groups when referring to what could be in fact the same concept. This was largely resolved by adopting the terminology used by Euromethod (EM) and translating the local terms into

EM terms. The result of this procedure was a set of concepts identified by both developers and their managers and also recognised by EM – see Figure 3.

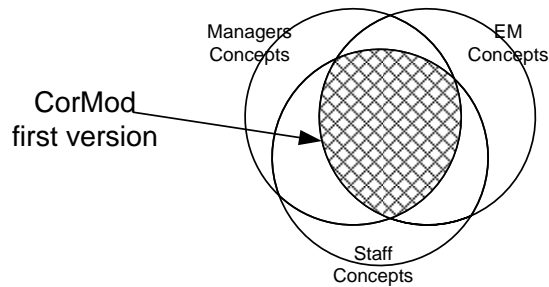


Figure 3. Combining EM, Managers and Staff diagrams.

This experiment produced a list of 10 concepts that were similar in all of the three diagrams as follows:

- Capability of project team: skills, experience, communication, understanding the environment, understanding the IS requirements.
- Ability of users: clarity of requirements, experience in IS project development.
- Project technology (HW, SW): maturity of technology, availability of technology.
- Changing requirements: how much the requirement is changing, evolving.
- Control of the project: monitoring and tracking the progress against target.
- Quality of delivered information system: correctness, completeness, up to user expectation.
- User participation: the amount of involvement in the project.
- Budget adequacy: the amount of budget toward the completion of the project.
- Supplier and customer relation: the level of partnership between the supplier who is developing the project and the customer who is the stakeholder and the end user.
- IS deployment according to schedule: project deployment ahead or behind schedule.

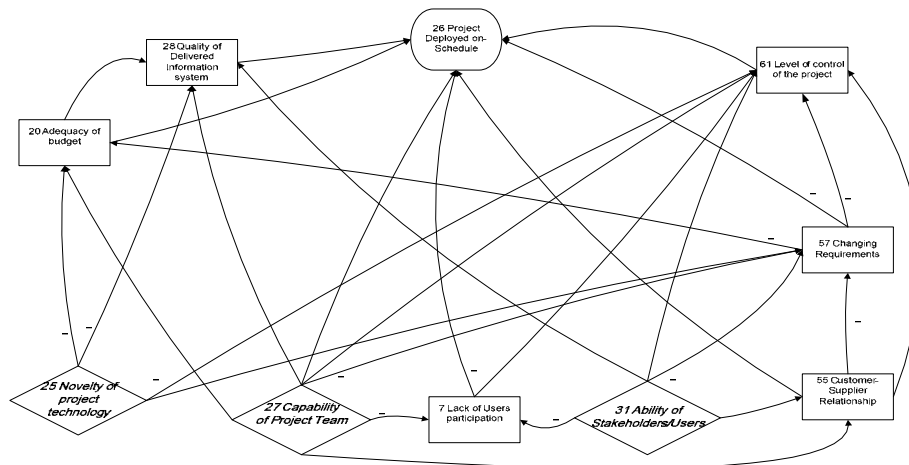


Figure 4 CorMod diagram.

Subsequent to identifying the concepts, the links were transferred to CorMod according to their existence in one of the three diagrams (Manager, staff or EM), resulting in the CorMod diagram of Figure 4. In this figure, diamond shape represents a ‘tail’ concept/variable, i.e. there are no links entering the concept but there are exiting links. Each rectangle represents a concept/variable, where the links enter and exit. The one oval shape in the model of Figure 4 is a ‘head’ concept/variable, where links only enter.

The tail concepts are considered to be the ‘initiators’ of CorMod. As in any software development project, the project begins with the existence of these concepts, namely the relative novelty of project technology, the capability of the project team, and the ability of the stakeholders/users.

6 EVALUATION AND DEVELOPMENT OF CorMod

The resulting CorMod has some theoretical attractions, particularly in relation to work on innovation and the learning curve. Rogers (1983) argues that while innovation in the longer term can reduce uncertainty by introducing improvements to products, tools and processes, in the shorter term it increases uncertainty. The process of overcoming this uncertainty is through learning both by the staff developing applications and those who will eventually be using them. This process is well-documented as the ‘learning curve’.

Evaluation has so far been of two kinds. The first has been of the explanatory power of the model when applied to the known outcomes of previous projects. Effective use of the model would also depend on perceived ease of use and the second type of evaluation has examined this.

CorMod was initially tested on scenarios from three pre-existing case studies. Two of these case studies were of projects classified as failures and the third was a success. The CorMod outcome closely reflected the documented events of each project. These results were encouraging, especially the potential for using CorMod as a prediction tool.

Data was then gathered via questionnaires from over forty different IS/IT projects from 12 different organizations in the fields of telecommunications, health, education, social security, aviation, military, oil sector and the software industry, and this is being used as the basis for ongoing evaluation of the CorMod.

The second type of evaluation involved the application of the framework at a software company in Kuwait. The company has been in the business of IS/IT for the past 18 years with many international partners, and was awarded a major software development contract recently to develop a web based internet/intranet system. The objective of the experiment was to conduct a workshop, comprising all of the project team, and apply CorMod at an early stage in the project. This would enable us to investigate the applicability of the process and the outcome of the model.

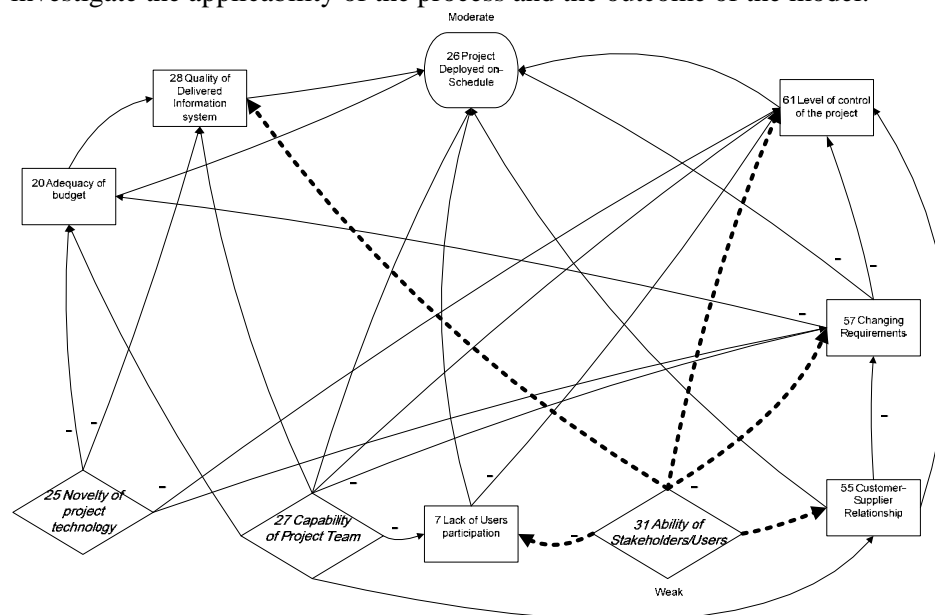


Figure 5. Workshop outcome and affected concepts.

In the workshop, participants were introduced to CCM and how it could be used in risk management. The participants were then introduced to CorMod and how it could be applied to the current software project. As a first observation, as soon as the CorMod was shown, it had a clear visual impact on the participants; reporting that they could observe how the project components are interrelated with each other. The workshop participants were in general agreement with all of the links, but added one more to the CorMod.

The participants rated the input to CorMod according to their perceptions within the context of the new project. Inter-group communication and collaboration was markedly improved with the existence of the common, shared visualization, and an agreed outcome resulted (see figure 5). The participants were aware that there were problems associated with the stakeholders' ability, but CorMod provided extra intuition on which other concepts would be affected by it. In Figure 5, the dotted arcs illustrate the 5 risk concepts affected. The participants took these concepts into consideration and entered them in the risk registry of the project.

7 CONCLUSION AND FURTHER WORK

As seen from earlier in this paper, there is evidence that formal risk management is not widely used because software project risk frameworks do not, in general, provide simple, pragmatic tools (Tiwana and Keil 2004). Our intention has been to create a qualitative risk identification and analysis tool. CorMod has been developed as a base foundation tool to support project managers and project team members in identifying different elements of the project and how they interact. CorMod does not use system dynamics modelling, but rather it captures a static 'snapshot' of the project behaviour and projects an expected outcome of the project based on this. Knowledge management has been facilitated in our case studies, with enhancements to the team's knowledge base.

In the case of the case study in Kuwait described in section 6, although the team members were aware that there was a potential problem with 'ability of users', they were surprised to see the effects such a problem might have on related risks. CorMod was able to demonstrate, in a visual sense, its consequences in a chain of cause and effect. The case study described in section 6 is on going and will in future form a longitudinal case study.

The literature is full of many risk factors that are not included in the model, but we believe that our model represents a viable and useful starting point for project managers and project members not only in the initial risk identification and assessment phase, but also during the lifetime of the project. CorMod may be tailored according to the particular circumstances of a project. From our research, benefits include:

- CorMod had a strong visual impact on the participants in project team group sessions.
- Project members can see the different components of the product and how they are interrelated.
- CorMod demonstrated its diagnostic capability, by indicating expected deficiencies in the software projects used in the study.
- CorMod demonstrated its simplicity in use by transforming the knowledge of the project participants into a single diagram representing the whole project.
- CorMod was successful in visualizing the expected path of cause and effect chain.

Although CorMod is able to provide an interactive, shared and accessible representation of IS/IT projects, there is a need to provide an execution engine that is capable of taking initiators values and making the required inferences to provide information on expected outcomes. We are currently developing software tools which are capable of animating the models produced, and defining measurement frameworks able to capture appropriate qualitative and quantitative values, both for risks themselves, and the relationships between them. This brings many challenges, including mitigation strategy identification, partial model enactment, and selective visualization, but initial experiments serve to demonstrate the potential for such developments.

References

- Al-Shehab, A., Hughes, R. and Winstanley, G., 2005a. 'Facilitating Organisational Learning Through Causal Mapping Techniques in IS/IT Project Risk Management', *Lecture Notes in Artificial Intelligence, LNAI 3782*, pp 145-154, April 2005, Springer, Germany.
- Al-Shehab, A., Hughes, R. and Winstanley, G., 2005b. Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping, *The Electronic Journal of Information Systems Evaluation*, Vol. 8, Iss. 1, pp 1-10.
- Bartlett, John., 2002. *Managing Risk for projects and programmes*, Project Manager Today Publications, Hampshire, Great Britain
- Bryson, M., Ackermann, F. and Finn, C., 2004. *Visible Thinking: Unlocking causal mapping for practical business results*, John Wiley & Sons, Ltd. 2004.
- Boehm, B., 1991. Software Risk Management: Principles and Practices, *IEEE Software*, pages 32-41.
- Charette, Robert, 1997. Managing the Risks in Information Systems and Technology, *Advances in Computers*, University of Maryland. Volume (44) page 1-58.
- Davey, Karl, 2000. Preventing Project Cost Escalation, *IEE Engineering Management Journal*, August 2000.
- Dorofee, A., Walker, J., Alberts C., Higuera, R., Murphy, R. and Williams, R. 1996. *Continuous Risk Management Guidebook*, Software Engineering Institute, Carnegie Mellon University, USA.
- Euromethod, version 1, July 1996, <http://projekte.fast.de/Euromethod/> [Accessed 28 April 2006]
- Gains, Brian, 2003. Concept mapping, *Software World*, v34, i4, p8(2), July 2003.
- Huff, A., 1990. 'Mapping Strategic Thought', Wiley and Sons, USA
- Hughes, B. and Cotterell, M., 2002. *Software Project Management*,. 3rd edition. McGraw-Hill Companies, UK.
- IEEE, 2001, 'Information technology-Software life cycle processes- Risk management', IEEE Std 1540TM-2001.
- Karolak, D. W., 1996. *Software Engineering Risk Management*, IEEE Computer Society Press, CA, USA.
- Keil, M., Tiwana, A. and Bush, A., 2002. Reconciling user and project manger perceptions of IT project risk: a Delphi study, *Information Systems Journal*, 12, pp103-119.
- McManus, J., 2004. *Risk Management in Software Development Projects*, Elsevier Butterworth Heinemann
- Narayanan, V.K. and Armstrong, D. J., 2005. *Causal Mapping for Research in Information Technology*, Idea Group Publishing, USA.
- Rogers, E.M., 1983. *Diffusion of Innovation*, The Free Press, New York, 1983
- Scavarda, A., Bouzdin-Chameeva, T., Goldstein, S., Hays, J. and Hill, A., 2004. A Review of the Causal Mapping Practice and Research Literature, *Second World Conference on POM and 15th Annual POM Conference*, Cancun, Mexico.
- Schmidt, R., Lyytinen, K., Keil, M. and Cule, P., 2001, Identifying Software Project Risks: An International Delphi Study, *Journal of Management Information Systems*, Spring 2001, Vol. 17, No.4, pp.5-36.
- Smith, J., 2001. *Troubled IT projects: prevention and turnaround*, Institution of Electrical Engineers, UK
- Tiwana, A. and Keil, M., 2004. The One-Minute Risk Assessment Tool, *Communications of the ACM* November 2004, Vol.47 No.11.
- Tsadiras, A. and Margaritis, K., 1997. Cognitive Mapping and Certainty Neuron Cognitive Maps, *Information Sciences* 101, pp109-130.
- Turner, P. and Jenkins, T., 1996. *Euromethod and Beyond, Open Frameworks for European Information Systems*, International Thomson Computer Press, London, UK.
- Watson, H., Haley, B.J., 1998. Managerial Considerations, *Communications of the ACM*, September Vol. 41, No. 9.