RADaR: A MODEL FOR IT ORGANIZATIONAL CHANGE FOUR AUSTRALIAN CASE STUDIES ON THE ADOPTION AND DIFFUSION OF METHOD ENGINEERING

Magdy K. Serour
University of Technology

Darryl Winder
InContext Solutions

Abstract

While there has been significant discussion of the techniques of method engineering, the current literature gives little attention to the adoption and diffusion process, firstly, how the organizations get ready for the adoption process then how they shift their behaviours from prior approaches to their new ones.
Utilising four Australian Case Studies, we investigate some factors contributing to the success or failure of a method engineering implementation and discuss a transition model (RADaR) that was designed and constructed to assist IT organizations to adopt and diffuse new approaches such as method engineering successfully and effectively. In this paper, we demonstrate and discuss some important aspects of the IT organizational process to adopt and diffuse a method engineering approach such as the organizational readiness and the impact of executive commitment on the outcome of the method engineering exercise. We also draw conclusions for practitioners looking to achieve 'successes' from the process.

Keywords*: Organizational Change, Method Engineering, Software Development Methodology, Human and Organizational Factors*

1 INTRODUCTION

Aligning Information Systems (IS) with business objectives has been a major focus of IS professionals and business executives alike for decades. The changing world of both Information Systems and businesses has significantly impacted and changed the relationships between these two domains. Businesses are looking for high-quality and reliable software systems and faster returns on their IT investments. They are eager to gain control of their information systems and adequate capabilities to easily customize and adjust their applications as needs arise. On the other hand, IS professionals have realized the need to gain better control over their development processes. They are urgently looking for new approaches and technologies to effectively manage both the ever-changing business requirements and their development process of their various projects and indeed, this is the focus of this paper.
In this paper, we explore the importance of the organizational adoption and diffusion of a well-constructed software development methodological framework that may well be instantiated and tailored to suit different software projects. Secondly, we present our engineered transition model, RADaR, which was specifically designed and constructed to aid IT organizations during their endeavour of adopting and diffusing new approaches to software engineering. Thirdly, we explore some of the collected findings from four case studies that were carried out in Australia with private and public organizations. Crucially we consider both successful and unsuccessful examples and attempt to investigate and examine the effect of various factors that contributed to those outcomes.

2 WHY DO WE NEED TO CHANGE OUR WAYS?

For organisations to solve an existing problem or enhance the way they do things, they need to change their existing way of thinking to be able to identify the most appropriate solution for the problem at hand and/or work out a better way to do things. To cope with today's ever-changing business requirements and to gain competitive advantage, organisations need to look for new and better ways to do their business and stay ahead of their competitors (Serour et. al., 2006, Platt 2006)

To do so, they need to plan to adopt and utilise new technologies or innovations that may enable them to achieve their goals. First, organisations need to understand their existing culture and work out the required cultural changes to become familiar and comfortable with the new environment to which they are aiming to move. Once they have succeeded with the shift to the new environment, they will be able to adopt and utilise new technologies much more effectively (a method engineering approach in this case).

3 THE METHOD ENGINEERING APPROACH

Method engineering (Kumar and Welke, 1992; Brinkkemper, 1996) is a rational approach to the construction of methods from method components stored in a method repository as the alternative to adopting a predefined method for all projects. The philosophical view underpinning the method engineering approach is the principle of constructing a method specifically configured to the situation of the project at hand (Brinkkemper, 1996) i.e. one that meets the requirements of a particular project or the requirements of an organizational suite of projects (Serour et. al., 2004). Then, instead of adopting one 'heavyweight' predefined methodology, IT organizations may adopt a method engineering approach where small pieces of a method are identified and stored as method fragments (Hofstede and Verhoef, 1997) in a repository or methodbase (Ralyté, 1999; Saeki, 2003). For each project, the development team, with guidance of a method engineer, selects appropriate method fragments from the methodbase. The method is thus "constructed" or engineered from its component parts in such a way that only relevant components are incorporated into the constructed method and those not useful can be safely bypassed (Brinkkemper et.al, 1999). The primary advantage of the method engineering approach is its ability to provide consistency across disparate environments without sacrificing flexibility or adoption effort.

3.1 The Fundamentals and Necessity of Method Engineering

With the growing complexity of today's business environment, IT organizations are facing an increasing pressure to respond to and fulfil business needs quickly and more effectively than ever before. The commercial pressures to produce more dependable software faster prompt management initiatives to improve their software practices. While many factors influence the success or failure of a software development project, two factors of high importance are the people involved and the methodological approach they use (Henderson-Sellers et. al., 2006). So, IT organizations need to secure the commitment, dedication and support of their people and also to look for new approaches and technologies to effectively manage and control the process of software development (Henderson-Sellers and Serour, 2005).

Based on the dynamic nature of today's business and due to the fact that software projects vary greatly depending on factors such as business scope, rules and constraints, it is increasingly

evident that there is no one approach or methodology/method suits all kinds of software projects even at the organizational level (Cockburn, 2000, Serour et. al., 2004). The traditional approach of adopting one predominant methodology as 'the solution' for all the organization's projects proved to be impractical and incompetent to respond to and fulfil business' needs. As a result, IT organizations are looking for new approaches to software development that offer flexibility and have the capacity to support a wide range of projects at the organizational or domain level. A new and highly recommended approach to offer such flexibility is that of Method Engineering (Brinkkemper et. al., 1998; Ralyté and Rolland 2001; Henderson-Sellers 2003).

3.2 Challenges in Adopting and Diffusing a Method Engineering Approach

Managers and academics in Information Systems continue to struggle with the critical challenges of how to integrate new technologies to transform organizational performance. There are many aspects that closely relate to these critical challenges mainly in relation with human, managerial, organizational and technical integration (Ovaska, 2005; Botezatu and Botezatu 2006; Serour and Younessi 2006).

People's natural resistance to change has proven to be the most problematical challenge that might face the IT organization during any endeavour to change their existing work culture, such as in the case of introducing a new approach to software development. The real threat of this challenge is all too often manifested in project failure. Managing the human factors such as gaining and securing support, commitment and dedication across the board, forms another major challenge that must be overcome for fruitful results. Once an organization comes to realize what it needs to achieve, and decides how it will accomplish its goals, the main challenge becomes the issue of effective and efficient management of the human factors. It is quite surprising to some to know that 80% of project failures are traceable back to mismanagement of human factors (Jacobson et al., 1995).

4 THE RADAR MODEL, INTRODUCTION

In this paper, we state that the process of adopting and diffusing a new approach to software development encompasses four major stages focusing on different aspects of the transition and aiming to achieve different goals. These major stages are: Readiness, Adoption, Diffusion and Retrospective (RADaR) and these stages are the underpinning fundamentals of our newly engineered model RADaR.

The RADaR model describes a process with a number of major stages, each with number of associated tasks. It starts with the readiness stage including the mandatory culture change, followed by the adoption and diffusion of the new work practices and measuring and evaluating the final outcomes through the retrospective stage and exploring improvements. The RADaR model aims to move IT organisations from their current common practice environment to their desired better practice environment painlessly and effectively. Table 1 illustrates the model stages with the associated tasks for each stage.

RADaR – The Organizational Transition Model			
Readiness	**Adoption**	**Diffusion**	**Retrospective**
Aim Define the compelling reasons for the change and specify the business objectives	**Assay** Examine and evaluate the appropriate technologies and tools required for the change	**Apply** Diffuse and put the adopted technology and tools into operation	**Assess** Assess and evaluate the new state and identify all the outcomes (losses and/or benefits)
Assess Inspect and assess the current state and plan for the change	**Acquire** Attain the assessed and appropriate technologies and tools for the change	**Advance** Advance people's competence by providing adequate training on the technologies& tools	**Aim** Based on the above assessment, set the goals for the next cycle
Activate Motivate and stimulate every one involved to gain their willingness, dedication and royalty for the change	**Adapt** Modify and/or tailor the acquired technology and tools to best suit the organization	**Assist** Abet and support people during their change by providing them with mentoring and just-in-time help	**Award** Reward and encourage the people involved for their effort and acknowledge their contribution
Allocate Identify and secure all the required resources that needed for a successful change	**Advance** Advance people's knowledge by providing adequate introduction and education on the adopted technology incrementally	**Assure** Observe people to make sure they are comfortable with their new culture and address any issues	**Act** Get ready to start the next change cycle to reassess the original business objectives and to achieve the new set of goals *[Back to readiness]*

Table 1 – RADaR, Stages and Tasks

4.1 Readiness

The transition starts with a focus on getting the organisation ready for the new paradigm. It concentrates on transitioning both management and development teams to a new work culture through the adoption and diffusion of a method engineering approach and achieving the necessary cultural changes. This stage involves a number of tasks aimed at making the organisation familiar and comfortable with the new software development environment before any attempt to implement and the new technology is made.

4.2 Adoption

This stage focuses on the assessment, acquisition, implementation, acceptance and dissemination of the new approach and technologies to be adopted along with associated tools. The tasks of this stage are mainly concerned with the assessment of appropriate technologies required for the change followed by acquisition of the selected technologies. Furthermore, the organization should carefully adapt their selected technologies to best suit their environment rather than adapting the organizational culture to suit the adopted technology. The last task of this stage pertains to the crucial activity of advancing people's knowledge by providing them with the appropriate and adequate introduction and education on the adopted technologies or techniques as a mandatory step towards managing people's resistance to change.

4.3 Diffusion

This stage concentrates on the utilization of the newly adopted technologies. It focuses on embedding the principles and concepts of the new approach, deploying and spreading out the adopted technology elements. Also, this stage aims to prevent people returning to their old habits of developing software by deploying a mentoring and rewarding approach that people must feel comfortable with. The tasks of this stage are primarily concerned with implementing the adopted technologies and providing people with adequate just-in-time training to enable them to utilize and master their new work practice comfortably and confidently. Moreover, during this stage, senior management must aid and support their people during their change by coaching them through their daily activities and acknowledge their achievements.

4.4 Retrospective

Through this stage, organisations will be able to assess the whole transition and draw attention to any area for improvement and keeping the entire IT organisation updated with the adopted approach and tools. So, this stage focuses on assessing and evaluating the transition achievements and finding any defects for corrections. It also aims to define any possibilities for further improvements, enhancement and optimisations. The major tasks involved in this stage pertain to evaluating the newly adopted and diffused approach and documenting suggested enhancements. Based on these improvement suggestions, the organization will be able to assess their next state then plan and set new objectives for another cycle.

5 CASE STUDIES OF SUCCESS AND FAILURE

5.1 Success and Failure, What do they Really Mean?

Due to the fact that there is no definitive definition of the terms 'Success' and 'Failure' and also owing to the importance of the meaning of these terms, It is appropriate to present our operational definition of these terms for clarification and consistency reasons only.
We define the term 'Success' in general as delivering a tangible business benefit in a timely fashion and at a price commensurate with the benefit achieved, this is the '*Return on Investment*' model of success and consequently, we define 'Failure' as not meeting all of the above conditions.

5.2 First Study Organization - Background and History

This organization is a leading publisher of legal and tax information. About two years prior to this study a decision was taken to change from predominantly print into an on-line publisher.

The Production Division (PD) within the organization was established as a key element in realising the organisational transformation. PD has been developing software since early 1995 when the development team consisted six people but now has around 85 people most of which are involved in some form of software development. This is around an order of magnitude increase in just over two years! This massive increase is compounded by the fact that there are two very different types of development going on: large systems development and web development.
The development team of PD is made up of a mixture of experienced and inexperienced permanent staff and a number of contractors. All software development is done in a fairly ad hoc way, and many of the staff unused to the tight deadlines inherent in the business of a commercial publisher.
These problems manifest themselves in a number of different ways: mismatches in communications when discussing different aspects of the system, time wasted due to informal on-the-job training and inconsistencies in the final design. Two projects were conducted at this organisation in order to transit the IT department to a new work culture with the adoption and diffusion of new Object technologies.

5.2.1 First Case Study

The focus of this first project was the introduction and establishment of a Software Engineering Process into the organisation to replace the existing ad-hoc process. Our team[1] with other staff members of the organization have proposed the idea of adopting a method engineering approach and the utilization of one of the contemporary methodological framework, OPEN Process Framework (OPF) (Firesmith and Henderson-Sellers 2002) for designing and constructing an organizational methodology that may perfectly fit the organization's needs. Senior executives and development teams accepted the proposal after conducting a number of education sessions regarding the Object Technology in general and the method engineering in particular. For this first project, the organisation selected one of its existing software applications (Search Engine) to be re-developed as a pilot project for the adoption process.

It soon became clear however that the wrong pilot project had been selected since it had little visibility and insufficient importance within the organisation (few would care whether it failed or succeeded) and that the involvement of end users was too low. Based on our observation and interviews with some of the key personnel, the following issues contributed strongly to the termination of this project:

- Lack of proper planning and the introduction of method engineering.
- Lack of time and resources to create a methodology to best suit the organisation's needs.
- Misinterpretation of the business objectives.
- Wrong selection criteria for choosing the pilot project (probably an important conclusion).
- Lack of resources despite having a budget for the project.
- Poor estimation of tasks required in adopting new technology (not surprisingly).
- Slower learning curve than originally anticipated.
- Strong resistance from several participants.

Shortly, it became evident that the commitment to the project was not total. Team members found they had other commitments; middle management, while being fully committed with a (reasonable) budget, was unable to identify and commit the necessary (human) resources. As a result, it was decided to terminate this first project with the organisation's management stating

[1] The first author of this paper was the head of the transition team whereas the second author was the IT Director

that an enormous amount had been learned and, thus, the overall project had been very successful.
However by our definition, the project was not successful since it was prematurely terminated and the end result was not the anticipated adoption of a method engineering approach software development.

5.2.2 Second Case Study with the Same Organization

The second project with the same organization was driven by a strong desire to enable the organization to compete effectively in their market place. A business case was approved by senior management to place approximately 60 databases online. A deadline of 12 – 18 months has been set for completion of this programme. A research project has been established to introduce OO and develop and promote a disciplined approach to software and systems engineering. The major business objectives are:

- Achieve the necessarily culture change to comply with OO.
- Deploy or construct a formal OO software development methodology into organisation.
- Enhance the organisation's maturity level in terms of software development (equivalent to CMM level 3. This means a process has been defined and incorporated into the organisation).
- Dramatically reduce lead times in the software development cycles without compromising quality.

The second pilot project was a new set of web-based software applications urgently needed for the organisation to offer their products on-line – a direct response to their competitors' actions, rather than rewriting (in OO) of an existing piece of software as in the first pilot.
Within the first four months of this project, a first draft document of the organisational methodology had been constructed and reviewed as an instance of one of the adopted OPF. The terminology was now beginning to be the lingua franca. Later the second draft documents of the organizational methodology were reviewed by a group of senior and programme managers, project leaders and system developers. Soon, final documents were constructed and a full presentation was designed to launch the new culture.

5.3 Second Study Organization - Background and History

The second study organization is a NSW government Department (GD) that provides public services to more than four million customers in the communities of Sydney and other cities. The GD has around 3,500 full time staff members in general and about 120 personnel in the IT section. Their main focus is developing and maintaining software applications for other sections within the GD including human resources, service management and service monitoring. About five years ago, a decision was taken to provide the community with some online services such as paying bills, submitting general enquires and viewing public reports to remain effective in the online services arena and improve operating efficiency.

5.3.1 Third Case Study with First Adoption Attempt

Senior management made a decision to adopt a formal (heavyweight) OO software development process along with a recognized modelling language and CASE tool to develop software applications for the new online service environment. The Rational Unified Process (RUP) (Kruchten, 1999) along with the Unified Modeling Language (UML) (OMG 2001) were selected for adoption.

The pilot project was the development of a Customer Relation Management (CRM) system due to its vital role in providing customer services to the public. This attempt was welcomed by top senior management but strongly (although passively) resisted by most IT developers. Firstly, the proposed software process (RUP) was considered too large to learn over a short period of time. It was soon identified by the development team as being much too "heavyweight" for transitioning the IT personnel to a new software development environment. Secondly, the selected pilot project was very large in size and critical in nature, requiring special skills and expertise for its implementation.
Senior management realized the problem and decided to outsource the entire CRM project to an external consulting firm over a period of two years. After three years, with the project budget exceeded and the system was still under development the project was cancelled.

From a research point of view, we can identify several factors that had a negative impact on this first adoption attempt and not surprisingly contributed significantly towards its termination:

- The inappropriate introduction of the new software approach, techniques and tools that strongly increased people's resistance to change.
- The lack of people's participation during the stage of assessing and selecting the most appropriate method for the organization's needs that negatively related to people's feeling of ownership and the sharing of values with their organization.
- The existence of a communication gap between top management and IT personnel.
- The lack of adequate and appropriate education and training that people needed to feel confident for adopting the new approach.

5.3.2 Second Adoption Attempt

A further attempt for a change was initiated by a senior IT manager and the web development project manger, who realized the immediate need for changing their existing work culture and requiring a method flexible enough to be customized to suit small and large projects. Following a strong recommendation from our team, the manager agreed to start the transition process with a small team first. The success of the small team could then influence and impact on other teams' transition. We call this technique "small wins", a technique successful in other projects (see Serour et al., 2002; Serour and Henderson-Sellers, 2004) and proven effective in changing people's culture and managing their resistance.

5.3.3 Solution with Method Engineering

Our team proposed a solution by creating a new agile method using the situational Method Engineering (SME) approach and the OPF in an incremental and iterative manner. The proposed solution came as a result of the findings and analysis of the current culture of the online team and also the use of the "sweet spots" technique (Cockburn, 2002). The method engineering and agility approaches were strongly recommended to the online team for the following reasons:

- Each organization must develop its own way of working rather than adapting to an existing way (Cockburn, 2002).
- The idea of adopting an existing method, even agile, was rejected due to their belief that they would have to adapt themselves to suit the adopted method.
- The method would provide manoeuvrability to deal with requirements changes and also enable the team to deliver software products faster to their customers.
- Agility not only provides a better way of developing software but can be enhanced by the use of SME to support method customization to best suit people's need.

- Getting customer involvement through the entire development lifecycle is a top priority – as in the Agile Manifesto (Agile Alliances 2001).

The development team selected method fragments that they understood and "felt happy with" with requirements engineering as the initial focus. This ability to "design their own method" proved to be a valuable asset in overcoming concerns related to method "ownership" as compared to "management's imposition of an externally-sourced methodology".

5.3.4 Signs of Success

Different use case templates were used for brainstorming to select and/or design the most suitable for them. One simple use case template was selected and refined by the whole team to suit the needs of the 'Tiny Projects' development. The new template was then sent to all customers for review. Very positive feedback was received with a few recommended changes. The customer feedback was delivered by a customer representative who attended the improvement team's regular meetings, a visible and healthy sign that they wished to be part of the improvement team.
Notably, the procedure to select and adapt the use case template proved very effective. It gave all a sense of ownership and promoted the value of the change. These feelings had been considered critical for reducing people's resistance to change and they were shown to be so on this project.

5.4 Third Study Organization - Background and History

The third study organization is a large IT Services organisation with regional headquarters in Sydney engaged in multiple projects for a wide range of clients. In contrast, this organisation started much more strongly with a clear mandate from senior executives and funding, 50% in the form of a research grant, for a multi-year program to define and construct a new software engineering approach.

5.4.1 Fourth Case Study with the Third Study Organization

As the project progressed however, the disparity between commercial and academic objectives became more apparent. When, as happens on long programs, one of the key initiators moved away from direct involvement, the program fell apart. Despite the fact that program was approved at the outset with a relatively long delivery cycle (three years); the practical situation is that any activity that runs more than six months without delivering runs a high risk of changes in direction and priority. It has been observed that the success rate for software development projects falls dramatically after six months.

6 CONCLUSION

In our case studies we have examined both successes and failures (as determined by our criteria of tangible business benefit) particularly for the purpose of identifying influencing factors. We suggest our observations are consistent with results in other domains whereby human factors have a major impact on success or failure of the exercise. Further we conclude that the method engineering approach, when it is itself applied in an iterative and incremental fashion has characteristics that are supportive of addressing the human factors. Most importantly we conclude that a significantly improved chance of success comes when the full spectrum of the transition: from Readiness, through Adoption, Diffusion and concluding with Retrospective (as in our RADaR model) is considered and particularly that choice of the participants (supportive & influential) and the pilot (important and timely) are critical.

REFERENCES

Agile Alliance, 2001, http://www.agilemanifesto.org accessed on 22nd February, 2007.

Botezatu, C., and Botezatu, C., 2006, New aspects of Software Development in Economy, Proceedings of ICCCC2006, Baile Felix, Oradea, Romania, pp. 100-104

Brinkkemper, S., 1996, 'Method engineering: engineering of information systems development methods and tools', Journal of Information Software Technology, Vol. 38, No. 4, pp.275-280.

Brinkkemper, S., Saeki, M. and Harmsen, F., 1998, 'Assembly techniques for method engineering', Proceedings of CAISE 1998, Springer Verlag, pp.381-400.

Brinkkemper, S., Saeki, M. and Harmsen, F., 1999, Meta-modelling based assembly techniques for situational method engineering, Information Systems, 24(3), 209-228

Cockburn, A., 2000, 'Selecting a project's methodology', IEEE Software, Vol. 17, No. 4, pp.64-71

Cockburn, A., 2002, An interview with Alistair, The Cutter Consortium,

Firesmith and Henderson-Sellers, 2002, The OPEN Process Framework. An Introduction, Addison-Wesley, 330pp

Henderson-Sellers B., Serour M.K., Gonzalez-Perez C. and Qumer A., 2006, Improving Agile Software Development by the Application of Method Engineering, Proceedings of Software Engineering, SE2007, Innsbruck, Austria Editor(s): W. Hasselbring 399 pp.

Henderson-Sellers, B. & Serour, M.K. 2005, 'Creating a dual-agility method: the value of method engineering', Journal of Database Management, vol. 16, no. 4, pp. 1-23.

Henderson-Sellers, B., 2003, Method engineering for OO system development, Comm. ACM, 46(10), 73-78 http://www.cutter.com/consultants/cockburna.html accessed on the 4th of October, 2006

Jacobson, I., Ericsson, M., and Jacobson, A. The Object Advantage, Business Process Reengineering with Object Technology, ACM Press, Wokingham, England, 1995, p.347.

Kruchten, Ph., 1999, The Rational Unified Process: An Introduction, Addison Wesley Longman Inc, USA.

Kumar, K. and Welke, R.J., 1992, 'Methodology engineering: a proposal for situation-specific methodology construction', in Challenges and Strategies for Research in Systems Development (eds. W.W. Cotterman and J.A. Senn), J. Wiley, Chichester, pp. 257-269.

OMG 2001. OMG Unified Modelling Language Specification, version 1.4. OMG documents formal/01-09-68 through 80 (13 documents). http://www.omg.org, accessed 12th February 2007.

Ovaska, P., Working with Methods: Observations on the Role of Methods in Systems Development, Proceedings of the 13th International Conference on Information Systems Development – Advances in Theory, Practice and Education (ISD'2004), Edited by Vasilecas, O., Caplinskas, A., Wojkowski, W., Wojkowski, G., Zupancic, J. and Wrycza, S., Springer

Platt, D., 2006, "Why Software Sucks…and What You Can DO About It", Addison Wesley Professional, 1st Edition, 272pp.

Ralyté, J. and Rolland, C., 2001, An assembly process model for method engineering, in K.R. Dittrich, A. Geppert and M.C. Norrie (Eds.) Advanced Information Systems Engineering), LNCS2068, Springer, Berlin, 267-283.

Ralyté, J., 1999, Reusing scenario based approaches in requirement engineering methods: CREWS method base, Proceedings of the 10th International Workshop on Database and Expert Systems Applications (DEXA'99), 1st International Workshop on the Requirements Engineering Process – Innovative Techniques, Models, Tools to support the RE Process (REP'99), Florence, Italy, 1-3 September 1999, IEEE Computer Society, Los Alamitos, CA, USA, 305-309

Saeki, M., 2003, CAME: the first step to automated software engineering, Process Engineering for Object-Oriented and Component-Based Development. Procs. OOPSLA 2003 Workshop, Centre for Object Technology Applications and Research, Sydney, Australia, 7-18

Serour, M.K. and Younessi, H., 2006, Towards Method Engineering for Agile Software Development: Theory and Practice, Proceedings of the IBIMA 2006 conference on CD, "Managing Information in the Digital Economy: Issues & Solutions" (ed., Khalid S., Soliman), ISBN: 0-9753393-5-4, Bonn, Germany, June 19-21, 2006

Serour, M.K., Dagher, L., Prior, J. and Henderson-Sellers, B., 2004, "OPEN for agility: an action research study of introducing method engineering into a government sector", Proceedings of the 13th Int. Conference on Information Systems Development. Advances in Theory, Practice and Education (eds. O. Vasilecas, A. Caplinskas, W. Wojtkowski, W.G. Wojtkowski, J. Zupancic and S. Wrycza), Vilnius Gediminas Technical University, Vilnius, Lithuania, pp.105-116

Serour, M.K., Henderson-Sellers, B., Dagher, L., 2006, "Augmenting an Existing Software Development Process with a Team Building Activity: A Case Study", CD-ROM Proceedings of the European and Mediterranean Conference on Information Systems (EMCIS) 2006, Alicante, Spain (eds. Z. Irani, S. Alshawi and O.D. Sarikas), pp.32-35

Serour, M.K., Henderson-Sellers, B., Hughes, J., Winder, D. and Chow, L., 2002, Organizational transition to object technology: theory and practice, in: Object-Oriented Information Systems, LNCS 2425, Z. Bellahsene, D. Patel and C. Rolland, eds., Springer-Verlag, Berlin, pp. 229-241.

Ter Hofstede, A.H.M. and T.F. Verhoef, 1997, On the feasibility of situational method engineering, Information Systems, 22, 401-422.