Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2005 Proceedings

European Conference on Information Systems (ECIS)

2005

A Paradox of the Change to User Stories: The Application of the Theory of Competing Commitments

John McAvoy
University College Cork, j.mcavoy@ucc.ie

Tom Butler
University College Cork, tbutler@afis.ucc.ie

Follow this and additional works at: http://aisel.aisnet.org/ecis2005

A PARADOX OF THE CHANGE TO USER STORIES: THE APPLICATION OF THE THEORY OF COMPETING COMMITMENTS

McAvoy, John, Business Information Systems, University College Cork, Ireland,
j.mcavoy@ucc.ie

Butler, Tom, Business Information Systems, University College Cork, Ireland,
tbutler@afis.ucc.ie

Abstract

Software development projects are about change, yet change is problematic in any situation. Individuals resist change and software developers are no different than other organizational actors in this regard. This paper describes a case study which examines the changes to the IS development environment wrought by the introduction of a new software development methodology. One aspect of the new methodology involves the use of user stories in place of traditional requirements documentation. The findings of this longitudinal study illustrated that developers' commitment to the use of user stories diminished greatly, ranging from initial commitment to skepticism, to virtual abandonment. In order to explain the underlying reasons for the reduction in commitment, the authors used the theory of competing commitments. Competing commitments are typically subconscious forces that work against behaviors and actions that social actors were previously committed to. While competing commitment theory has been applied in other fields, it has not been applied previously in the field of IS to understand IS-based phenomena. Further to the use of the competing commitment process, this paper's analysis of the software development project suggested the presence of hierarchical group think influencing the diminishing commitments.

Keywords: Resistance to change, Competing commitments, Project management, Software development.

1   INTRODUCTION

Change within software development projects is an area of importance to the success of the project, as projects, by their very nature, are about change. Although Cushway and Lodge (1999) emphasise the importance of managing change, their description of change management, is a restrictive one. For them, the concern is in developing strategies and structures. No mention is made of the teams and individuals who will effect, and be affected by, the change. The sole mention of the employees is a list of expectations, or required, behaviours such as: roles must be carried out in a dependable fashion; and there must be innovation in achieving organizational objectives.

This study describes a case study, undertaken by the authors, which examined the change involved in introducing a software development methodology. The case study is based in a software development project to develop a knowledge management system for the Irish Government. A longitudinal study of the development project was undertaken, initially using participant observation as its primary method. It concentrates solely on the software project team, as opposed to involving the various high-level project sponsors. One aspect of agile software development used in the project is the use of user stories. Rather than relying on complex design documents, agile espouses the writing of customer requirements in simple language. The stories should describe what is required of a part of the final software project. The longitudinal research into the software development project highlighted a problem with the change to this new process.

This paper reports on this longitudinal study of change. The change investigated was the change to the agile process of user stories. Although the developers were initially committed to the use of stories, this commitment diluted over time. The investigation into this dilution of commitment became a two-phase process. In phase 1, to determine the reasons behind this reduction in commitment to the change, Kegan and Lahey's (2001a,b) competing commitments process was followed. This process aims to determine the reasons, often subconscious, why a change that was originally committed to is not successful. These reasons are known as competing commitments as they work against the original commitment to change. Analysis of these competing commitments process was still insufficient in explaining the lack of success of the methodology change. Therefore, in phase 2, the output of the competing commitment process was then aligned with observations from the longitudinal case study and existing research literature on groupthink to determine a cause. This cause, the explanation for the failure to adopt user stories, is then elaborated on.

2   RESISTANCE TO CHANGE IN AGILE DEVELOPMENT

Cushway and Lodge (1999, p. 180) emphasise the fact that "probably one of the key skills required of managers in today's organizations is the ability to manage change." Lafleur (1996) stresses the fact that change is a constant in a project. Interestingly, many others regard projects as a usual method of implementing change (Boody & Macbeth 2000, Alsene 1999, McElroy 1996, Pellegrinelli 1997, Clarke 1999, Turner & Muller 2003).

2.1   *The relevance of change to Agile software development projects*

Metzger (1981) describes how change is part of every software project. Software development projects, which follow the agile methods (of which user stories is one aspect), regard change as one of their core aspects. Beck (1999) describes agile techniques as embracing change. Agile manufacturing, which shares many principles with agile software development, "*is the ability to thrive and prosper in an environment of constant and unpredictable change*" (Maskell 2001, p. 5). The agile manifesto, the core values of agile software development, is listed below:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

(Lindvall *et al* 2002, Fowler and Highsmith 2001)

The fourth point shows the importance of change in agile projects. Change is not only accepted but also encouraged. The first point shows the importance of the developers and their interactions (the team's social interactions) to an agile project. Each of these has relevance to this study and impacts the findings.

2.2 *Resistance to change in a software development project*

Humphrey (1989) describes how process change in software development will only be successful with continual commitment from, not just senior management, but everyone associated with the development of software. While further, Thomsett (1998) agrees that change is only effective when all impacted by the change are involved.

Nader (1993) describes the problems that arise when people resist change. People require a sense of security and this is achieved through stability. Changes imposed on an individual remove this stability. Whitehead (2001) describes how change can evoke stress and an emotional reaction against the change. Change, though, cannot be avoided in software development projects. "*Everything in software changes. The requirements change. The design changes. The team changes. The business changes. The team members change. The problem isn't change, per se, because change is going to happen; the problem, rather, is the inability to cope with change when it comes*" (Beck 2000, p. 28). Zmud (1983) adds to the list of areas that change in software development projects by discussing changes to processes (such as user stories in this research). Zmud argues that trying to implement process change by changing people will lead to resistance. Rainwater (2002) describes the danger for software projects of not assessing the impact of change.

Resistance to change can indicate a lack of commitment to the change, but there can be other explanations (Bowe *et al* 2003a). Software developers are a good demonstration of this resistance. Programmers' resistance to change is shown in a description of programmers from the 1960's "*like converts to a new religion, they often display a destructive closed-mindness bordering on zealotry*" (Zachary 1994, p. 13). Yourdon (1993) describes change as one of the major dislikes of programmers. Changing anything, ranging from procedures to hardware platforms to methodologies, will result in complaints. Resistance to change is not helped with various attitudes to change, demonstrated in others work, such as change being the enemy of the project (Birmingham 2002), and change being something that must be coped with and its disruptive impact minimised (Field & Keller 1998).

Returning to the second point of Bowe *et al* (2003a) – there can be other explanations for resistance than a lack of commitment. Lawrence (1969) believes that this resistance to change can highlight the fact that something is being overlooked. Rather than resistance to change having negative connotations, it should be regarded as an indicator that the change itself needs further examination. Kegan and Lahey (2001a) propose that some resistance is easy to explain (for example the stress of learning a new skill) but other resistance is not as easily explained. A paradox exists where people show a commitment to, and support, the change, yet still resist the change. Robbins and Finley (1998) state that resistance can be a subconscious act. We can all agree on an idea, but then do nothing to implement it. It was necessary to use a method which would identify and explain the causes for this paradox.

3   THE THEORY OF COMPETING COMMITMENTS

To understand the need for a new approach to determine why a change was resisted and failed, it is necessary to understand the context in which the resistance occurred.

3.1   The case study: an Agile software development project

A longitudinal study was performed on a software development team who were changing over to an Agile software development process. The project was the design and development of a knowledge management system for the Irish Government. A team of seven developers and one project manager were involved in the project.

One of the new processes was the use of user stories, which involved the creation of user stories in place of traditional requirements documentation. During the longitudinal study, the authors noted that the initial commitment to the use of user stories decreased. At the start of the development project there appeared to be full commitment to the use of user stories but, as time passed, the commitment effectively disappeared. The investigation of this decrease in commitment involved the use of the Competing Commitment Theory which, to the best of the authors knowledge, has not been applied before in the IS field. This theory was used as existing theory provided an inadequate explanation for the failure of the change to user stories.

3.2   The inability of existing theories on change to explain the observations

The paradox of initial commitment to the use of user stories and their ultimate rejection is not adequately explained by existing theories on change. Burnes (2000) lists the three schools of thought in the field of change management. The Individual perspective states that change is achieved through stimuli which influence human behaviour. There are two camps within this school: Behaviourists and Gestatlt Field. The Behaviourists believe that change is achieved through reward. The Gestalt-Field psychologists believe that individuals accept change through the use of reasoning. If an individual is helped to understand the need for change, then behaviour will change.

Neither explains the resistance to change seen in this development project. Rewards were present– the financial reward of successfully completing the project on time. The agile software development methodologies are aimed at successful and timely completion of a project, and user stories are part of these methodologies. Similarly the use of reasoning does not appear to explain the paradox, as user stories were explained, and their benefits described, after which the team accepted their usefulness.

The Open Systems School examines change through organizations and their subsets. Any change in one subset will impact the others. It is necessary, therefore, to take a holistic view of the organization when implementing change. Boody and Macbeth (2000) add subtly to this by stating that change in one area, needs to be accompanied by (as opposed to 'will cause') appropriate changes elsewhere. Hunt and Thomas (2000) refer to this as non-orthogonal systems. Birmingham (2002) describes these small changes affecting other areas as having a ripple effect or cascading effect. A small change in one area can affect the software in a seemingly independent area. Again, this does not adequately explain the paradox seen. User stories involve changes in other areas, such as how testing of the product is performed, but these changes were initially implemented in the project.

The Group Dynamics School, of importance when discussing small teams, emphasise bringing about change through groups. As individuals work in groups, changes occur through changing the group's norms and practices. One of the reasons people attach themselves to groups is to shield themselves from change – "*an insurance mechanism coping with uncertainties*" (Alkire and Denevlin

2002, p.21). Initial considerations of this do not explain the paradox. The team was committed to the use of user stories, yet it was the team that failed to implement them. If the team was initially willing to adopt the use of user stories then it implies that group norms and practices would not present an obstacle. The term "initial considerations" is used now, in hindsight, as the use of the competing commitments theory did show the relevance of the group dynamics school of thought.

Kegan and Lahey, as organisational psychologists (or adult development psychologists) propose that resistance to change, in these paradoxical situations, does not imply opposition – it implies the existence of a competing commitment. Competing commitments are hidden in a person's mind, and are observed as energy being unwittingly applied against the commitment already made. In this research, a resistance to their usage was undermining the commitment to user stories. This resisting force was seen as sighs when user stories were mentioned, an unwillingness to take ownership of the stories, and the fact that they were not applied properly. This resisting force is caused by commitments that act against the initial commitment to the user stories. It was necessary to determine what these competing commitments were.

Andersen and Jessen (2003) refer to projects as a venue for change, while Duck (1993) states that people achieve changes to processes. Competing commitment theory, in this research, explains the change paradox by examining the people involved in the project.

3.3   Competing Commitments Theory

Firstly, Kegan and Lahey (2001a, 2001b) prefer the term immunity to change in place of resistance to change. Resistance implies knowingly working against something – competing commitments are not obvious, even to the individual who has them. Banerjee (2003, p.74) describes competing commitments as *"self-defeating behaviour."* These behaviours, even if subconscious, act against change. Competing Commitments, also known as the Big Assumptions, Theory proposes a process through which the competing commitments, that effect change, can be identified. This process was originally proposed in Kegan and Lahey (2001a), and further discussed and demonstrated in Kegan and Lahey (2001b), Sparks (2002), Nash (2002), Bowe *et al* (2003a, 2003b), and Banerjee (2003). Competing commitments have some similarity with the view of Milgram (1971) who argues that public declarations of adherence to group decisions do not imply that the individual will translate this adherence into action. Kegan and Lahey's competing commitments describe the reasons why this initial acceptance is not acted upon. It should be pointed out, though, that Millgrams experiments showed that adherence can be translated into action.
Kegan and Lahey, the originators of the theory, do not restrict the domain of its application. While Nash (2002) and Bowe *et al* (2003a, 2003b) apply the process in the filed of medicine and medical educational, and Banerjee applies it at the organisational level, this paper applies it the context of a software development project. To the best of our knowledge, this process has not been applied in the IS field before. The focus of this paper is specifically an examination of why the change to a new software development process, although initially supported by the development teams, never materialised in the project.

The suitability of this approach is identified in Bowe *et al* (2003b, p723) which describes the technique being used to examine why problems arose *"during implementation when unanticipated or unaddressed organizational resistance surfaces."* Nash (2003, p.592) describes the use of the competing commitments process to go beyond *"buy-in"*. The same problems arose during the software development project being studied in this paper. User stories were initially supported – there was buy-in - yet never fully implemented. Kegan, in a interview in Sparks (2002) describes, what Nash refers to as "buy-in", as espousing commitment. Bowe *et al* (2003a) described the problem as *"like many new years resolutions, sincere intent to change may be short lived and followed by a return to old behaviours."* Again, the analogy of New Year's resolutions applies in this project. At the beginning of the project, the team felt that some of the problems with a previous project highlighted the need for more process, or new processes within the project. Although the previous project was successful, the

developers were able to acknowledge failings that they would like to overcome in the new projects. This in itself is noteworthy as most teams, and individuals, find it hard to acknowledge their own faults as described by cognitive dissonance theory (Weinberg 1971; Festinger and Carlsmith 1959; Schelling 1989; Harmon-Jones 1998) and self-justification theory (Keil *et al* 2000). The success of the previous project may explain part of the problem, though not all. Arrow *et al* (2000) highlight the seemingly contradictory theory that successful teams are more problematic when it comes to change. A team that has failed in the past is more likely to adapt new responses to change.

4 THE PROCESS OF UNCOVERING COMPETING COMMITMENTS

Kegan and Lahey developed a technique, which is used to determine competing commitments. Various authors describe this technique (Kegan and Lahey 2001a; Kegan and Lahey 2001b; Sparks 2002; Nash 2002; Bowe *et al* 2003a, 2003b). The technique comprises six steps, in the form of questions, although different authors merge some steps. The examples used below are those given as examples in Kegan and Lahey (2001b).

Step	Question	Example Response
1	What problem are you experiencing in work – a gripe or complaint?	My team do not tell me what's happening in a project
2	The complaint identifies something about you. What commitment does it imply?	I am committed to maximising the flow of information within the project.
3	What am I doing or not doing that goes against this commitment?	Sometimes I don't go out of my way to find out what is happening.
4	What do you think would happen if you were not doing what you described in question three – if you did the opposite of the undermining behaviour? What would worry you about this?	I might find out things from my team that I can do nothing about, something I can't fix.
5	What does this worry imply that you are committed to?	I am committed to not learning about things I can't control.
6	Inverting the answer from step five, and making it into the beginning of an assumption, complete the sentence. i.e. I assume that if I ….	I assume that if I learned about thing I couldn't control, people would realise that I am not able to do my job.

Figure 1.   Determining competing commitments.

In the examples above, the process has moved us from a complaint about a team not keeping the individual in the loop; to the big assumption that people will think the person incompetent if they cannot do everything. The individual states a commitment to full communication, yet the competing commitment - not learning about things they can't control – effectively works against their commitment to full communication. The process, as described in the examples above, was applied in the software development case study.

5 OBSERVATIONS AND ANALYSIS

This phase involved two sub-phases. Phase one involved discovering the competing commitments of the developers. This provided the cause of the failure to change, but did not explain the why. Phase two took these competing commitments and explained why they caused failure to change, by using

data from the longitudinal study of the software development project's change to an agile methodology. Phase 2 also required a review of existing literature on groupthink to guide the analysis.

5.1   Phase 1: Competing commitments identified

Five developers were interviewed out of the group of seven. One was not interviewed as he joined the project late - by joining late, he was not part of the initial expression of commitment to user stories. The other developer not interviewed was simply not available. Although this is a small group, the argument for its use is that the proposition of this research is the effect that small teams have on change.

Each developer went through the interview process and discovered two competing commitments. Although the process aims to find one competing commitment, each developer felt that two areas were important. Out of the five interviewed, five had a competing commitment directly related to the group. Each felt that a good relationship with their co-workers was vital, and that they were committed to ensuring that this relationship remained. These commitments expressed are listed in table 1.

Developer	Competing Commitment One	Competing Commitment Two
1	A good relationship with co-workers	Results are more important than process
2	To be accepted as part of the team	Do not want to be associated with a failed project
3	Recognition from team and project manager	The path of least resistance
4	Respect from the team	Avoiding conflict
5	Being part of the team	Ensuring the boss sees the work I am doing

Table 1.   The developers' competing commitments

Of the interviewee's second competing commitment (second in terms of display – it is not meant to imply a level of importance), two of these had relevance to relationships within the team. One was specific about avoiding conflict within the development team, while the other was specific about recognition from the project manager. Each of the original problems that the developers had with user stories, and their related original commitments, were unique to each developer. For each of these to evolve into the importance of team relationships do show its relevance. It should be pointed out though, that although developers accepted that it was important to them, it was not always their ultimate commitment.

- "It plays a part but is not the major one."
- "They are valid but not complete"
- "They are not commitments, they are more like traits"

The fact that intra-group relationships were not the ultimate competing commitments, but still important commitments, is summed up by one developer who stated that even if the major problems were sorted out, influences like intra-group relationships could still cause user stories to fail.

5.2   Phase 2: Competing commitments in context

From this we can see that the commitment to the team was having an effect, even if at times subconscious, on the change to the use of user stories. What was still not clear was how the social

relationships within a group had impacted the change. A further literature review revealed a possible explanation.

The importance of group relationships pointed to the possibility of groupthink having an impact. One of the main symptoms of groupthink (described in Janis 1972, Griffin 1997, Cartwright 2002, Moorhead *et al.* 1991) is a pressure to conform to the group's views. This appeared to be relevant to this study as initially there was group agreement on the benefit of changing the process to use user stories. Over time there was group consensus that they were not beneficial. The group's opinion changed as a group. Added to this is the proposition that the ultimate factor in groupthink is a highly cohesive team (Janis 1972, Griffin 1997, Kim 2001, Martin 1991, Neck 1996). "*Group cohesiveness refers to the degree to which members of the group desire to remain in the group*" (Kim 2001, p.175). Martin (1991) refers to cohesiveness as the forces between group members that keep them as a unit, cohesiveness been generally regarded as highly beneficial. Observations of the team in operation, over an eight-month period, showed them to be a highly cohesive team. Baron *et al* (1999, p.8) refer to cohesion as "*the overall strength of positive relationships within the group*." The competing commitments of the team members all mention the importance of relationships with team mates. There was still an incomplete explanation of why the change to user stories was not effective. Groupthink appeared to be having an effect, yet it was unclear why. A study of field notes made during the eight month observation of the project provides an explanation.

The project manager was liked and respected by the team. Unlike some project managers, the project manager in this case was regarded as part of the development team, as opposed to an outsider. Several developers specifically stated this. The majority of the team had worked under the project manager on a previous successful project. The project manager initiated the change from traditional development methods to agile software development, of which user stories are a major part. At this stage, the development team were committed to the use of user stories. Over time, the project manager himself allowed the use and application of user stories to be diluted. The project management approach, or style, was against the rigid interpretation of the method of writing and using user stories, and tended to apply a less rigid application. One of the reasons that the project management approach was liked and respected was his willingness to let the developers "get on with it", avoiding anything that was perceived to slow them down. An example of this dilution of the new methodology was seen during the prioritisation of user stories. One of the core concepts of user stories is that each story must be prioritised and ranked numerically, the most important story being developed first, and so on. The project manager was against this and argued for a rating of high, medium, and low. Further examples of the dilution of the application of user stories occurred, such as customer involvement in the stories being minimal. Concurrent with the project manager's dilution of the application of stories, the commitment from the developers diminished.

This matches what is described as hierarchical groupthink, as opposed to the commonly described peer groupthink. Maoz (1970) and Wright and Schaal (1978) argue that groupthink originates from a desire for concurrence with the group's ideas or the leaders ideas. Cartwright (2002) specifically differentiates, and names, two types of groupthink. Peer groupthink originates in a need for conformity and close integration within a team. Huczynski and Buchanan (1991) argue that the synergy and loyalty, which are regarded as team's greatest benefits, are the same factors that lead to groupthink. Hierarchical groupthink originates in a desire to please a leader, specifically the desire not to disagree with them. It is similar to approval-seeking behaviour as found in Lippitt *et al.* (1968) and ingratiation through conformance with the leaders view described in Jones *et al.* (1968) and Hurwitz *et al.* (1968). Kohl (1975) presents the argument that the dominance of Kissinger in Nixon's cabinet could have lead to groupthink as his opinion tended to be sacrosanct. Neck (1996) lists leader preference for a certain option as a potential factor in groupthink.

As the project manager changed from a firm advocate of user stories, to a diluted advocate of user stories, the developers commitment decreased. This appears to be an example of hierarchical groupthink – a desire to please the leader. Returning to the developers' competing commitments, this hierarchical groupthink can be seen in the competing commitments of two developers who specifically

stated that one of their competing commitments was a desire for recognition from the project manager. The other competing commitments, involving good relationships with the team, also have relevance because, as already noted, the project manager was regarded as part of the team. Another developer's competing commitment was that results are more important than process. This is practically verbalising the reason for the project manager's dilution of the user stories process. Hierarchical groupthink is therefore seen to have an impact on the ineffective change to the use of user stories. The role of the project manager as a bureaucracy buster, while beneficial in many aspects of the project, eventually assisted in the failure of the change to a new methodology.

6 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This paper does not posit that the role of the project manager is not to remove unnecessary overheads (bureaucracy busting). Nor does it imply that a project management who is respected by the team will cause problems. The longitudinal aspect of the overall research (much of which lies outside the domain of this paper) showed the benefits that this style of project management brings. What is clear, though, is that the project management role/style adopted can have a negative impact. The two main findings of this paper are:

- In this study, the project management style, which was one which the developers themselves preferred, was found to negatively impact the change to a new process (specifically the change to the use of user stories). This particular style of project management (and the roles this involves) is prevalent in the software development industry (Highsmith 2000).

- Another striking observation is that the fact that the development team was so cohesive (usually regarded as a major benefit to a project) that it ultimately assisted in the failure to change. The cohesiveness of the team, and the groupthink that evolved from it, was a factor in the failure to change to a new methodology. What is most striking about this is the fact that the change was a change to an Agile software development methodology. The Agile proponents emphasise the importance of a cohesive team. One of Agile's main benefits was also one of the reasons the Agile methods were not adopted.

To summarise, two desirable aspects of any software development project – a cohesive team, and a respected project manager – were factors in the failure of a change initiative. This would go against traditional views and findings in IS research.

At present, another case study in a different organization is taking place examining the changeover to agile software development. Again, the change to user stories was seen to be problematic, after initial commitment from the developers. Hierarchical groupthink does not appear to be an issue, so further research is required. It is planned to use the competing commitments theory to determine the causes for failure in this second case study.

Although previously not used in the IS area, the use of competing commitments as a research approach in the IS field has its advantages. Much research in IS involves the investigation of adoption of processes, tools, systems, etc. Failure of IS projects has been well reported so, is we accept the human impact on these failures, competing commitments could be used as another research tool to determine why IS adoptions can fail.

References

Alkire, S. and Denevlin, S. (2002). Individual motivation, its nature, determinants, and consequences for within group behaviour. In Heyer, J., Stewart, F., Thorp, R. (eds) Group behaviour an development.. Oxford University Press. NY, 51-73

Alsene, E. (1999). Internal changes and project management structures within

enterprises. International Journal of Project Management, 17 (6), 367-376.

Andersen, E. and Jessen, S. (2003). Project maturity in organisations. International Journal of Project Management. 21 (6), 457-461.

Arrow, H., McGrath, J. and Berdahl, J. (2000). Small Groups as Complex Systems. Sage Publications. CA.

Banerjee, R. (2003) Organisational character: issues, imperatives and practices.

International Journal of Human Resources Development and Management. 3 (1), 72-83.

Baron, R., Kerr, N., Miller, N. (1999) Group process, group decision, group action.

Open University Press. Buckingham, UK.

Beck, K. (1999) Embracing change with extreme programming. IEEE Computer. 32 (10), 70-77.

Beck, K. (2000) Extreme programming explained. Embrace change. Addison

Wesley. NY, USA.

Birmingham, D. (2002) Software development on a leash. Springer-Verlag.

NY, USA.

Boody, D., Macbeth, D. (2000) Prescriptions for managing change: a survey of their effects in projects to implement collaborative working between organizations. International Journal of Project Management. 18 (5). 297-306.

Bowe, C., Lahey, L., Kegan, R., Armstrong, E. (2003a) Questioning the big assumptions. Part I: addressing personal contradictions that impede professional development. Medical Education. 37 (8), 715-722.

Bowe, C., Lahey, L., Kegan, R., Armstrong, E. (2003b) Questioning the big assumptions. Part II: recognizing organizational contradictions that impede organisational change. Medical Education. 37 (9), 723-733.

Burnes, B. (2000) Managing Change, a Strategic Approach to Organisational Dynamics. Pearson Education Ltd. Harlow, UK.

Cartwright, R. (2002) Mastering team leadership. Palgrave Macmillan. Wales.

Clarke, A. (1999) A practical use of key success factors to improve the effectiveness of project management. International Journal of Project Management. 17 (3), 139-145.

Cushway, B., Lodge, D. (1999) Organization Behaviour and Design. Kogan Page Ltd. London, UK.

Duck, J. (1993) Managing change: The art of balancing. Harvard Business review. 71 (6), 109-118.

Festinger, L., Carlsmith, J. (1959) Cognitive consequences of forced compliance. in Coats, E., Feldman, R. (2001) Classic and contemporary readings in social psychology. Third Edition. pp 194-203. Pearson Education. NJ, USA.

Field, M., Keller, L. (1998) Project management. Thomson Learning. London, England.

Fowler, M. Highsmith, J. (2001) The agile manifesto. Software Development. 9(8). CMP Media

Griffin, E. (1997) A first look at communication theory. McGraw-Hill. NY, USA.

Harmon-Jones, E. (1998) Towards an understanding of the motivation underlying dissonance effects: Is the production of adverse consequences necessary? in Coats, E., Feldman, R. (2001) Classic and contemporary readings in social psychology. Third Edition. 204-213. Pearson Education. NJ, USA.

Highsmith, J. (2000) Adaptive software development: A collaborate approach to managing complex systems. Dorset House Publishing. NY, USA.

Huczynski, A., Buchanan, D. (1991) Organizational behaviour. An introductory text. Second Edition. Prentice Hall. NY, USA.

Humphrey, S. (1989) Managing the software process. Addison-Wesley. NY, USA.

Hunt, A., Thomas, D. (2000) The pragmatic programmer. Addison Wesley

Longman. MA, USA.

Hurwitz, J., Zander, A., Hymovitch, B. (1968) Some effects of power on the relations among group members. in Cartwright, D., Zander, A. (eds) Group dynamics. 291-297. Harper & Row Publishers. NY, USA.

Janis, I. (1972) Victims of groupthink. Houghton Mifflin Company. MA, USA.

Jones, E., Gergen, K., Gumpert, P., Thibaut, J. (1968) Some conditions affecting the use of ingratiation to influence performance evaluation in Cartwright, D., Zander, A. (eds) Group dynamics. 278-289. Harper & Row Publishers. NY, USA.

Kegan, R., Lahey, L. (2001a) How the way we talk can change the way we work. Jossey-Bass. CA, USA.

Kegan, R., Lahey, L. (2001b) The real reason people won't change. Harvard Business Review. 79 (10), 85-89. Harvard Business School Publishing. MA, USA

Keil, M., Mann, J., Rai, A. (2000) Why software projects escalate: An empirical analysis and test of four theoretical models. MIS Quarterly. 24 (4), 631-660.

Kim, Y. (2001) A comparative study of the 'Abilene paradox' and 'groupthink'. Public Administration Quarterly. 25 (2), 168-187.

Kohl, W. (1975) The Nixon-Kissinger foreign policy system and US-European relations: Patterns of policy making. World Politics. 28 (1), 1-43.

Lafleur, R. (1996) Project management. getting control and keeping control of complex projects. American Programmer. 9 (4), 24-28.

Lawrence, B. (1996) Unresolved ambiguity, The silent source of risk in your project. American Programmer. 9 (4), 18-22.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., Zelkowitz, M. (2002) Empirical findings in agile methods. Extreme Programming and Agile methods – XP/Agile Universe Chicago 2002. 197-207.

Lippit, R., Polansky, N., Redl, D., Rosen, S. (1968) The dynamics of power. in Cartwright, D., Zander, A. (eds) Group dynamics. 236-250. Harper & Row Publishers. NY, USA.

Maoz, Z. (1990) Framing the national interest: The manipulation of foreign policy decisions in group settings. World Politics. 43 (1), 77-110.

Martin, R. (1991) Working in groups. in Smith, M (ed) Analysing organisational behaviour. 154-177. Macmillan Press. London, UK.

Maskell, B. (2001) The age of agile manufacturing. Supply Change Management. 6 (1), 5-11.

McElroy, W. (1996) Implementing strategic change through projects. International Journal of Project Management.. 14 (6), 325-329.

Metzger, P. (1981) Managing a programming project. Prentice Hall Inc. NJ, USA.

Milgram, S. (1971) Group pressure and action against a person. in Taylor, D. (ed) Small groups. 157-170. Markham Publishing Company. Il, USA.

Moorhead, G., Neck, C., West, M. (1991) The tendency towards defective decision making with self-managing teams: The relevance of groupthink. Organizational Behaviour & Human Decision Process. 73 (2), 327-351.

Nader, D. (1993) Concepts for the management of organisational change. in Mabey, C., Major-White, B. (eds) Managing change. Second edition. 5-19. Paul Chapman Publishing. London, England.

Nash, D. (2002) Beyond buy-in. P&T Journal. 27 (11), 617-618.

Neck, C. (1996) Letterman or Leno: A groupthink analysis of successive decisions made by the national broadcasting company. Journal of Managerial Psychology. 11 (8), 3-17.

Pellegrinelli, S. (1997) Programme management: organising project-based change. International Journal of Project Management. 15 (3), 141-149.

Rainwater, J. (2002) Herding cats: A primer for programmers who lead programmers. Springer-Werlag. NY, USA.

Robbins, H., Finley, M. (1998) Why change doesn't work. Orion Publishing. London, England.

Schelling, T. (1989) The mind as a consuming organ. in Bell, D., Raiffa, H., Tversty, A. (eds) Decision making. Descriptve, normative, and prescriptive interactions. 343-357. Cambridge University Press. Cambridge, England.

Sparks, D. (2002) Inner conflicts, inner strengths. Journal of Staff Development. 23 (3), 66-71.

Thomsett, R. (1998) The care and feeding of Project Managers. American Programmer. 11 (2), 5-11.

Turner, J. Muller, R. (2003) On the nature of the project as a temporary organization. International Journal of Project Management. 21 (1), 1-8.

Weinberg, G. (1971) The psychology of compter programming. Van Nostrand Reinhold Co. NY, USA.

Whitehead, R. (2001) Leading a software development team. A developers guide to successfully leading people and projects. Addison-Wesley. London, England.

Wright, J., Schaal, D. (1988) Groupthink: The trap of consensus investing. Journal of Financial Planning. 1 (1), 41-44.

Yourdon, E. (1993) Decline and fall of the American programmer. Prentice Hall Inc. NJ, USA.

Zachary, G. (1994) Show-Stopper! The breakneck race to create Windows NT and the next generation at Microsoft. The Free Press. NY, USA.

Zmud,R. (1983) The Effectiveness of External Information Channels in Facilitating Innovation Within Software Development Groups. MIS Quarterly. 7 (2), 43-58.