The Influence of Intra-Team Relationships on the Systems Development Process: A Theoretical Framework of Intra-Group Dynamics

C.C.H. Rosen

University of Derby
Derbyshire Business School
University of Derby
Kedleston Road
Derby, DE22 1GB
UK

c.rosen@derby.ac.uk

Abstract. Improving the efficiency and effectiveness of developing software systems is something like the search for the Holy Grail. Tools have been developed, processes and procedures proposed, environments created. These have unquestionably improved the production process. Yet the aspiration of producing reliably, high quality systems on time and to budget is yet to be achieved. The reasons for this are well known. System development is a socio-technological process. Unless human factors are taken into account, in particular, interpersonal relationships that affect the operation of the process, a vital component is being overlooked. Remarkably, little is known about this process.

This paper reports case study findings examining the social-dynamic factors that affect the software development process. It proposes that a systems development environment must be understood as a dynamic system which intricately interconnects personal, interpersonal and intra-group factors. These factors can be observed using a framework of six interdependent, orthogonal polarities of intra-group dynamics. Interventions in the system can be assessed according to how they affect the balance between polarities.

INTRODUCTION

The discipline of Software Engineering has concentrated much of its effort on developing the tools and processes required to produce software. Significant progress has been made in this area. Models of the process are available, tools, languages and some theories of process that appear to accelerate production have been developed. However, in the drive to understand and assist systems developers, a fundamental element of the process has been largely overlooked. For all the automation of software production that is available, systems development remains a highly labour intensive activity. Consequently, the human factors are not only an important consideration in the process, they are critical to its success. The majority of commercial software is developed by teams of people, rather than by one person in isolation. The dynamics of their interactions play a major role in the success or failure of a software project. This paper contributes to the understanding of how relationships within the development team affect the production process.

Part of the problem has been the limitations in theoretical understanding of the systems development process. Solutions have been proposed in a relative vacuum of accurate conceptual understanding, and therefore, assertions have been made that cannot be substantiated. This paper addresses this omission by proposing a framework of intra-group dynamics that can be used to examine the intra-team relationships that affect the development process. The framework presents a structure for researchers and managers for observing interpersonal and intra-team relationships providing a basis upon which research questions can be framed and interventions assessed. It has been developed from three studies; two in commercial software development environments and one of student group projects which used a range of interpretive investigative techniques. The framework consists of six non-orthogonal polarities:-

1. Acceptance of personal responsibility vs. blaming / distancing
2. Personal interests vs. group interests
3. Task orientation vs. social awareness
4. Leadership (initiative taking) vs. followership (passivity)
5. Coherence and consistency of conceptual vision vs. cognitive diversity and valency
6. Need for flexibility vs. need for structure

Research Issues

There is little work on how interpersonal relationships affect the systems development process, particularly the actual production of software, even though the problem has been recognised for some time. Weinberg [39] first studied psychological factors in programmers. Zmud [40] recognised the issue, but proposed procedural solutions rather than address interpersonal relationships. Curtis [7], [8] and others [9], looked at individual differences between programmers and at organisational affects [22], [10]. DeMarco and Lister [12] reported anecdotal evidence. Rockwell [30] proposed that the development process be seen as a communications process and Rodwell et. al. [31] examined the role of communications on productivity. Sawyer and Guinan [36] observed that social processes had more influence on software production than methodologies or automation and Sawyer [35] reported that the variance between the best performing teams and the worst could largely be accounted for by intra-group conflict. There has been a growth in research in recent years examining the relationship between developers and users since Dagwell and Weber's [11] and Cronan and Means' [6] studies; Barki and Hartwick [2] being a recent example. There has however been little investigation into interpersonal or intra-team relationships within the development team itself. Wastell's [38] case study of the introduction of SSADM into an organisation and Hall and Fenton's [16] study of software metrics programmes are probably the best examples of such work.

This paper draws on classic theory from management ([1, 24, 25, 27, 37]) organisational theory [28], communications theory [17] and group theory ([4, 23]) to inform the observational data.

Methodology

The research consisted of three studies. Two were conducted at commercial software development organisations, the third with second year computing students during a nine month group project module at the University of Derby. The first commercial site produced tools for managing large corporate systems. Two teams were chosen consisting of between eight and twelve staff. It was not possible to contact all team members, but semi-structured interviews were conducted with seven members of Team A and five from Team B. The interviews were recorded and transcriptions made. Interviews lasted between 35 minutes and one and a half hours. A person centred approach [26] was taken to the interviews. A total of 207 hand written pages were transcribed. These were analysed according to Klein and Myers' [21] principles using discourse psychology [13] and narrative analysis [29]. 24 pages of evidence were codified qualitatively and organised into tables from which the framework of polarities emerged.

The second study observed student group projects. Eight projects were chosen for analysis by selecting a range of attainment and observed anxiety levels. The primary data was collected from the facilitated peer assessment exercise [32] that forms part of the assessment. This is a face to face meeting of the whole team with a tutor in which the group decides upon the allocation of available marks. The role of the tutor is to facilitate the students' decisions. These meetings were recorded. 88 type written pages of transcription were produced.

The final study took place at a commercial organisation producing embedded software for the communications industry. Construct repertory grids [20] were collected from nine members of staff and analysed using "idiogrid" [15]. Follow up laddering interviews were conducted which were recorded. These analyses provided corroborative evidence for the framework of polarities.

The evidence presented in this paper is taken from the initial study only, in order to comply with publication restrictions, although all the data analysed in the other two studies complemented that from the initial study.

Findings

The roles of participants are summarised in Table 1.

Table 1. Team Organisation.

Team	Pseudonym	Position
Team A	A1	Project manager
	A2	Senior designer
	A3	Recently joined programmer / designer (home based)
	A4	Designer / programmer
	A5	Designer / programmer
	A6	Designer / programmer
	A7	Designer / programmer (part time)
Team B	B1	Project manager (not interviewed)
	B2	Unofficial team leader
	B3	Designer / programmer
	B4	Designer / programmer
	B5	Designer / programmer (home based)
	B6	Designer / programmer

In study 1 the two groups were remarkably similar in many respects. Both products were providing services to help to manage large systems. Both groups consisted of eight to ten people. The majority of the people in both groups had been with the company for a long time (over 15 years in a number of cases). Most team members had received extensive training in quality processes. They had been through a period in the 1980s when conformance to quality processes had been rigorously audited. This had been followed by a period when little concern for process had been evident. All the staff, on both teams, felt that the initial period had been too rigorous, and that the subsequent period had been too lax. All stated that conformance to process was important, but that departure from documented process should be permissible. The similarities between the two groups helped highlight the contrasts between them and bring into sharp relief the effects of interpersonal differences. These differences are summarised in Table 2.

Table 2. Comparison of Team A and Team B.

Team A	Team B
Clear leadership (managerial / technology)	Managerial ambiguity
Structure provided by quality system and organisational hierarchy	Processes maintained by individuals, egalitarian decision making
Product scope explicitly managed to satisfy timescales	Product scope determined by pragmatic expediency
Acceptance of personal responsibility extended throughout project	Acceptance of personal responsibility restricted to own technical domain and competence
Active co-operation between team members.	Team members working more in isolation of each other
High level of mutual respect amongst team members	Tendency to be critical of other team members

The data was collected at the individual level. This enabled observations to be made at the team level and the interpersonal level. Comparing the two groups allowed group level observations to be made. These three levels are intrinsically interconnected. Individual behaviour affected other people and the group environment. Simultaneously, the environment influenced the behaviour of individuals. This made it impossible to consider the one without regard for the other. This suggests that understanding the inter-referential nature of the development process, rather than trying to disaggregate it, is essential.

At the personal level, staff expressed a unanimous commitment to producing quality products. This was often expressed as pride in the product, a desire to produce the best product possible, and an anxiety about introducing faults into the product. The existence, or even the perceived existence by customers of a fault, was taken personally. This was taken as personal criticism of professional competence.

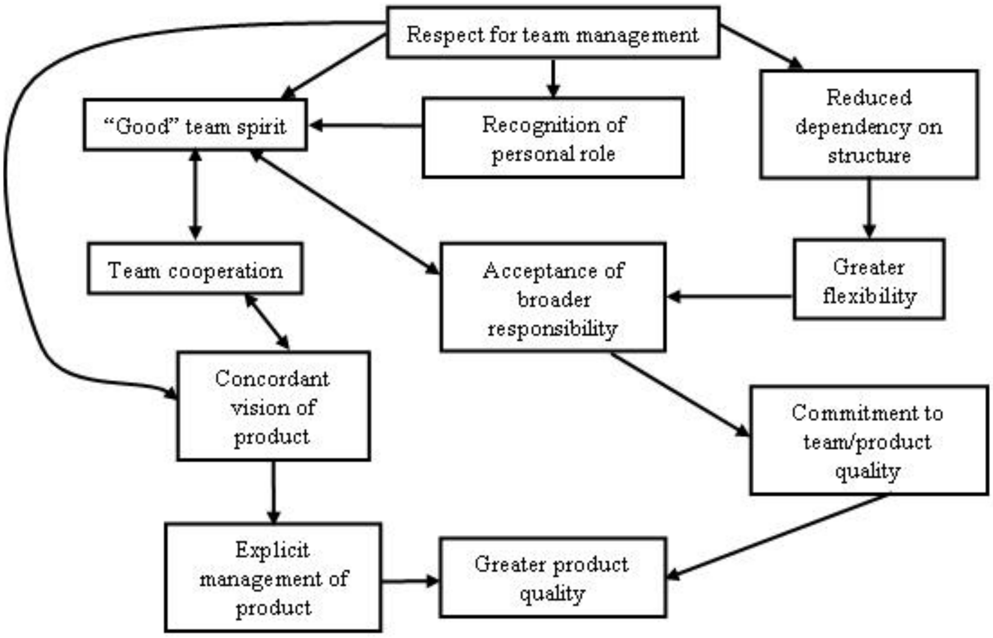
> "You have a pride in the product your doing even if you're just putting together somebody else's. You've still got a pride that you want it to work first time you don't want somebody to find a hole in it" (A3 14:50)

Team members' desire to do the best job possible was most often confounded by timescales. An interesting distinction between developers was their acceptance, compliance with, or rejection of this inevitable compromise. Team A adopted a covert strategy for managing this compromise by deliberately leaving aspects of the requirements specification vague. This allowed them to redefine the requirements to exclude some features thus enabling them to meet their targets. Team B employed a similar strategy, but as they did not have a requirements document they found it more difficult to manage the changes in product scope. Team A members seemed happier to accept that some features might be sacrificed in the released product, provided that, what was there was still thoroughly tested. In Team B there was more dissent and more discontent.

An obvious difference between the two groups was the style of management. Both teams had two people at the head of the team, but in the case of Team B this was far less hierarchical, and far more ambiguous. B1 was the appointed project manager of Team B, but was more concerned with external affairs. He exerted little influence over the team. He was seen as laissez faire, if not absentee. B2 assumed responsibility for day to day team leadership, but had not been appointed to the position. He felt that he had very little authority. The rest of the group accepted his coordination activities, but did not regard him as an authority figure. Consequently, Team B members were more emotionally distanced from each other, and had less involvement with each other technically than Team A.

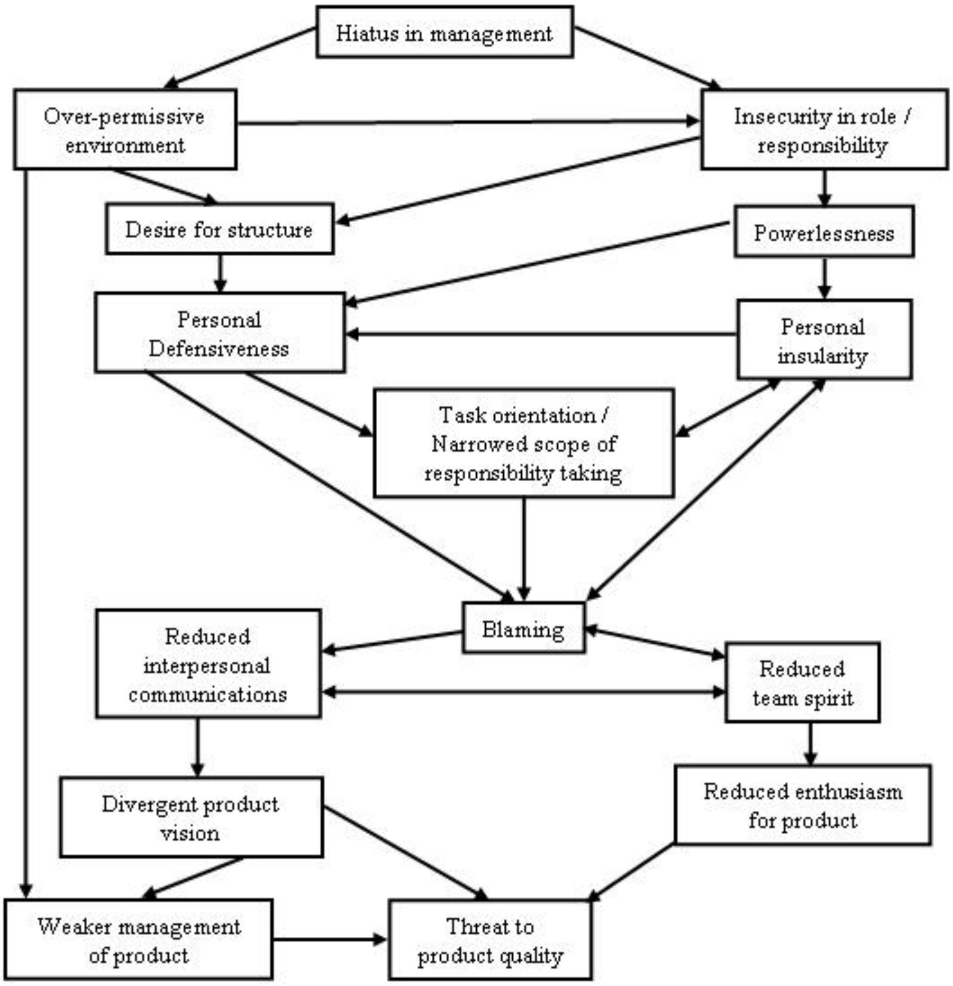
Table 2 seems to imply that each of the observations is distinct, clearly defined and independent of each other. In fact, a good deal of ambiguity is inherent and there is much interdependence between the concepts. Figures 1 and 2 model the dynamic interactions within each group.

Fig. 1. Team A Process Model.


Respect for team management
"Good" team spirit
Recognition of personal role
Reduced dependency on structure
Team cooperation
Acceptance of broader responsibility
Greater flexibility
Concordant vision of product
Commitment to team/product quality
Explicit management of product
Greater product quality

The similarities and differences between the teams pointed to six meta-observations. These have been termed "polarities of intra-group dynamics".

Fig. 2. Team B Process Model.


Hiatus in management
Over-permissive environment
Insecurity in role / responsibility
Desire for structure
Powerlessness
Personal Defensiveness
Personal insularity
Task orientation / Narrowed scope of responsibility taking
Blaming
Reduced interpersonal communications
Reduced team spirit
Divergent product vision
Reduced enthusiasm for product
Weaker management of product
Threat to product quality

Polarities of Intra-Group Dynamics

1. Acceptance of personal responsibility vs. blaming / distancing
2. Personal interests vs. group interests
3. Task orientation vs. social awareness
4. Leadership (initiative taking) vs. followership (passivity)
5. Coherence and consistency of conceptual vision vs. cognitive diversity and valency

6. Need for flexibility vs. need for structure

These polarities should not be seen as independent factors readily open to quantifiable analysis. As can be seen from the descriptions below, they are intricately connected and interdependent with each other. They are designed to provide a structure for observing relationships within teams and the preferences of individual team members.

Acceptance of personal responsibility vs. blaming / distancing

In any environment people are constantly faced with decisions about intervening in a situation. Their decision will require a judgement regarding their ability to influence the outcome, and their sense of culpability for the outcome. "Culpability" is a more appropriate word here than "responsibility" because it does not imply any sense of duty. If a person believes they can influence the outcome and that they also feel culpable in some way, they are likely to accept responsibility and take action. If they feel powerless to influence outcome, yet also feel culpable, they are most likely to feel anxiety. Blaming others is a way of protecting themselves from this anxiety. The polarity between accepting responsibility and blaming is therefore a means of assessing levels of anxiety in people and in teams. Generally, a manager would want her/his staff to accept responsibility and avoid blaming. However, people can feel stressed if they accept responsibility for things that they are powerless to affect and this can have a detrimental impact on the task.

Personal interests vs. group interests

It is difficult to define 'personal interest'. There is an argument that pure altruism is impossible because ultimately people only ever do something for our own benefit. However putting other colleagues' interests before their own, or being willing to put themselves out, are indicators of commitment to the team ideal. Team B expressed a lot of loyalty to the team. They had been together a long time. There was a commitment to the team's survival. There was however little evidence of helping each other out in the data. Their approach to work was more individualistic than Team A's. Team A on the other hand, seemed to have adopted a much more cooperative style of working.

People tend to put their own interests first when they have become disengaged from the team. This is a defensive reaction that often signals dissatisfaction with the organisation [1]. Personal ambition is however a powerful and dynamic motivator. Also a team which is too united can act in the best interests of the team at the expense of the rest of the organization [19]. As with the other polarities, this polarity can play both ways.

Task Orientation Vs Social Awareness

Task orientation versus people orientation was noted as an important dimension in work groups by Fiedler et. al. [14] and Hersey and Blanchard [18] in the context of leadership and is inherent in Belbin's team roles taxonomy [3]. In this research, evidence to support the significance of the team as a social unit is observed in the process models. Team A had a strong team orientation whereas Team B members were more individualistic. B3 for example was detached, and somewhat cynical. Team meetings were rare in Team B.

> "B1, God bless him, is not really into meetings." (B2 18:37)

There was a relationship between peoples' willingness to help others and the degree to which people focussed on their own work. Argyris (op. cit.) refers to this as a "defensive routine" in which a team member retreats from active participation with the group. The consequence of this is a reduced level of interpersonal communications. On the other hand, too much socialising may result in less work actually being done.

Leadership (initiative taking) vs. followership (passivity)

Leadership has long been recognised as an attribute of management theory.

> " In Stogdill's (1974)* marathon survey of over 3000 studies of leadership …the influence exerted by others emerges as a consistent pattern" [34]

It was unsurprising therefore that it was highly significant in this study.

The concept of leadership is intimately associated with ideas of authority and power. It is often only applied to managers. In this polarity, the concept of leadership has been extended to include initiative taking, hence enabling it to be applied to all group members, and reducing the emphasis on authoritarian power in favour of personal authority.

The two groups provided a vivid contrast in management style. This affected the interrelationships within the teams. Team A's project manager (A1) was actively involved in managing timescales and staff development. He was seen to be an active member of the group. More significant however, he was seen as, and saw himself as holding the boundaries for the group [33]. This was not just in terms of timescales, but, more importantly, the vision of the product and process definition. A2's role supported A1. She provided technical leadership and held a more detailed vision of the product. She also had the confidence to challenge A1 when necessary, and was encouraged to do so.

By modelling the acceptability of challenging authority, A1 and A2 encouraged initiative taking in other team members.

In contrast, in Team B, B1's leadership style was laissez-faire. Consequently there was a vacuum of leadership that B2 attempted to fill. However, B2 felt that he did not have the authority to lead, so his leadership lacked confidence. The team attempted to operate on consensus which worked to a certain extent, but resulted in a greater divergence of product vision, less effective communications and lower levels of initiative taking. Initiatives were still taken in Team B, but they were limited to

* Stogdill R.M, 1974 Handbook of Leadership Free Press New York

circumstances in which people felt secure. More challenging initiatives were avoided as people did not feel confident to take risks.

In a systems development context, initiative taking is essential. However, if a person goes beyond their authority, or acts without coordinating with other team members, this can be highly counterproductive.

Coherence and consistency of conceptual vision vs. cognitive diversity and valency

This concept was alluded to in the previous paragraph. It is, by nature intangible so difficult to provide explicit evidence. Certainly, the range of concerns expressed in Team B was broader than Team A. B2 was concerned with release processes and planning, B3 with testing and support, B4 with requirements valency, B6 with team membership and communication and B5 with getting his own work right. B4 expressed problems with cognitive diversity in the following passage:-

> "…problem of trying to control a group of people that aren't all in the same place so you have a problem of trying to communicate with this diverse set…you need a slightly closer group of people there to throw ideas between yourself." (B4 3:04)

In Team A, A2 was the final arbiter of technical decisions and hence, ultimately, holder of the conceptual vision. It was not clear where this resided in Team B.

Clearly it is essential that a team has a consistent vision of what is expected and the direction the team is going in. However, divergent thinking can lead to greater clarity of understanding and avoid groupthink [19].

Need for flexibility vs. need for structure

Several people in each of the teams expressed needs for both more structure and more flexibility. In Team A structure was provided by the process definition and by the reporting hierarchy. Requirements however were fixed, until they were explicitly changed. A2 managed requirements as a deliberate strategy for meeting timescales. Definitions of some requirements were deliberately kept vague so that they could be redefined to exclude features. The formal development process was also applied flexibly. Even the management hierarchy was applied flexibly in that authority was open to challenge. In Team B, both B2 and B3 expressed a desire for more structure. B2's need drove him to developing a project plan and producing some checklists. B5 complained about trying to hit a moving target. In his case the lack of overt structure led to retreat into task and followership.

Structure and flexibility are inherently contradictory and ambiguous. The more structure that exists, the less flexibility is possible and hence the less initiative taking. However a lack of structure can lead to insecurity and anxiety that militates against taking initiatives and risk. This is a fundamental dilemma with which organisations and managers must struggle.

Discussion

These polarities are associated with individuals within the team. They are both influenced by and influence the organisation and culture of the team and other team members. It is perhaps best to consider the team as a system. The system is affected by changes in the external environment and also by internal changes. Furthermore, whilst some fairly consistent patterns may exist, the whole system is dynamic. This means that if problems with the operation of the team exist, potential intervention should be considered by how they might affect the system. The polarities are value free in that they make no judgement of good or bad, right or wrong. All such judgements are context dependent and require subjective assessment.

The polarities are a means of analysing a system. As such the polarities interact with each other in the same way that elements within the system interact with each other. For example, task orientation may limit the scope of a person's conceptual vision, and lead her / him to expect other people to provide resources and information deemed outside the self defined scope of their responsibility. Having chosen a passive role for themselves, they might criticise other people if the expected resources is not provided. Appreciating interconnectivities of this type is an objective of the analysis rather than a distortion of it.

Conclusion

It has long been recognised that personal [7, 39], interpersonal [5] and group [12] factors affect the systems development process, but little work has been done in understanding the process by which this occurs. This research suggests that personal, interpersonal and organisational factors are intricately intertwined. Understanding these mechanisms is essential if one wishes to intervene. All too often in the past researchers and academics have proposed solutions to improving software quality or productivity without appreciating the complexity of the environment (including the interpersonal environment) in which they were operating. Successes have been claimed for technologies or methodologies that cannot be replicated in different contexts. This paper suggests that this is because the development environment is a system in which people and their interrelationships are integral. The suggested framework of polarities provides a means of observing the system in order to understand it. In this way, the framework can help to identify how the potential of the team is dissipated by the intra-group dynamics and interventions can be assessed in terms of the affects they have on these dynamics. The framework provides a solid foundation for evaluating interventions and contributes towards a more comprehensive understanding of the software and systems development process.

References

1. Argyris, C., Defensive Routines, in *Organization Theory*, D.S. Pugh, Editor. 1985, Penguin, Harmondsworth Middlesex. 416-432.
2. Barki, H. and J. Hartwick, Interpersonal Conflict and Its Management in Information Systems Development. MIS Quarterly, 2001. **25**(2): 195-228.
3. Belbin, R.M., Management Teams : Why they Succeed or Fail. Heinemann, London (1981).
4. Bion, W.R., Experiences in Groups. Routledge, London (1961).
5. Boehm, B.W. and P.N. Papaccio, Understanding and Controlling Software Costs. IEEE Trans. On Software Eng., 1998. **14**(10): 1462 -- 1477.
6. Cronan, T.P. and T.L. Means, System Development : An Empirical Study of User Communication. Data Base, 1984. **15**(3): 25 -- 33.
7. Curtis, B., The Impact of Individual Differences in Programmers, in *Working With Computers: Theory Versus Outcome*. 1988, Academic Press, London. 279-294.
8. Curtis, B. Three Problems Overcome with Behavioral Models of the Software Development Process. in *11th Int. Conf. on Software Management*. 1989: ACM.
9. Curtis, B., H. Krasner and N. Iscoe, A Field study the Software Design Process for Large Systems. Comms. of the ACM, 1988. **31**(11): 1268 -- 1287.
10. Curtis, B. and D. Walz, The Psychology of Programming in the Large : Team Organizational Behaviour, in *Psychology of Programming*, Samurcay, Editor. 1990, Academic Press, London. Excellent article on the problems of communication in large s/w development. Good references.
11. Dagwell, R. and R. Weber, Systems Designers' User Models : A Comparative Study and Methodological Critique. Comm. Of the ACM, 1983. **26**(11): 985 -- 997.
12. DeMarco, T. and T. Lister, Peopleware : Productive Projects and Teams. Dorset House, New York (1987).
13. Edwards, D., Discourse and Cognition. Sage, London (1997).
14. Fieldler, F.E., M.M. Chemers and L. Mayhar, The Leadership Match Concept. Wiley, New Jersey (1978).
15. Grice, J.W., Idiogrid: Software for the management and analysis of repertory grids. Behavior Research Methods, Instruments, & Computers, 2002. **34**(3): 338 - 341.
16. Hall, T. and N. Fenton, Implementing Effective Software Metrics Programmes. IEEE Software, 1997(3): 1-14.
17. Hargie, O.D., Interpersonal Communications : A Framework Theory, in *A Handbook of Communication Skills*, O.D. Hargie, Editor. 1986, Routledge, London. 29 -- 63.
18. Hersey, P. and H. Blanchard, Management of Organisational Behavior: Utilising Human Resources. Prentice-Hall, Englewood Cliffs NJ (1977).
19. Janis, I., Groupthink. Psychology Today, 1971: 43 -- 76.
20. Kelly, G.A., The psychology of personal constructs. Routledge, London (1991).
21. Klein, H.K. and M.D. Myers, A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly, 1999. **23**(1): 67-94.
22. Krasner, H., B. Curtis and N. Iseco. Communication Breakdown and Boundary Spanning Activities on Large Programming Projects. in *Empirical Studies of Programmers (2nd Workshop)*. 1987. Washington D.C.: Ablex.
23. Lewin, K., Field theory in Social Science. Harpers& Row, New York (1951).
24. Mayo, E., The Human Problems of an Industrial Civilization. Macmillan, Cambridge Mass. (1946).
25. McGregor, D., The Human Side of Enterprise. McGraw-Hill, London (1960).
26. Mearns, D. and J. McCloud, A Person Centred Aproach to Research, in *Client Centred Therapy and the Person Centred Approach*, J.M. Shlien, Editor. 1984, Harper and Bros., New York. 370-389.

27. Menzies-Lyth, I., The Functioning of Social Systems as a Defence Against Anxiety, in *Containing Anxiety in Institutions*, I. Menzies-Lyth, Editor. 1959, Free Association Press, London. 43-85.
28. Perrow, C., A Framework for Comparative Analysis of Organizations. American Sociological Review, 1967. **32**(2): 194 -- 208.
29. Riessman, C.K., Narrative Analysis. Qualitative Research Methods. Vol. 30. Sage, Newbury Park C.A (1993).
30. Rockwell, R., Software Development is a Communications Process. Lecture Notes in Computer Science, 1992. **635**: 61 -- 62.
31. Rodwell, J.J., R. Kienzle and M.A. Shadur, The Relationships Among Work-Related Perceptions, Employee Attitudes, and Employee Performance: The Integral Role of Communications. Human Resources Management, 1998. **37**(3&4): 277 - 293.
32. Rosen, C. Individual Assessment of Group Projects in Software Engineering : A Facilitated Peer Assessment Approach. in *9th International Conferece on Software Engineering Education*. 1996. Daytona Beach Florida.
33. Rosen, C.C.H.: The Facilitation of Experiential Training Groups and Non-Task Oriented Groups, in *Adult Education*. 1992, Manchester University: Manchester.
34. Rosenfeld, R.H. and D.C. Wilson, Managing Organisations. McGraw-Hill, London (1999).
35. Sawyer, S., Effects of Intra-group Conflict on Packaged Software Development Team Performance. Information Systems Jou., 2001. **11**(2): 155-178.
36. Sawyer, S. and P.J. Guinan, Software Development: Processes and Performance. IBM Systems Jou., 1998. **37**(4): 553 -- 569.
37. Taylor, F., Scientific Management, in *Organisation Theory*, D.S. Pugh, Editor. 1912, Penguin, London. 275 - 295.
38. Wastell, D.G., The Fetish of Technique: Methodology as a Social Defence. Information Systems Jou., 1996. **6**: 25 -- 40.
39. Weinberg, M.W., The Psychology of Computer Programming. Van Nostrand Reinhold, New York (1971).
40. Zmud, R.W., Management of Large Software Development Effort. MIS Quarterly, 1980. **6**: 45-55.