From Individual Contribution to Group Learning:
the Early Years of Apache Web Server

Hala Annabi[1], Kevin Crowson[2], Robert Heckman[2]

1 University of Washington, The Information School
Box 352840, Seattle, WA 98195-2840
(206) 616-8553, hpannabi@u.washington.edu
2 Syracuse University, School of Information Studies
Syracuse NY 13244, (315) 443- 1676
crowston@syr.edu, rheckman@syr.edu

Abstract. Open Source Software (OSS) groups experience many benefits and challenges with respect to the core group's effectiveness. In order to capitalize on the benefits and minimize the challenges, OSS groups must learn not only on the individual level, but also on the group level. OSS groups learn by integrating individual contributions into the group's product and processes. This paper reports on the characteristics of the learning process in OSS groups. The study utilized an embedded single case study design that observed and analyzed group learning processes in the Apache Web server OSS project. The study used learning opportunity episodes (LOE) as the embedded unit of analysis and developed and utilized three content analytic schemes to describe the characteristics of the learning process and the factors affecting this process.

1   Introduction

Open Source Software (OSS) groups, by their very nature (distributed, often voluntary, and having a potentially large number of submitted bug reports and fixes from outside of the core-development team) experience many benefits and challenges with respect to the core group's effectiveness. Since OSS teams are distributed, they have access to a larger pool of experts, have better load balancing, and are able to train developers (Grinter, Herbsleb, & Perry, 1999). However, the geographical distance between the members challenges the groups in the following ways: miscommunication, problems in product and process management, coordination difficulties, low self-efficacy, low self-sufficiency, and knowledge

management problems (Bélanger & Collins, 1998; Carmel & Agarwal, 2001; Herbsleb & Montra, 2001; Jarvenpaa & Leidner, 1999; Kraut, Steinfield, Chan, Butler, & Hoag, 1999).   These challenges are especially critical in the case of software development where communication and coordination are paramount. Large scale software development requires knowledge from multiple domains, thinly spread among different developers (Curtis et al. 1988). Thus this is an environment requiring a high degree of knowledge integration and coordination of efforts on the part of multiple developers (Brooks 1975). This is of particular importance for OSS teams as contributions come not only from the core developers but from all the co-developers and active users as well.

Accordingly, to minimize the negative effects mentioned above, OSS core groups must learn effective communication and coordination practices suitable to their new environment. In their study of distributed cross-functional teams, Robey et al. (2000) suggest that to be successful, distributed groups must learn. This learning has to occur on both the individual and group levels (Senge 1990; Lin and Lin 2001). However, research and practitioner communities know little about the processes of learning suitable for distributed teams (Orlikowski, 2002; Robey et al. 2000). Thus it is important to first understand the learning processes of distributed groups. According to Maier, et al. (2001), "Knowledge about the process, or the know how, of learning facilitates corrections that simulate or accelerate learning" (pg. 16).

The study had two objectives. The first was to address the gap in the literature and develop a theoretical approach to study learning process distributed groups. The second was to describe the learning processes in OSS groups and identify factors that enhance or impede this process. This paper will report on the second objective.

2   Learning in OSS Groups

To study learning in OSS groups, we draw on Huber's definition of learning: "An entity learns if… the range of its ***potential behaviors is changed***" (Huber, 1991). The term behavioral potential recognizes the fact that not all outcomes of learning will be observable immediately in behavior. Rather, they will only be observable if and when appropriate circumstances arise. For example, airline pilots train to handle emergencies, but are rarely called upon to exhibit these behaviors.

To conceptualize a group's behavioral potential, we draw on Grant's (1996) knowledge-based view of the firm. In this view, a group is a structure for integrating the knowledge of its members. A group creates coordination mechanisms, such as rules and routines to economize on communication, knowledge transfer and learning (Grant 1996). In this view, rules and routines structure how members coordinate their tasks efficiently and effectively. Therefore, changes in the behavioral potential of a group will be observable in changes in these rules and routines (Hayes and Allinson, 1998).

Argyris and Schön (1978) and Brown and Duguid (1991) suggest that for a group to create or change rules and procedures it is key that its members have shared

understanding or shared mental models. Shared mental models, as defined by Cannon-Bowers et al. (1993), "are knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and in turn, to coordinate their actions and adapt their behavior to demands of the task and other team members" (pg 228). The importance of shared mental models comes from the fact that for rules and routines to be effective coordination mechanisms, they have to be interpreted consistently on the group level. Without shared mental models individuals may interpret tasks differently based on their backgrounds (Dougherty, 1992). Shared mental models are manifested in common language, communication patterns, and consistency in interpreting and executing the rules.

In summary, we define group learning as *the process by which group members share knowledge and information and integrate it into the group's implicit and explicit rules, leading to changes in the behavioral potential of the group*. Group learning is operationalized as changes in explicit and implicit rules. We focused on changes in rules and procedures as specific indicators of explicit rules, and on changes in shared mental models as specific indicators of implicit rules. The following are the specific research questions of the study:

RQ1: What are the characteristics of group learning process in OSS groups?
　　　More specifically:
　　　RQ1a: How do OSS groups change rules and procedures?
　　　RQ1b: How do OSS groups change shared mental models?
RQ2: What are the factors that impede or enhance group learning?

3   Research Framework

In order to guide data collection and analysis, we integrated the definition of group learning, and concepts from multiple area of study including organizational learning (OL), group research, shared mental models (SMM), and asynchronous learning networks (ALN) to develop an initial theoretical framework. The initial framework was modified and refined as more data was analyzed. We represent the learning process of a group in terms of the input-process-output framework illustrated in figure 1. The model includes group structure, organizational level, and group design inputs. These inputs affect the nature of learning opportunity episodes (LOE) (triggers, process and outcomes) in the group which include the group learning process. The learning process results in group and individual learning. The framework indicates that outcomes of learning recursively affect group structure inputs.

Input Variables:

Input variables in this model include organizational context and group design variables as suggested by both Hackman (1986) and Gladstein (1984). Group structures include rules, shared mental models, and role structure. Group structure input variables are affected by outcomes of the learning process. Organizational context represented by corporate participation, which was controlled for in this case study (one case with no corporate participation). Additionally, the framework included group design variables, represented by group composition and task to

illustrate effects of members' skills and knowledge and how it may influence the group process. This is also influenced by nature of task.
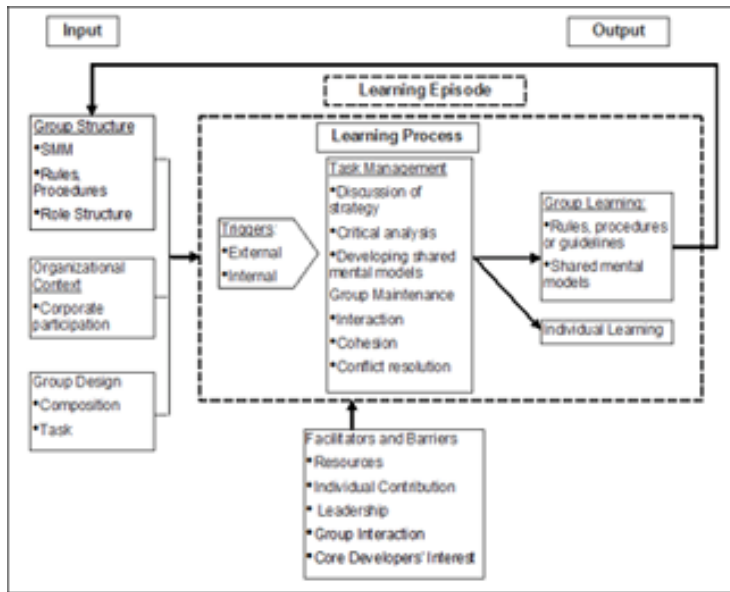

Figure 1 Refined Theoretical Framework for Learning Process in OSS Groups

Learning Opportunity Episodes (LOE):

The theoretical framework conceptualized group learning using an episodic view to bound the phenomenon as suggested by Miles and Huberman (1994). A ***learning opportunity episode (LOE) is a group event that occurs over time as a result of a learning trigger. It may or may not lead to changes in the behavioral potential of the group***. The framework suggests that input variables affect group LOEs that include learning triggers and learning process, and that episodes may or may not lead to learning.

Learning Triggers:

Walton and Hackman (1986), propose that all groups must satisfy a number of important group functions (social; interpretive, task, agency, and regulative). If any of the group functions are not met, or can be met more effectively or efficiently, the group has an opportunity to learn. We refer to this opportunity as a learning trigger. As described in the results section below, an important finding of this study was a better understanding of the nature and impact of different types of learning trigger. At this point we will briefly note that learning triggers differed in terms of the type (internal or external) and focus (product or process). Internal triggers occur within

the core group (e.g. errors, inconsistent interpretations). External triggers come from the external environment or the core (e.g. new technology, user requests).

Group Learning Process:

Learning is the process by which the group's potential behavior changes. We conceptualized the learning process in terms of **task management** and **group maintenance** functions. Figure 1 indicates that task management includes three aspects of group behaviors discussion of strategy, critical analysis, and developing shared mental models. Group maintenance behaviors included interaction, cohesion, and conflict resolution behaviors.

Facilitators and Barriers

Research question 3 identifies factors that impede or enhance group learning process. We included factors identified in out empirical analysis in the framework as the facilitators and barriers to LOE. These factors include resources, leadership, individual contribution, group interaction, and core developers' interests. Space does not permit a detailed presentation of these factors in this version of the paper.

Process Output

The focus of this study was on changes in rules as an explicit indicator of learning, and changes in shared mental models as an implicit indicator of learning. Observation of these outcomes was used to assess whether or not a particular episode resulted in group learning. Another outcome of group learning identified in the conceptual framework is individual learning. Due to the retrospective nature of this study, individual learning was beyond the scope of this study.

4   Methodology

This study employed a qualitative case study design to better understand the phenomenon of learning in a work setting as suggested by Miner and Mezias (1996). As Yin defines it, a case study is "an empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used" (Yin, 1984, pg. 23). More specifically, we employed a single embedded case study design, based on theoretical sample strategy for case selection. The case for this study is the Apache httpd Project. The embedded unit of analysis LOE defined earlier.

Theoretical selection criteria in this study were group size and group effectiveness. We selected a group having more than seven core developers, a lower-limit sample as suggested by Hare (1976). The literature suggested that learning leads to effectiveness (Maier et al., 2001). This increases our chances for observing learning, the research selected an effective group previously identified as successful in the OSS literature: Apache Web Server.

A continuation of the httpd server developed by Rob McCool and the National Center for Supercomputing Applications (NCSA) "the Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems" (Apache.org). After McCool left NCSA in 1994 eight of the

developers started collaborating via private e-mail in 1994 and in early 1995 established a Web presence and mailing list to continue their development effort. The Apache Web server has been the most widely used Web server on the Internet since 1996, holding 64% market share in 2003 according to Netcraft Web Server Survey (http://news.netcraft.com). We observed the Apache httpd project between its inception (February 1995) and the first stable release, Apache 1.0 (December 1995); tracking the group movement from alpha to beta to stable.

We chose to bound the learning process using LOE as suggested by Miles and Huberman (1994). Behavioral potential is manifested in changes in explicit rules (from which we focused on changes in rules and procedures) and implicit rules (from which we focused on shared mental models). We considered a LOE to have no change if one month passed without a direct response to that trigger (the average between LOE times four). Explicit learning outcome was measures by identifying a change in rules or procedures in the group. Implicit learning outcome was measured by identifying group shared mental models evident in change in the code, change in agreement or course of behavior.

Figure 2 illustrates the nature of the LOE. An LOE can be selected by identifying learning triggers, indicators of learning process, or identifying explicit changes to rules. Once any of these elements was identified as being part of the LOE the related interaction messages and documentation were collected. The interaction data was analyzed using Atlas-ti, and the documentation was reviewed.
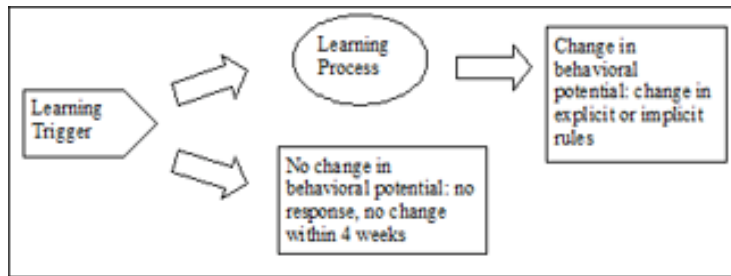

Figure 2 Learning Opportunity Episodes

4.1   Data Collection and Analysis

The study collected data surrounding each LOE from multiple related sources including interaction data, documentation, and primary and secondary source interviews. Interaction and documentation data was publicly available on Apache.org. The study also included one e-mail interview with a core developer and secondary interviews with and articles written by core developers.

We used three content analytic schemes to analyze group interaction data from mailing list. One scheme analyzed group learning process, the second the learning

triggers, and the third LOE. Interviews and documentation served to corroborate findings from interaction data. The content analysis process followed Miles and Huberman's (1994) interactive model. We started the data analysis using initial content analytic schemes, but modified these schemes as new indicators emerged. Intercoder reliability tests were conducted and modifications made to the content analytic schemes until the various coders reached acceptable intercoder agreement (Baker-Brown et al., 1990) (LOE scheme (containing learning triggers scheme): 89.6% agreement; learning process: 91% agreement.)

5   Results and Discussion

The Apache group had no formal role structure, procedures, or guidelines to guide group membership, rules for task management, coding style and structure, system requirements or work plans at the start of the project in February of 1995. Individuals interested in the project joined a mailing list (new-httpd@hyperreal.com) where members contributed ideas, code, bug report and bug fixes based on needs and interests. During the period of observation, 6,649 messages were posted to the mailing list, and the group produced 38 versions of Apache as a result of 236 of patches, bug fixes, bug reports, and documentation. Figure 3 displays activity level (number of postings in the mailing list), project's stage of development and major releases overtime.
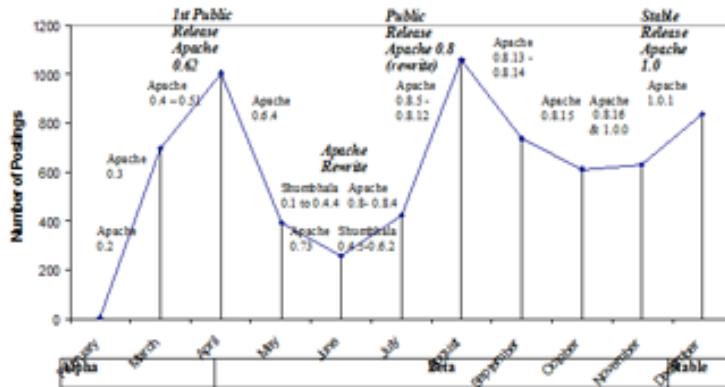

Figure 3 Group Activity in Mailing Lists over Time

Messages posted to the mailing list came from eight core developers and 46 active (co-developers) and occasional (active users) contributors[1]. Code submissions

---

[1] Note: the number of co-developers and active users is estimated based on our analysis of messages in learning-opportunity episodes and not the total number of messages during this period

were made by the eight core developers and 24 co-developers and active users. During this period of observation we identified 178 LOE

Due to space limitations, in remainder of this section we will present selected findings that illustrate the nature of learning opportunity episodes, and the events that trigger them.

1. Group learning requires interaction

Figure 4 suggests that the distribution of LOE is correlated with the distribution of level of activity over time (number of LOE was scaled up in figure 4). This further suggests that level of group activity is important for group learning. Periods marked by limited group activity (interaction between the group members) are also associated with periods of fewer learning opportunities. A significant example of this occurred in the period between May and July. During this period, the level of group activity (measured by the number of postings to the mailing list) suggested that there was little group activity taking place. However, documentation and the content of the messages revealed that individuals were independently developing code. These individuals might have learned and gained insight about the code, but the group did not learn as a result of individual knowledge. In fact, the group witnessed the least number of learning opportunity episodes during the period. Not until the group started to interact again to integrate the individual contributions did the group learn. To integrate the individual contributions, the group developed new shared mental models and coordinating mechanisms, as evident in the increase of learning in July. .
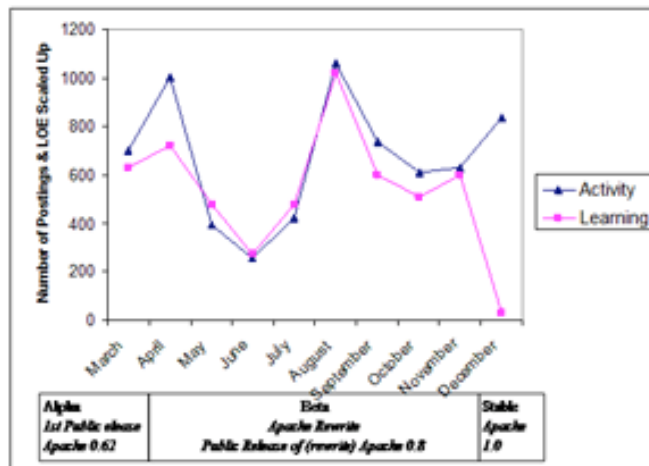

Figure 4 Distribution of Learning Opportunity Episodes vs. Level of Activity Over Time

2. *A majority of learning opportunities had a product focus*

In our analysis we discovered that learning opportunities had a focus on either developing the group ***product*** (e.g. writing code and documentation), developing ***processes*** for producing the product (e.g. contribution guidelines, voting procedures), or developing ***both*** product and process. Table 1 indicates that 72% of the episodes focused on developing the product. In comparison, 56% of episodes are focused on the process. This suggests that the group activities are less focused on developing processes and more focused on developing the product.

Table 1 Focus and Type of Learning Opportunity Episodes

Focus of Episode (Product vs. Process)	Type of Learning (Shared Mental Models or Rules)			Total: focus of episode
	SMM	**Rule**	**Both**	
Process	11	21	18	50
Product	64	4	11	79
Product and Process	10	0	39	49
Total: Type of Learning	85	25	68	178

3. *Some opportunities produced no learning*

As presented in table 2, of the 178 episodes collected, 150 led to change in behavioral potential and 28 led to no change. Most group activities provided opportunities for developing shared mental models of product and process. It is no surprise that 91% of episodes leading to learning resulted in developing shared mental models as indicated in table 2. Only 9% of learning outcomes strictly developed rules and guidelines. Developing rules and guidelines is present in 38% of episodes leading to learning. However, it is important to note that even the episodes that only lead to changes in rules displayed SMM behaviors as will be discussed later in this paper.

Table 2 Learning Outcomes

Outcome	Number	Percentage of Total Number of Episodes	Percentage of Episodes that Led to Learning
No learning	28	16%	0%
Change in shared mental model	93	51%	62%
Change in rule	14	8%	9%
Change in both rule and shared mental model	43	25%	29%
	178		

4. Product-focused episodes produce different learning than process-focused episodes

Product- focused episodes appeared to have different learning outcomes than do episodes focused on process or both process and product. Process-focused episodes, for example, have a higher probability in leading to no learning (32%) than product-focused episodes (11%) or both process and product-focused episodes (6%). This suggests that the group is more likely to ignore learning triggers that are process-focused and respond to triggers that are product-focused. Also, process-focused episodes are more likely to lead to both shared mental models and rules (33%) than product-focused episodes (13%).

Table 3 Learning Opportunity Episodes Focus and Learning Outcomes

Focus of Episode (Process or Product)	Learning Outcome				Total of focus of episode
	No Learning	Shared Mental Models	Rule	Both Rules and Shared Mental Models	
Process	16	10	10	14	50
Product	9	63	2	5	79
Product and Process	3	20	2	24	49
Total Learning Outcome	28	93	14	43	178

5. The majority of learning triggers were internal

We identified a total of 13 different types of internal and external triggers (table 4). In the 178 LOE 75% of the learning triggers were internal and 25% were external. External learning triggers were more likely (23%) to lead to no learning than internal learning triggers (13%). Internal triggers are more likely (88%) to generate complex learning episodes. This suggests that this group was less attentive to learning stimuli generated by users on the periphery.

Table 4 Frequency of Learning Triggers

Learning Trigger	Indicator	Number	Percent
External		44	25%
	User need or request*	13	7%
	New technology*	3	2%
	External expectation/ requests *	11	6%
	Offer to contribution or new member (Grant, 1996)	6	3%
	Error*	11	6%
Internal		134	75%
	Misrepresentations or gaps in understanding*	29	16%
	Conflict (Gladstein)	0	0%
	Lack of resources (Hackman)	0	0%
	Error (Argyris & Schön, 1978)	25	14%
	Share information of code and product status*	35	20%
	Efficacy of the process (Anderson et. al.)	17	10%
	Innovation in the process*	16	9%
	Innovation in the product*	12	7%

6. The group devised its own learning mechanism

An important learning trigger was the mechanism developed by the group to share information on code and product status. This mechanism was developed to ensure every member had the same understanding. A member, (often a release coordinator) would provide the group with a summary of the code and the patches with the intention of generating a discussion to clarify understanding. Other members contributed information to correct errors or omissions provided in the summary. This was an important mechanism for learning, as it addressed shared mental models about the code and about who is doing what, as well as providing grounds for deciding on to-do lists and timelines. This trigger generated 20% of all learning episodes (the largest percentage for any one learning trigger) and 35% of complex episodes (episodes focusing on both product and process, and both shared mental models and rules).

6   Conclusion

This study had both theoretical and pragmatic implications. The theoretical implications of this research raised new insights into the study of learning in distributed work groups, and addressed several issues concerning the definition, content, outcome and process of learning in these groups. Extending what prior research suggests (Simon, 1991; Grant 1996)), the study discovered that individual

learning is not sufficient to change group behavior. It must be integrated into group process and product for learning to occur. This integration is accomplished through an information intensive process that relies heavily on building shared mental models.

We also discovered that learning episodes were more likely to be triggered when the group was focused on its primary mission: (that of writing code) than when it was focused on working processes. This may not be surprising, given that in voluntary organizations such as open-source projects, many important group functions (e.g. role, status) are dependent on the action of writing code. A result that is perhaps more surprising, given claims about the egalitarian and democratic nature of open-source communities, is the fact that learning triggers originating in the periphery of the community where less active users reside were less likely to be attended to by the core. These findings suggest that deeper investigation of the social dynamics of open-source projects may reveal surprising results. Project leaders and management of distributed groups in general, may use findings from this study to improve the management and design of their groups.

References

Argyris, C. and Schön, D. (1978) *Organizational Learning,* London: Addison-Wesley.

Baker-Brown, G., Ballard, E., Bluck, S., De Veries, B., Suedfeld, P., & Tetlock, P. (1990). *Coding Manual for Conceptual/Integrative Complexity*: University of British Columbia University of California, Berkeley

Bélanger, F., & Collins, R. (1998). Distributed Work Arrangements: A Research Framework. *The Information Society, 14*(2), 137–152.

Brooks, F. P., Jr. (1975). *The Mythical Man-month: Essays on Software Engineering*. Reading, MA: Addison-Wesley

Brown, J. S., & Duguid, P. (1991). Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. *Organization Science, 2*(1), 40–57.

Cannon-Bowers, J. A., & Salas, E. (1993). Shared Mental Models in Expert Decision Making. In *Individual and Group Decision Making,* Castellan, N.J. (Ed.). Lawrence Erlbaum Associates: Hillsdale, NJ, 221-246.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *CACM, 31*(11), 1268–1287.

Dougherty, D. (1992). Interpretive Barriers to Successful Product Innovation in Large Firms. *Organization Science, 3*(2), 179–202.

Grant, R. M. (1996). Toward a knowledge-based theory of the firm. *Strategic Management Journal, 17*(Winter), 109–122.

Gladstein, D. (1984). Groups in Context: A model of task group effectiveness. *Administrative Science Quarterly, 29*(4), 499-517

Grinter, R. E., Herbsleb, J. D., & Perry, D. E. (1999). The Geography of Coordination: Dealing with Distance in R&D Work. In *Proceedings of the GROUP '99 Conference* (pp. 306–315). Phoenix, Arizona, US.

Hackman, J. R. (1986). The design of work teams. In J. W. Lorsch (Ed.), *The Handbook of Organizational Behavior* (pp. 315–342). Englewood Cliffs, NJ: Prentice-Hall.

Hayes, J. and Allison, C. (1998) Cognitive Style and the Theory and Practice of Individual and Collective Learning in Organizations, *Human Relation,* 51(7).

Herbsleb, J. D. & Moitra, D. (2001). Global Software Development. *IEEE Software* (March/April), 16–20.

Hala Annabi 15/1/06 4:21 PM
Formatted: English (US)

Huber, G.P. (1991). Organizational Learning: The contribution processes and the Literatures. *Organization Science.* 2(1): 88-115.

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and trust in global virtual teams. *Organization Science, 10*(6), 791–815

Kraut, R. E., Steinfield, C., Chan, A. P., Butler, B., & Hoag, A. (1999). Coordination and virtualization: The role of electronic networks and personal relationships. *Organization Science, 10*(6), 722–740.

Lin, F. & Lin S. (2001) A Conceptual Model for Virtual Organizational Learning, *Journal of Organizational Computing and Electronic Commerce,* Vol. 11(3), Pg. 155-178.

Maier, G.W., Prange, C. and Rosenstiel, L. 2001, Psychological Perspective of Organizational Learning, in M. Dierkes, A. Berthoin Antal, J. Child, and I. Nonaka (eds.), *Handbook of Organizational Learning and Knowledge*, New York, Oxford Press, 14-34.

Miles, M. B., & Huberman, A. (1994). *Qualitative Data Analysis: An Expanded Sourcebook.* Thousand Oaks, CA: Sage

Miner, A.S. and Mezias, S.J. (1996), Ugly Duckling No More: Pasts and Futures of Organizational learning, *Organization Science,* 7(1): 88-99.

Orlikowski, W. J. (2002). Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. *Organization Science, 13*(3), 249–273

Robey, D., Khoo, H. M., & Powers, C. (2000). Situated-learning in cross-functional virtual teams. *IEEE Transactions on Professional Communication* (Feb/Mar), 51–66.

Senge, P. (1990). *The Fifth Discipline: The art and practice of the learning organization*. London: Random House.

Simon, H. A. (1991). Bounded Rationality and Organizational Learning. *Organization Science, 2*, 125-134

Walton, R. E., & Hackman, J. R. (1986). Groups Under Contrasting Management Stratedies. In P. S. G. a. Associates (Ed.), *Designing Effective Work Groups* (pp. 168-201). San Francisco, CA: Jossey-Bass.

Yin, R. (1984). *Case Study Research*. Beverly Hills, CA: Sage Publications Inc.