How Does Radical Collocation Help a Team Succeed?

Stephanie Teasley
School of Information
University of Michigan
1075 Beal Avenue
Ann Arbor, MI 48109-2112
+1 734-763-8124
fax: +1-734-936-3168
steasley@umich.edu

Lisa Covi
School of Communication, Information, and Library Studies
Rutgers University
4 Huntington St.
New Brunswick, NJ 18901-1071
+1 732-932-7981
covi@scils.rutgers.edu

M. S. Krishnan
University of Michigan
Business School
710 Tappan St.
Ann Arbor, MI 48109-1234
+1-734-763-6749
fax: 734-936-6631
mskrish@umich.edu

Judith S. Olson
School of Information &
University of Michigan
Business School
701 Tappan St.
Ann Arbor, MI 48109-1234
+1-734-647-4606
fax: 734-764-2475
jsolson@umich.edu

ABSTRACT

Companies are experimenting with putting teams into warrooms, hoping for some productivity enhancement. We conducted a field study of six such teams, tracking their activity, attitudes, use of technology and productivity. Teams in these warrooms showed a doubling of productivity. Why? Among other things, teams had easy access to each other for both coordination of their work and for learning, and the work artifacts they posted on the walls remained visible to all. These results imply that if we are to truly support remote teams, we should provide constant awareness and easy transitions in and out of spontaneous meetings.

Keywords

Rapid software development, collocation, team rooms, warrooms, metrics, productivity.

INTRODUCTION

Intense teamwork is very difficult. The literature on software development makes this point very clear. About 2/3 of software projects substantially overrun their schedules [11, 17] and exceed their budgets by half [13]. A number of tools have been adopted as potential solutions to this difficulty, but the field is still plagued with poor schedule, cost and quality numbers. Some of the major difficulties recognized in intense teamwork have to do with communication delays and breakdowns [20]. To overcome some of these, companies are beginning to experiment with putting an entire project team in one room for the duration of the project, a strategy that we call *radical collocation.*

In this paper, we report on a field study in which we studied 6 teams that worked under this new strategy. By being collocated, communication is likely to be easier; the answers to many questions are at hand, producing better productivity and faster schedules. On the other hand, living together destroys privacy, quiet, and perhaps concentration at moments when the cognitive demands are highest. It is this tradeoff that we sought to examine.

In this field study, we compared the productivity of the teams working in radical collocation with metrics collected on teams at the same company doing software development in the traditional office arrangement. Since collocation can not work for any size group, this experiment was conducted under conditions in which the project was scoped to fit a team that could fit into a single room, a project that fit 6-8 people. In addition, the company adopted a "timeboxing" approach to scoping the project--the project was not only to be done by 6-8 people, but the scope was limited to what they could accomplish in about 4 months of work. Timeboxing has the purported advantage that during this time it is unlikely that the world would have changed enough to make the original feature set irrelevant, a problem that has plagued traditional software development for years.

In the strict sense, this is a confounded experiment: the teams were both collocated and the scope was "timeboxed." As will become evident, the productivity enhancements are astounding--and we know we cannot attribute this success to either the collocation or timeboxing individually. However, we have a rich data set and believe we can talk informatively about which aspects of the changes in work were caused by collocation itself.

Most of the focus in this investigation is on the communication patterns of the individuals on the team. Past studies have indicated that less than 30% of a programmer's time is spent on traditional programming tasks and less than 20% of the time is spent on coding [4]. The rest of the time is spent on meetings, problem

resolution with the team, resolving issues with customers, and product testing, etc. And, we know that the further away people are who have to communicate, the less they talk with each other. A distance of 30 meters is equivalent to being truly remote [2]. When collocated, team communication is a subtle dynamic. Team members coordinate their actions around various artifacts and arrangements of people in space [Hutchins, Suchman and Heath & Luff, in 9]. Hutchins calls this phenomenon *distributed cognition*, whereby teammates exploit features of the social and physical world as resources for accomplishing a task [15].

A number of other studies have suggested that collocation will help software development teams and increase productivity. Poltrock and Englebeck [15] described the advantages of physical collocation within teams via scheduled meetings and opportunistic interactions. Sawyer and his colleagues found that team rooms helped focus the activities of work groups and isolated them from interruptions [26]. There are additional studies since Allen [2] showing that distance between team members matters [19]. But none of these studies show the direct relationship between collocation and productivity, with rich data showing what the work interactions are like. In this paper we show such a relationship.

RESEARCH OVERVIEW

The research reported here is a combination of case study and empirical evaluation at a Fortune 50 automobile company. The company had already decided to innovate in software development methodology by using iterative development, putting the customer on the development team, and timeboxing. Having seen the success reported about the facilities in the Sun JavaFactory [27], the company adopted the idea of using warrooms as facilities to support good communication and collaboration among team members. For purposes of this paper, the facility is called the Rapid Software Development Center (RSDC). At the time of the study, the final facilities were in the process of being designed and built. At issue were the particulars about the size and physical arrangements within and nearby the rooms, and how these factors influenced team communication. To inform the decisions about details, the company conducted a trial with six pilot teams and asked us to join in the evaluation. The pilot teams occupied an existing older facility that was modified to house six team rooms, as well as nearby conference rooms and hotelling areas.

We collected a number of measures:

- Productivity indicators: standard measures including time to market and function points per staff month, described later.

- Questionnaires administered at the start and end of the project, asking team members to predict their reactions to various facilities and then assess their experience with them.

- Observations of two teams in depth, visiting them about 8-10 hours a week for the duration of the project.

- Interviews of the members of the two teams studied in depth, at project completion.

- Questionnaires administered to all six teams at project completion assessing the satisfaction of the team, customer, and sponsor.

The setting

The facilities at the RSDC included a dedicated team room for each software development team, conference rooms nearby, and various hotelling cubicles for more private work away from the team. Figure 1 shows the general layout of the rooms, and Figure 2 shows the interior of one of the rooms.
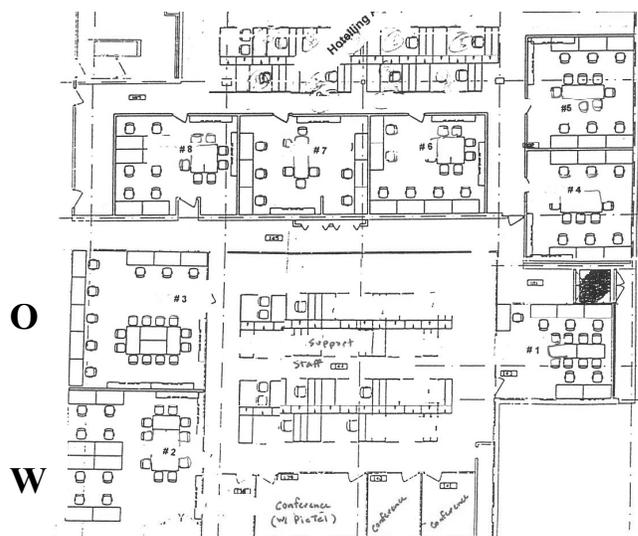

Figure 1. Floorplan of the RSDC with two major types of layouts on the left-hand side, the O and the W.

The teamrooms were outfitted with individual workstations for each team member. Workstations were in one of two arrangements: a W shape of two bays of wall-less cubicles, and an O shape, where PCs were arrayed along the outside walls of the room. In the center of the rooms were central worktables. Walls had whiteboards, and flipcharts were available as needed. Several rooms had printing whiteboards. Near the team rooms were several conference

Figure 2. One of the teamrooms, in an O configuration.

rooms, available on a first come first served basis. Other cubicles nearby were designated as hotelling space; no one owned them, but they could be used as needed for quiet, private time away from the team. Cubicles were outfitted with computers and telephones.

The teams

Six teams participated in this pilot. Each team consisted of 6-8 people: a manager, 4-5 contract programmers, and 1-2 customers (from within the company). The teams shared the services of a methodologist, technical architects, database experts, and testing specialists, called in when needed. For the duration of the RSDC projects, team members were not involved in any other project.

The RSDC projects

The projects in this initial trial ranged from web-based services, to client server applications, to mainframe systems. The projects came from a number of areas within the company, including manufacturing, finance, marketing and sales, and purchasing. Projects ranged in size from 326 to 880 function points (a standard measure of complexity described later).

The RSDC Method used

All teams used a development method based on Fusion. These six pilot teams used Fusion with the following "accelerators." First, all projects were within 600-1,000 function points and staffed to be done in 24 staff months (usually 6 people for 4 months). Second, projects used "timeboxing," which attempts to hold the time and staffing constant, and requires customers to determine which of a full set of desired features they most highly value. Finally, the team members were radically collocated.[1]

MEASURES

We looked at a number of measures, including productivity, satisfaction, attitudes, and use of time.

---

[1] Obviously, not all projects can fit within this bound. The hope is that with the large productivity increase noted here, that large projects can eventually be divided into smaller pieces, and artfully knitted back together in the end.

Productivity

There are a number of options to measure productivity: lines of code, function points, feature points, object points, etc. [5,1,3,18]. Since some of these are specific to a language and domain, we chose one that is relatively even-handed. Function points was the measure of choice [18,1]. Function points are the weighted sum of five factors: inputs, outputs, logical files, queries, and interfaces. We used the approach specified by the International Function Points User Group, version 4.0 to compute one of the measures of productivity in the study. We used the overall measure of [adjusted] function points per staff month.

A second measure of productivity was the cycle time, the number of months from the start of the project to the time when the project is completed. We normalize the cycle time for the size of the projects: the number of months per 1000 function points.

Satisfaction

Our team satisfaction measures included satisfaction with the organization of the team and with the roles of each individual member, with the warroom facility, and with facilities provided outside the warroom.

Our customer satisfaction measures include satisfaction the customers have of the system data, system performance, functionality, ease of use, system documentation, and training.

Our sponsor satisfaction measures include overall satisfaction with the project, cost, schedule, quality and periodic information updates about the project.

Team experience and preferences

We administered questionnaires to individual team members at the beginning of the project assessing prior experience with various facilities and technologies, using 5-point Likert scales. We asked for their predictions of how frequently they would use the new facilities and technologies, and how well they liked to work in various facilities and tools and in various styles. Nine items assessed people's preferences for working with various characteristics we thought the new facilities were going to have, such as being busy and collaborative.

An exit questionnaire contained many of the same items, but worded in terms of experience instead of expectations. In addition, team members in two teams were interviewed to better assess their experiences, attitudes, and activities in this environment.

Two of the teams were also observed about 8-10 hours per week over the 6 weeks of the projects. We sat in on meetings and conference calls, watched teams solve various kinds of problems, and photographed them using various artifacts and tools.

RESULTS

We review the productivity data first, because without finding some improvement, the other 'color commentary'

and data is less meaningful. The question is whether teams working in this environment were better.

Productivity

Teams in this warroom environment with the software development accelerators were much more productive than standard teams, both at this company and in the industry as a whole. Table 1 shows the function points per staff month and cycle time results for these teams, compared with both the company's standard scores and the industry as a whole.

Table 1. Comparative statistics on productivity measures.

	Pilot Teams	Company Baseline	Industry Standard
Function points per staff month (higher is better)	29.49	14.35	20.00
Cycle Time (lower is better)	7.64	19.47	24.00

In simple terms, the pilot teams produced double the number of function points per staff month from the previous baseline for the company. The time to market (cycle time) dropped to almost 1/3 as compared to the company baseline, and even lower than that produced by the industry as a whole. Both these results are statistically significant at $p < .001$ using z scores against the single baseline number.

Satisfaction

The team, sponsor, and end user satisfaction measures are shown in Table 2.

Table 2. Satisfaction measures for the pilot teams.

Team Satisfaction	4.15
Sponsor Satisfaction	4.56
End User Satisfaction	3.68

On a scale from 1-5 where 5 is "very satisfied", these are good marks. Although we do not have a baseline to compare these with, the overall satisfaction was high.

Is there a sample selection bias?

There is a worry that the teams selected for this pilot program were special in some way. To explore this question, we compared many of the same productivity and satisfaction measures for pilot teams against 11 teams using this facility following the pilot. (These were measures provided by the company that were not complete for all 11 subsequent teams, hence the *df*s reported in the analyses below vary somewhat.)

The productivity results are shown in Table 3. Follow-on teams were even more productive than the pilot teams. Function points per staff month doubled again, while cycle time stayed about the same. (* for *df* show where the analyses were adjusted for unshared variance.)

Table 3. Comparison of pilot with follow-on teams

Productivity Measures	Pilot Teams	Follow-on Teams	*Sig*	*df*
Function points per staff month	29.49	51.32	*p<.01*	15*
Cycle Time	7.64	6.58	*n.s.*	12*

The satisfaction of the subsequent teams remained consistently high, as shown in Table 4.

Table 4. Comparison of satisfaction of pilot teams and follow-on teams.

Satisfaction Measures	Pilot Teams	Follow-on Teams	*Sig*	*df*
Team	4.15	4.30	*n.s.*	13
Sponsor	4.56	4.29	*n.s.*	8
End User	3.68	3.97	*n.s.*	4

Behavioral results

We report on the data from the questionnaires, interviews and observations, organized thematically.

Team members liked the warrooms

Team members reported that they had very little *experience* with warrooms prior to being assigned to work at the RSDC, rating a 2.17 on a 5 point scale, where 5 = very frequent experience. Working at the RSDC increased their *preferences* for warrooms and decreased their preferences for working the old way (in cubicles).

Table 5. Comparison of entry vs. exit questionnaire data.

Preference Measures	Entry	Exit	*Sig*	*df*
for warrooms	3.53	4.0	*p<.01*	5
for cubicles	3.86	3.42	*p<.04*	5

How did the teams work in the warrooms?

Warrooms were found to be supportive of interactive, continuous communication between team members. The proximity of the spatial arrangement allowed team members to overhear each other. If someone was having difficulty with some aspect of the coding or design, others walking by, seeing the activity over their shoulders, stopped to help an individual. Figure 3 shows the movement of one team from two separate discussions to one joint meeting, all done on the fly. Also, when one team member was explaining things to the other team members, others could overhear and spontaneously interject commentary, clarifications, and corrections.
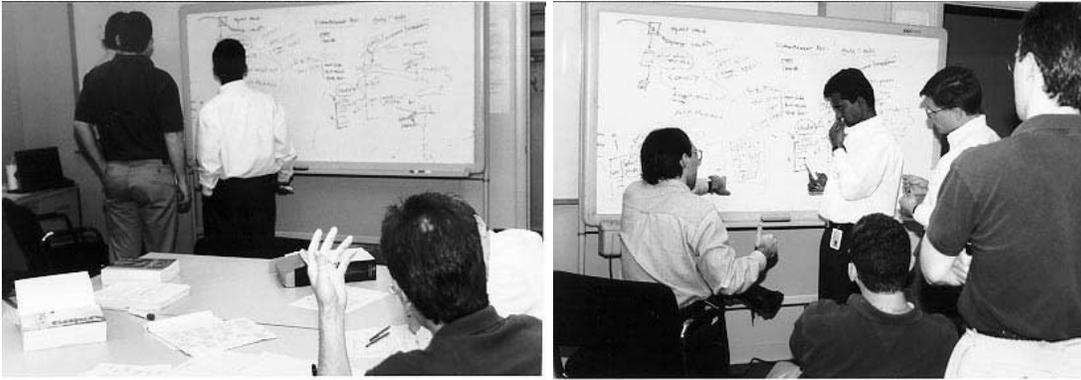
Figure 3. Two phases of interaction in the warroom: The team moving from two separate meetings (left panel) to one central meeting (right panel), simply from overhearing what the other group was talking about.

Our observations showed that the teams chose to work in different spaces available to them in different major subtasks. We identified nine different kinds of work, listed in Table 7.

Table 7. Nine kinds of work.

1. Discussion to acquire customer input
2. Discussion of a political issue
3. Problem solving at the whiteboard
4. Status meeting using the to-do list (usually on a flip chart or the whiteboard).
5. Team building discussion (social)
6. Training
7. Simultaneous problem solving meetings (subsets of team members)
8. Working solo (typically coding)
9. Private conversations with outsiders.

Being in a warroom supported most of the above kinds of work because the space was large enough to monitor each other, but not to have the noise of the other conversation totally distract one. On some occasions, subsets of team members went to nearby conference rooms. This happened less in the larger rooms, where distance reduced the overall noise level.

The downside of continuous, interactive communication was the distraction. Overhearing distracts those that are doing work that requires concentration. Programmers described their desire to work in what they called a "flow state." Sometimes they retreated to the hotelling area, but found that they did not have all the material they needed there to get their work done.   Some programmers reported that they occasionally came in after hours (both before and after) in order to work in the room with some peace and quiet.

How the facilities influenced collaboration and communication

The entry and exit questionnaires told a story about the reactions to the room over time. Table 6 shows that people in these facilities were less distracted working with each other continuously than they had thought they would be.

Table 6. Changes in reported attitudes about activity in the warrooms.

	Entry	Exit	*sig*	*df*
Susceptibility to distraction	3.37	2.68	*p<.01*	5
Concern about individual recognition	2.87	3.29	*p<.04*	*5*

Team members reported that they learned to work in the collocated environment, tuning in and out of the simultaneous activities. They also reported that they increased familiarity with their teammates so that the lack of privacy and increased visibility became less of a problem.

Interestingly, however, the team members reported increased concern about whether their managers would recognize their individual contribution to the project. Clearly, the incentive/performance metrics need to be aligned with the new way of working.

Privacy was a second concern. Private conversations with outsiders and conversations about personnel or sensitive issues typically moved outside the warroom. In fact, the.

Figure 4. Three views of the new RSDC rooms with walls of whiteboards, projection equipment and large distances between collocated work areas.

most frequent use of the hotelling space was to make private phone calls (e.g. to a banker about a loan, or to a physician about a medical issue).

What artifacts did the team use to support their collaboration?
The whiteboards and flip charts located in the warrooms helped the communication by providing large public workspaces that served as a visible permanent record of group activity and decisions. Teams used the whiteboards to keep current the status of the project plan visible to everyone. "To-do" lists were posted on whiteboards or flipcharts to note when a team member was responsible for an item. The fact that it was visible made it accessible to everyone at a glance.

FOLLOWUP
The success of these rooms has encouraged the company to invest heavily in propagating warrooms. The company has built a new facility with 112 such rooms in the US, and has plans for other sites in Europe, all to reap the measured benefits of both adopting accelerators for the development method and collocating the team. The significant investment in the new physical environment is an indication of the company's belief that the software development accelerators alone can not produce the benefits found here, nor are they limited to only specialized kinds of development teams.

In the new sites the warrooms are clustered into neighborhoods in which 8 rooms reside on a hall, with a central area housing support services, (e.g. the database expert, the methodologist, etc.). The rooms are both large and configured in the W style that was used by several of the pilot teams. The entire end wall is a whiteboard (indeed they asked for stepladders so they can use the entire surface). In informal training or status meetings, teams project a laptop using a portable projector on a cart, projecting the image on the whiteboard, which has a semi-gloss surface suitable for viewing. Figure 4 shows three views of one of these new rooms.

WHAT HAVE WE LEARNED ABOUT COMPUTER SUPPORTED COOPERATIVE WORK?
There are two classes of lessons to be taken from this experience: lessons for collocated work and lessons for remote work.

Implications for collocated work
The teams had desktop PCs on which they ran programming environments to support their design and coding. Team members used much of the whiteboard space to support their problem solving meetings, yet when at closure on a particular solution, they had to transfer the material to the desktop environment. Some things on flipcharts had to be typed in. People talked about wanting other things that were generated in the desktop environment (e.g. an object hierarchy or a diagram of the architecture) to be shown to the group in more permanent display. One could imagine printing out a desktop image on a flipchart size paper and posting it. There was a call for more electronic surfaces to support teamwork. The new rooms have projected laptops, but even in these rooms, the participants draw on the whiteboard onto which the projection is made, annotating the drawing. This, of course, is not recorded in the original. What is needed is a smoother transition between the shared, sometimes informal, often heavily edited displays that groups use and the centrally stored linked objects used in both individual coding and in the final product.

As shown in Figure 5, we can imagine this transition being supported today by combinations of

- electronic whiteboards like the SmartBoard™ ,

- desktop or tablet personal workstations,

- markerboard capture devices like Mimio™ or Ebeam™ (which detect where specially housed whiteboard or flipchart markers are in 2-D space, capturing the drawing as it emerges and storing it in a connected PC),

- large scale printers to print out flip-chart size paper of viewing stored objects that do not change often (like to-do lists, contact phone numbers, etc.), and

- scanners to take input of drawings, etc. that come in paper form.

Work along this line has begun already with Knight, a system that interprets electronic whiteboard sketches and drawings and appropriately puts them into a software development environment [8].
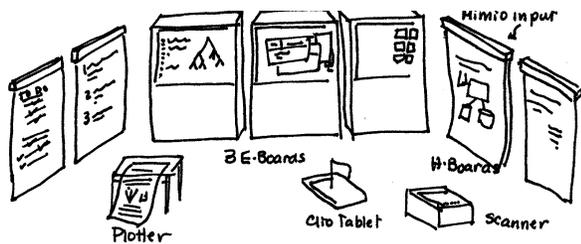

Figure 5. A sketch of a technology supported warroom with electronic whiteboards, pen capture devices, tablets, scanners and plotters, supporting a mixture of editable and more permanent work surfaces.

To date we know of no technology to aid with reducing distractions, aside from distance itself and using headsets with music. When someone wishes to speak privately, the only solution is departure to the hotelling area. No one yet has actualized Get Smart's cone of silence.

Implications for remote work

What does this say about trying to support remote team members in a way that produces the same kinds of productivity enhancements? This will be difficult. One of the main drivers of success was the fact that the team members were *at hand,* ready to have a spontaneous meeting, advise on a problem, teach/learn something new, etc. We know from earlier work [2] that the gains from being *at hand* drops off significantly when people are first out of sight, and then most severely when they are more than 30 meters apart. What, therefore, can we do, either with design of the process or with technology, to attempt to bring people close enough to be *at hand?*

First, remote teams must overlap in time as much as possible. They cannot be *at hand* if they are not working at the same time. People do work asynchronously, even successfully, but likely not at the kinds of productivity levels witnessed here.

Second, other members must be visible and audible at the apparent distance of 12-15 feet, with enough resolution in both vision and sound to mimic reality. This calls for an open video and stereo audio system, much like the open video connection in the Portland [23] and the video wall at Bellcore [10].

Third, the team members need to overhear each other, and, when they overhear something of importance, to meet in a more directed interaction. This requires easy use of collaborative-shared objects near the video/audio connection (e.g. remotely connected SmartBoards and Pictel units,

available today). Again, subtleties of gestures and eye contact might be important, as seen in the ClearBoard [16]. We are not arguing here that these rich gestures and sounds are critical, but anything that makes the interaction seamless and natural will at least have the possibility of reproducing the easy communication seen in radical collocation. The popular press is calling electronic forms of awareness and object sharing as the next killer app [12].

SUMMARY

Our study of six teams that experienced radical collocation showed that in this setting they produced remarkable productivity improvements. Although the teammates were not looking forward to working in close quarters, over time they realized the benefits of having people at hand, both for coordination, problem solving and learning. They adapted to the distractions of radical collocation, both by removing themselves to nearby hotelling areas when they needed privacy, and by zoning out, made possible because of the distance between people in the larger rooms. Of the nine kinds of activities the team engaged in, only two were best done individually and separate from the rest of the team.

While achieving great productivity in this environment, the work was not trouble free. Collocated groups could benefit from electronic walls that have easy transition to individual workstations. Today's common use of flipcharts, whiteboards and workstations is far from what's possible even today with Electronic Whiteboards and pen-capture technologies. Blending these informal sketch-based tools with more formal PC based tools is emerging as well.

This study also tells us what might be needed for virtual collocation to be as good as radical collocation. The teams primarily benefited with everyone being at hand. This means that remote teammates, to enjoy these benefits, should not be so far as to be in different time zones, or they should at least have significant overlap during the workday. And, it likely requires video walls with stereo sound, as well as group-based electronic whiteboards, like the ones available today. Although, with remote groups the issues of common ground and culture will pervade, we can likely go a far ways down the road to radical collocation with the proper awareness tools [22].

ACKNOWLEDGMENTS

REFERENCES.

1. Albrecht, A., and Gaffney, J. (1983) "Software Function, Source Lines of code, and Development Effort Prediction: A Software Science Validation," *IEEE Transactions on Software Engineering*, (SE-9:6), November, pp. 639-647.

2. Allen, T. J. (1977) *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*. : MIT Press, Cambridge, MA.

3. Banker, R. D., Datar, S., Kemerer, C. F., and Zweig, D. (1993) "Software complexity and software maintenance costs," *Communication of the ACM*, (36), November, pp. 81-93.

4. Barstow, D., (1987) "Artificial Intelligence and Software Engineering," *Proceedings of the 9th International Conference on Software Engineering*, (9:5), pp. 541-561.

5. Boehm, B. W. (1981) *Software Engineering Economics*, Prentice-Hall, Inc., NJ.

6. Brinck, T. and Gomez, L (1992) A collaborative medium for the support of conversational props. *CSCW'92.* Pp.

7. Covi, L. M., Olson, J. S., and Rocco, E. (1998) A room of your own: What do we learn about support of teamwork from assessing teams in dedicated project rooms? In N Streitz, S. Konomi, and H.J. Burkhardt (Eds.) *Cooperative Buildings.* Amsterdam: Springer-Verlag. Pp. 53-65.

8. Damm, C. H., Hansen, K. M., Thomsen, M. (2000) Tool support for cooperative object-oriented design: Gesture based modeling on an electronic whiteboard. *Proceedings of CHI'2000,* 518-525.

9. Engestrom, Y., and Middleton, D. (eds.).(1996) *Cognition and Communication at Work*. Cambridge University Press, Cambridge.

10. Fish, R. S., Kraut, R. E., Root, R., & Rice, R. E. (1993) Video as a technology for informal communication. *Communications of the ACM, 36.* 8-61.

11. Gibbs, W. W. (1994) "Software's chronic crisis," *Scientific American*, September, pp. 86-95.

12. Giglio, C. M. (1999) Business is war, but you can fight back with a killer app--a virtual war room. *Infoworld,* (21:18), p. 72.

13. Hoch, D. J., Roeding, C. R., Purket, G. and Lindner, S. K., (2000) Secrets of Software Success, *Harvard Business School Press*, Boston, MA.

14. Hutchins, E. (1995) *Cognition in the wild.* Cambridge, MA: MIT Press

15. Hutchins, E. (1997) "Constructing meaning from space gesture, and speech," in *Discourse, Tools, and Reasoning: Essays on Situated Cognition*, L. B. Resnick, R. Saljo, C. Pontecorvo, and B. Burge (eds.), Springer, Berlin, pp. 23-40

16. Ishii, H., Kobayashi, M., Arita, K., and Yagi, T (1997) Iterative design of seamless collaboration media. In K. Finn, A. J. Sellen, and S. B. Wilbur (Eds) *Video-mediated Communication.* Mahwah, NJ: Lawrence Erlbaum Associates. Pp.435-456.

17. Johnson, J., "CHAOS: The dollar drain of IT project failures," *Application Development Trends,* (20: 1), 1995, pp. 41-44.

18. Jones, C., (1996) *Applied Software Measurement: Assuring Productivity and Quality*, McGraw-Hill Book Co., New York.

19. Kraut, R.E., Egido, C., & Galegher, J. (1990) Patterns of contact and communication in scientific research collaborations (Pp. 149-171). In J. Galegher, R.E. Kraut & C. Egido (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work*. Hillsdale, NJ: Lawrence Erlbaum Associates.

20. Kraut, R.E. and Streeter, L.A. (1995) "Coordination in Large Scale Software Development," *Communications of the ACM*, (38:7), pp.69-81..

21. Mark, G., Grudin, J., and Poltrock, S.E. (1999) Meeting at the Desktop: An Empirical Study of Virtually Collocated Teams *Proceedings of ECSCW'99*, The 6th European Conference on Computer Supported Cooperative Work, 12-16 September 1999, Copenhagen, Denmark.

22. Olson, G. M. & Olson, J. S. (in press) Distance matters. *Human Computer Interaction.*

23. Olson, M.H., & Bly, S.A. (1991) The Portland experience: A report on a distributed research group. *International Journal of Man-Machine Studies*, 34, 211-228.

24. Poltrock S. E., and Engelbeck, G. (1999) "Requirements for a virtual collocation environment," *Information & Software Technology*, (41:6), pp. 331-339.

25. Saund, E. (1999). Bringing the marks on a whiteboard to electronic life. In N. Streitz, J. Seigel, V. Hartkopf, & S. Konomi (Eds.), *Cooperative Buildings: Integrating Information, Organizations, and Architecture*; Proceedings of the Second International Workshop (CoBuild'99). Heidelberg: Springer.

26. Sawyer, S., Farber, J. and Spillers, R. (1997) "Supporting the social processes of software development teams," *Information Technology and People*, (10:7), pp. 46-62.

27. *Sun Journal* (1999) How to jump-start Java computing initiatives. November 5, 1999.
http://www.sun.com/SunJournal/v1n4/global2.html

28. Whittaker, S. & Schwarz, H. (1995) Back to the future: Pen and paper technology supports complex group coordination. *Proceedings of the Conference on Human Factors in Computing Systems, CHI'95.* NY: ACM Press. pp. 495-502.