Computing Environments for Flexible Teams

Danielle Fafchamps, Pankaj Garg
Software Technology Laboratory
HPL-94-105
November, 1994

teamwork, CSCW, process re-engineering, concurrent engineering, programming environments

Flexible teams are a new type of organizational entity that will become even more prevalent in the future. We define the concept of a flexible team, present selected attributes of such teams (composite membership and roles, diverse disciplines and skills, rapid communication alignment, and rapid process alignment), address the impact of these attributes on different categories of tools, and discuss implications for the design of computing environments to support flexible teams.

Computing Environments for Flexible Teams

Danielle Fafchamps and Pankaj Garg

Hewlett-Packard Laboratories
P.O. Box 10490
Palo Alto, CA 94303-0969
garg@hpl.hp.com
(415) 857-4709

Abstract. Flexible teams are a new type of organizational entity that will become even more prevalent in the future. We define the concept of a flexible team, present selected attributes of such teams (composite membership and roles, diverse disciplines and skills, rapid communication alignment, and rapid process alignment), address the impact of these attributes on different categories of tools, and discuss implications for the design of computing environments to support flexible teams.

1 Introduction

Flexible teams are emerging as a new type of entity within the electronics industry. Such teams are less permanent than traditional product teams but less temporary than task forces or ad hoc groups. In this paper we provide some answers to the following questions: What are flexible teams? How do their collaborative practices differ from those of other types of teams? What computing environments do they need?

Our approach is problem driven and grounded in observations of teamwork within Hewlett-Packard Company (HP). We draw on Computer-Supported Cooperative Work (CSCW) concepts and approaches to examine the type of computing environments required to support the collaborative needs of flexible teams.

Software engineers and CSCW researchers share a common agenda: To design systems that facilitate and potentiate human activities. CSCW researchers, however, focus on activities involving some form of teamwork. They integrate work from the social sciences, organizational development, and computer science to understand how professionals work together, and to articulate technical solutions that are socially and organizationally desirable. To quote from Grudin's article[Gru90]:

> "The term *user interface* was not needed in the beginning, when most users were engineers and programmers; it may again become inappropriate when more applications are written for groups than for individuals. But there is a continuity from the outward movement of the computer interface to its external environment, from hardware to software to increasingly higher-level cognitive capabilities and finally to social processes. As the focus shifts, the approaches to design and the skills required of practitioners change."

The CSCW community has researched the particular collaborative needs of many professional domains (e.g., consulting [Orl92], research [KEG90], medical [Cic90, FYT91, EW92], navigation [Hut90], air-traffic control [BHR+92], education [SHLRA92], flight crews [BCP90]), and the needs of different types of teams (e.g., product development teams [AC90] and ad hoc teams [FSK90, EB88]), but we did not find published work about the type of teams we call flexible. This is not surprising.

In many companies, product teams constitute the basic unit of the hierarchical structure. Entities that do not fit neatly within that structure stick out as unmanageable oddities. On the other hand, flexible teams are not very amenable to experimental research studies that inevitably consume valuable project time. Many factors justify their shyness: They already have high organizational visibility, they work under very tight schedules, they have to show results in the early stage of their association, and they have to quickly adjust to the working styles of individual members without the benefit of a getting-acquainted grace period.

No published work exists on flexible teams, but the rich CSCW literature is useful to begin our investigation. This initial investigation is presented as follows: (1) an overview of the technological, business, and social changes that motivate the emergence of a new team structure; (2) an illustration of why current computing environments are unsuitable for this new team structure; (3) a description of selected attributes of flexible teams; (4) the technological implications of these attributes; and (5) directions for future research.

2 Understanding Flexible Teams

In the divisional structure common to many electronics companies, the product team forms the basic organizational unit and includes a manager and specialists dedicated to the delivery of a product. Product teams are grouped into product lines, which are often geographically dispersed.

Members of a product team come together to design and develop a product, and they remain together for the duration of the products life. Their computing needs are well defined and relatively stable. If the computing environment evolves during the lifetime of the product, all members are affected in the same manner at the same time. However, recent changes on the technological, business, and social scenes are altering the nature of work and collaboration.

2.1 Technological scene

Fast-paced technological innovations challenge the electronics industries' ability to adapt quickly to new requirements. For example, an increasing proportion of HP's revenues is from recent products. Flexible teams are one means to increase adaptiveness. In a flexible mode, individuals with desirable expertise leave their home groups to become full- or part-time members of a team for the duration of an assignment that is shorter than the traditional life span of a product team but longer than typical task force or ad hoc teams.

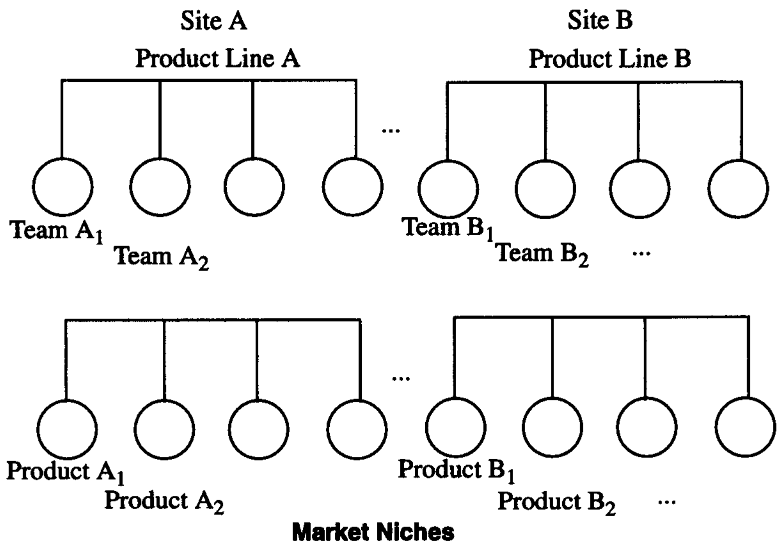

Fig. 1. A Typical Divisional Structure

2.2 Business scene

Customer needs have changed drastically over the last five years. The demand for integrated products is forcing electronics industries to reconsider established market niches, and hence the charters of product-focused teams. The products in a suite must "talk" to each other and must exhibit the same "look and feel". Successful integration requires purposeful interdependencies among product teams, but interdependencies are difficult to achieve: Social tensions arise between teams who formerly worked autonomously; technological obstacles delay time to market because different product teams often work in different, sometimes incompatible, computing environments; and alignment of products is sporadic when each team claims that its own brand of core components should set the standards for all [Faf94].

Flexible teams are one means to ease the development of integrated solutions. For example, a multinational HP division created three cross-site flexible teams: One team handles the technical aspects of the product suite; another handles interface standards; and a third team designs components shared by all products in the suite. The structure of the other product teams is unchanged.

2.3 Social scene

Industrial forecasters predict that contractors will constitute about 70% of the work force by the year 2000. These contractors will share membership in flexible teams with other contractors and with employees.

Flexible teams are likely to become familiar elements in the organizational landscape, but computing legacies and preferences will be a major obstacle to the timely collaboration of experts from different venues. One solution is to put the burden on individuals and to force them to frequently adapt to new computing environments. In this scenario, productivity is likely to be poor for at least two reasons: (1) steep learning curves will delay time to market, and (2) quality is likely to suffer because different professionals perform better in some environments than in others. Instead, the solution we advocate is to design computing environments that enable individual choice.

3 A Simple Motivating Example

Consider the example of two people, A and B, working on a new common document, say D. A standard problem is that some form of concurrency control must be exercised on D. Traditionally, software engineering version control systems, such as RCS [Tic82], SCCS [Roc75], DSEE [LR84], or ClearCase, have dealt with this problem for one situation: A and B access a shared repository where versions of D are stored. If A wants to make modifications to D, she obtains a lock on D and starts making modifications. While A has a lock on D, B cannot obtain a lock on D and, therefore, cannot enter his modifications. There are minor variations to this basic solution, such as optimistic concurrency control, but major aspects of the situation remain implicitly assumed.

This traditional approach works well with stable teams, but it starts to break down when flexible teams are formed. For purpose of illustration, let's assume that A and B are now part of a flexible team and that they are accustomed to different tools. If D is a text document, then A might want to use Microsoft Word while B prefers to use FrameMaker. In this case, the assumed model of the situation (i.e., one copy of D can be used for concurrency control) breaks down and a variety of scenarios are possible.

Scenario #1 A and B elect to use one data format and they learn the appropriate tools reflecting that choice. In this case, we go back to the traditional solution, and previous version control systems are still applicable.

Scenario #2 In the more realistic scenario, A and B cling to their preferred tools. There is no longer a unique copy or representation of D; a version of D exists for the tools of both A and B. Translators will be required to translate from one representation to another, and concurrency control will still have to be somehow enforced. The possible solutions are:

Solution 1 A and B negotiate the timing and manner of their modifications. This form of agreement is not enforceable and, therefore, not of interest to us from a tools support viewpoint.

Solution 2 A mutex object (file) can be set up that guards modifications to any one of the various representations of D. Before making modifications, A or B must acquire the mutex object, and then release it after modifications. Only one person is allowed to acquire a mutex at any one time, therefore ensuring mutual exclusion. In addition, once a file has been modified, all the different representations need to be updated.

Note that solution 2 applies only to cases where translators from the different representations are available. But users should not be responsible for locating the translators and for checking their availability. This flexible team should be able to specify their new situation, in a declarative manner, and the environment should be able to determine the needs of the situation and the availability of the appropriate tools. In order to identify and characterize such situation variables, we need to better understand the variability in the computing environments that flexible teams will need to confront. The next section provides a framework to understand such variability.

4 Characteristics of Flexible Teams

Currently, our data on flexible teams include project retrospectives available as internal documentation, data repositories of various observational studies, and opportunistic discussions with engineers and managers. From these data we abstracted four attributes of flexible teams: composite memberships and roles, diverse disciplines and skills, rapid communication alignment, and rapid process alignment.

4.1 Composite memberships and roles

A flexible team may include on-site, full-time experts. For example, one HP division determined that the expected pay off from the work of a flexible team was worth the costs of a two-year relocation for out-of-state team members. However, not all flexible teams included full-time membership. Figure 2 illustrates the four recurrent types of membership in the data. The mix of membership types varied across teams depending on the problem they were to solve, the availability of desirable experts, the expected length of the assignment, and the contribution of a particular expertise.

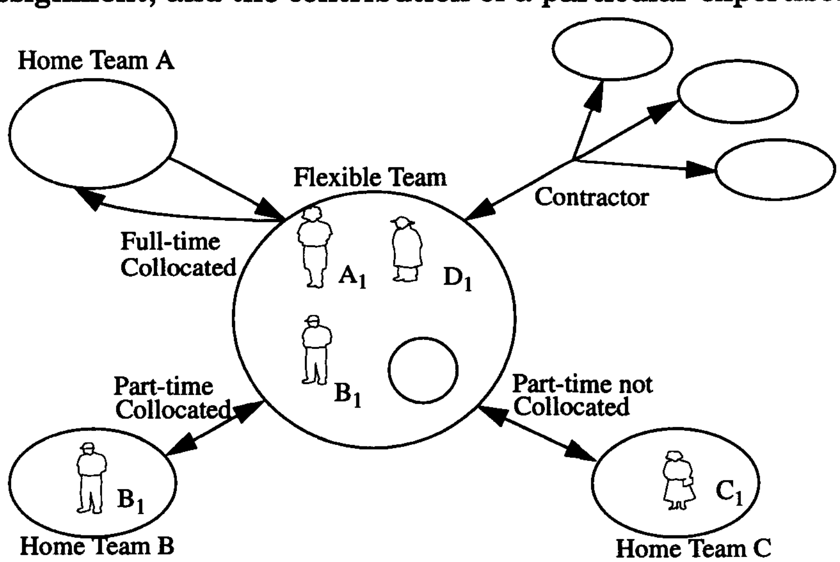

Fig. 2. Types of membership in a flexible team

Full-time collocated. A_1 is the sole full-time member of the flexible team, and she assumes the role of project manager. The computing environment available for the project may be similar to or different from A_1's preferred environment.

Part-time collocated. B_1 shares his time between his home team and the flexible team. At the end of the project, he will resume full-time work with his home team. In the worst-case scenario, B_1's work may require frequent switching between two radically different computing environments.

Part-time not collocated. C_1 works part-time and is geographically remote from the flexible team. Her expertise is required only for fixed durations, such as the early stage of the project and the last stage of the project. Between engagements with the flexible team, C_1 will work full-time with her home product team.

Contractor. D_1 is a contractor who divides her time between the flexible team and other teams in different locations.

In the worst-case scenario, a new team could face a spectrum of possible computing environments the size of which increases exponentially with the size of the team and the number of tools.

4.2 Diverse disciplines and skills

Flexible teams may include members who belong to the same professional practice, or they may be cross-disciplinary. Different specialists have different world views, different work styles, different representations, and different preferences for the tools they find appropriate to the enactment of their respective expertise [Kim90, VAL90]. Programmers have clear preferences for a variety of tools such as a particular e-mail application, editor, program shell, or configuration. Similarly, professionals from different disciplines have clear preferences for the tools that are best suited to their trade. For example, although the favorite editor of many programmers, "vi" is not adequate to the work of a marketing person, a physician, an interface designer, or an anthropologist.

4.3 Rapid communication alignment

The stability of product teams enables them to refine over time the modes of communication with which members are most comfortable. The selection of modes of interaction depends on the channels available, as well as on the personality, interactive styles, and preferences of team members. Flexible teams have little time to experiment with modes of interaction. Our data show that difficulties encountered in figuring out the best way to communicate, especially when members were geographically dispersed, increased time to market and involved much effort, frustration, and potential for conflict.

Research has identified two main models of interaction for within-team communication [FSK90]. In the "wheel" model, the manager assumes a central position in the interaction of the team members, while in the "circle" model, members communicate with each other in a decentralized fashion. The studies of Finholt and colleagues [FSK90] suggest that the circle model works better when competence is evenly distributed through the group. Flexible team members are typically recognized experts, and our data so far indicate that flexible teams indeed prefer a decentralized communication structure.

4.4 Rapid process alignment

Even when particular work processes are enforced within an organization, the observable processes of stable teams will deviate from the norms. Deviations are often implicitly shared as the culture of the team and are not explicitly articulated or managed [GMP+94, HC93].

Flexible teams, on the other hand, need to articulate their processes at the outset and to quickly reach a consensus on how they will work together in a manner satisfying to all. Individuals who belong to multiple teams are likely to need to adapt to different processes, and sometimes a teams processes will conflict with the processes of another team. For example, during design reviews, one design team might want to use presentations with little or no formal documentation, while another team might prefer to have well-documented designs before conducting design inspections or reviews. Furthermore, the processes of stable teams may or may not be applicable to flexible teams. For example, stable teams may spend more time in the planning stage when they define the problem and engage in training activities, but the recognized experts of a flexible team are expected to produce results faster.

5 Impact on Computing Environments

Variations in the attributes of flexible teams (composite membership and roles, multiple disciplines and skills, rapid communication alignment, and rapid process alignment) will affect computing requirements. To understand the impact, we have identified four categories of tool — communication management, professional work management, time management, and artifact management — that address the basic needs for work management within a team.

Communication management tools are used to exchange information within the team and with others. Such tools include electronic mail, phones, faxes, and printers. We include printers, faxes, and phones in this category because software interfaces to these systems are increasingly situation dependent. For example, an individual may prefer to send faxes by first creating a simple text document and electronically mailing it to a fax server. A new site may not provide such services.

Professional work management tools support particular professional expertise. Examples in this category are tools such as slide-making tools, databases, multimedia editing tools, CAD systems, and word processors. Members in a stable product team typically belong to the same professional group (e.g., computer science), and they allocate time to examine the selection of work tools at the setup of the team. Flexible teams are likely to be multidisciplinary, and they need to compensate for the heterogeneity in their tool set.

Time management tools are used to organize the time of individuals and to coordinate the team activities. Current tools are not built for flexible teams — for example, the Synchronize [Cro92] tool assumes a stable team model.

Artifact management tools include users' file systems and any configuration management tools that are used over the file system (e.g., ClearCase). Current artifact management tools also assume a stable team structure. For instance, OSF's Distributed Computing Environment (DCE) assumes a hierarchical team structure of cells within cells, which is very tedious to modify after it has been defined.

Flexible team members will use combinations of these four categories of tools to accomplish their tasks. A member's computing environment is defined as a particular collection of tools. The enterprise that wants to support both stable and flexible teams must have a versatile, protean computing environment where team members are able to: (1) add, modify, or delete tool configurations, (2) maintain parallel tool configurations, and (3) rapidly switch across tool configurations. Table 1 illustrates the broad-scoped impact that flexible teams will have on requirements for tool configurations.

Table 1. Some issues about flexible computing environments

Tools vs. Attributes	Membership/ Roles	Disciplines/ Skills	Communication	Processes
Communication Management			Rapid tool reconfigurations	
Professional Work Management		Heterogeneous tool integration		
Time Management				Integrated cross-team time management
Artifact Management	Information sharing across boundaries			

Even with a high-level analysis, it is clear that the attributes of a flexible team will affect all categories of tools and their configurations: geographically distributed team members will need flexible, shared artifact management; multidisciplinary teams will require data sharing across different applications; the rapidly changing communication needs of flexible teams will require communication tools to be easily reconfigured; and process alignment will require time management tools to operate with multiple changing processes. Although not definitive, this table gives an idea of the complexity of support features needed for the computing environments that will support flexible teams.

6 Future Research

From a CSCW perspective, we need answers to questions such as:

- What kinds of roles are performed in flexible teams? What types of information access, retrieval, manipulation, and representation are needed for these roles? What kinds of tools are preferred to perform these roles?
- What types of information access, retrieval, manipulation, and representation do individuals from different disciplines prefer? What tools do they select?
- What types of information access, retrieval, manipulation and representation are needed for the internal and external interactions of flexible teams? What tools are available and how are they used?
- What are the process needs of flexible teams and what types of process support tools are available?

From a software engineering perspective, one of the main challenges is to define a configuration language that will allow flexible team members to define new situations simply and easily. This language will allow team members to specify the attributes of their flexible teams, that is, their work style, tool, communication, and process preferences. For a set of specifications, a computing environment should be able to generate an environment that is suited to the particular attributes of a flexible team. The work of Karrer and Scacchi on the meta-environments concept [KS93] should provide useful guidance on how to design such computing environments.

References

[AC90] D. Ancona and D. Cadwell. Information Technology and Workgroups: The case for new product teams. In J. Galegher, R. E. Kraut, and C. Egido, editors, *Intellectual Teamwork: Social Foundations of Cooperative Work*, pages 173–190. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[BCP90] I. Benson, C. Ciborra, and S. Proffitt. Some Social and Economic Consequences of Groupware for Flight Crews. In *Proceedings of CSCW'90, Conference on Computer Supported Cooperative Work*, pages 119–129, Los Angeles, California, October 1990. ACM Press.

[BHR+92] R. Bentley, J. A. Hughes, D. Randall, T. Rodden, P. Sawyer, D. Shapiro, and I. Sommerville. Ethnographically Informed System Designs for Air Traffic Controllers. In J. Turner and R. E. Kraut, editors, *Proceedings of CSCW'92, Conference on Computer Supported Cooperative Work*, pages 131–129, Toronto, Canada, October-November 1992. ACM Press.

[Cic90] A. Cicourel. The Integration of Distributed Knowledge in Collaborative Medical Diagnosis. In J. Galegher, R. E. Kraut, and C. Egido, editors, *Intellectual*

Teamwork: Social Foundations of Cooperative Work, pages 1–20. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[Cro92] CrossWind Technologies, Inc., 6630 Highway 9, Suite 201, Felton, CA 95018. *Synchronize: User's Guide*, 1992.

[EB88] J. Eveland and T. Bikson. Work Group Structures and Computer Support: A Field Experiment. In *Proceedings of CSCW'88, Conference on Computer Supported Cooperative Work*, pages 324–343, Portland, Oregon, September 1988. ACM Press.

[EW92] E. Egger and I. Wagner. Time-Management: A Case for CSCW. In J. Turner and R. E. Kraut, editors, *Proceedings of CSCW'92, Conference on Computer Supported Cooperative Work*, pages 249–256, Toronto, Canada, October-November. 1992. ACM Press.

[Faf94] D. Fafchamps. Organizational Factors and Reuse. To Appear in it IEEE Software, September 1994.

[FSK90] T. Finholt, L. Sproull, and S. Kiesler. Communication and Performance in ad hoc Task Groups. In J. Galegher, R. E. Kraut, and C. Egido, editors, *Information Technology and Workgroups: The case for new product teams*, pages 291–325. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[FYT91] D. Fafchamps, C. Y. Young, and P. C. Tang. Modeling Work Practices: Input to the Design of a Physician's workstation. In P. Clayton, editor, *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care*, pages 788–792, Washington, D.C., November 1991. McGraw-Hill.

[GMP$^+$94] P. K. Garg, P. Mi, T. Pham, W. Scacchi, and G. Thunquest. The SMART Approach for Software Process Engineering. In *International Conference on Software Engineering*, pages 341–350, Sorrento, Italy, May 1994. IEEE Computer Society Press.

[Gru90] J. Grudin. The Computer Reaches Out: The Historical Continuity of Interface Design. In *Proceedings of CHI'90, Conference on Human Factors in Computing Systems*, pages 261–268, Seattle, Washington, April 1990. ACM Press.

[HC93] M. Hammer and J. Champy. *Re-engineering the Corporation: A manifesto for business revolution*. Harper Collins, New York, 1993.

[Hut90] E. Hutchins. The Technology of Team Navigation. In J. Galegher, R. E. Kraut, and C. Egido, editors, *Intellectual Teamwork: Social Foundations of Cooperative Work*, pages 191–220. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[KEG90] R. E. Kraut, C. Egido, and J. Galegher. Patterns of Contact and Communication in Scientific Research Collaborations. In J. Galegher, R. E. Kraut, and C. Egido, editors, *Intellectual Teamwork: Social Foundations of Cooperative Work*, pages 1–20. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[Kim90] S. Kim. Interdisciplinary Cooperation. In B. Laurel, editor, *The Art of Human-Computer Interface Design*, pages 31–44. Addison-Wesley, Menlo Park, California, 1990.

[KS93] A. S. Karrer and W. Scacchi. Meta-Environments for Software Production. *International Journal on Software Engineering and Knowledge Engineering*, 3(1):139–162, 1993.

[LR84] David B. Leblang and Robert P. Chase, Jr. Computer-Aided Software Engineering in a Distributed Workstation Environment. In *Proceedings of the ACM SIGSOFT / SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, pages 104–112, Pittsburgh, Pennsylvania, April 1984.

[Orl92] W. Orlikowski. Learning from Notes: Organizational Issues in Groupware Implementation. In J. Turner and R. E. Kraut, editors, *Proceedings of CSCW'92, Conference on Computer Supported Cooperative Work*, pages 362–369, Toronto, Canada, October-November 1992. ACM Press.

[Roc75] Marc J. Rochkind. The Source Code Control System. *IEEE Transactions on Software Engineering*, SE-1(4):364–370, December 1975.

[SHLRA92] R. G. Schwab, S. Hart-Landsberg, S. Reder, and M. Abel. Collaboration and Constraint: Middle School Teaching Teams. In J. Turner and R. E. Kraut, editors, *Proceedings of CSCW'92, Conference on Computer Supported Cooperative Work*, pages 241–248, Toronto, Canada, October-November 1992. ACM Press.

[Tic82] W. Tichy. Design, Implementation, and Evaluation of a Revision Control System. In *6th International Conference on Software Engineering*, pages 58–67, Tokyo, Japan, 1982.

[VAL90] L. Vertelney, M. Arent, and H. Lieberman. Two Disciplines in Search of an Interface. In B. Laurel, editor, *The Art of Human-Computer Interface Design*, pages 45–55. Addison-Wesley, Menlo Park, California, 1990.