Floor Lay-out Design for Effective Software Production:
Applying the Implications from the Optimal Communication Pattern of a Software Project Team

Oscar Hauptman

Working Paper
Management in the 1990s

Massachusetts Institute of Technology
Sloan School of Management
Floor Lay-out Design for Effective Software Production:

Applying the Implications from the Optimal Communication Pattern of a Software Project Team

Oscar Hauptman

90s: 86-024

June 1986

Sloan WP # 1798-86

© Massachusetts Institute of Technology

Management in the 1990s
Sloan School of Management
Massachusetts Institute of Technology
Management in the 1990s is an industry and governmental agency-supported research program. Its aim is to develop a better understanding of the managerial issues of the 1990s and how to deal most effectively with them, particularly as these issues revolve around anticipated advances in Information Technology.

Assisting the work of the Sloan School scholars with financial support and as working partners in research are:

- American Express Travel Related Services Company
- Arthur Young and Company
- British Petroleum Company, p.l.c.
- BellSouth Corporation
- Digital Equipment Corporation
- Eastman Kodak Company
- General Motors Corporation
- International Computers, Ltd.
- MCI Communications Corporation
- United States Internal Revenue Service
FLOOR LAY-OUT DESIGN FOR EFFECTIVE SOFTWARE PRODUCTION:
APPLYING THE IMPLICATIONS FROM THE OPTIMAL COMMUNICATION
PATTERN OF A SOFTWARE PROJECT TEAM

Oscar Hauptman

Management of Technological
Innovation Group
Sloan School of Management
Massachusetts Institute of
Technology
Cambridge, Massachusetts 02139

Production Operations
Management Group, Morgan Hall,
Harvard Business School
Harvard University
Soldiers Field,
Boston, Massachusetts 02163

ABSTRACT

The article takes the design of the IBM Santa Teresa software
development laboratory, as described by McCue [13], and suggests
modifications for the floor lay-out of a departmental module.
These modifications are based on recent findings by the author
[8], [9] about the information flow which seems to be most
effective in enhancing performance of software development and
production project teams.

THE INFLUENCE OF PHYSICAL LAY-OUT ON COMMUNICATION AT THE WORK-PLACE

Architectural lay-out of organizations has been found to be one of
the most powerful ways to influence the organizational communication
structure. For instance, an analysis of changes in the communication

1 The research presented in this paper has been sponsored by the
Management in the 1990s Research Program, Sloan School of Management,
MIT. This study has been facilitated by an exceptional level of
support by ICL, UK. My gratitude goes to the Applied Systems members
whose continuous participation in this study made it possible. Special
acknowledgement to Hugh Macdonald (Technical Directorate), Asa Lanum
(Director Applied Systems), and Ken Bodenham (C&TS) of ICL, for their
unwavering support. My gratitude to Sloan School, MIT Professors
Thomas Allen and Michael Scott Morton.
2 The author completed his Ph.D. studies at Sloan School of
Management, MIT, June 2, 1986, and joined the faculty of the Graduate
Business School, Harvard University July 1, 1986.
patterns of a technology-based organization by Allen and Fusfeld [2] shows that weak or non-existent communication links were strengthened or reinforced by architectural change and relocation of departments from one building to another. Subsequent studies corroborated these findings in different organizations, industries, and countries. For instance, Tomlin's data [17] from the Republic of Ireland Research Institute of the Food Industry, Allen's study [1] of the non-territorial office at the IBM Watson Research Center, Hauptman's results [9] from International Computers Limited, United Kingdom all show the direct influence of physical separation on communication patterns in these organizations. Obviously, the influence of physical lay-out is mediated by the design of the organizational formal structure (e.g., [1], [17], [9]) - members of the same project team will communicate with each other more frequently than members of different projects, controlling for physical separation, because they are more interdependent in their daily tasks. The same applies to larger organizational units such as departments and divisions. It is logical for organizational structures to reflect task interdependence by teaming together those members of the organization who are involved in similar or complementary tasks.

Before the emergence of the new information-communication technologies such as electronic mail, organizational and physical re-design could be considered one of the few effective means for the modification of communication patterns and daily interactions of organizational members. Although there is significant interest and research effort concerning the power of information-communication technologies to modify organizational communication patterns, e.g.,
the Management in the 1990s Research Program at the MIT Sloan School of Management, or the Social Science Research in Computing Program at Carnegie-Mellon University, it is yet to be evidenced through empirical and reliable research. Consequently, this paper is based on the premises and findings presented above, from the pre-information-communication technologies era.

After establishing the basis for the use of architectural lay-out as a communication modifying tool, the objective of this paper is to apply some of the findings [8], [9] about what could be considered an effective information flow for software development and production project teams to the architectural lay-out of a typical software development environment.

**OPTIMIZING THE INFORMATION FLOW TO THE SOFTWARE DEVELOPMENT TEAM**

The findings from an empirical study of the software development division of International Computers Limited, UK [8], [9] indicate that the relationship between communication and performance of software development projects requires a specific communication pattern from the development team. The prescriptions which were derived from this study can be summarized in the following terms: first, the intra-project communication among managers, designers, and programmers should be minimized. This finding is very much in line with Brooks's experience described in the seminal *Mythical Man-Month* [4], which suggests that there is a high cost to be paid for coordination and mutual re-training of members of large project teams ([15], [14], [7]). Second, open and intensive communication among the managers of
a department, which is usually based on a common technology such as graphics, data base access tools, or language compilers, is conducive to high project performance. Third, the frequency of communication of the project team members beyond their department with the members of the business center, which usually addresses a specific market application, e.g., management support software, CAM, or public administration software, seems inconsequential as far as its performance is concerned. Finally, communication beyond the business center with the members of other business centers should be minimized and in the same time controlled by a few managers -- the project managers, and the head of the department.

These prescriptions are contingent upon various qualifiers such as the complexity of the software, or its comparative difficulty. They probably do not apply to all types of software development projects, but primarily to those projects which produce moderately novel application software. For this type of projects, in view of the findings presented in [8], [9], software "development" is a misnomer. More specific constructs of R&D and production should be used for its composites. But more than that -- these composites should be managed differently. While programming or coding should probably be managed as a production or manufacturing activity, software engineering and design probably will benefit from informal communication, which will be the familiar vehicle of technological innovation of non-software R&D [1].

It should be noted here that the suggestions above imply a more formal phase demarcation between design and coding, which seems to
some extent to ignore the recently achieved feasibility of their integration [6]. It is possible that the recently evolving software CAD and code generators, as well as fourth generation languages (4GL), will be able to replace the organizational demarcation, suggested here, by technological means. Nevertheless, at present most of the software producing organizations still operate without these technologies. Furthermore, as it is convincingly argued by Yourdon [18], the availability of these technologies does not make the principles of phase demarcation or structured design and programming obsolete.

Assuming that this is the type of software we have to deal with, the following section provides some recommendations concerning the architectural floor lay-out design for a typical software development and production department.

MODIFYING THE ARCHITECTURAL LAY-OUT OF IBM SANTA TERESA SOFTWARE DEVELOPMENT LABORATORY

The Santa Teresa IBM Software development center provides the lay-out, which will serve as a strawman for the ideas derived from the studies of the relations between communication and performance of software development teams ([8], [9]). The center was designed in what might be considered one of the most thoughtful and careful attempts to create a comfortable and functional working environment for software development and production. McCue [13] describes the design process in detail. It started by defining the task as project work, which includes operating systems and application programming and
design, documentation, testing, and support. The work had to be conducted in teams of two to five members, and was to be intensive and creative. Every two to four teams were incorporated into a department with a manager and administrative support. The designers of the lab assumed that the members of the project teams would spend 30% of their time alone, 50% in groups of two and three, and 20% in larger groups. This time allocation is quite similar to McCabe's [12] empirical findings. The most essential characteristics of the individual level work environment, which were derived from several surveys at IBM include:

1. Personal work area to concentrate, screen distraction, and discourage interactions; it should also contain a terminal and sufficient space for storage of documentation and print-outs;
2. Proximity to common terminal rooms for team work and small meetings;
3. Proximity to conference rooms;
4. Proximity to computer rooms.

The team level requirements were:

1. To encourage effective communication-interactions within teams and departments;
2. Flexible spatial arrangements;
3. Integration of the administrative area with the programming area.

McCabe found [12] that members of software development teams spent 30% of their time working alone, 50% interacting with other, and 20% non-productively, on travel and administration.
Figure 1: IBM's Santa Teresa Software Development Laboratory: A Departmental Module [13]
The designers of the Santa Teresa center were aware of the numerous contradictions these requirements contain. For instance, the lay-out has to achieve privacy for the individual programmer concomitantly with open office planning; individual work space close enough to aggregate work areas; small scale identity close to central services; informal atmosphere in a large scale laboratory. The final results of this design endeavor are provided in Figure 1.

These conflicting requirements at Santa Teresa Lab are not atypical of software development elsewhere; Deutch and Taft [5] show in their empirical study that there is little consensus about the comparative importance of the dimensions of what should be a good programming environment. There are several reasons for this situation: first, although most recognize at the present the differences between design and coding, (e.g., [5], [16]), very few are ready to formalize this separation organizationally and architecturally. But probably the most central cause of the implicit confusion about the optimal programming environment is the incomplete understanding of what "effective communication-interactions within teams and departments" really means. The way it is worded in [13] implies, without much empirical evidence, that more communication is better. Based on my empirical findings [8], [9], this is not quite true for intra-project communication; on the contrary, we should actually try to minimize this type of communication as a costly coordinative overhead (e.g., [16], [10]). And while the

---

1 Howlett [10] formulates the total output of an n members team working jointly on a totally divisible task as W(n)=n-kn(n-1), where k is the constant proportion of time consumed by coordinative.
Figure 2: Lay-out of a Software Development Department: Applying the Results About the Optimal Information Flow [9]

Legend:
DM - Department Manager
PM - Project Manager
Mr - Marketing Expert
V - Validation
TA - Technical Author
D - Designers' Offices
P - Programmers' Offices
PS - Project Secretary
S - Secretary
C - Conference Rooms

Project Module

Project's Common Area

Stairs

Corridor to Another Module
intra-departmental communication is conducive to project performance, the intra-business center communication is inconsequential, and the inter-business center communication even has a negative effect on project performance.

In addition, because the domain of the project was found to be so local, this implicitly suggests that a diverse, multi-functional staff at the department level would be more effective. Consequently, such important components of the software product development and production process as the validation team, the technical author, and possibly the marketing expert should be incorporated in the department.

The floor lay-out, which takes the Santa Teresa software development house, and modifies it according to the findings about the optimal information flow prescribed above is presented in Figure 2. This lay-out should facilitate a better technical coordination among the key members of the department by bringing the designers, managers, validation staff, a marketing expert, and a technical author physically together at the center of the floor. At the same time it should reduce the amount of organizational coordination between designers and programmers, and among the programmers themselves, through physical isolation: first, the programmers will occupy individual offices, as suggested by the initial design of the Santa Teresa Lab [13], and not an open space floor with partitions so widespread in the software industry (e.g., Applied Systems, International Computers Limited, UK). Second, the designers are to be communication. Consequently, the net contribution of a new member added to the team, after integration and training is \( W(n+1) - W(n) = 1 - 2kn \).
located closer together, somewhat separately from the programmers, towards the central core of the floor. Although this separation is not very restrictive, it should emphasize the differentiation between designers' and programmers' tasks, and it should also enhance the structured design-programming approach - the completion of software design before the product is moved to the coding stage.

Concerning the intra-business center, and the inter-business center communication, the initial design of the Santa Teresa departmental module fits the information flow requirements described above: its location in a separate module, on one floor, naturally isolates the department from other departments of the business center, and from other business centers. On the other hand, the suggested location of the managerial staff of the department close to the hub of the building, by the stairs, will make the managers of different departments and business centers more mutually accessible.

**SUMMARY**

The modified lay-out described in Figure 2 puts the coders, who can be regarded as the software manufacturing personnel, onto the periphery of the floor, and the software engineering specialists - the designers, towards the center of the floor. The managerial and the support staff such as validators, technical authors, and marketing specialists are concentrated as closely as possible to the managerial and support staff of other departments and business centers. It also creates the small group environment for the development team, in which the rest of the organization is not quite visible and accessible to
most of its rank-and-file members. In a sense, the isolation of the project team developing a software product enhances its productivity, in contrast with the non-software R&D project teams (e.g., [3], [11]).

It is important to emphasize at this stage that the empirical findings on which the architectural recommendations are based ([8], [9]), stem from a study of a single organization, in this case—the Applied Systems division of ICL. They should be corroborated by additional research in other organizations in the USA and possibly, Japan. In addition to this word of caution, the architectural recommendations for floor lay-out of a software development department presented in Figure 2, have never been tested empirically at IBM Santa Teresa or anywhere else. Consequently, they can serve as general conceptual guidelines about architectural design of software development facilities. The influence of the suggested environment on software project teams communication and performance should be tested empirically by managers of software development and production.
REFERENCES


