Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

Experimenting with Virtual Laboratories for Distributed Software Development Teams

John Nosek
Computer and Information Sciences, Temple University, nosek@cis.temple.edu

Murad Mandviwalla
Computer and Information Sciences, Temple University, mandviwa@vm.temple.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis1996

Recommended Citation
Nosek, John and Mandviwalla, Murad, "Experimenting with Virtual Laboratories for Distributed Software Development Teams" (1996). *AMCIS 1996 Proceedings*. 140.
http://aisel.aisnet.org/amcis1996/140

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1996 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Experimenting with Virtual Laboratories for Distributed Software Development Teams

John Nosekand Munir Mandviwalla
Temple University, Computer and Information Sciences
nosek@cis.temple.edu, mandviwa@vm.temple.edu

Introduction

Software development is increasingly complex, distributed, and collaborative in nature. The VLab (Virtual laboratory) provides any time-any place process support for software development teams working on complex "real world" problems. The goal of VLab is to shift the way software development teams are taught, and provide insights on how to support distributed/mobile work teams. The VLab is being tested in a pilot research project that started in Spring, 1995. We present early results from our project including a process based model of complex software development and learning, an instrument designed to capture the interactions in complex software development and the applicability of process support technology. We also discuss early qualitative results of matching process support technology with specific project objectives.

Background

Organizational work increasingly demands the completion of complex tasks by distributed teams. Systems development is becoming increasingly complex and the pressure to build fully-tested, safe information systems in less time is growing (Abdel-Hamid, 1990). However, the skills needed for complex systems development can be distributed across the department, organization, country, or the globe. Even if a development team is based in the same physical location, scheduling problems, and the option to telecommute make it difficult for a team to meet.

Traditional college instruction that presents material in scheduled, formal, once or twice-a-week chunks and emphasizes measurable, individual learning does not succeed in preparing students for the work place of the future. Current learning approaches must be redesigned to apply process support technology to teach students:

- how to work in distributed teams to solve complex tasks;
- how to use technology to structure group problem solving;
- and how to use technology efficiently and effectively to support distributed work processes.

VLAB

In 1994 Temple University was awarded a grant from the National Science Foundation to create virtual laboratories for students working in software development teams. The VLab (Virtual laboratory) provides any time-any place process support for software development teams. Nosek and Mandviwalla (1995) describe the proposed goals of the project. This paper reports on the continuing research effort.

The VLab consists of notebook computers connected through an ad-hoc peer-to-peer wireless LAN, dial-in wide area connectivity, and group decision making (GroupSystems), information sharing (Lotus Notes), and desktop conferencing software (Intel Proshare). (A detailed technical discussion is available).

First, we present a model that focuses on the processes involved in complex systems development and learning and which helps to inform the approaches that have evolved to support and evaluate these processes. The next section describes the action-research development of the work activity and interaction log being used to measure important process variables. The last two sections describes the early qualitative results of matching technology with specific project objectives and future expectations and directions.

Model

A useful way to envision the process of teaching complex systems development by distributed teams is to view it as a process within a process (see Figure 1). Student teams are engaged in the process of developing a computerized software system for a "real-world" problem (the inner process). The instructor and students are also in engaged in the process of learning how to develop a computerized software system for a "real-world" problem (the outer process). Some of the outcomes of the inner process, such as the quality of the developed system and user documentation, are used as outcomes of the learning process, e.g., the better the quality of the system and documentation, the better the students have learned how to develop systems. The inner process, for the most part, has been hidden from instructors and therefore the opportunity to effect changes in how students actually perform the systems development process has been limited.

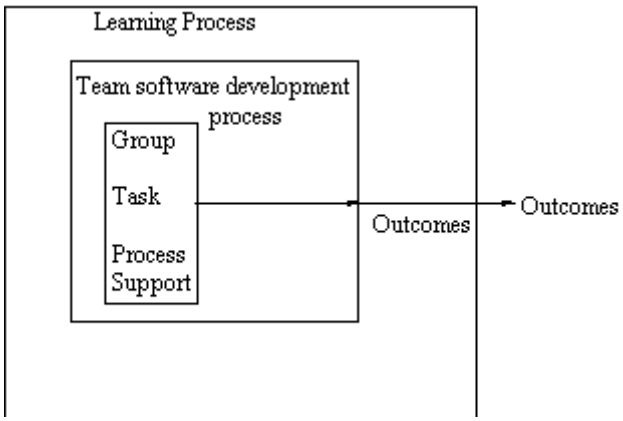

Figure 1: Process-within-a-process Model

Processes, Obstacles, and Redesign

Using the "process redesign" approach of Mandviwalla and Hovav (1996) we are decomposing the basic processes involved in systems development (the inner process), and the associated learning context (the outer process).

Examples of systems development processes:

- scanning and filtering
- interpreting user needs and synthesizing requirements
- evolving a shared mental model
- applying textbook techniques to real world scenarios
- interacting with users, team members, and manager
- understanding instructions from the manager

Examples of generic teaching and learning processes:

- questioning users, instructors, and team members
- discussing facts with users and team members
- sharing documents
- facilitation by the instructor when a problem arises
- lectures and demonstrations by the instructor
- encouragement from the instructor

Problems associated with the above processes:

University Structure: It is difficult for team members to meet, especially in an urban university setting. The problem of team members meeting is exacerbated when team members must not only find opportunities for which they can meet, but now they are restricted by end-users who are only available during the normal work day.

Laboratory Structure: Fixed location, computer-based laboratories limit access and do not support team efforts at off-campus work sites.

Course Structure: Formal, once or twice-a-week instructor-centered presentations are poor choices for showing students how to apply development techniques The structure limits the opportunities for questioning and discussion between instructor and student and among students. However, interpretation of user needs and evolving a shared model usually requires extensive interaction.

Evaluating Team Work: Evaluation of team efforts is extremely difficult. Team effort is mostly limited to evaluation of deliverables. Individual and team process activity is hidden and poorly evaluated.

Instructor Participation: There is little opportunity for instructors to "observe" the process or to "demonstrate by example." The instructor evaluates outcomes, but has limited ability and opportunity to observe and correct the process. This implies that:

- instructors have less influence on the student's process of system development.
- instructors can not evaluate who on the team may be stronger or weaker.

Scalability of learning: The scalability of instructor resources should be possible through better collaborative learning (Johnson & Johnson, 1991). By nature, team members will seek other team members to solve problems, especially lower level procedural questions, which do not require instructor intervention.

Work Activity and Interaction Log Development

This section describes the action-research development of the work activity and interaction instrument called the "log." The objectives of the instrument include:

1. Develop a base line of individual and group work, including interactions, prior to the introduction of process support technology.
2. Open up the team systems development process:
    - understand the individual/group activities and interactions
    - understand the current problems with activities and interactions
    - better evaluate individual and group effort
3. Based on research. For example, longitudinal testing of Daft and Lengel's information richness theory.
4. Easy to use.
5. Capture statistically useful information.

Initial pilot testing of the log was conducted in a systems development course during the Spring 1995. Baseline information was gathered and the log was modified to improve ease of use. In Fall 95, weekly log completion and submission were made part of the course requirements:

1. The class was actively engaged as part of the research activity. For measurement validity purposes, the students were not informed of the specific theories that guided the development of the instrument. However, students were told of the VLab project goals and the need to obtain good information from the logs.

2. Students were encouraged to make modifications to the logs and submit these suggested changes on the logs. Feedback from the students improved the instrument and clarified the correct way to complete log entries. For example, the comment section was increased in size to allow room for contextual comments needed to clarify entries.

The revised final log captures the following: Information is checked for each work activity or interaction. For each interaction, the following information is obtained: Date; begin and end time; scheduled/ad hoc; whether a computer was used; the software used; the initiator and others involved in the interaction; location (campus/work/home); the medium (phone, email, FTF…); the degree there was a need for shared screen/view; pointing; shared files/documents; satisfaction with the medium, process and outcome, definition of the task and comments.

Technology Deployed and Early Results

After pilot testing a range of technologies last Spring and Fall, several technologies have been in use by the students since the beginning of the Spring semester. Table 1 is a current reflection in matching a set of process objectives for improving the efficiency and effectiveness of systems development and teaching/learning processes with VLab technology. The table also provides outcome measures for the process improvements and early qualitative comments.

Goal	Process Support	Outcomes
Increase access to people	Wireless communications; Desktop conferencing	Count contacts
Increase access to technology	Wireless communications; notebook computers	Log Usage: Time and Location
Increase interactivity	Desktop conferencing, E-Mail, and Bulletin Boards	Count frequency of interaction on a topic
Increase information sharing	Document Sharing	Count volume of information shared
Improve task definition	Group Decision Support	Test students on how well they feel tasks are defined
Shared understanding of task	Group Decision Support	Test students on task

Table 1: VLAB Goals, Technology, and Measures

Early results

In terms of increasing access to people and interactivity most students only have access to one phone line. Modems that allow simultaneous voice and data on one line are only available currently as internal cards for desktop machines. There are external modems available that allow switching from voice to data. However, the ones that we have tried do not work. This limits the value of the desk top conferencing. Wireless and dial-up connectivity is currently being tested for use by all the students.

Increasing access to technology and document sharing has worked extremely well. Major benefits to date are that students have the same hardware and software available any and everywhere. This has greatly enhanced progress on system development. The wireless LAN is especially helpful of transferring data between systems so that all group members are working on the same version. Students are still in experimental stage with technology for improving task definition and shared understanding of the task.

Summary and Future Directions

Organizational work increasingly demands the completion of complex tasks by distributed teams. Traditional college instruction does not develop the skills that the work force of the future will need. VLab is one effort to match the learning requirements of the non-traditional student engaged in distributed software development with process support technology. What we learn in VLab should improve our

understanding of how to better match process support technology with learning requirements and provide guidance for these kinds of problems within and outside academia.

References available on request from first author