Using an Issue-Based Model in a Team-Based Software Engineering Course

Allen Dutoit, Bernd Bruegge and Robert Coyne

EDRC 05-97-95

Using an issue-based model
in a team-based software engineering course

Allen Dutoit (dutoit@edrc.cmu.edu)
Bernd Bruegge (bruegge@cs.cmu.edu)
Robert Coyne (coyne@edrc.cmu.edu)
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

Communication in software engineering projects becomes a bottleneck as the number of participants increases. As today's software systems grow in complexity and size, teaching effective communication skills in software engineering courses becomes a critical issue.

This paper is an experience report on the use of a issue-based model for teaching meeting procedures in a team-based software engineering project course (7 teams, 25 students). We observed that, when carefully introduced in the classroom, the use of an issue-based model provided significant benefits, even with such limited tool support as a word processor template. More specifically, we observed that students conducted meetings more efficiently, that they maintained a more complete record of the issues under discussion, and that intra-team communication was significantly improved[1]

1. Overview

Our educational focus has been to provide students with a realistic software engineering experience. We have done this by immersing students in a single, team-based, system design project targeted to build and deliver a complex software system for a real client[1][2][3][5]. We grade students according the results of the client review. We evaluate our success in teaching software engineering according to how well the students perform.

As the number of students taking the course has grown larger (from 12 in 1989 to 65 in 1995), communication between project participants has become a bottleneck, as others have observed[7][12][13]. We observed that propagating and maintaining up-to-date information across the project becomes increasingly difficult, due to the amount of information to maintain, the increased rate at which it evolves and the increased latency of the communication between participants.

In our context, the communication problem is exacerbated by the distribution in time and space of our organization: the software engineering course is typically only one of four courses that students take during the semester. Students have different schedules and work from different computer clusters, making informal, spontaneous interpersonal interactions less frequent. This has a significant impact on the overall project communication given that informal interactions are usually the primary mode of communication in non-distributed organizations[13].

We initially dealt with the distributed nature of our organization by the use of electronic bulletin boards (bboards) in Fall'91[1]. We then taught students to post agendas and minutes of their weekly meetings on their team bboard. This provides the team with a record of their decisions and provides the other participants with access to the state of the team, often before other means of communication. Finally, we introduced teams whose role was to integrate various aspects of the work produced by each teams (Fair93[5]). These cross-functional teams were composed of representatives of each subsystem teams.

We observed that, given the distributed nature of the course, a significant number of decisions are made and communicated during weekly team meetings. Although these meeting procedures address part of the problem of capturing and rapidly propagating information, we have also observed that it has several serious shortfalls. First, there is a wide variance in the quality of the meeting process (e.g. communication bandwidth, number of decisions) and the quality of their record (e.g. readable minutes, action items). Second, the chronological and free style of minutes makes it difficult to extract specific issues and decisions for participants not attending the meeting. Generally, the lack of information structuring makes browsing and searching information through minutes difficult.

Issue-based models (such as IBIS presented in [10]) have been promising for addressing communication issues in distributed time contexts (e.g. gIBIS for capturing design rationale [16]), distributed space (e.g. rIBIS for capturing real-time conversations [15]) and both (e.g. scientific research data management[8]). In other cases, issue-based models are also used for process support (e.g. requirements elicitation[14]).

We attempted to address the challenge of teaching more efficient meeting procedures through the use of an issue-based model (hereafter refer to as itWEB[2]). We extended the IBIS model[10] to include a resolution construct, and integrated it with a simple task model for documenting meeting action items. We then designed an agenda and minute document template describing which aspects of the meeting should be captured. We also defined meeting procedures describing how the agendas and minutes should be written and posted. We provided very simple tool support (i.e. a word processor

---

2. itIWEB stands for "indented text Information WEB". See Section 3.2. for the genealogy of this acronym.

template) to minimize artifacts associated with introducing a new technology in the classroom and to keep the focus on the issue model.

We introduced itIWEB in the Spring'95 course. After a training period, we then left the students with the choice of continuing its use or not, in order to observe whether the students themselves perceived it as a useful (they did). We then analyzed the communication traffic and the surveyed the students in order to get better insight in the benefits and drawbacks of itIWEB.

We observed that the use of itIWEB lead to a more efficient meeting process and a better record of meetings, compared to a previous course. We also observed improvements in the intra-team communication. However, we were not able to measure significant inter-team communication improvements.

This paper is structured as follows: Section 2. provides background information about the software engineering courses we teach and its communication infrastructure. Section 3. describes itIWEB and its introduction into the Spring'95 software engineering course. Section 4. presents a discussion of the benefits and drawbacks which we observed. Section 5. presents our future plans for subsequent courses.

2. Communication in software engineering courses at Carnegie Mellon University

2.1. Undergraduate SE courses at CMU

Our goal in teaching software engineering in the School of Computer Science of Carnegie Mellon University has been to provide a realistic software engineering experience to students. We have done this by immersing students in a single, team-based, system design project targeted to build and deliver a complex software system for a real client. In Spring'95, the students built an emissions modeling system (called JEWEL[3]) for the Environmental Protection Agency of the US Government.

In these courses, students are exposed to a broad range of issues relevant to software engineering: technological issues (e.g. object-oriented databases, speech recognition), process issues (e.g. object-oriented methodologies), and organizational issues (team communication, meeting organization and capture). Given that most of these issues are still subject to research and do not have "right" solutions, we involve the students in the decision making and customization process required to set up the project. We provide the students with training, tools, techniques and advice, while they are left with the choice of what they feel works best for the project. The only non-negotiable items in the project are the client, the deliverables and the final deadline.

We teach two four month courses per year. The Fall course is a senior level software engineering course aimed at giving students a basic understanding of the most important issues in

software engineering. While these students typically have a strong programming and theoretical background, they haven't been exposed to team work and organizational issues. They also typically have not been exposed to the construction of a large piece of software. The Fall course is usually composed of 40 to 60 students.

The Spring course is an advanced software engineering class. Typically, about half of its students have taken the Fall course. The Spring course addresses issues such as software re-engineering, rapid prototyping and code reuse. The starting point of the Spring course is the set of deliverables left by the preceding Fall course. The Spring course is usually composed of 12 to 25 students. We often try new technologies and procedures in the advanced course, as students are more familiar to technological and organizational issues.

Given that the software engineering course only represents one of the four courses that students take during the semester, we view the project as an effort distributed both in time and in space. Except for class meetings, students have different schedules, work on the project at different hours and from different computer clusters. Given the part-time and distributed nature of the course, we organize students in small teams (3-6 students) and mainly rely on meetings and electronic bboards for communication.

2.2. Organizational & communication structure

The class is divided into teams, each of which is responsible for a subsystem (both from the coding and documentation point of view). Initially, the class also included an additional team of students which functioned as a service provider to other teams (e.g. revision control tool installation, makefiles development, etc.). In FalF93, we replaced service provider teams by cross-functional teams, which were composed of one representative from each subsystem team. In Fall'93, we also introduced cross-functional teams whose task was the integration of design, code or documentation. We found that cross-functional teams accomplished system support and integration tasks much better than specialized service provider teams due to a better knowledge of team specific requirements and problems. In Spring'95, cross-functional teams were: an *architecture* team who was responsible for the system design and subsystem interface definition; an *integration* team, responsible for code change requests and configuration management; an *information* team, responsible for the integration of documents, the development of world wide web services and the delivery of the prototype to the client.

Each team is also assigned a teaching assistant or instructor, who functions as a coach. The role of team leader is a rotating role and is always filled by a student. The team leader plays also the role of facilitator during team meetings.

The main medium of communication has been a hierarchical structure of electronic bulletin boards (bboards). Each team is assigned a set of bboards which are primarily used for their internal communication. Students are also strongly encouraged to read the bboards of the other teams, which frequently leads to team bboards also being used for team to team

---

3. Joint Environmental Workshop and Emissions Laboratory

communication. In addition to team bboards, we provide two bboards for project-wide discussion and course announcements.

Students meet twice per week during class time. At the beginning of the project, class time is spent mostly for training and organizational setup. As the project progresses, class time is used for internal reviews, and finally, for client reviews. All teams (including cross-functional teams) meet formally every '- week. For students functioning as liaison, this represents up to two team meetings pet week in addition to two class meetings. We found that, given the distributed nature of the project, the weekly team meetings were a very critical component in the project communication. Meetings became even more critical once the cross-functional teams were introduced into the class organization.

2.3. Meetings: organization & capture

Given that students have not been exposed to team work prior to the Fall course, we spend additional time at the beginning of the semester to teach students how to conduct and capture meetings.

Each meeting has a facilitator and a minute taker. The facilitator writes the meeting agenda, posts it on the team bboard a day ahead of time, and ensures that all items of the agenda are addressed during the meeting. The minute taker is responsible for recording the discussion and posting these minutes no later than a day after the meeting. Each team is provided with a laptop computer allowing the minute taker to type minutes in real time (or at least, to take notes allowing for writing minutes after the meeting)

Until Fall'94, we provided a text template for agendas and minutes. The agenda template was composed of a header identifying the meeting (date, time, location and audience), the goal and expected outcome of the meeting, a list of issues to be addressed during the meeting and a list of status item that team members were to report on. The format of minutes was left to the students. We provided some guidelines on the content of minutes, i.e. that minutes should contain a record of the discussion on the agenda items (in addition to any other item brought to the meeting), a summary of the team status and a list of action items to be carried out by individual team members.

Although most students followed these procedures, we observed a wide variance in the quality of the meetings and their minutes. As ti^e project size has grown, we find it impossible to build a consistent picture of the project based only on posted minutes.

2.4. Meetings: observations & issues

We found three problem areas with the meeting process: its preparation, the way it is conducted, and the capture of its content and its surrounding context.

When the number of students grows and their background is diverse (e.g. cross-functional teams), significant time is spent during the meeting, negotiating the agenda and understanding the context surrounding the agenda. During the meeting, the facilitator's role is critical in maintaining the scope of the meeting within the limits set by the agenda. Only few students with good leadership skills managed to succeed in that role.

The role of the minute taker suffers similar problems: when the facilitator was not able to fill its role, recording minutes in a coherent form is difficult. Most of the minute takers recorded minutes as a sequential list of statements made by various meeting attendees. The less organized the meeting, the less readable the minutes. Again, only a few students would take time reorganizing their minutes before posting them.

Another problem most minutes suffered was the lack of context: reading minutes without attending the meeting made it difficult to understand the issues being discussed. Team meetings are usually spent exploring alternatives and agreeing on solutions, however specific issues are usually implicit and known to the attendees, thus not mentioned in the minutes.

Finally, minutes posted by the students usually recorded the formal part of the meeting: starting when all attendees were present and stopping when the meeting was adjourned. The hallway discussions before and after the meetings were always omitted from the minutes. These omissions are significant knowing that some of the task assignments was frequently done after the meeting, on the way to the computer cluster.

In summary, conducting and capturing meetings is generally a challenging task for students, although a few of them succeeded in that task. In teaching die meeting process in our course, we were looking at techniques to improve the general process and make it more efficient for a majority of students. A case study of the use of IBIS in an industrial context [16] reported improvements in consistency and completeness when recording a design rationale. Given the similarities of the problems being solved, we took IBIS as a starting point, as described further in the next section.

3. itIWEB: an issue-based model for capturing meetings

3.1. IBIS discussion model

IBIS[10] is a method and an issue-based model, which has been developed for capturing design rationale on large, complex, design problems. IBIS has been used with success both in research and industrial contexts, with various level of tool support[8][14][15][16]. The observed benefits of the method were that: a) it improves consistency in the way information is captured (i.e. independence from who captures the information); b) it improves the completeness of the information captured; c) it improves the accessibility of information, by providing structure.

IBIS structures information as a graph with three types of nodes: *issues, positions* and *arguments.* Each node is a piece of natural language text. Nodes refer to other nodes via links of different types: for example a position has a *responds-to* link to an issue. An argument must be linked to a position via either a *supports* or *objects-to* link. Issues may be organized by linking them to other issues via *specializes, generalizes, replaces* or

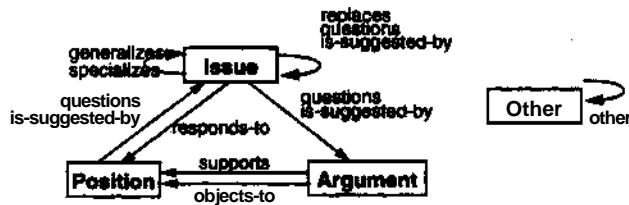*questions* links (see Figure 1).

Figure 1: The IBIS model

In order to capture information which does not fit in the issue-position-argument paradigm, IBIS provides a "catch all" node type called *other*. It may be used to refer to information supporting a discussion, such as excerpts from a customer requirement document, a piece of source code and so on.

3.2. IWEB meeting model

I WEB [6] (Information WEB) is an information modeling and management system targeted to supporting team-based software engineering. It is part of n-dim[9][ll], an ongoing project at the Engineering Design Research Center whose intent is to address the broader information management problem in engineering design. IWEB addresses communication issues through the integration of an issue-based model, free form notification and other models (e.g. task, artifact and organization models) into a single information management environment. itIWEB ("indented text IWEB", inspired from itIBIS[16]) is the low tech implementation of a subset of the IWEB models targeted specifically at addressing the meeting issues we previously described. itIWEB is a stepping stone in the IWEB research effort.

The IWEB issue-based model is similar in essence to the IBIS model. We renamed the links and node with terms which are used in the object-oriented world to express similar concepts. We also renamed the position node to *proposal* node to indicate more strongly that the proposed solution and its supporting rationale should be kept separate in proposal and argument nodes. We also added a *resolution* node to capture the decision which was made on the issue. The resolution node refers to the issue or issues it addresses through a *resolves* link. It also refers to any number of proposals through a *based-on* link.We differentiate proposals from resolutions because we found (through our use of issue models) that resolutions are often a synthesis of several proposed solutions. We were interested to capture this in the structure of our model. We also kept the concept of a "catch all" node and link of IBIS to refer to other models in IWEB (e.g. a task model, described below).

After several attempts to capture our own meetings with this issue model, we felt the need for an explicit task model to document the outcome of the meeting. Resolutions were used to capture decisions, but we needed an additional construct to document how these decisions were to be implemented. We introduced a simple task model composed of a single node, called action item, containing the name of the person who is assigned the task and description of the task. The action item is implicitly to be accomplished by the next meeting. The action

item may also contain references to other nodes in the issue-model (e.g. such as the resolution which prompted this action item). At the next meeting, the discussion of the action item status is recorded using the issue model.
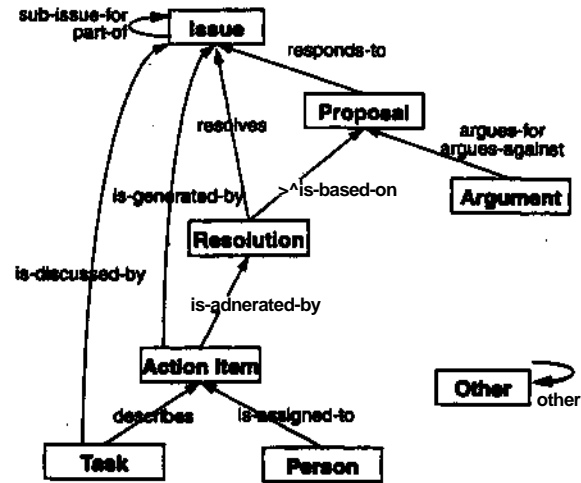

Figure 2 IWEB discussion and task model

Finally, we organized the issue and task model for each meeting into an agenda and a minutes. We kept the main sections of the previous agenda and meeting templates and only modified the way their content was structured. An agenda is composed of a header identifying the meeting it refers to (i.e. date, location, audience), a list of action items to be accomplished by the meeting (i.e. the action items which were generated in the previous meeting) and a list of issues which will be discussed during the meeting. The minutes contain the same header, the discussion of the action items recorded as issues, proposal, arguments and resolutions, and the discussion of the issues listed in the agenda. The minutes may also contain any additional issues which were generated during the discussions. The last section of the minutes contains the action items generated as a result of the discussion. The new action items refer to the resolutions and the issues which generated them. The discussion and task model of IWEB is described by Figure 2.

3.3. itIWEfl

For our first test of IWEB in a software engineering course, we used the IWEB model in a textual form. The rationale for using a low technological support was: a) to minimize the transition effort by not introducing yet another new tool; b) to facilitate the integration of the model with the existing communication infrastructure (i.e. bboards); c) to isolate the impact of the model vs. tool issues.

We represent each node as tagged paragraphs. The tag represents the type of node (I: for issue, P: for proposal, A+: for argument for, etc.). The tag of a node may also contain a number in brackets for the purpose of reference (e.g. I[21]:). Links are represented by the relative position of the node in the text and by its indentation level. For example, proposals responding to an issue are located below the issue and indented

to the right. Finally, in order to capture non-hierarchical structures (e.g. a resolution based on a set of proposals) we introduced a reference tag. Figure 3 is a small itIWEB example illustrating these concepts.

1(0): Which policy should be used for retrieving obj^ect attributes from the database?

    P[l] : On demand per attribute.

      A-[2] : Lower throughput.

      A+[3]: Simpler.

    P(4) : On demand per object (prefetch all attributes).

      A+(5): Overall better performance: during emission calculations, we often need all or most the attributes anyway.
      {ref: 1/31 emission meeting}

R(6) : Implement P[4]. However, the caching should be implemented in the database layer, allowing use to hide the fetching policy. If all else fails, we will fall back on P(3).

AI(7): For: Tom. Task: Modify fetching policy without modifying current API.

<center>Figure 3 itIWEB example</center>

We also refined the meeting process used in previous courses to minimize the overhead introduced by the issue model. We asked the facilitator to post the agenda at least a day in advance such that not only the team members can be informed of the content of the meeting, but also the minute taker, who then can transfer the agenda on the team laptop prior to the meeting. The on-line version of the agenda could then be used as a template for the minutes. As the facilitator followed the agenda during the meeting, the minute taker recorded the discussions at the appropriate places in the minutes. During the last five minutes of the meeting, we taught students to "wrap-up", that is, for the minute taker to summarize which issues were addressed and resolved, and compile a list of action items based on these resolutions. After the meeting, the minute taker cleaned up the minutes and posted them to the team bboard within a day of the meeting.

It can be observed that we did not drastically changed the essence of our meeting procedures. Instead, we designed a formalism and adapted its use to our previous meeting procedures, hoping that it would serve as a better teaching vehicle and improve the overall meeting efficiency.

3.4. Pedagogical issues & technology transition

Based on our previous experience introducing modeling techniques into the classroom[l][5] and on others* use of IBIS in industrial prqjectsflO], we took great care planning the introduction of itIWEB. Generally, we found that the mere availability of a technique, however useful and well-supported by a tool, is not sufficient for students to use it effectively. It is only when students are fluent in the use of the given technique and when they are convinced that it makes their life easier that

the technique is used effectively.

The initial barrier we have to overcome in such transition is the resistance of students to change existing work practices. In our case, more than half of the students of the Spring course took the Fall course and had some experience conducting and capturing meetings.

A second barrier is the artifacts associated with tools supporting the technique. The introduction of yet another tool in the classroom represents more training time, more computing resources and more frustrations associated with the integration of the tool with already accepted tools (e.g. a word processor, the bboard communication infrastructure).

Finally, the transition process of different tools and techniques compete with each other and with the development procfess itself. Typically, students are only willing to spend time learning about new tools and techniques at the initiation of the project, and will generally focus on the ones which are easier to learn.

We carefully addressed these three issues when we introduced itIWEB in the Spring'95 course. We planned the introduction of itIWEB as a four step process:

1. a lecture describing the communication problems in general and motivating itIWEB as a solution;
2. a homework allowing students to practice the syntax of itIWEB without interfering with their current meeting process;
3. a tutorial conducted in the context of each team, during which the team learned practices the meeting procedures required by the introduction of itWEB;
4. a follow up lecture at midterm reporting how well itIWEB was used by the class and which benefits the instructors observed.

The lecture was given at the beginning of the semester, before teams were able to setup their meeting procedures. The first three steps were completed within three weeks of the beginning of the course, such that the introduction of itIWEB would not compete with the development process. The homework and tutorial gave the opportunity for the students to practice the technique, and potentially foreseen its benefits. The follow up lecture (in this case, reporting that itIWEB was yielding more benefits than expected) gave the final push which led the students to use the technique almost all the way through the end of the project

We addressed the tool support and integration issue by providing simple word processor templates for writing agendas and minutes. We also provided a template for the word processor used in the course (FrameMaker). Given that students were using bboards as their main communication infrastructure, the integration with their current tools was trivial.

4. Evaluation

We evaluated itIWEB along three axis: the acceptance of the meeting procedures and of die issue model, its impact on the meeting process and its impact on bboard communication. Given that we left the students with the choice of accepting or

<center>5</center>

rejecting the use of itIWEB, we interpreted its acceptance as a strong indicator of its perceived usefulness. We evaluated the impact of itIWEB on the meeting process in order to observe if itIWEB was a better vehicle in teaching more effective meeting procedures. Finally, we studied the impact of itIWEB on the bboard communication to see if the use of itIWEB had broader consequences on communication.

4.1. Approach

We observed the course from three perspectives: we recorded our observations as instructors during the course; we conducted a survey of the students at the completion of the course, and we collected quantitative data from the bboard traffic.

We conducted the student survey in a semi-structured fashion: we submitted them a set of questions about the class organization, methodology, technological support and communication infrastructure. Students were strongly encouraged to be as critical as possible and to suggest any improvements they could think of for subsequent courses. The survey was not anonymous, since we wanted to interpret student answers in the context of their team and their background. However, we reviewed the survey only after the academic end of the semester and we informed students that their answers would not influence their grade. In past courses, we found that this approach was successful for gathering critical and constructive suggestions from students.

In the Spring'95 survey, we asked two questions related to the use of itIWEB and the meeting process:

- "What were the pros/cons of using itIWEB to structure minutes and agendas with respect to inter/intra-group communication?"
- "What group roles (minute taker, leader, etc.) did or didn't work for your group?"

18 of 25 students answered these two questions, averaging a third page answer for each question.

In order to confirm (or infirm) our observations and those of the students, we collected data on bboard traffic. We also attempted to compare it with data from the Spring'94 course which we used as a baseline. The Spring'94 and Spring'95 courses were of similar size and organization. The project was different, however the architecture of both systems was similar and so was their implementation technology. Finally, both courses produced a successful outcome. However, given that these two courses do not represent enough data to build a statistical case, this data is only presented in this paper for its anecdotal value. We still feel these results are worth reporting given the number of students who used the issue-based model.

4.2. Acceptance

Out of 18 surveys, 16 students reported using itIWEB in their process. Out of these 16, 4 reported that they found it very useful, 9 reported that it was moderately useful, while 3 students mentioned they perceived it as a time sink. 2 students admitted not using itIWEB.

From a quantitative perspective, we compared the number of agendas and minutes written using the model with the total number of minutes and agendas. Figure 4 display these numbers for each subsystem team (*database, emission* and *ui*) and each cross-functional team (*information management, architecture, integration* and *prototyping*).

In Figure 4, we observe that all subsystem teams, the information team and the architecture team consistently posted agendas for their meetings. The lower number of agendas posted for the prototyping and integration team was due to a smaller number of formal meetings. On one hand, these teams started work only after midterm (due to the nature of their task), on the other hand, these teams were event driven, and therefore, solved issues as they occurred through e-mail and hallway discussions. In general, we found in previous courses that event driven teams have much more difficulty setting up formal procedures.
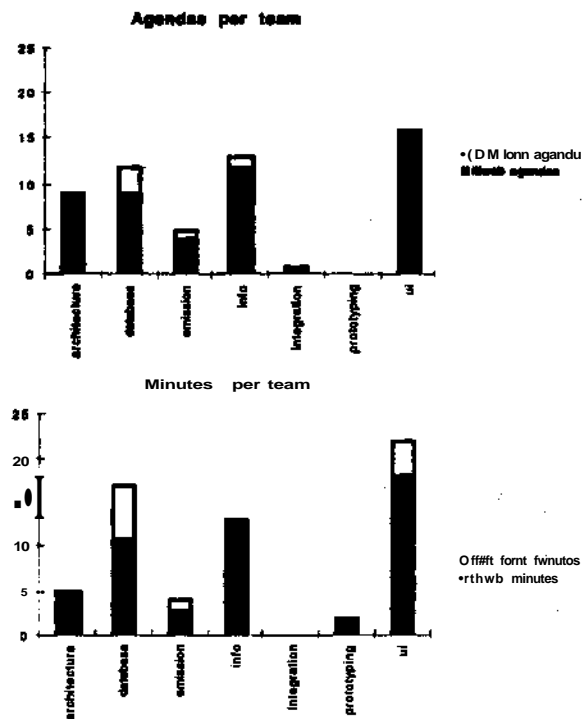

Figure 4 Number of meeting documents per team vs. free form meeting documents.

Considering only subsystem teams, we can observe that the emission team posted relatively few agendas. A closer observation of their meeting process revealed that the attendance rate of their weekly meeting was lower than average (3 out of 6) due to schedule conflicts and job interviews. For this reason, the rotating role assignment did not work and quite often, either the facilitator or the minute taker (or the laptop) were absent from the meeting. Near the end of the semester, the emission team fell back on spontaneous crisis driven meetings held on weekends. It is to be noted that from all subsystem teams, the emission team delivered the least amount of working code (emission: 3 Klines, database: 13.9 Klines, ui: 10.4 Klines).

For the teams which followed the weekly meeting procedure,

we observe that the use of itIWEB was widely accepted. Overall 91% of all posted agendas were written using itIWEB. Moreover, a closer examination of the 5 agendas which were not written using itWEB reveals that they were posted during the transition period, before the completion of the tutorials.

We observed similar patterns for posted minutes (see Figure 4). The integration and prototyping teams posted less minutes than the other teams; the emission team, compared to the other subsystem teams posted less minutes. All the other teams 'consistently posted their minutes. 83% of all posted minutes were written in itIWEB.

, * We also observed that more minutes were posted than agendas. Direct observation showed that often the facilitator would bring hard copies of the agenda to the meeting. Other times, the agenda was similar enough to the agenda of the previous meeting, and therefore was not posted. In both cases, we observed that it did not impact the capture of minutes in itIWEB.

The gap between the number of structured and unstructured minutes is not only due to minutes posted before the completion of the itIWEB tutorial (as for the case of the agendas), but also to minutes posted during the integration phase (i.e. when the project suffers a time crunch). The student survey showed that posting structured minutes took more overhead than free form minutes, and that unformatted notes were posted at the postprocessing step in order to save time.

4.3. Impact on the meeting process

Students perceived the impact of itIWEB on the meeting process as the most significant one. 10 out of 18 students mentioned that the use of an issue-based model increased the efficiency of the meeting. The contributing factors were that:

- prior to the meeting, the facilitator was able to produce an agenda using the minutes from the previous meeting (i.e. which action items needed to be reviewed, which open issues were still to be addressed, etc.),
- during the meeting, the facilitator had a better tool for keeping the scope of the discussion on track,
- during and after the meeting, the minute taker was better able to organize the minutes according to issues and action items, given that the structure was available from the agenda.

In contrast, two student perceived this structuring in a negative t effect, mentioning that it made brainstorming sessions harder.

Three students elaborated that when the preparation, the . facilitating and the recording of the meeting were done "according to the procedure, the roles of the facilitator and the minute taker were made easier than in the previous process. More emphasis was put on the preparation of the meeting, which encouraged team members to be familiar with the issues prior to the meeting, and saved time at the beginning of the meeting.

On the other hand, three other students mentioned that the use of itIWEB was more difficult if any of the facilitator or the minute taker does not fill his role. Producing an agenda with little record of the previous meeting, and capturing a meeting

that was not facilitated on an issue basis increased the overhead in the meeting process. We observed that in these cases, the minute taker fell back on taking chronological notes as before and tried to coerce them into the itIWEB syntactical format.

Finally, three students mentioned that the wrap up phase of the meeting and the action item section of the minutes encouraged the team to assign tasks at the meeting and to record the task assignment. They stated that it improved their productivity by assigning responsibility of specific tasks to individual students, and by reminding students of their responsibilities via the team bboard.

4.4. Impact on bboard communication

Out of 9 students who raised the bboard communication issue, 5 students reported that the itIWEB template improved intra-team bboard communication. The structured agendas improved agenda negotiation while structured minutes made it easier for students who missed a meeting to catchup with the decisions the team made in their absence. The 4 other students did not perceive a significant improvement in intra-tcam bboard communication.

Scanning through the agendas and minutes, we observed that meeting minutes were more comprehensive and more readable than in previous courses. In order to confirm these observations, we compared the word size of the meeting documents between Spring'94 and Spring'95, and the number of replies to each agenda and minutes.

To compare sizes, we removed all e-mail header information and all syntactical constructs associated with the use of itIWEB. We observed that the sizes of the agendas and the minutes was significantly larger when itIWEB was used. The relative size difference was larger for agendas than minutes, indicating a better meeting preparation (see Figure 5). A careful reading of messages in both courses revealed that the information content was indeed larger in the agendas and minutes of Spring'95, and that this increase in size was not due to redundances or verbosity.
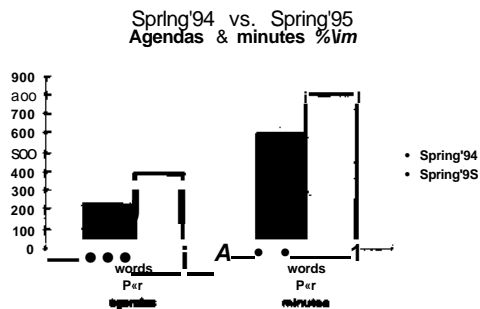

Figure 5 Spring'94 vs. Spring'94: average word size of meeting documents

In an attempt to quantify readability, we counted the number of replies for each agenda and minutes. Figure 6 displays the average number of replies for agendas and minutes for the Spring'94 and Spring'95 classes. We observe that the number of replies for agendas doubled. A closer examination to the replies to the agendas, we realized that the subject of the

replies also changed: in Spring'94, all of the replies to agendas were organizational in content, (e.g students indicating they could not attend the meeting, notices informing of new location of the meeting). In Spring'95, replies also included reaction to the content of the agenda (e.g. update on the status of an action item, additions to the agenda, etc.). Moreover, replies to agendas in Spring'95 often included the relevant excerpt (in the form of an itIWEB action item or issue) the message was responding to. This reinforces our previous observation that meetings were better prepared in Spring'95.
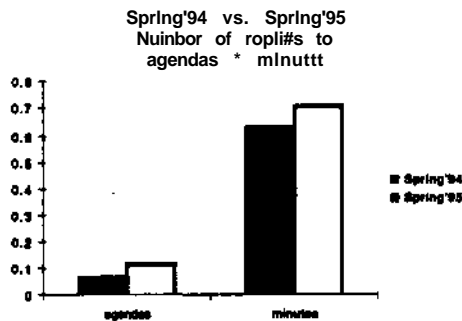

Spring'94 vs. Spring'95
Nuinbor of ropli#s to
agendas * mlnuttt

Figure 6 Spring'94 vs. Spring'95:
average number of replies per meeting documents

We made similar observations for minutes. However, the increase in the number of replies was not as drastic for minutes as for agendas. After analyzing the replies for both semesters, we concluded that the deeper and larger structure of the bboards in Spring'95 discouraged students to follow other teams' bboards, thus decreasing the responses to minutes from other teams[4]. This assumption was confirmed by the Spring'95 survey in which 7 students (out of 7 who raised the issue) admitted not following other teams bboard on a regular basis. Overall, the number of replies to minutes from team members increased significantly.

5. Conclusion & future plans

We found that itIWEB was widely used and accepted, and generally perceived as an improvement. itIWEB also had a impact on the meeting process for those teams which applied the procedures consistently. In these cases, the meeting process was much more structured, better captured and overall more efficient. From the minute taker's perspective, that efficiency sometimes came at the cost of post-processing the minutes. itIWEB also significantly improved intra-team communication.

However, itIWEB did not have as much impact on the inter-team communication as we had hoped. We found that we need to improve the communication infrastructure itself before the inter-team communication improves.

Overall, we found the use of an issue-based model for teaching meeting procedures to be an excellent vehicle. Our plan is to continue the use of itIWEB. For the Fall'95 course (53

---

4. In Spring'94 there was only one bboard per team, whereas in Spring'95, each team had a bboard for minutes, a bboard for discussion and a bboard for announcements.

participants), we replaced the bboard infrastructure with Lotus Notes. We designed database templates in Lotus Notes to not only make it easier for students to conduct and capture meetings, but also to carry discussions about these issues on-line, using the itIWEB model.

6. References

[I] B. Bruegge, J. Blythe, J. Jackson & J. Shufelt, "Object-Oriented System' Modeling with OMT", Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'92), Vancouver, Canada, ACM Press, pp. 359-376, October 1992.

[2] B. Bruegge & R. Coyne: "Model-based software engineering in larger scale project courses", Proceedings of the IFIP Working Conference on Software Engineering Education, Hong Kong, September 1993, Elsevier Science , Publisher.

[3] B. Bruegge & R. Coyne: 'Teaching iterative and collaborative design: lessons and directions", Software Engineering Education, Lecture Notes in Computer Science, J. L. Diaz-Herrcra (Ed.)., Springer Verlag, 7th. SEI Conference on Software Engineering (CSEE'94), San Antonio, Texas, USA, January 1994.

[4] L. Constantino 'Teamwork Paradigms and the Structured Open Team," Proceedings: Embedded Systems Conference, San Francisco (1989).

[5] R. Coyne, A. Dutoit, B. Bruegge & D. Rothenberger 'Teaching More Comprehensive Model-Based Software Engineering: Experience With Objectory's Use Case Approach", In, Proceedings of the 8th Conference on Software Engineering Education (CSEE'95), New Orleans, April 1995.

[6] R. Coyne, A. Dutoit, J. Uzmack & K. O'Tool, "Requirements Analysis for Information Web (IWEB): An Issue-based, Information Modeling and Management Environment for Software Engineering", EDRC Technical Report 05-87-94, Carnegie Mellon University, Pittsburgh, PA, 1994.

[7] T. DeMarco & T. Lister, "Peopleware: Productive Projects and Teams, New York, Dorset House Publishing 1987.

[8] Won Kim, Yongmoo Suh & A.B. Whinston, "An IBIS and Object-Oriented Approach to Scientific Research Data Management", Journal of Systems and Software, vol. 23, no. 2, November 1993, pp. 183-97.

[9] S. Konda et al., "Shared Memory in Design: A Unifying Theme for Research and Practice " Research in Engineering Design, vol. 4 (1992): pp 23-42.

[10] W. Kunz & H. Rittel, "Issues as elements of information systems", Working Paper No. 131, Inst. Of Urban and Regional Development, UC at Berkeley, California, 1980.

[I1] S. Levy et al., "An Overview of the n-dim Environment," Tech. Report, EDRC-05-65-93, Engineering Design Research Center, CMU (1993).

[12] M. Lubars, C. Potts, & C. Richter, "Developing Initial OOA Models", Proceesdings of the International Conference on Software engineering, IEEE CS Press, Los Alamitos, CA, 1993.

[13] D.E. Perry, N.A. Staudenmayer & L.G. Votta, "People, Organizations, and Process Improvement", IEEE Software, July 1994, pp. 36-45.

[14] C. Potts, K. Takahashi & A. Anton, "Inquiry-Based Requirements Analysis", IEEE Software, vol. 10, no. 5, September 1993, pp. 19-28.

[15] G.L. Rein & CA. Ellis, "rIBIS: a real-time group hypertext system", International Journal of Man-Machine Studies, val.34, no. 3, March 1991, pp.349-67.

[16] B. Yakemovic& J.Conklin,"Report on a development project use of an issue-based information system". Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW), Los Angeles, CA, October 7-10,1990, pp. 105-118.