Case Study Report

Executive Summary

1. Research question	Can the theory of Attention Investment predict which source of TaskTracer-related information programmers will choose to consult?
2. Propositions	A. TaskTracer programmers will choose to consult other programmers rather than the online documentation. B. TaskTracer programmers will choose to consult other programmers with whom they are already familiar. C. TaskTracer programmers will choose to consult other programmers with whom prior contact has been positive.
3. Type of case study	Primarily explanatory, but also descriptive.
4a. Units of analysis and units of observation.	Units of Analysis/Observation: TaskTracer developers.
4b. Data collected.	A. Semi-structured interviews with developers. B. Observations of developers' activities. C. Participant observations in meetings with other programmers. D. Information posted in the online documentation. E. E-mails to the TaskTracer mailing list. F. Entries in the JIRA bug-tracking system. G. TaskTracer source code.
5. Design type.	Single-case holistic

Introduction

Communicating knowledge efficiently and effectively is important in any software development team. However, one could argue that such communication is particularly important in the university setting due to the high turnover associated with relying on student employees. This study examines the means by which students programmers working on the research-based TaskTracer software development project choose to acquire knowledge related to the project and asks whether or not those means are, or are meant to be, as efficient as possible.

The theory of Attention Investment proposed by Alan Blackwell [Blackwell 2002] suggests that TaskTracer developers will choose to acquire knowledge in the most efficient means possible in terms of attention units. In other words, with the scarcity of their attentional resources as their main motivating factor, developers will weigh the cost and benefits of each source of information and will tend to choose the source which they *perceive* has the lowest cost/benefit attentional ratio either in the present or in the future.

Though this theory seems valid, anecdotal evidence suggests that it may not fully describe developers' motivations in this case. In reality, developers may choose sources of information based on any number of other factors, including (but not limited to) the personal relationships developers may have with other developers or a general dislike of the online documentation. Along these lines, we expect that despite the motivation described by attention investment, developers will choose to consult (1) other developers over the online documentation, and (2) developers with whom they are familiar and (3) (3) developers with whom prior contact has been positive over those who may be able to more efficiently and/or effectively communicate the desired information.

Design

We will study the 12 student members of the TaskTracer software development team at Oregon State University. The team includes 6 graduate students and 6 undergraduate students. Since we anticipate sufficient similarity across developers with respect to how they choose to acquire knowledge, we have designed a single-case study in which the developer is the unit of analysis (the "case"). This unit of analysis will then be replicated 12 times – one time for each developer.

What follows is our Case Study Database. The database is comprised the procedures for collecting each form of data and the data collection sheets which will be used. Each set of procedures includes a brief description of why we are collecting the form of data described and what the investigator should keep in mind while collecting it. I will also describe briefly here why each form of data is being collected and how it is related to our propositions:

1. Email data. The goal of this data is to confirm which TaskTracer developers are in charge of which aspects of TaskTracer. To illustrate the importance of this data, consider the following scenario:

> Developer A consults developer B concerning the TaskPredictor component of TaskTracer, but developer C is the only one who ever commits code related to TaskPredictor.

In this case, one must wonder, why is developer A consulting developer B and not developer C? Wouldn't attention investment predict that developer C would be the most attentionally efficient source of information? Why or why not?

2. Interviews. These will provide us with the best approximation of the developers reasoning behind whom they consult and when. This will allow us to determine whether or not developers are considering a cost/benefit analysis in terms of attentional units (as attention investment would suggest) when deciding who / what to consult when they are in need of TaskTracer-related knowledge.

3. Participant Observations. These will give us an idea of the team dynamic in TaskTracer and will help us to answer the questions:
   a. Is there anyone team members just don't like talking to?
   b. Is there anything inherently wrong with the online documentation? If so, does what is wrong lead to decreased efficiency in using it? Or is it bothersome in some other respect not related to attention?

   If the answers to any of these are yes, then that may indicate that developers consider factors other than attention when choosing which source of information ton consult.

Case Study Protocol

Table of Contents

Procedures:

Data Collection Sheets:

Schedule

Week 0 – Study Design / Preparation

1. Finalize study design outline and protocol.
2. Make appointments for Interviews. See interview protocol for instructions on requesting appointments with TaskTracer developers.

Week 1 – Data Collection

1. Conduct interviews with TaskTracer developers.

Developer	Date	Time
Twinkle Lettkeman		
Anh Tran		
Mike Goodman		
Wei Wu		
Lida Li		
Jiangiang Shen		
Xinlong Bao		
Frederic Carl		
Stephen Kolibaba		
Ian Oberst		
Brenton Kirschner		

2. Conduct interview with Jed to get his opinion on the best sources of information and why.

3. Collect e-mail data (CVS commit e-mails, JIRA-generated e-mails, other e-mails).

4. Collect participant observation notes.

Week 2 – Data Analysis

1. Coding / Organizing Data.
2. Building explanations and linking them to the data and the main and rival theories.
3. Analyze study validity.

Week 3 – Composing Report

1. Completing the case study report.
2. Preparing a presentation about our case study.

Email Data Collection Procedures

Purpose of Analyzing Emails (*keep this in mind*)

We anticipate that e-mails sent to the TaskTracer mailing list will indicate on which topics developers have expertise. For example, we *assume* that if a developer is the only developer committing code in the TaskTrail project, that developer will be a good person to go to for TaskTrail-related information. Thus, the question we intend to answer using TaskTracer e-mail is:

Who / what does TaskTracer e-mail indicate is the best source of information concerning the tasks currently assigned to TaskTracer developers?

Email Types

Communication Emails

Communication e-mails are those which are generated by TaskTracer developers and not by the JIRA or CVS systems. Most of these e-mails are sent to the cs-tasktracer@engr.orst.edu mailing list, of which each TaskTracer developer is a member. However, some e-mails concerning projects on which Erin is working and not relevant to all TaskTracer developers may be sent directly to Erin. All e-mails will saved for analysis.

Each time a "communication e-mail" is received, information about that e-mail should be recorded in the Communication Email Log (EmailLog.xls) on page 7 of the Case Study Protocol.

JIRA Emails

JIRA e-mails are those sent automatically to cs-tasktracer@engr.orst.edu by the JIRA system when a JIRA issue is created, modified, resolved, or commented upon. We will ignore e-mails sent when JIRA issues are created because JIRAs are too often created by the software's "end-users."

Each time a JIRA e-mail is received, information about that e-mail should be recorded in the JIRA Email Log (JIRALog.xls) on page 8 of the Case Study Protocol.

Code Commit Emails

Code commit emails are sent automatically to cs-tasktracer@engr.orst.edu when a programmer commits TaskTracer source code via the CVS versioning system. It should be safe to assume that programmers committing code related to a given TaskTracer component should have expertise in that component.

Each time a code commit email is received, information about that e-mail should be recorded in the Code Commit Email Log (CommitLog.xls) on page 9 of the Case Study Protocol.

Interview Procedure

What is our fundamental line of inquiry?

Why do TaskTracer developers choose the sources of information they do? Do they choose based on a desire to minimize attentional effort? And/or do they choose based on other factors?

How will interviews be conducted?

10-15 minute interviews will be conducted with each of the 12 TaskTracer developers. During each interview, all three investigators will be present. We will endeavor to have two interviewers taking notes and one interviewer focusing entirely on the interviewee at all times.

Since Erin is most familiar with the interviewees, she will be in charge of the warm up and cool down parts of the interview. Then the other investigators will be free to interject any questions as needed. Note that these are semi-structured interviews. If you have any questions not on the interview data collection sheet, feel free to ask them. However, when asking questions on the fly, be careful that the questions you ask are not leading or somehow offensive to the interviewees.

The interviews will proceed roughly as follows:

1. Introduction
   a. State who you are.
   b. State why you're there. If asked, say "We are trying to understand how knowledge is communicated in the TaskTracer team."
   c. Reassure them that anything they say will be kept confidential.
   d. Ask their permission to conduct the interview. E.g. "Do you mind if we take notes during your interview? Are you ready to begin?"
2. Warm-up – ask easy, non-threatening questions just to get them used to giving your answers and to help them feel comfortable (e.g. "How is your project going? How long have you been working with TaskTracer?")
3. Main Interview
   a. Go from easy to hard questions. The questions in the Interview Data Collection Sheet are laid out in this manner.
   b. Use a logical sequence when asking questions.
4. Cool off – ask easy questions such as "So what is your favorite part about working with TaskTracer?"

5. Closing – Say thank you, inform the interviewee that the interview has ended, etc.

Who will we interview?

See schedule.

When?

See Schedule.

How will we gain access to them?

E-mail the interviewee at least 1 week in advance to the desired interview data. Be flexible in accommodating the schedule of the interviewee. We will send the following e-mail to each of the potential interviewees:

Dear TaskTracer Developer,

We are conducting a brief study on communication in the TaskTracer team, and we would like to interview to learn more about how you communicate with respect to TaskTracer. If you have 10-15 minutes available between May 14 and May 18, 2007 and you are willing to participate in the study, please contact Erin Fitzhenry at fitzheer@eecs.oregonstate.edu. Note that any data collected during the interviews will be kept confidential in the sense that all interviewees will be referred to in the study report by number and not by name.

Thank you very much,

Erin Fitzhenry
Wei Lin
Chandan Sarkar

What should we remember to do / not to do?

1. Avoid long, complex questions, to reduce possibility of misunderstanding, or not getting the answer
2. Avoid leading the interviewee (e.g. “Why do you like wiki so much” would be bad).

3. Be precise in recording the answer. This will be difficult given the brevity of the interviews, but having multiple note-takers during the interview should help us in this.
4. Remember to have at least one investigator focusing solely on the interviewee at all times in order to make him feel more comfortable.

Participant Observation Procedure

Purpose of Participant Observations

Participant observations are meant both to *verify data* obtained from other sources and to *collect data* which cannot be obtained from other sources. With this in mind, pay particular attention to the following during participant observations:

1. The interpersonal relationships within the TaskTracer group.
2. Comments made informally about the knowledge transfer process, particularly by the newest team members, and by one of the oldest, who is currently in the process of handing his project over to other developers before he leaves TaskTracer this month.

How to Conduct Participant Observations

1. Take notes after each TaskTracer-related meeting in the Participant Observation Notes document. Mark the scenario as “meeting.” In these notes, describe:
    a. Those present at the meeting.
    b. The goal of the meeting.
    c. The tone of the meeting.
    d. Whether or not the meeting’s goal was accomplished.
    e. Any information that may pop up about who may know the most about which aspects of TaskTracer.
    f. Anything else that seems interesting.

2. When talking to other TaskTracer team members, or simply when doing your own work, record any information you think is relevant to the study in the Participant Observation Notes log. Pay particular attention to the interpersonal relationships in the team and other aspects of TT development which cannot be ascertained in interviews, etc.

Notes on Particpant Observations

1. Eliminating bias in this process will be extremely difficult. When you believe your notes are biased, make a note of that in the margins, but still include the notes in the case study database, as their seemingly biased ideas may be verified by other data.

2. Since the other investigators cannot fully participate in this process, be diligent about explaining your findings.

Appendix

Data Collection Sheets

Interview Data Collection Sheet

Date: _____________________

Developer: ________________

Investigator: _______________

Questions:

1. (Warm up) What is your name?

2. (Warm up) How long have you been working with TaskTracer?

3. What is your agenda for today in terms of TaskTracer?

4. Do you have any specific components you tend to work on?

5. What have you worked on in the past week?

    a. Was there any information related to that task that you didn’t have when you started?

        How did you acquire that information? Why?

Did you consider consulting the wiki for that information? Why / Why not?

Who do you think knows the most about that aspect of TaskTracer? Did you consult that person? Why / why not?

6. What do you plan on working in next week? (this question should be analyzed skeptically because the above questions make our line of inquiry pretty clear, and the developers may at this point start just telling us what we want to hear).

Do you believe you will need any information that you don't already have?

Will you consider consulting the wiki for that information? Why / Why not?

Who do you think knows the most about that aspect of TaskTracer? Will you consult that person? Why / why not?

7. Do you have any developers you prefer to work with? Why?

8. (Cool off) So what’s your favorite part about working on TaskTracer?

JIRA Emails Log

ID	JIRA ID	From	TT Component	Issue Type	Issue Status

Possible Issue Types
Bug
Improvement
Task

Possible Issues Statuses
Closed
Resolved
Commented
Edited

Communication Emails Log

This log is for e-mails between developers. Please record as much of the following information as possible about each e-mail.
Also, if any information that is not included here seems interesting or surprising, record that as well in the "comments" section.

ID	Date	To:	From:	Subject Line:	Topic:	Who is expert?	Comments

CVS Commit Emails Log

ID	Date	From	Files Committed	Related TT Components

Participant Observation Notes

Date:	Scenario:	Notes: