Studying the Effects of ClearType on Knowledge Worker and Programmer Productivity

May 9, 2003

Submitted to Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Submitted by
Dr. Randolph Bias
School of Information
The University of Texas
Austin TX 78712-1276
Executive Summary

For at least a quarter of a century, psychologists and information scientists have investigated the relative readability of text from computer screens and compared it to the readability of text from paper (see Gould, Alfaro, Finn, Haupt, and Minuto, 1987). This work (summarized in Dillon, 1989) has had as its fundamental goal the identification of the variables that, if manipulated, would allow on-screen text to be as close to hard-copy text in readability.

One recent attempt at maximizing the readability of on-screen text is Microsoft’s ClearType font rendering technology, “an unprecedented innovation in font display technology that dramatically improves font display resolution and marks a genuine breakthrough in screen readability” (Microsoft ClearType web site). HCI researchers from the University of Texas at Austin propose to conduct a follow-on set of evaluations of ClearType. The aim is to identify and quantify the effects of ClearTyped text on computer user performance, satisfaction, and productivity. This follow-on work is intended to take the recent findings of positive effects of ClearType on on-screen readers’ preference and performance on short tasks and extend them into the area of computer programmer productivity. The planned work will be conducted over a 13-month period by a research team consisting of research specialists and Ph.D. students led by Associate Professor Randolph Bias. The series of studies will involve a combination of formal, lab-based tests and evaluations conducted naturally in the office environment. The target users for this research are computer programmers.

Results will be reported directly to Microsoft in a series of quarterly progress reports, with a white paper on the results being delivered at the end of the project. The results will also be disseminated to the wider HCI and usability community through refereed journals and conferences.
Context

In the 2002-2003 study funded by Microsoft Research and conducted by this team, “Studying the user response to ClearType,” ClearType was found to be preferred by approximately 80% of test participants, across a variety of fonts and type sizes, and lengths of test stimuli up to a few paragraphs. In a second study covered by that first contract we are studying the effects of ClearType on user performance in simple office tasks. (Data available by June, 2003.) Microsoft has contracted with other research labs to identify the low-level effects on the psychology of reading of using ClearType. The explicit goal of this study is to extend this rudimentary reading research into a more naturalistic, though still controlled setting, and to measure the effects of ClearType on productivity.

Statement of Work

We propose to conduct a series of user-based evaluations of ClearType font rendering technology for a range of specified users performing a range of programming tasks. This research will emphasize evaluations of real-world tasks in both controlled (lab) and naturalistic (field) environments. The aim of the work is to provide statistically reliable behavioral and affective measures of user productivity and preference for CT-enhanced or control screens.

In all cases, ClearType will be paired with a font to be provided by Microsoft (perhaps Andale) that has been optimized for ClearType use. Whenever the phrases “ClearTyped text” or “ClearType enabled” appear below, the reader is to assume that the text is in this font. This will be compared with nonClearType text in the default font of the application being used at the time.

Methodological and theoretical orientation

As stated above, the key objective of this line of study is the quantification of the effects of CT on productivity. In The Magic of Reading, Hill discusses the “flow experience” of reading and programming:

“On a Roll”

Another researcher’s perspective on the flow experience appears in the paper A Theory of Productivity in the Creative Process (Brady, 1986), which examined how computer
programmers achieve the state of maximum efficiency and creativity we call “being on a roll.” The key to achieving the “roll state” is that concentration is not broken by distractions. “Interruptions from outside the flow of the problem at hand are particularly damaging… because of their unexpected nature.” This data on the flow experience will resonate when we come to consider the psychology and physiology of reading and the typographic analysis in subsequent sections. (p. 59)

Does CT’d text permit programmers to get “on a roll,” or to stay on one longer? Might CT’d text allow people to stay at their tasks longer, more happily? Might reading CT’d text be closer to reading hard-copy print, than standard on-screen text, and thereby allow users to extract information from on-screen sources more readily, and perhaps reduce the expenditure of time and other resources required by printing documents? These are questions we intend to answer.

Usability measures

We will select a range of programming tasks to examine, including both creating and debugging computer programs. These will be checked with the development team at Microsoft to ensure relevance. Once a set of tasks has been agreed on, we will proceed to engage in criteria derivation to establish the form of measures we will take. To this end, we will exploit the ISO 9241 part 11 definition of usability, which places emphasis on three broad criteria:

- Effectiveness: the extent to which a task is completed or goal is attained
- Efficiency: the resources used (cognitive and temporal) to attain effectiveness
- Satisfaction: the affective reactions of users to their experience on task

To this usability approach we will add attention to productivity (a well-accepted secondary effect of good usability). Good usability yields real, tangible benefits for organizations (see Bias and Mayhew, 1994), with increased productivity being a key result. Research data will, depending on the precise experiment, include such measures as:

Behavioral (performance) measures:
- task completion,
- time on task,
- number of steps,
- measures of productivity (e.g., lines of code, function points),
• decision to cease participation in the study (because the test participant insists on working full-time with or without ClearType).

Affective (satisfaction) measures:
• ratings of preference,
• estimates of performance
• quality of user experience

We will couple these usability indices with data derived from more qualitative techniques such as:
• Verbal protocols
• Diary records
• Interviews
• Direct observation

The approach we will adopt is to seek, wherever possible, several measures of an important dependent variable (such as productivity) so that we can gain the most accurate estimates of effects.

**Experimental form**

We propose to use two distinct forms of experiment: lab and field.

A lab-based study will enable us to control and manipulate key variables in a manner that yields rigorous performance data for statistical analysis. Typical data collection will involve capturing and analyzing individual user interactions that can be recorded and time-stamped. However, in this series of studies we are trying to simulate more closely real-world computer use, and so even the lab study (Experiment 1) will have a naturalistic flavor to it.

Field-based data collection will aim to provide us with naturally occurring data from a relevant group or mix of users who will be exposed to the ClearType font rendering technology over a period of at least weeks. Data will be collected via logging tools, interviews, diary-keeping, self-report, surveys, and professional analysis of computer programs.
Research Objectives

The aim of the work is to develop detailed behavioral (performance) and affective measures of user responses to ClearType that will support the derivation of quantitative estimates of ClearType's impact on user productivity in programming tasks.

Experimental details

We propose to conduct the following two experiments. It is important to note that the exact details of these studies will change as pilot tests are conducted and it is anticipated that discussion of the experimental design with the ClearType team will also lead to some modifications.

Lab study

Study 1: Programmer productivity

Goal: to measure programmer productivity differences between ClearType enhanced and standard displays. Post-task interviews and surveys will additionally measure programmers’ perceptions of the differences.

Method

Ten users (programmers), drawn from an appropriate sample and hired for four days at a rate of $150 per day, will be run independently in a lab environment. Users will be presented with a different programming task each day. Each programming task will be APPROXIMATELY one day’s worth of work. (Clearly, there is wide variance in programmers’ ability and speed. We will endeavor to identify four programs to code that will average a day’s worth of work for an average programmer. If some test participants finish early and some don’t quite finish in a day, this will not materially affect the results.) The users will have ClearType turned on for two of their projects, and turned off for the other two, interleaved. Half the users will begin on a “ClearType” day (Days 1 and 3) and half will begin on a nonClearType day (thus, ClearType text will be on Days 2 and 4. Users will be required to code (or debug) each at their own pace. The specific programming tasks will be offered to half the subjects as a CT task, and the other half of the subjects as a non-CT task. There will three measures of programmer productivity, lines of code, bugs per lines of code, and the number of function points.
“Lines of code” is an old, and largely outmoded measure of programmer productivity. Indeed, it is a poor measure with which to compare two programmers for job performance reasons (or compare one programmer across two different points in time). Perhaps obviously, a programmer could pad code with meaningless, or even detrimental (and hard to test) lines of code. However, for the purposes of this test, lines of code is a fine measure. However efficient a programmer may or may not be, if ClearType makes interacting with text easier, and tends to keep a programmer “in the zone,” then he/she should generate more lines of code (including even comment lines) in the same period of time.

Bugs per lines of code is an improved measure, taking into account quality as well as quantity of the code.

Function points are a metric for measuring developer productivity that is an improvement on lines of code. Instead of counting the number of lines written, we count the number of input/output calls. The number of function points should be the same if a programmer writes inefficient code or slim, elegant code. Analysis of function points will require the help of a professional programmer who will evaluate blindly (as to experimental condition) the output of each programming test session.

Three other dependent variables will be collected in Study 1:
1 – Cognitive effort measures will be taken from the TAM scales (Davis et al. 1989) or NASA’s Task Load Index (TLX), two reliable and valid measurement tools, or an equivalent. In previous studies (Dillon and Morris 1999) TAM’s Perceived Ease of Use and TLX’s Effort measures correlated negatively at -.049. (High ease of use relates to low cognitive effort, as expected).
2 – A non-invasive behavioral measure of visual fatigue (e.g., sensitivity contrast function test – see Priez et al., 1998).
3 – Satisfaction data, gathered via a daily questionnaire.

At the end of the four-day session users will be shown two screens, demonstrating the effect of the CT enhancement and asked to comment on perceived differences, state their preference, and to rate their confidence in their answers to the post-task questions.

To summarize:

Behavioral measures taken:
- Lines of code
• Bugs per thousand lines of code
• Function points
• Time on task
• Success rate
• Visual fatigue

Affective measures taken:
• Confidence level in answer
• Preference for display
• Perceived differences between screens

Informal hypotheses

We anticipate that day-long programming tasks undertaken with ClearType turned on will lead to quicker, more efficient coding, less visual fatigue, and increased preference, as compared to tasks undertaken without ClearType.

Field study

Studying users in the field yields far more natural response patterns than lab-based studies but such work brings with it its own set of problems. We propose to initiate a field-based study to complement the findings from the experiment described above and from our 2002-2003 study.

The first field study (Study 2) will involve more naturalistic study of programmers.

Study 2 – Naturalistic study of programmers

The aim of this study is to determine how well the controlled lab findings on programmers generalize to real-world use.

Method

Complementing the lab test of programmers, this field study will entail asking groups of University of Texas at Austin programmers to conduct their work for a period of time (two weeks to one month) with ClearType enabled, and for an equal period of time with ClearType disabled. We will counterbalance for order of condition (CT vs. nonCT). We will ask programmers to maintain a
diary of their computer use, and we will periodically interview all test participants and present them questionnaires. Users will serve as their own controls for ClearType vs. nonClearType assessment (i.e., we will use a within-subjects design).

Measures and indices that will be captured naturalistically will be:

- Perceived benefits
- Frequency of print out
- Ratings of productivity
- Willingness to read lengthy text on screen
- Records of reading on screen (where possible)
- Visual fatigue
- Reactions to having ClearType removed
- Point in time when/if the programmer refuses to continue in the test (and demands CT be turned on or off, permanently)

Measures and indices that may be captured through set tasks:

- Speed on task
- Accuracy of task completion
- Comprehension or memorability of material
- Length or duration of reading on screen
- Perceived cognitive effort
- Perceived quality of code

Informal hypotheses

We anticipate that day-to-day programming tasks undertaken with ClearType turned on will lead to quicker, more efficient coding, less visual fatigue, and increased preference, as compared to tasks undertaken without ClearType.

Proposed Schedule &Timeline

Month 1. Establish test set-up
Gain Human Subjects approval at UT
Establish suitable field site(s) and group(s)
Recruit test participants for lab test
Identify programming assignments
Identify professional programmer to assess function points
Agree (with Microsoft and UT-Austin CFO) about exact measures we’ll collect

**Months 2-3 Initiate Data collection**
Conduct experimental study
Analyze results and submit progress report to ClearType team
Kick-off two naturalistic studies

**Months 3 – 6, further experiments**
Conduct field study (Study 2)
Write up Study 1 for Microsoft

**Months 7 – 9, data analysis**
Analyze field data for Study 2
Produce data report
Make presentation to Microsoft team
Make presentation (and recommendation) to University of Texas teams

**Months 10 - 13, research report**
Generate journal article submissions

**Microsoft Technologies used in work**

**Software**
- Use of Office instrumented version
- IE Hack – CT with sub pixel positioning

**Conclusion**

We propose to conduct a series of lab-based and field investigation of computer programmer performance and affect, with the MS ClearType Font rendering technology. These studies will be
conducted on computer programmers performing a variety of programming tasks. The results are intended to provide scientifically robust data on the impact on user productivity of ClearType.

**Personnel**

Project Lead: Dr. Randolph Bias  
Ph.D.-level Research Assistants: Ph.D. students to be named later.

**Deliverables and Ownership of Results**

Microsoft would allow Dr. Bias and his team to publish any and all findings from this research project in the academic, peer-reviewed literature if they so choose. Microsoft would not make any proprietary claims on the results of the research. The data and the results will be owned by the University of Texas and freely usable by Microsoft.
BUDGET

Personnel:
Lead researcher (1 FT summer salary, $21,667 plus indirect costs) $32,501
1 PhD student, 20 hours per week:
    stipend ($15000 plus indirect cost) and tuition $5283 $27,783
Programmer to judge function points ($4500 plus indirect costs) $6750
Ten programmers for 4 days at $150 per day (plus indirect costs) $9000
Fringe benefits: lead researcher (faculty): $8450
Fringe benefits: Ph.D. student $6750

Travel
$7500
Visits to Redmond to report results 1 person x 1 visit.
Two people to attend conferences to present results.

Materials and Supplies $1200
    Disks, paper, copying costs, mail shots (plus indirect costs)

Total project costs $99,934
References


Signature of Project Director:

Associate Professor Randolph Bias