8-15-1997

The Process of Software Maintenance: A Comparison of OOP and 3GL

Michael A. Eierman
University of Wisconsin -Oshkosh

Mark T. Dishaw
University of Wisconsin -Oshkosh

The Process of Software Maintenance: A Comparison of OOP and 3GL

Michael A. Eierman
Mark T. Dishaw
College of Business Administration
University of Wisconsin - Oshkosh
800 Algoma Blvd.
Oshkosh, WI 54901

Abstract

This paper describes an on-going field study to examine the differences and similarities of software maintenance in the Object Oriented and conventional 3GL programming environments. We expect our results will show that the maintenance process will differ between environments in that programmers working in Object Oriented environments may need to spend more time understanding and coordinating their changes and less time making the changes.

Introduction

Software maintenance is the last and longest phase of the software life cycle. Maintenance is the process of changing existing software to add functionality to the program, enhance existing functions, or fix errors (bugs). Software that is newly developed in several years may be in production for ten's of years and may undergo hundreds of changes to meet changing business requirements. Maintenance is the primary function of many MIS organizations. An average of 70% of software budgets are devoted to maintenance (Swanson & Beath, 1989).

Object-oriented programming (OOP) languages have been in used in research and, to a limited extent, in practice for approximately fifteen years. However, it was not until recently that OOP received considerable attention from the practitioner community. The object-oriented paradigm has been hailed because of its natural relationship to the real world, ease and speed of development, and ability to reuse objects in different programs. These capabilities are thought to produce major productivity gains over programming in 3GLs. However, given the importance of maintenance in the MIS organization, the decision to develop systems using OOP must include evaluation of the relative effectiveness of maintenance in that environment. Unfortunately, there is little research that investigates the maintainability of object-oriented languages.

The objective of this study is to examine the maintenance of systems developed in objected-oriented programming languages. This examination will compare the process used in object-oriented languages versus that used in 3gls. This comparison is based on the assumption that a difference in the process is the basis for a difference in productivity or effectiveness of maintenance in the different environments.

This paper is organized in the following manner. The next section will present the research model used in the study. The final section presents the research method used in the study and progress to date.

Research Model

The research model for the maintenance process is developed based on Vessey's Software Maintenance Activity Model (Vessey,1986), the CASE Tool Function Model (FCTM) developed by Henderson and Cooprider (1990), and the program understanding literature. Basic software maintenance tasks in the model include Understanding, Transformation, and Coordination.

Program Understanding

Program understanding can be viewed as a problem of representation development, manipulation, and testing. At the heart of the process of understanding a program is building a representation of the problem to be solved. A problem representation is an mental model or conception of the problem (Pennington, 1990). The development of a program representation appears to be an essential part of the software understanding process.

Letovsky and Soloway (1986) view program understanding as the process of recognizing plans or intentions of the code. This process is more difficult when the plans are delocalized or spread over the module, or even between modules.

Vessey (1986) divides maintenance debugging activities into three categories. These are planning activities, knowledge building activities, and "bug-related" activities. Planning activities center around the management of the task and the coordination of other activities towards the completion of the task.. Knowledge building activities center around the procedures involving the acquisition and management of information needed to accomplish the goal at hand. Knowledge retrieval is the accessing of technical information regarding the language of the program or related system software from memory or from other external sources. "Bug-Related" activities in the Vessey maintenance activity model center around three types of actions, clue generation and management, hypothesis generation and evaluation, and error management.

The first two categories of "bug related" activity are the essential elements of diagnostic problem solving (Araki, 1991; Pennington, 1990). We separated Vessey's "bug-related" activity into two sub-activities and renamed them Diagnosis and Treatment.

The final category in this group is error management, during which the actual error in the program is "repaired". This is more appropriately considered a "Transformation" activity.

The understanding task, as described above is composed of three sub-tasks or activities: planning, knowledge building, and bug-related (diagnostic) activities. The key, as noted above, to understanding is the development of an appropriate representation and its manipulation.

Program Transformation

Program understanding is not the entire story of software maintenance support. The programmer must be able to actually change software and document the effects of that change. The change cycle includes making a change in a source module, compiling the module, and testing of the changed program.

Program modification may be performed during the understanding process. In this process the programmer arrives at a point where he or she is about to test an assumption about the software to be changed. The task being performed in this process is diagnosis. In these activities a hypothesis is tested and confirmed or rejected.

Coordination

In most MIS organizations the programmer must initiate a production release process which may include documentation updates and testing for standards adherence. The later process corresponds roughly to the coordination function in the FCTM. Project management software which facilitates coordination of maintenance activities involving dependencies with other activities is also included in this category. These activities include control and cooperation.

Control focuses on creating and enforcing rules and standards during the development or maintenance process. Programmers are frequently required to submit programs about to be released for production for analysis to determine compliance with "shop" standards. Program or project managers use project management tools to communicate tasks and deadlines to their staff of programmers and analysts.

Cooperation technology allows users to exchange information with other individuals (Henderson & Cooprider, 1990). Programmers frequently use e-mail and other coordination devices in an effort to keep their work consistent with other projects or activities which are occurring simultaneously. The most significant of these tools are library and configuration management tools. While many maintenance tasks or projects are relatively short in time duration and may be undertaken by a programmer or analyst working alone, no maintenance project is undertaken in isolation from other software or programmers. Although the coordination task is a normally a small part of the maintenance task, such activities ultimately have a very significant impact on the success of the maintenance project.

In summary, maintenance consists of: (1) Understanding, which includes Knowledge Building, Planning, and Bug-Related activities, (2) Transformation, and (3) Coordination, which includes Control and Cooperative activities.

Research Method

This research tests several hypotheses about the nature of maintenance tasks in object-oriented programming versus the nature of maintenance in 3GL's. Smalltalk represents the object-oriented paradigm in this research while Cobol represents the 3GL paradigm.

This research project will employ the field study as its principal method. The subjects for the study are working programmer analysts performing maintenance tasks. The projects to be included in the study will be selected from the existing maintenance backlog.

The data for this study will come from two different sources. Data on the nature of the maintenance task in the 3GL environment has been collected in a previous study (Dishaw, 1994). This data was collected from three Fortune 50 organizations. All have large MIS applications groups who expend a large proportion of their annual budgets on software maintenance. All three organizations use COBOL as their 3GL programming language and all are responsible for maintaining millions of lines of code.

The second source of data is from local organizations that use Smalltalk as an object-oriented programming environment. This data will be collected for the proposed study via questionnaire from organizations involved in the maintenance of systems developed in Smalltalk.

The basic unit of analysis for this study is the individual maintenance project. The definition used for "project" may vary slightly from that used by the subject organization. Project, in the context of this study, is taken to mean a change to an existing function, or group of related functions that can be accomplished by a single programmer analyst.

Research Questions

The literature concerning the nature of maintenance was primarily developed based on research of maintenance in 3GL environments. The questions generated in this section primarily concern a comparison of the maintenance task in the 3GL environment versus the nature of the task in the object-oriented environment. The primary goal is to determine if the maintenance involves the same, or different activities in the object-oriented environment.

The research model consists of six types of activities divided in three classes. The first class is Understanding, consisting of Bug-Related Activities, Knowledge Building, and Planning activities. The second class is Transformation and the final class is Coordination, consisting of Control and Cooperation activities. The questionnaire used in the study, developed and tested in Dishaw (1994), demonstrated an adequate evidence for discriminant and convergent validity in identifying these activities.

The primary proposition concerns the difference in maintenance activities between maintenance in an object-oriented environment and in a 3GL environment.

Proposition 1: The proportion (relative frequency) of maintenance sub-activities is the same for maintenance in OO and 3GL environments.

In either environment, the programmer must understand, plan, and coordinate their changes to the program with other programmers before they make actual changes in the code. However, as noted in the discussion of the maintenance process, programs with

"de-localized plans" are more difficult to understand for programmers (Letovsky & Soloway, 1986). Also, complex systems with many modules and links to other systems invite errors of omission and incomplete understanding (Letovsky & Soloway, 1986).

The object-oriented environment embodies de-localization and interaction with other systems because programming is the linking of independent objects that perform different functions. This suggests that programmers in an OO environment may need to spend more time understanding and coordinating their changes before actual enhancements or bug-fixes can be undertaken. On the other hand, the OO environment is described as being easy to understand because of its natural relationship to the real-world. This may result in less frequent need to undertake planning and knowledge building activities. Propositions 1a and 1b examine this question.

Proposition 1a: The proportion (relative frequency) of Understanding activities is more for maintenance in the OO environment than the 3GL environment.

Proposition 1b: The proportion (relative frequency) of Transformation activities is less for maintenance in the OO environment than the 3GL environment.

Proposition 1c: The proportion (relative frequency) of Coordination activities is more for maintenance in the OO environment than the 3GL environment.

We will test for differences in the two populations (OO and 3GL) using MANOVA. A longer version of this paper with references is available from the first author.