Vern ppl

TO: Administrative Distribution

FROM: David M. Jordan

DATE: January 22, 1973

SUBJECT: Analysis of Multics Assurance Practices

Introduction

The recent installation on Multics of the new directory control system resulted in a large number of problems. In this case, the problems centered on two major areas: the user interface and system performance. Unfortunately, the new directory control installation is not an unusual case but is, instead, an obvious example of the general situation of system installations. In an effort to analyze the problems involved and to understand the complexities of their solutions, this report presents a post-mortem of the new directory control development and installation, a group of generalized conclusions, and a brief series of recommendations.

New Directory Control

In order to demonstrate the wide range of systems assurance problems associated with the development and installation of a large system change, I have broken the new directory control change into several separate phases. I have attempted to demonstrate several problems with each phase, but two things should be noted: first, thtat the list of problems is not

intended to be complete, but merely representative; and second, that many of the problems mentioned have always been problems and thus have been unsolved in previous installations as well. Also, I want to stress that, by current standards, the new directory control was one of the smoothest installations, in view of the magnitude, that has been performed since I've been here.

Design

During the design phase of the new directory control, little effort was spent updating the specifications of each module to be modified. This resulted in several problems later, most notably that an audit of the new code was essentially impossible because the specifications were not available. This also contributed to a second failure during the design phase: no major effort was spent searching out unexpected implications of the design. This, in turn, resulted in such incidents as an on-the-fly redesign of the status_ primitives because of problems discovered after installation.

In addition, there were insufficient brainstorming type sessions in which the designers, the implementers, and any other interested parties could get together and go over the proposed changes module by module. This led to a certain lack of understanding on all sides as to what everyone else thought was really going on.

Another problem which should have been solved during the design phase was that of finding the non-access control

procedures (especially in the online libraries) which needed to be changed. It is unfortunate that some required changes were not made until it was discovered that the procedures involved either didn't work right or just didn't work at all.

Implementation

The implementation phase suffered from one major problem: a lack of man-power. Two very overworked people did all of the programming for this 80-90 module modification and, although they did an outstanding job, they were forced by the magnitude of the change to eliminate certain important steps. One major step which was avoided was a complete audit of the new code. Part of the reason this audit was not performed was the lack of module specifications, but the importance of code audits, even without proper specifications, can be seen in the changes made to the Salvager for the new directory control. These changes were more closely supervised and the code was, to some degree, audited before installation. The result has been no serious Salvager problems.

Another major problem encountered by the implementation effort involved the distinctions (or lack of distinction) between the design, implementation, and checkout phases. In the case of the new directory control, the design effort was completely separate from the implementation effort and, in fact, the designers were away during most of the implementation. This resulted in solutions to problems being without any effort to

re-evaluate the design. On the other side, the implementation and checkout phases were done almost completely by the same overworked people. In addition to making the job that much larger, this meant that some problems were never discovered because they were overlooked by the same people in both the implementation and checkout phases.

Testing

The testing phase of the new directory control suffered from three major problems. The most basic of these was the pressure put on to get the system installed. This resulted in the system eventually being installed without a complete and sucessful test with a large number of users. A second major failure was that no effort was made to allow subsystem users to check their systems before installation. Several problems (such as the problems with status_) would have been found and fixed before installation had subsystem users been given this opportunity. The third major failure was that no serious analysis of performance changes was done. This resulted in several surprising performance degradations that were only discovered after installation.

Documentation

The original user documentation of the new directory control (MCB 969) provided a reasonably good overview of the changes but, unfortunately, did not provide the detailed, step by step information needed by users to prepare for the new system. Such

statements as "This change will not affect most users.", while undoubtably true, are insufficient and misleading and resulted in several subsystems ceasing to work after the installation. In addition, the user documentation was never modified as changes is the design or implementation were discovered to be needed.

Also, the changes required to the MPM and SPS documentation have not, as yet, been completed. This problem seems to be with us all the time and is the more troublesome due to the amount of time required to get the changes published once they have been made.

Follow-up

After the new directory control was installed and the problems which led to repeated system crashes had been fixed, very little programming support was available for finding and fixing the 'non-fatal' problems which 'only' resulted in user programs and subsystems not working. The development priorities, with the follow-on development considered to be most important, meant that instead of 5 programmers working as a task force for four or five days to find and fix problems there were 1 1/2 - 2 programmers working for over a month. The cost of this course of action in terms of user aggravation and ill-will is unknown, but must be extremely high. In addition, there existed no individual or group whose sole responsibility was to provide aid to users during this period. This resulted in unnecessary delays and problems as users first tried to find some one to talk to and

then found themselves shuffled from one person to another until someone was willing and available to help them.

Conclusions

In looking at the problems and types of problems encountered by the new directory control installation, several basic deficiencies in the area of Multics system development can be seen:

1) The overriding importance of maintaining the operation of a service system for the benefit of users is often ignored.

2) Errors in the estimates of resources required to implement changes are made repeatedly.

3) The importance of complete design and user documentation is often overlooked.

4) The absense of a well defined development organization structure contributes to delay, inefficiency, and lack of useful communication.

5) The lack of sufficient user impact evaluation, code auditing, and testing has predictable results.

It appears clear that the solution to these problems is dependent on a strong commitment from each of the organizations involved in the development and use of Multics. This commitment may be measured in man-power, machine time, or money, but, if the

kinds of problems encountered in the new directory control are to be avoided in the future, the commitment must be made.

Recommendations

First, a more or less complete re-evaluation of the structure and responsibilities of the various development and maintenance groups involved with Multics must be undertaken.

In order to begin to solve the problems involved, I expect that the following kinds of commitments are going to be required:

1) An enlargement of the Honeywell design and development staff, not for the purpose of producing more system changes, but to provide much more redundency in the design, implementation, and checkout phases of system changes.

2) An enlargement of the Programming Development Office Multics assurance staff in order to convert this staff from its current limited functions of installing system changes and analyzing system crashes to the function of assuring the continuity and stability of the system.

3) The creation and training of a user services group within the Information Processing Center to act as the primary source of aid and assistance to users.

4) The institution of a regularly scheduled system testing period on the service system, preferably during normal working hours, during which users would be allowed to run without charge to

assure that a proposed new system has no unexpected problems.

As I stated previously, these are the types of commitments required. The actual details of these commitments and the method of arriving at these goals will have to be worked out as a part of a major re-evaluation, but it is crystal clear that all groups involved must be prepared to make significant increases in their current monetary commitments if the kinds of problems experienced during the installation of the new directory control are to be avoided in the future without completely paralyzing the development effort.