SOFTWARE MAINTENANCE ASSIGNMENT

- ASHISH KUMAR
- 2K18/SE/041

Ques: Do the following -

1. Briefly explain SM challenges in today's era.
2. Mention technology support to aid SM.

Note: Ma'am I searched a lot about technology support, but not able to find quite correct answer for that. So, I am writing what I have found on internet.

Ans: Software maintenance is one of the major concerns of software development. Good maintenance process is very essential to maintain the quality of software. Most problems that are associated with software maintenance can be traced to deficiencies of the software development process.

There are several problems encountered while maintaining software:

A. Measurement Issues

Surveyed systems of very similar sizes showed very different patterns of maintenance activity, with the surveys revealing the roles of the user and manager in maintenance activities and suggesting that measurement of the software be external, as well as internal, to the operation of the system. To explore sources of potential change, we consider the environment of an application system. The environment consists of four factors, each of which can affect a system.

i) Technological Change

Technological change can affect applications by making possible new options. Distributed data processing or new, more intelligent terminals can split an application system across multiple computer systems. Other technology may make it possible to join or tie together separate applications.

ii) Managerial Factors

Management frequently exerts pressure to control costs and modify schedules. This pressure can directly impact the quality and quantity of the maintenance effort. In particular, this pressure often causes the documentation of changes to be done insufficiently or not at all. Because managerial pressure focuses on the short term, it excludes attention to fundamental application system rework using new techniques.

Who wants to expend the effort to rework something that works? This exclusion, in turn, prevents the use of productivity aids. Systems grow in size and complexity as enhancements accumulate, and, not unexpectedly, the surveys reveal that systems become more complex and difficult to maintain as they age. Contributing to the difficulty is the fact that the original staffs, who know the application, leave the organization.

iii) User-External Environment

The user-external environment includes legislation, competitive pressures, and social and cultural factors. It also includes the user's organization and its staffing. To explore the user-external environment, quantitative factors can be gathered. For instance, the requests for change can be classified by their ultimate source, the number of users actively working with the system can be counted, and user-developed systems can be assessed.

iv) Marketplace

Just as technology produces new techniques and tools, the marketplace produces new products and services. The marketplace also creates competition for personnel, exerting pressure on the maintenance staff to leave their current jobs. Another effect is that new products and services often spur users to request more enhancements.

B. Organizational Issues

Traditionally, general interest has centered on the organization of the maintenance function within a systems group. But given the rising interest in and the impact of the user community on maintenance issues,

The role of users is a major issue for systems groups in general. Users may play a role in filling the gap between supply and demand. Because of this growing trend, the user's role in maintenance, enhancement, and operations needs to be more precisely assessed. A separate, but related, organizational issue is that of system controls. Our surveys in the commercial sector reveal that many controls supported in education and theory are not used in practice, usually because of a trade-off between benefits of the controls and the cost of their implementation. Also, many organizations lack the technical implementation aids to make such controls feasible.

C. Productivity issues

A main research focus has been the productivity of programmers and, to a lesser extent, analysts in the system organization. Of the variety of techniques for increasing productivity, the surveys reveal only limited use. Furthermore, even in cases in which such techniques are employed, the results are not significantly different from those cases in which only traditional methods are used. It should be emphasized that the surveys provided no verification or control of the techniques among the respondents. Thus, if a productivity technique can be found to facilitate users' functions, its effect will be multiplied far more than a technique for programmers. It may be easier to increase the productivity of users with less complex tools, than to increase the productivity of programmers with more complex tasks.

A second area for which productivity tools have been developed is that of the analysis and design stages of system development and enhancement. These tools are designed to improve design correctness and completeness, since by explicitly specify in the requirements, the system will theoretically be easier to maintain and be more compatible with user needs. Today the situation has changed. Management pressures users to automate in order to control organization costs. Requirements no longer remain static. In many areas there are substantial yearly changes that necessitate major enhancements and retrofitting.

D. Conceptual Issues

In the surveys, we defined maintenance to include enhancements and operational support, as well as routine debugging and problem identification and resolution. More specifically, the questionnaires in the larger commercial survey and the naval survey included emergency fixes, routine debugging, accommodations of changes in file and data input, accommodation to hardware and software change, enhancements, documentation improvement, and recoding for efficiency. In all three studies the research has shown that enhancements for users are the major activity (50-70 percent of all maintenance and enhancement). Adaptation to new technology as a major activity surfaced only in the weapons systems survey. Emergency fixes and recoding for efficiency utilized relatively few resources (less than 20 percent) as did documentation (less than 5 percent).

Maintenance is related to the continued development of the application system. For many systems there appears to be no single linear life process, but rather a repeating life cycle. The present-day and one-year-previous data indicated that once development was completed and the system stabilized in operational use, programmers began making enhancements individually or in groups of changes. The data indicated that while total maintenance in the organizations is at about the same level as development, maintenance on particular systems first declined as the initial operational errors were fixed, and then increased as users requested enhancements. However, the survey data were not sufficient to fully support the hypothesis that, as users request new enhancements, a new developmental cycle is begun.

The conceptual framework of maintenance needs a complete classification of the tasks and work done under the umbrella term maintenance. For example, a classification method could be used to assist designers of project control systems. Even the rudimentary classification of perfective, adaptive, and corrective maintenance is already being used in a number of organizations, including system groups who want to charge their costs back to the user organization. The classification method has proved beneficial in estimating costs by task and by type of system.

These are the some **tools (or softwares**) that can be used for continuous monitoring of software for software maintenance:

1. Zabbix

Strengths:

- Collecting numerous metrics including application metrics, such as availability, response time, storage space utilization, memory utilization.
- Diverse visualization options (e.g., dashboards, graphs, maps) for displaying collected metrics.
- Monitoring multiple servers from any location.

2. Nagios

Strengths:

- Collecting diverse software metrics (e.g., memory used, data read/written, events over time).S
- Simple integration with third-party software (e.g., incident management software, databases due to multiple APIs.
- Configuration wizards that allow users to add new devices, services and applications.

3. Datadog

Strengths:

- Capabilities for collaboration on software monitoring and analytics.
- Customizable dashboards showing various metrics like application uptime, total web traffic, and more.
- Easy-to-use tool for records and logs search.

References:

1. Research Gate paper: https://www.researchgate.net/publication/220566142_Issues_in_Software_Maintenance
2. International Journal of Computer Science Engineering (IJCSE) Paper: http://www.ijcse.net/docs/IJCSE15-04-01-037.pdf
3. https://www.scnsoft.com/services/software-development/software-support-and-maintenance