Iqra National University Peshawar Pakistan

Department of Computer Science

Spring Semester, Assignment, May, June 2020

Subject:	**TSE**	**Issue Date:**	**15/May/2020**
Program:	**MS (CS)**	Submission Date:	**10/June/2020**
Teacher Name:	**Dr. Fazal-e-Malik**		
Student Name	**Rooh Ullah Jan**	ID	**6611**

What is software maintenance?

Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software application after delivery to correct faults and to improve performance.

Summary of case study on software maintenance:

To design, implement, and operate a successful software development process, exposure to similar existing systems is invaluable. The objective of this paper is thus to document and analyze an existing, moderate size, software maintenance project

In summary, the following were found to be the strengths and weaknesses of the project maintenance process, as compared to the practices defined in the

Capability Maturity Model.

Strengths:

- The process of assuring correctness and completeness of baselines is effective.
- Change control is well defined and managed through an effective CR process.
- The SCM function is separate and independent of the development function.

Weaknesses:

- Lack of documentation (procedures, plans, etc.).

- Lack of formal training.

- Lack of tracking (e.g., on cost and schedules).

- Lack of independent audits. Most of the above weaknesses do not deal with the day-to-day process issues as modeled, but rather with issues which affect the longer term operation. For example, if there were a major movement of personnel out of the project, the lack of training and documented procedures would become a more vital issue.

Summary of case study on software maintenance:

It is apparent from case histories of the mini computer upsurge that the firm purchasing a micro computer can no longer afford to ignore the costs and responsibilities of software maintenance. Hard cash must be allocated to the repetitive needs of backup and program maintenance. What is the cost of failing to do this? Perhaps that Is the best way to pose the question. For if your business depends on your computer, it doesn't matter how small the computer Is or what the dollar figures in the receivables are. You will still not be able to get at them if you have no way to restore your system from a backup. Is it really "less costly" for a 10 employee firm to go bankrupt? Certainly not to those ten peoples.

Summary of case study on software maintenance:

One major finding is that the maintenance group frequently has to support software which is not developed with maintainability in mind. Schedule pressures often result in software being transferred to the maintenance phase before all deliverables are completed. Consequently, the maintenance group devotes significant time to issues related to supporting software which is poorly designed, coded, tested and documented; that is, code which was not designed for maintenance. Thus we heard a consistent message that tools for such activities as reverse engineering and testing were a high priority and that several groups were not aware of the state-of-the-art in tools. We also heard that the software to be maintained does not have effective documentation to support it, either in written reports or embedded in the code. We do not believe that it is the intent of the original developers to produce incomplete work. Rather, the pressures to meet unrealistic schedules force both in-house developers and contractors to release their software before it is ready. We also saw evidence that schedule pressures within maintenance projects result in the same problem with maintained code. Thus the problem is being perpetuated in the maintenance phase. To overcome these problems, we recommend a two-phase approach. In the short term, there is a strong need for better tools to support reverse engineering, testing, configuration management, and documentation. These tools will help deal with the problems of existing code. In the longer term though, there is a need to improve the quality of the developed software. This involves such issues as developing schedules that provide more time to develop a quality product, writing effective software documentation, adhering to quality assurance standards, providing effective training on technical and non-technical issues, and coordinating communication between development and maintenance groups. We found that communication was less effective than it could have been between different organizational entities. First, little communication appears to occur between different project personnel regarding software engineering issues. Because many of the problems confronted are common across projects, the sharing of experiences, mistakes made, lessons learned, etc., can be invaluable. Therefore communications ought to be encouraged through such means as informal reports, seminars, a software engineering bulletin board, etc. Second, we found that communication on technical issues from the project level to higher management could be improved. Some project personnel we interviewed felt that upper management was not aware of the depth of some of the technical problems they were facing. We have therefore recom- 14 CMU/SEI-93-TR-8 CMU/SEI-93-TR-8 13 overriding consideration. In the long run, we firmly believe that the quality improvement actions we have described will reduce costs, but an up-front financial investment is clearly required. Thus, some form of cost-benefit analysis seems appropriate in order to establish priorities. Also, establishing a task force that is responsible for developing, implementing, and monitoring the improvement plan will help keep the project on track.

What is software maintenance costs ?

Software maintenance cost is derived from the changes made to software after it has been delivered to the end user. Software does not “wear out” but it will become less useful as it gets older, plus there WILL always be issues within the software itself.

The software industry has had significant progress in recent years. The entire life of software includes two phases: production and maintenance. Software maintenance cost is increasingly growing and estimates showed that about 90% of software life cost is related to its maintenance phase. Extraction and considering the factors affecting the software maintenance cost help to estimate the cost and reduce it by controlling the factors.

Summary of case study on software maintenance costs:

Based on interviews, 32 factors were identified in the cost estimation of medical software maintenance and were approved by informatics specialists. Using AHP model parameters, *6* groups were ranked. Since in each research a problem is stated and examined and at the end solutions are proposed, in this study, we also provide solutions to reduce maintenance costs. What the Informatics experts agree on for reducing maintenance costs, is that “with respect to some important factors such as accuracy in HIS projects feasibility, along with complete documentation and helping the design and implementation mechanisms in the maintenance phase ,favorable results can be achieved in reducing the cost.“

Generally we can conclude that for an accurate assessment and reduce the cost of software maintenance, software maintenance factors determining is essential. This will lead to the longer life of software. Evaluation of these factors and their influence on each of the maintenance costs, help the project manager in making decisions and planning, and is essential in the success of software maintenance. Project managers must consider these factors for success in their projects and decisions:

* HIS software is generally in a network and for giving a better service to applicants, data collection is done on the central server. As a result, software should be developed in a network and maintainers should give their service in a network. In other words, if the software is single that costs less, but for network applications, computer network costs are added to the costs. So in designing this software these costs should also be noted.

* To reduce maintenance costs and increase the longevity of HIS software determining the cost estimation factors is necessary, this can help to increase productivity and provide a native model to estimate the system maintenance cost. It will make the project manager able to estimate the real cost at any time in the system.

Summary of case study on software maintenance costs:

The bulk of the cost of software development is due to the human effort, and most maintenance cost estimation methods focus on this aspect and give estimates in terms of person-months. Accurate software maintenance cost estimates are critical to both developers and customers. They can be used for generating request for proposals, contract negotiations, scheduling, monitoring and control. Underestimating the costs may result in management approving proposed systems that then exceed their budgets, with underdeveloped functions and poor quality, and failure to complete on time. Overestimating may result in too many resources committed to the project, or, during contract bidding, result in not winning the contract, which can lead to loss of jobs. Accurate maintenance cost estimation is important because

• It can help to classify and prioritize development projects with respect to an overall business plan.

• It can be used to determine what resources to commit to the project and how well these resources will be used

. • Projects can be easier to manage and control when resources are better matched to real needs.

• Customers expect actual development costs to be in line with estimated costs. But there is no such model for estimating the maintenance cost of CBS.

----------------**-----------------------