SOFTWARE MAINTENANCE (SE - 409)

DELHI TECHNOLOGICAL UNIVERSITY

PROJECT REPORT

ON

TOPIC: Software Maintenance Management

Submitted To:	**Submitted By:**
Priya Singh	Ashish Kumar (2K18/SE/041)
	Kevin Tirkey (2K18/SE/074)

TABLE OF CONTENTS

CERTIFICATE

This is to certify that this project titled, "**Software Maintenance Management**" submitted by "Ashish Kumar & Kevin Tirkey" students of "Software Engineering" branch at the "Delhi Technological University". I certify that this project is up to my expectations and has been made under my supervision and guidance.

Date: 11 November 2021

Priya Singh

(Assistant Professor, Software Engineering, DTU)

ACKNOWLEDGEMENT

This project report required a lot of help & leadership and we are extremely fortunate to receive such assistance in completing our project report. Whatever our group has done is only due to able guidance and assistance and we will always be remembering them for their help.

We respect and thank Priya Singh Ma'am for giving us this opportunity to do this Mid-Term Project and providing us support and leadership which aided us in finishing this project report successfully on time.

ABSTRACT

This report contains a comprehensive evaluation of a survey regarding Software Maintenance Problems & issues and how to deal with these problems by proposing new models and various hypotheses. In the end, the best management techniques and recommendations are made based upon the results of the study. This file contains the components of our project report i.e. Introduction, Methodology, Results, Limitations, conclusion and references.

Number of Tables: 3

Number of Figures: 1

INTRODUCTION

The process of software maintenance is very difficult and also the software development process. In the maintenance phase, software is customized according to the problems that are faced by the customer and improvement requests from the customers and then iteratively upgrading the software and after resolving issues it is released to customers. This is a very complicated task and therefore Software maintenance management plays a very crucial role in the software maintenance phase.

This report contains brief introduction to Software maintenance management and it consists of comprehensive evaluation of a survey conducted in 1977 regarding Software Maintenance Problems & issues followed by the recommendations to management which resulted from the study and how to deal with these problems by proposing new models and various hypotheses based on the 1977 study.

Now let's get into what is **Software Maintenance Management**?

Software Maintenance Management forms the base for effectively managing the software maintenance activities and enforces management rules based on the performance of systems so as to increase the performance of a whole organization. Software Maintenance management helps organizations to be proactive rather than reactive with how it approaches maintenance and how to fix the issues during the maintenance phase.

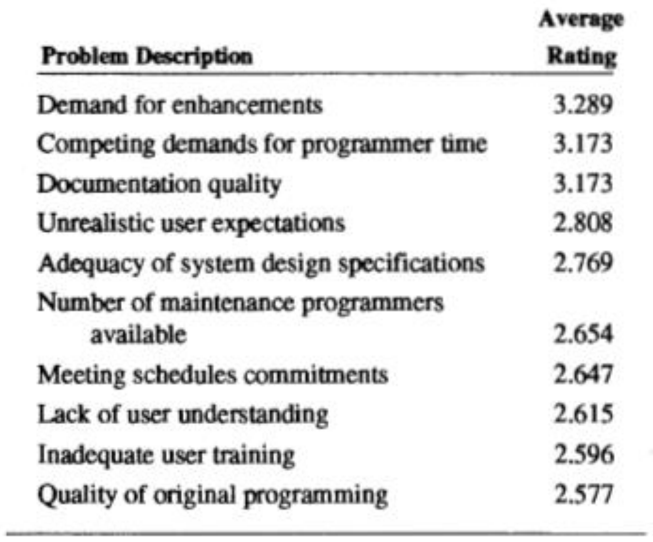
METHODOLOGY

In 1977, Lientz and Swanson performed a study of software maintenance in an organization. From their study, they identified the major issues of concern in software management, and made several specific recommendations for management concerning software maintenance.

Analysis was done and 26 maintenance problems were identified. These are as follows:

1. Maintenance personnel turnover
2. Documentation quality
3. System hardware and software changes
4. Demand for enhancements and extensions
5. Skills of maintenance programmers
6. Quality of original programming
7. Number of maintenance programmers available
8. Competing demands for programmer time
9. Lack of user interest
10. System run failures
11. Lack of user understanding
12. Program storage requirements
13. Program processing time requirements
14. Maintenance programmer motivation
15. Forecasting maintenance prog. Requirements
16. Maintenance programming productivity
17. System hardware and software reliability
18. Data integrity
19. Unrealistic user expectations
20. Adherence to programming standards
21. Management support
22. Adequacy of system design specs
23. Budgetary pressures
24. Meeting scheduled commitments
25. Inadequate user training
26. Turnover in user organization

The major issues that Lientz and Swanson had identified on the basis of average rating of these above mentioned maintenance problems:



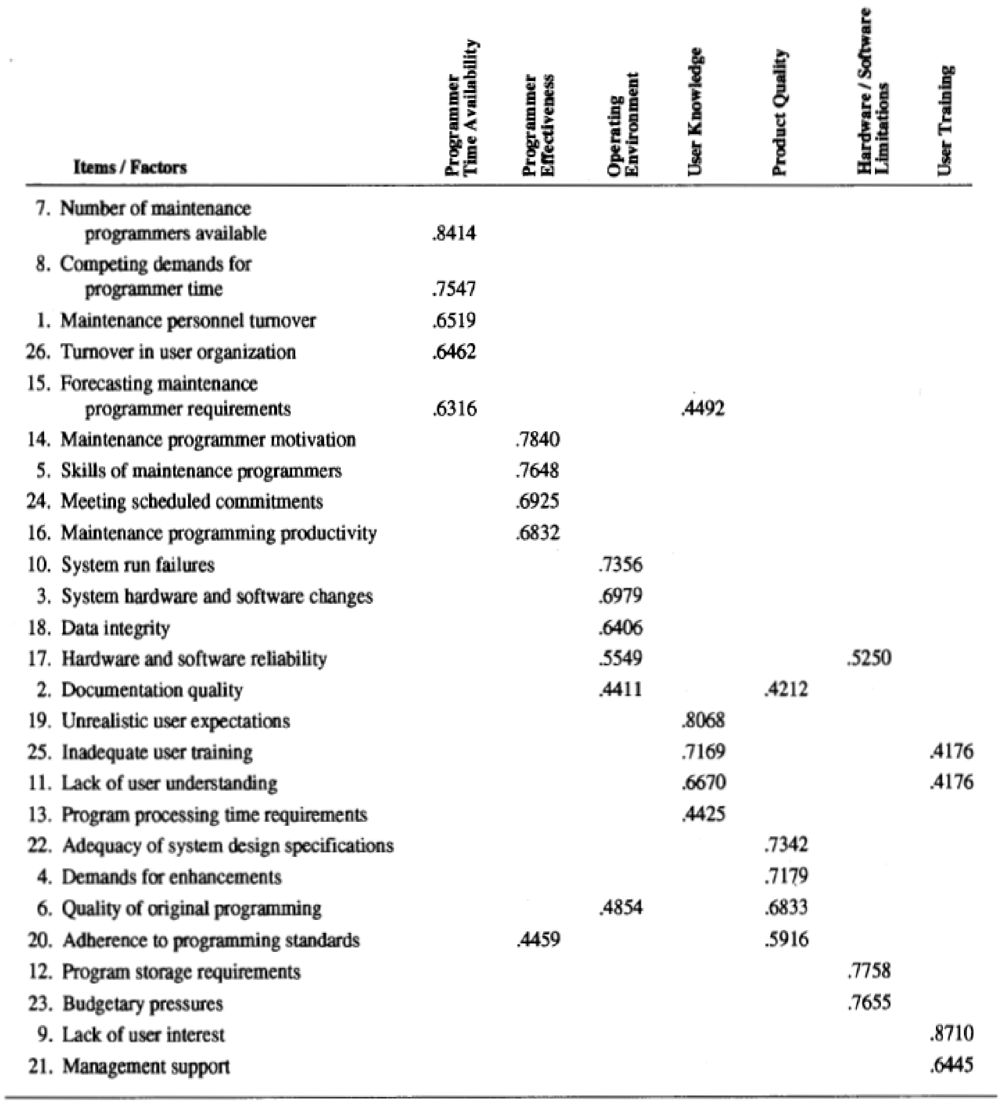
Problem Description	Average Rating
Demand for enhancements	3.289
Competing demands for programmer time	3.173
Documentation quality	3.173
Unrealistic user expectations	2.808
Adequacy of system design specifications	2.769
Number of maintenance programmers available	2.654
Meeting schedules commitments	2.647
Lack of user understanding	2.615
Inadequate user training	2.596
Quality of original programming	2.577

Table: Most Serious Maintenance Problems

First and the fourth problems in the above table are referring to user needs and expectations.
Second, sixth and seven problems in the above table are referring to challenging demands on programmer schedules and meeting schedule deadlines.
Third, fifth and tenth problems in the above table are related to the system quality. Eighth and ninth problems deal with the lack of user understanding and user training.

A factor analysis of the survey responses on the twenty-six problems was carried out in order to conclude the problem dimensions and to aid contingency analysis. Lientz and Swanson also conducted a factor analysis where they have identified following 7 dimensions:

1. Programmer Time availability: The things under this dimension refer to shortage of maintenance programmers, and their increasing demands.
2. Programmer effectiveness: This dimension refers to programmer skills, usefulness, and dedication.
3. Operating environment: This dimension refers to the issues caused due to hardware and software reliability, system failures, data consistency, documentation.
4. User knowledge: This dimension refers to issues that are caused due to user requirement, lack of user training and their understanding.
5. Product Quality: This dimension refers to the quality of the system. It consists of design specifications, quality of original programming, and following of rules & procedures.
6. Hardware / Software Limitations
7. User Training



Items / Factors	Programmer Time Availability	Programmer Effectiveness	Operating Environment	User Knowledge	Product Quality	Hardware / Software Limitations	User Training
7. Number of maintenance programmers available	.8414						
8. Competing demands for programmer time	.7547						
1. Maintenance personnel turnover	.6519						
26. Turnover in user organization	.6462						
15. Forecasting maintenance programmer requirements	.6316			.4492			
14. Maintenance programmer motivation		.7840					
5. Skills of maintenance programmers		.7648					
24. Meeting scheduled commitments		.6925					
16. Maintenance programming productivity		.6832					
10. System run failures			.7356				
3. System hardware and software changes			.6979				
18. Data integrity			.6406				
17. Hardware and software reliability			.5549			.5250	
2. Documentation quality			.4411		.4212		
19. Unrealistic user expectations				.8068			
25. Inadequate user training				.7169			.4176
11. Lack of user understanding				.6670			.4176
13. Program processing time requirements				.4425			
22. Adequacy of system design specifications					.7342		
4. Demands for enhancements					.7179		
6. Quality of original programming			.4854		.6833		
20. Adherence to programming standards		.4459			.5916		
12. Program storage requirements						.7758	
23. Budgetary pressures						.7655	
9. Lack of user interest							.8710
21. Management support							.6445

Table: Factor Matrix

The Factor analysis is carried out with principal components analysis as the extraction technique and varimax as the method of rotation. Rotated factor loadings of + .30 are considered significant, + .40 more significant, and + .50 very significant.

Proposed Model

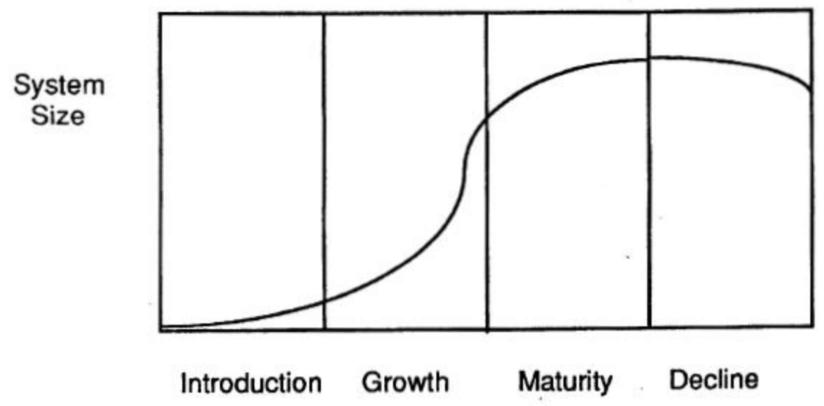
The motivation of this proposed model is to determine the impact of scale of effort factors such as system size, age and system complexity upon various software maintenance types, problems, and issues. This model considers 3 types of software maintenance process which are corrective, adaptive, and perfective maintenance.

S-curve model and Systems concepts were used to produce a base model of contingency relationships. The S-curve model can help us to identify what kinds of maintenance techniques were required on the basis of system age.

The below diagram describes the IS (information systems) growth, maturity and decline lifecycle.

FIGURE 1

S-curve Model of IS Growth



We have described few hypothesis below based on the proposed model and these are as follows:

1. System age: In general older systems may cause more problems due to lack of user knowledge, product quality, and programmer time along with h/w and s/w limitations.

   Based on this, we have mentioned various hypotheses which are written in table below:

Hypotheses number	Description
H1	System Age is positively correlated with corrective maintenance.
H2	System Age is positively correlated with user knowledge problems.
H3	System Age is positively correlated with product quality problems.
H4	System Age is positively correlated with programmer time availability problems.
H5	System Age is positively correlated with hardware/ software limitation problems.
H6	System Age is positively correlated with programmer effectiveness problems.
H 7,8,9	System Age is positively correlated with the use of cost benefit analysis (H7), chargebacks (H8), and logging procedures (H9)

2. System size: It basically tells us about the no. of programs and statements present within the system. As the system size gets larger, it gets quite complex. Accordingly, we have mentioned various hypotheses which are written in table below:

Hypotheses number	Description
H10	System Size is positively correlated with corrective maintenance.
H11	System Size is positively correlated with product quality problems.
H12	System Size is positively correlated with user knowledge problems
H13	System Size is positively correlated with hardware/software limitation problems.
H14,15,16	System Size is positively correlated with the use of formal maintenance control procedures like cost benefit analysis (H14), chargebacks (H15), and logging procedures (H16).

3. Database size: It tells us about the no. of files and the no. of char(s) in the database. A large size database would require more changes in input files. Also, the users must be less knowledgeable when it comes to using a large database.

   Thus, we have mentioned various hypotheses which are written in table below:

Hypotheses number	Description
H17	Database Size is positively correlated with adaptive maintenance.
H18	Database Size is positively correlated with user knowledge problems.

4. Staff size: It tells us about the no. of people in the development and the maintenance staff. A large system would require large no. of staffs which would require a considerable amount of resource allocation which would in turn make the system more complex.

   Therefore, we have mentioned various hypotheses which are written in table below:

Hypotheses number	Description
H19	Staff Size is negatively correlated with corrective maintenance
H20	Staff Size is negatively correlated with product quality problems.
H21	Staff Size is positively correlated with user knowledge problems.
H22,23	Staff Size is positively correlated with the use formal maintenance control procedures like cost benefit analysis (H22) and chargebacks (H23).

5. Percent of maintenance budget to overall IS budget: It tells us about the budget w.r.t maintenance activities. So systems that are provided with more resources tend to have less no. of problems.
   Therefore, we have mentioned various hypotheses which are written in table below:

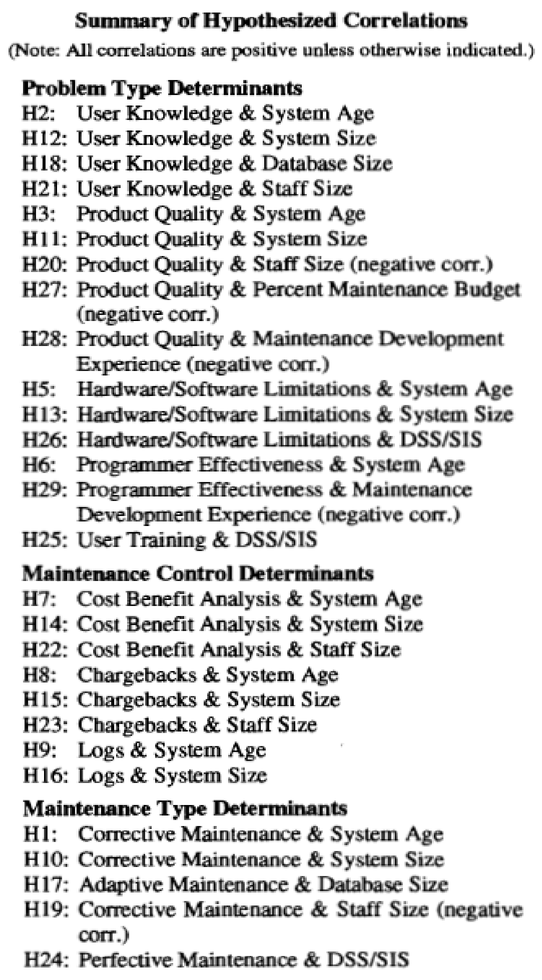
Hypotheses number	Description
H27	The percent of IS budget devoted to maintenance is negatively related to product quality problems.

6. Development Experience of the Maintenance Staff: It tells us about system development w.r.t maintenance staff's experience, so with greater experience there should be less problems with productivity and quality of the product.
   Accordingly, we have mentioned various hypotheses which are written in table below:

Hypotheses number	Description
H28	Development experience of maintenance staff is negatively related to product quality problems.
H29	Development experience of maintenance staff is negatively related to programmer effectiveness problems.
H30	Development experience of maintenance staff is negatively related to programmer time availability problems.

RESULT

After formulating these hypotheses it is essential to validate it, so we have regrouped the hypotheses based on each dependent variable (Problem Type, Maintenance control, Maintenance effort). Each of these hypotheses that we have mentioned in the proposed model was determined by calculating Pearson's correlation coefficients. Not all the hypotheses are supported but many of them contribute to the validity of the model.



Summary of Hypothesized Correlations

(Note: All correlations are positive unless otherwise indicated.)

Problem Type Determinants

H2: User Knowledge & System Age
H12: User Knowledge & System Size
H18: User Knowledge & Database Size
H21: User Knowledge & Staff Size
H3: Product Quality & System Age
H11: Product Quality & System Size
H20: Product Quality & Staff Size (negative corr.)
H27: Product Quality & Percent Maintenance Budget (negative corr.)
H28: Product Quality & Maintenance Development Experience (negative corr.)
H5: Hardware/Software Limitations & System Age
H13: Hardware/Software Limitations & System Size
H26: Hardware/Software Limitations & DSS/SIS
H6: Programmer Effectiveness & System Age
H29: Programmer Effectiveness & Maintenance Development Experience (negative corr.)
H25: User Training & DSS/SIS

Maintenance Control Determinants

H7: Cost Benefit Analysis & System Age
H14: Cost Benefit Analysis & System Size
H22: Cost Benefit Analysis & Staff Size
H8: Chargebacks & System Age
H15: Chargebacks & System Size
H23: Chargebacks & Staff Size
H9: Logs & System Age
H16: Logs & System Size

Maintenance Type Determinants

H1: Corrective Maintenance & System Age
H10: Corrective Maintenance & System Size
H17: Adaptive Maintenance & Database Size
H19: Corrective Maintenance & Staff Size (negative corr.)
H24: Perfective Maintenance & DSS/SIS

1. Problem Type Determinants: From our findings, system size was +vely associated with user knowledge (H12: p = .067). Also the user training factor was +vely correlated with system size (p =.075). Even though this finding wasn't considered in above mentioned hypotheses, a probable justification is that systems that generally have a large size are mostly complex, so the amount of training required on the system by the user is not usual, it requires a high amount of training. In contradiction to the above hypothesis (H20), the staff size is +vely associated with the product quality (H20: p=.03).
   The probable reason for this finding might be that huge no. of staff works on larger & complex systems and also they are harder to maintain which results in problems in

product quality. Therefore, staff size is not an independent variable but an intermediate variable. The %age of budget allotted to maintenance is -vely related to product quality (H27: p=.031. Additionally, the development experience of maintenance staff is -vely related to problems with programmer effectiveness (H29: p=.029).

2. Maintenance control determinants: In General, large size systems are associated with excessive utilization of maintenance controls as there is a wide use of user request logs in large size systems (H14: p =.000), & of chargebacks (H15: p=.093). Another finding from our analysis is +ve correlation between system age and chargebacks (H8: p = .062).

3. Maintenance effort distribution: Larger systems are +vely associated with the corrective maintenance (H1: p= .088) as the systems grows larger in size the no. of errors would significantly increase, therefore increasing the amount of corrective maintenance that is required. Systems with larger databases are +vely associated with adaptive maintenance (H17: p=001) as Large size databases might require a considerable amount of change in data.

RECOMMENDATIONS

Even after technological advancements problems still continue in software maintenance. These problems generally are associated with the procedure & rules related to maintenance management. So, on the basis of our analysis we come up with the below recommendations:

1. From our study it is made evident that maintenance is considered to have lower significance than new development which is one the main problems that are mentioned in this report. So we need to focus on the management view on the maintenance part by giving maintenance and equal status to new development. Another recommendation in this regard is that we can split systems into two categories: one is Installed Systems & another one is Future Systems. The main idea behind this is that we can achieve a seamless split between maintenance & development.

2. From our study we also observed that the maintenance staff not having enough experience in their domain aggravated the problems related to staff's efficiency, so another recommendation is that we can improve the working environment of the organization and also by hiring skilled & experienced folks in the maintenance department.

3. Another suggestion is to include allocating additional budget to maintenance activities as the budget has an important role in improving product quality and also by changing management techniques.

4. User knowledge is considered as the major problem which includes unrealistic user expectations and users not having enough experience in training and understanding. So to address this problem we can give user training on a monthly basis so as to adjust them with the system operations. Another solution that might work on unrealistic user demands by introducing user developed systems.

5. The maintenance activities are directly proportional to quality design, so it is essential to follow quality standards, proper documentation (SRS) and good programming skills while developing a system to avoid any problems in future.

6. Monitoring of maintenance: There should be an effective database that records the no. of hours & resources allocated to a maintenance activity to efficiently manage the organization.

CONCLUSION

In this study, we have overlooked the software maintenance issues and provided a brand new perspective to tackle these problems. A similar study was conducted in 1970's by Lientz & Swanston and the same methodology was used to gather data for the study. In addition to this, a model and various hypotheses have been proposed to analyze and find the cause behind these problems. One of the conclusions that have been discovered is that even after technological advancements, the same issues that prevailed a decade ago, continue to scourge maintenance activities and their management even today. With the amount of resources being spent on maintenance, it is crucial to shift our attention on addressing the maintenance issues. These issues are further worsened by the existing notion that new systems require a considerate amount of attention from the technical and management staff.

The most effective way to overcome these issues is to identify the problems and enforcing good management strategies that ensure system availability & reliability along with user satisfaction and timely responses to problem reports which can overall improve the productivity of an organization.

REFERENCES

We have used 4 research papers from various citations and 1 book by James A. McCall, Mary A. Herndon, and Wilma M. Osborne.

1. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nbsspecialpublication500-129.pdf
2. https://link.springer.com/article/10.1023/A:1013156608583
3. https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.4360020303
4. http://www.software-supportability.org/Docs/strategies.pdf
5. http://jitm.ubalt.edu/VI-3/article2.pdf