SOFTWARE MAINTENANCE

(SE – 409)

Software Maintenance Management

Submitted by:
Ashish Kumar (2K18/SE/041)
Kevin Tirkey (2K18/SE/074)

Submitted to : Priya Singh

PRESENTATION OVERVIEW

- ❑ Introduction
- ❑ Methodology
- ❑ Proposed Model
- ❑ Result
- ❑ Recommendations
- ❑ Conclusion
- ❑ References

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 presentation 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 model and various hypotheses.

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 organization to be proactive rather than reactive with how it approaches to maintenance and how to fix the issues during maintenance phase.

METHODOLOGY

In 1977, Lientz and Swanson performed a study of computer application software maintenance in data processing organizations. 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:
Maintenance personnel turnover, Documentation quality, System hardware and software changes, Demand for enhancements and extensions, Skills of maintenance programmers, Quality of original programming, Number of maintenance programmers available, Competing demands for programmer time, Lack of user interest, System run failures, Lack of user understanding, Program storage requirements, Program processing time requirements, Maintenance programmer motivation, Forecasting maintenance prog. Requirements, Maintenance programming productivity, System hardware and software reliability, Data integrity, Unrealistic user expectations, Adherence to programming standards, Management support, Adequacy of system design specs, Budgetary pressures, Meeting scheduled commitments, Inadequate user training, Turnover in user organization.

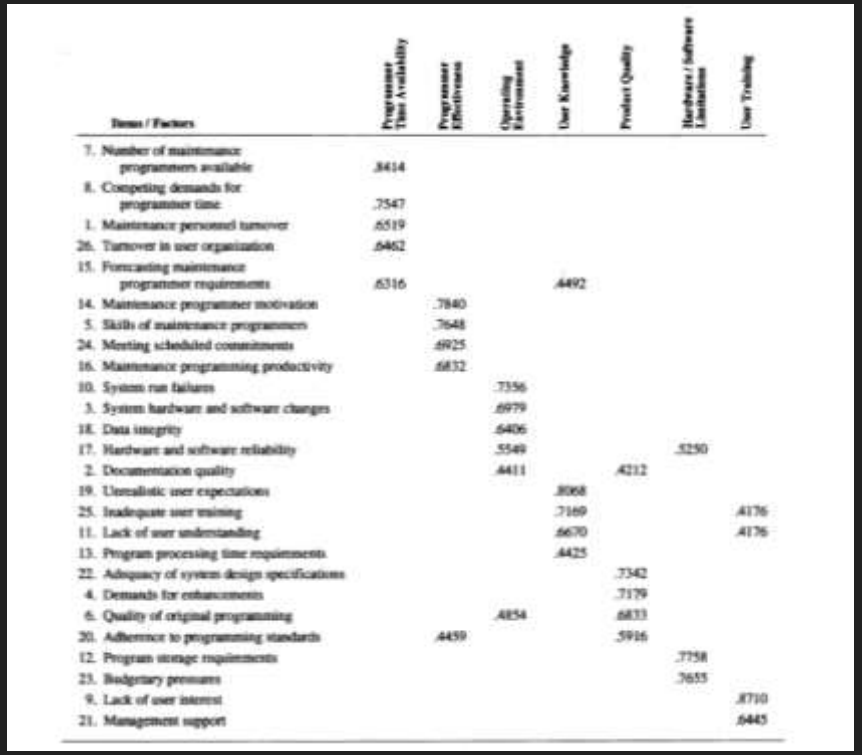
These were the 10 major issues that Lientz and Swanson had identified on the basis of average rating of the mentioned maintenance problems :

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**

A factor analysis of the survey responses on the 26 problems was carried out in order to identify the problem dimensions and to aid contingency analysis. So from the factor analysis, Lientz and Swanson have identified following 7 dimensions:

1. Programmer Time availability
2. Programmer effectiveness
3. Operating environment
4. User knowledge
5. Product Quality
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 analysis result**

PROPOSED MODEL

The motivation of this proposed model is to determined impact of scale of effort factors such as system size, age and system complexity upon various software maintenance types, problems, and issues.

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 .

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.

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.

4. Size of staff: 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.

5. Percent of maintenance budget to overall 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.

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.

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.

1. Problem Type Determinants: From our findings, system size was positively associated with user knowledge (H12: p = .067). Also the user training factor was positively 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 less than adequate, it requires a high amount of training. In contradiction to the hypothesis (H20), the staff size is positively 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.

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).

3. Maintenance effort distribution: Larger systems are positively associated with the corrective maintenance (H1: p = .088). Systems with larger databases are positively associated with adaptive maintenance (H17: p = .001).

TABLE 6

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

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.

2. From our study we also observed that the maintenance staff's lack of experience in their domain aggravated the problems related to staff's efficiency, so our 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.

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 hypothesis 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. 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

THANK YOU