Software Maintenance Productivity measurement: how to assess the readiness of your organization

Alain April[1], *Alain Abran*[2], *Reiner R. Dumke*[3]

[1] École de Technologie Supérieure, Montréal, Canada

[2]École de Technologie Supérieure, Montréal, Canada,

[3]Otto von Guericke University of Magdeburg, Germany,

aapril@ele.etsmtl.ca, aabran@ele.etsmtl.ca , dumke@ivs.cs.uni-magdeburg.de

Abstract:

Software maintenance constitutes an important part of the total cost of the lifecycle of software. Some even argue this is the most important fraction of the cost (50-80 percent according to Tony Scott [14], *75% according to Rand P. Hall* [5]*). The added value of software maintenance is often not fully understood by the customer leading to a perception that software maintenance organizations are costly and inefficient. A common view of maintenance is that it is merely fixing bugs. However, studies over the years have indicated that in many organizations the majority, over 80%, of the maintenance effort is dedicated to value added activities (Sommerville*[15], *Pressman*[13], *Pigoski*[12]*). To improve customer perceptions of software maintenance it is important to provide them with better insights into the activities performed by the maintenance organization and to document such performance with objective measures of software maintenance activities.*

In this paper the prerequisites for Software Maintenance productivity analysis are described with the use of the experiences at the Bahrain Telecommunications Company (Batelco) during the years 2001-2. First the differences between software maintenance activities and IS development projects are described. Then a basic trend model is applied as well as ways to manage the expectations of the customers. To conclude, some remarks are made regarding the application of productivity analysis for the software maintenance managers.

Keywords

Software Maintenance, productivity measurement, process improvement

1 Introduction

The software life cycle can be divided into two major parts: (1) initial software development and (2) software maintenance. Well documented by *Abran* [2, 10] and *Pfleeger* [11], software maintenance is different from software development. It is different because of the following characteristics (*Abran et al)*[1]:

- The size and complexity of each maintenance work request are such that it can usually be handled by one or two resources;
- Maintenance work requests come in more or less randomly and cannot be accounted for individually in the annual budget-planning process;

- Minor enhancements (adaptive) work requests of the enhancement category are reviewed with customers and can be assigned priorities;
- The maintenance workload is not managed using project management techniques, but rather with queue management techniques;
- Maintenance has a broader scope of configuration management with more operational considerations.

The software maintenance workload is user-services-oriented and system-responsibility-oriented. Priorities can be shifted around at any time, and work requests of system correction often take priority over other work in progress.

The four major aspects that software maintenance focus on are (*Pfleeger)*[11]:

- Maintaining control over the system's day to day functions;
- Maintaining control over system modification;
- Perfecting existing acceptable functions;
- Preventing system performance from degrading to unacceptable levels.

An important aspect of software maintenance work done to a system or environment might not be directly visible to the customer; therefore customers often feel that IS does not provide enough value with respect to the budget spent on maintenance. Such a negative perception is basically due to a lack of information made available to the customers to explain all the services exchanged, and their contributions to the organization. Most software maintenance organizations have not progressed yet to a transparent operation where they share detailed management information with their customers in terms they understand [4].

This is also based on a premise that the software maintenance processes are documented, mature and stable across the organization and that they are well understood by the maintainers themselves.

Before addressing productivity analysis it is important to present the reader with the prerequisites that should be operational in the software maintenance organization as a prerequisite to this measurement. The Software Engineering Institute (SEI) has identified that there is a basic set of common measures that can be used by all software organizations: size; effort; schedule and quality. To enable the measurement of these four measures the following software maintenance processes must be reviewed across all maintenance organizations: (1) software maintenance activity definitions, (2) software maintenance requests tracking process, (3) software maintenance personnel time recording. Most software maintenance organizations will need to initiate process improvement activities to implement, document and review these three basic processes before initiating productivity measurement.

During 2001, a process improvement activity was initiated in the Bahrain telecommunications Company to implement a maintenance request tracking process and information system. Each customer request sent to software maintenance organizations need to be recorded, dispatched and tracked formally. For this process to be clear to every software maintenance employee involved, it was necessary to review and standardize the many software maintenance activities and services offered in this organization. This improvement project ensured that the demands for software maintenance services could be orderly recorded, processed, measured and analysed. A second process improvement activity was initiated to enhance the time recording system to track maintenance personnel efforts according to (1) “billable” (*also called productive activities*) and (2) “non billable” (*also called non-productive activities*) administrative activities. This second process improvement project ensured that the “demand” for software maintenance services could be measured and analysed by ensuring that the time records reflect accurately the effort spent by each member of the software maintenance organization. With both the billable and non-billable perspectives being tracked, insights into productivity would be possible in the future.

2 Prerequisites

This section describes, in more detail, the prerequisites to productivity measurement of software maintenance.

2.1 Software maintenance activities

For an organization to mature it is necessary that its processes (or set of activities) for carrying out the software maintenance activities be defined. *Lientz and Swanson* [9] where the first researchers to propose categories of software maintenance. Since then, standards for software maintenance *(ISO/IEC 14764* [8] *and IEEE Std. 1219-1998*[6]*)* have documented the international consensus on software maintenance work categories. These standards divide software maintenance into four work categories (Fig.1). Work categories are defined in terms of: (1) the timing of the change (proactive or reactive) and (2) the goal of the change (correction or enhancement).

	Correction	Enhancement
Proactive	Preventive	Perfective
Reactive	Corrective	Adaptive

Figure 1: **ISO 14764** Software Maintenance categories of work

Software Maintenance organizations typically work on software systems developed internally with access to the source code. Currently, with the implementation of many Commercial Off-The-Shelf (COTS) software solutions (i.e. SAP/R3, Oracle finance/human resources and PeopleSoft), the nature of some of the software maintenance activities differs. A third party then owns the source code and a maintenance contract with a third party is required to formalize the third party supplier Service Levels. For the COTS, the software maintenance personnel then become the main interface to the vendors, which in turn guarantee the software service levels in the area of functionality corrections and enhancements. These additional maintenance interactions with third parties lead to more interfaces and hand-over, making maintenance work even more difficult to understand for the software maintenance customers.

2.1.1 Software Correction dimension

The two main work categories identified in the ISO14764 software maintenance international standard are corrections (preventive and corrective) to the software. These corrections aim at keeping the software at an agreed quality level in conformity to the established service levels.

The first work category of corrections is preventive software maintenance. It provides activities, after the initial delivery of the software, to prevent failure of software system by detecting and correcting latent errors before they become effective faults. Examples include:

- Y2K and Euro changes;
- Correcting edit rules to prevent recurrence of failures;
- Implementing and using monitoring tools to detect problems before they impact the client.

The second work category is a software correction which is a reactive modification needed for the elimination of an error condition in a software system that is impacting the operability of a production system.

This service covers all associated coding, testing, change control, software distribution, documentation, job rerun, and file recovery required to fix the error condition. In summary, these activities include:

- Provide immediate assistance to Help Desk and assist in problem (trouble ticket) prioritisation;
- Provide immediate support and priority on system failure;
- Providing system recovery from outages;
- Fixing job failures;
- Fixing production system problems;
- Correcting erroneous data, data fill and flow problems.
- Provides feedback and status report to Help Desk until problem is resolved and problem is closed;
- Analyse system fault information and conduct root cause analysis;
- Discuss and review with system developers, system designers operations personnel and third party suppliers future upgrades, changes and quality standards;
- Coordinate problem resolution with Help Desk, Operation Management and Service Partners.

Corrective maintenance is necessary and often caused by hidden faults in the software system. These faults, not uncovered by the many verification and validation processes of software development should be dealt, initially, by the organization supplying the software (as a form of project or contract warranty). Once the new software stabilized, the software maintenance organization will take over the responsibility of the software system to operate within the service levels agreed and established in the SLA (Service Level Agreement).

2.1.2 Software Enhancement dimension

The two work categories in the enhancement dimension seem less familiar and are defined in terms of results rather than activities.

Proactive enhancements (perfective and adaptive maintenance) are usually initiated by the IT organization to enhance the maintainability of the software and improve software maintenance effectiveness. This type of change does not change the information system functionality and should therefore be highlighted as such in the SLA.

The functionality enhancements required by the customer do have a large impact on his perception of the software maintenance value and therefore must be tracked closely. This is mainly because they are not to be considered as maintenance of software when their size requires more than one individual.

The first work category, the perfective maintenance, provides minor functional improvements, quality, maintainability and operability improvements for a specific system or the whole system portfolio with an objective of reducing the current level of resource consumption. Examples include:

- Optimising code and resources;
- Restructuring code logic;
- Clarifying and improving system documentation;
- Minor functionality enhancements.

The second work category in this dimension, the Adaptive maintenance, provides activities required to adapt the system to a change in the operating environment: hardware, operating system, or volume where no new or changed functionality is required. Examples include:

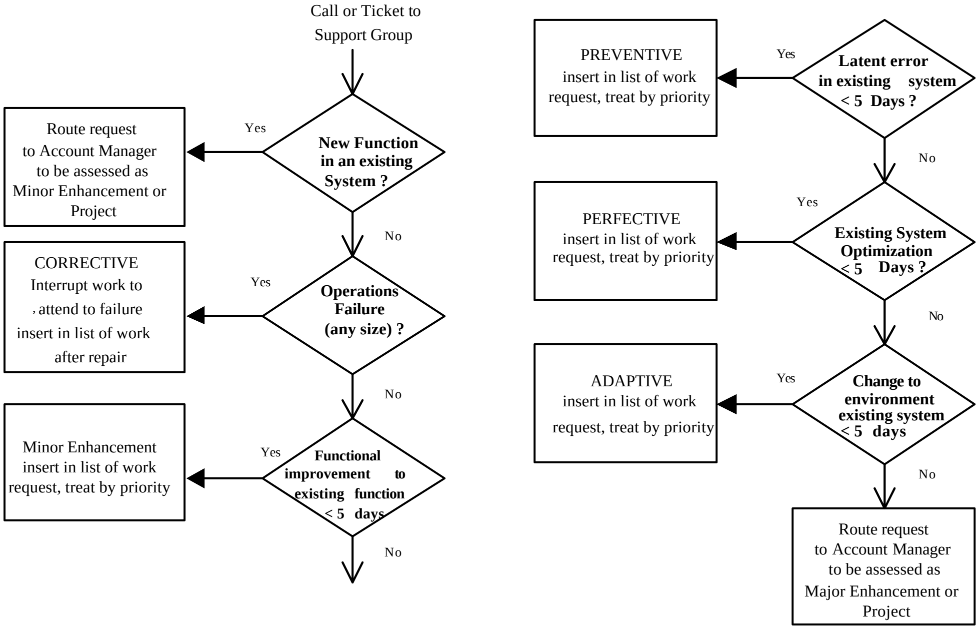
- Compiler and utility changes;
- Hardware upgrades;
- Media conversions;
- Making adjustments to accommodate changes in load;
- Evolving the System Recovery Manual and testing of Disaster Recovery Plans.

These *ISO14764* definitions should be used to categorize each work request and track personnel time against these categories. Each operational software should have management reports using these categories.

As well, the organization should ensure that all the Software Maintenance personnel track their time (know as the *effort*) accurately against these internationally accepted software maintenance categories of work.

2.2 Work Request Process and Time Recording of Software Maintenance

For an organization to mature it is necessary that its processes (or set of activities) for carrying Software maintenance procedures differ according to the type of maintenance requests. At Bahrain telecommunications four major sources of maintenance requests have been identified: software customers, software operators, project managers completing their projects and requests for re-engineering studies. Because the size and complexity of each work request to be assigned to a maintenance team is handled by one or two resources only it is essential that at the outset of a work request it be assessed and, when it is outside of the maintenance scope in terms of either required time and resources and levels of authority, be re-directed to the development organization for review, prioritisation, budgeting staffing, planning and execution. This is often a source of confusion in IS since it is initially difficult to form a consensus on how much effort a minor-enhancement (*perfective request*) to be done by maintenance, and what is the threshold for a handover to a software project organization the development [4].



Call or Ticket to Support Group
New Function in an existing System ?
Yes
Route request to Account Manager to be assessed as Minor Enhancement or Project
No
Operations Failure (any size) ?
Yes
CORRECTIVE Interrupt work to , attend to failure insert in list of work after repair
No
Functional improvement to existing function < 5 days
Yes
Minor Enhancement insert in list of work request, treat by priority
No
Latent error in existing system < 5 Days ?
Yes
PREVENTIVE insert in list of work request, treat by priority
No
Existing System Optimization < 5 Days ?
Yes
PERFECTIVE insert in list of work request, treat by priority
No
Change to environment existing system < 5 days
Yes
ADAPTIVE insert in list of work request, treat by priority
No
Route request to Account Manager to be assessed as Major Enhancement or Project

Figure 2: Acceptance/Rejection of work requests in software maintenance

The procedure illustrated in Fig. 2 was developed and a threshold of 5 person/days was selected as the maximum effort that the software maintenance personnel could dedicate to a perfective request (*also often called a minor-enhancement*). This five day limit is also recognized by the United Kingdom Software Metrics Association (UKSMA). They state that: ‘The distinction between maintenance activity of minor enhancements and development activity of major enhancement is observed in practice to vary between organizations. The authors are aware that in some organizations activity as large as to require 80 workdays is regarded as maintenance, while in others the limit is five days. Initially it is proposed that the ISBG and UKSMA will adopt the convention that work requiring five days or less will be regarded as maintenance activity'. [7] This threshold is very important in organizations as it states clearly what is software development versus software maintenance.

The work requests larger than the approved threshold of 5 days are re-directed to IS account managers for budgeting review and planning including as part of major enhancement or bundled into small development projects.

The work request tracking system, often automated using problem ticket commercial software (i.e. *Remedy, Vantive, etc),* supports the creation, follow-up and closure of an individual work request (*also called a ticket*). Both the Time Recording System and the work request tracking system should be integrated to ensure it is entered only once by the maintenance personnel. The use of a formal work request procedures and toolset should be agreed upon with the customers to ensure awareness of the priority assignment rules for each work request. An audit of this process, system and list of active/pending requests by the customers, should be done at least once a year to ensure and raise their confidence in the work request processing.

Since the software maintenance service levels are based on tracking effort for each service request it is essential that the IS Time Recording System (TRS) and reporting also be accurate. A review of the TRS associated to software maintenance work requests is necessary to ensure completeness and accuracy of reporting. If it is found that software maintenance does not account for close to 90% and more of their effort/cost a correction/improvement plan should be put in place to ensure completeness and accuracy of these records.

The solution to incomplete or deficient time tracking accuracy is not technically difficult: it is a software engineering management issue. It requires, on the one hand the set up of a standard chart of accounts (Fig 3.) for the daily tasks performed on software maintenance work requests and, on the other hand, a data collection procedure for all maintainers. To explore economic productivity in ways that lead to insights for customer charge back or improvement process, a reasonably granular chart of accounts is required. Details of time reporting on time spent per work request and maintenance category should be available on demand to both IS and customer organization. It is essential that the data provided be assessed and validated to ensure integrity and credibility both within IS and customers organizations.

Activity	Definition
Admin.	General Admin, Housekeeping, Office work, communication, Reporting.
Break/ Non-Productive	Lunch, tea, personal work, non-productive task, sick.
Classify and update work-request	Preventive, Corrective, Perfective, Adaptive or Customer Request for information.
Analysis	Search for causes, replicate/verify.

Modification	Code and unit test the change.
Regression Testing	Verification of fixes and changes.
Migration & Configuration Management	Source code moves, Migrations.
Reviews	Verification of change. Surveillance days after changes.
Training	Attending training, presentation & conference.
Documentation	Updating Technical Documentation and D/R.

Figure 3: Software Maintenance Time Recording Chart of Accounts

Definitions of collected data and validation of that data are essential steps that IS must take to ensure that the data integrity is present. It should be verified that the supporting information systems used by IT allows for all the categories and sub-activities described in this section since measurement will be based on that data.

3 Maintenance Measurement

3.1 Studying the demand for Software Maintenance services

The monthly demand for software maintenance is the total of all requests in a given month (Correction work requests + Enhancement work requests). The software maintenance demand reporting was based on this approach during 2001 (Fig. 4).

Work Requests Demand	Corrections	Enhancements
From last month	33	4
New this month	204	3
Solved this month	219	6
Carried forward	16	3

Figure 4: Back Office November report

While these totals provide high-level information aggregated for all the systems in this organization, they do not provide enough information on maintenance work requests and they are not conducive to productivity analysis and comparisons. A first improvement was to identify the maintenance personnel effort associated with each request and report it (Fig. 5). Monthly supply for software maintenance can be derived from the total of effort, by work categories, reported in a given month (S work request time reports).

Demand	October	November	December	Total
Demand(#Requests	218	209	180	607

+ #problems)				
Supply (days)	870	1111	1222	3203

Figure 5: Back Office supply vs. demand

Since distinct system have both different technical characteristics as well a customer driven demand peculiar to their own domains, it is also necessary to report software maintenance work requests by categories and by system (Fig. 6):

Software Maintenance work requests	Information System A	Information System B	Information System C
Correction	59	35	34
Enhancement	95	101	96
Customer Query	59	70	56
Unclassified	2	0	0

Figure 6: Back Office details **IVQ01**

This data highlights that most of the effort of software maintenance is carried out in enhancements (5 days of less), corrections to production problems and customer query/support of production system. We then proceed to do this for all the back office system. With this information the maintenance manager can create a graph representing the service trends by software maintenance services. (Fig. 7)

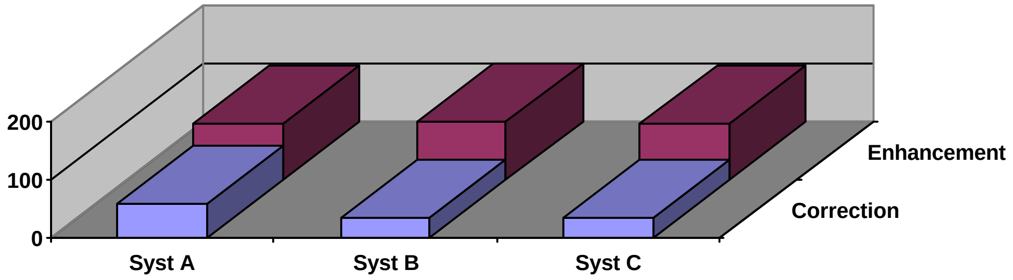


Figure 7: Back Office comparative data

Understanding why each system requires more or less maintenance expenses is necessary for pro-active management of the software maintenance portfolio. Such a chart can also be produced by cost per category for budget view. Once the customers will have worked with this detailed information provided by IS there are additional measures that can be introduced when required. Lines of code and Function point measures can be used as normalization factors to compare the customer systems between themselves and to the industry. However, obtaining these measures from COTS such as SAP/R3, Oracle or PeopleSoft can be quite difficult, if not impossible.

3.2 Studying the supply of Software maintenance services

Figure 5 documents that IS software maintenance personnel have recorded 3203 days of work effort for this period of time. This represents 16 work-years for this organization (*assuming 200 work-days in a year, with 7.5 hours of billed work per day*). It also represents an average of 5.2 days of effort per work request. This measure can be an interesting one to compare with other maintenance organization.

Comparing and studying the percentage of distribution of work per category can also give you insights about overall distribution of effort compared to other maintenance organizations. It is important to understand how much time an average request takes per category for budgeting reasons. It is interesting for the IS customers to have an idea of where are the software maintenance hours being spent in comparison with their demands. This measurement of the workload distribution is critical to maintenance managers in order to evaluate if the trends are going in the direction that the 'paying' customer wants.

3.3 Using trend analysis for Software Maintenance analysis

With three months of work effort distribution by software maintenance category, maintenance managers should be able to start identifying trends. In this experience we observed that the customer appreciates downward trends in the corrective requests while upward trends in perfective requests (small enhancements) are also appreciated.

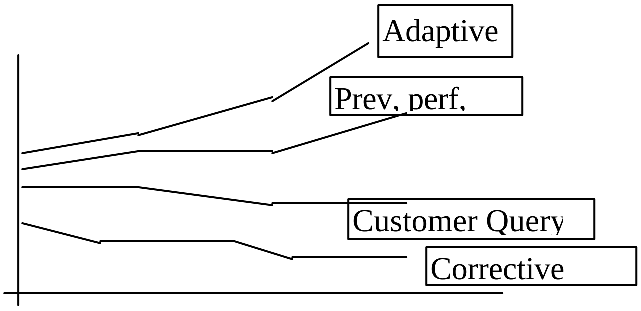


Figure 8: Typical trends of Software Maintenance work

Trends shown in Figure 8 demonstrate to the customers that progressively less effort is required to control the existing software (e.g. corrections and tuning take less effort out of his budget) while more functionality and "value added" support is done. Figure 8 represents trends that a typical customer would like to observe from a software maintenance supplier on their software.

As stated earlier, for that analysis to be valuable, the software maintenance personnel should take great care on the assignment of work requests to a specific category. Also an indirect benefit of measuring these elements could be to collect historical data for work request early effort estimation as we collect more data. The following section describes where this type of data gathering could also be beneficial for maintenance managers:

Software Corrections: Whereas a fire fighting mentality was predominant beforehand there is now more professional and manageable approach to presenting the software maintenance contributions. Comparisons between software systems are also made possible.

The information gathered from analysing the categories of work can be used to define quality indices and targets by system, for external supplier providing maintenance and or software development teams before handover of software. Such quantitative figures can also be used to initiate preventive maintenance programs and justify system re-engineering's efforts.

Enhancements: The ratio of adaptive maintenance is often an indicator of mandatory enhancements and the introduction of new products. It is also an indication of the rate of introduction of changes and new technology in the organization. It could also lead to insights into identifying suppliers that have the most impact on the organizations' resources (effort required by products).

Preventive, Perfective and Portability maintenance: The data collected for each sub-category can be used to evaluate the quality and efficiency of the software. It can also be used to compare the costs and level of system tuning required by each software. Insights from such analysis can lead to re-engineering decisions and provide insights on comparing potential initiatives on the software maintenance portfolio.

Customer Queries: Too much support effort identified for a software might point to a root-cause in a previous life-cycle phase that was shortened, missing training, poor quality of a third party supplier, or not enough investment made to obtain expected stability. This measure can help maintainers identify software development weaknesses.

4 Conclusion

The subject of productivity in software maintenance can only be envisaged by organizations that have stable and mature processes. Many basic definitions of work and data collection activities are required as a pre-requisite to the measurement. A number of process improvement projects can be initiated to ensure the proper processes and information is in place to successfully conduct this and other needed measurement. Progress towards productivity analysis can only be considered once the software maintenance organization has achieved a higher level of maturity (*equivalent to SM^{mm} level 3 (April et al.)*[3]*)*.

Acknowledgement

This study could not have been done without the help of Dhiya Al-Shurougi formerly of Batelco Business Planning. Sincere thanks are given to all the software maintenance managers and third party who have contributed in the software maintenance productivity measurement project.

References

1. Abran, A; Nguyenkim, H. “Measurement of the Maintenance Process from a Demand-based Perspective,” *Journal of Software Maintenance: Research and Practice*, Vol. 5, no 2, p. 63-90, 1993.

2. Abran, A; Maya, M. “A Sizing Measure for Adaptive Maintenance Work Products,” *International Conference on Software Maintenance - ICSM-95*, Opio, France, 1995.

3. April, A; Abran, A; Dumke, R, "SM^{cmm} to Evaluate and Improve the Quality of Software Maintenance Process: Overview of the model,"SPICE 2004 Conference on Process Assessment and Improvement, Critical Software SA, The Spice User Group, Lisbon (Portugal), Apr. 27-99, 2004, pp. 19:32.

4. April, A; Bouman, J; Abran, A, Al-Shurougi D, “Software Maintenance in a Service Level Agreement: Controlling the Expectations,” *Fourth European Software Conference, FESMA*, Heidelberg, Germany, May 2001.

5. Hall, R.P, “Seven Ways to Cut Software Maintenance Costs,” *Datamation*, Barrington, Jul 15, 1987.

6. IEEE Standard for Software Maintenance, Std 1219-1998, Software Engineering Standard, Volume 2, 1999.

7. (ISBG) International Benchmarking Software group. Data Collection Questionnaire Application Software Maintenance and Support, Version 1.0, OnLine http://www.isbsg.org.au/html/index2.html

8. ISO/IEC-14764 Information technology – Software Maintenance, Standard, First edition, Nov 11, 1999.

9. Lientz, B. Swanson, E. Software Maintenance Management, Reading, Mass., Addison-Westley, 1980, 214 pp.

10. Maya M; Abran A; Bourque P; “Measuring the Size of Small Functional Enhancements to Software,” *6th International Workshop on Software Metrics*, University of Regensburg, Germany, 1996.

11. Pfleeger, S.L; Software Engineering – Theory and Practice. Prentice Hall, 1998.

12. Pigoski, T.M; Practical Software Maintenance: Best Practices for Managing your Software Investment. Wiley, 1997.

13. Pressman, R.S; Software Engineering: A Practicioner’s Approach. McGraw-Hill, fourth edition, 1997.

14. Scott, T; Farley, D; Slashing Software Maintenance Costs ; *Business Software Review;* Indianapolis; Mar 1988.

15. Sommerville, I; Software Engineering. McGraw-Hill, fifth edition, 1996.