JOURNAL OF INFORMATION SYSTEMS ENGINEERING & MANAGEMENT

Journal of Information Systems Engineering & Management, 2017, 2(4), 23
ISSN: 2468-4376

lectito

A Case Study on the Strategy of Maintaining Commercial Software with a Large Number of Users

Hellen Christine Seródio Thomazinho[1]*, Alexandre L'Erário[1], José Augusto Fabri[1]

[1] *Federal University of Technology - Parana*, BRAZIL

***Corresponding Author:** hellen.serodio@gmail.com

Citation: Thomazinho, H.C.S., L'Erário, A. and Fabri, J.A. (2017). A Case Study on the Strategy of Maintaining Commercial Software with a Large Number of Users. *Journal of Information Systems Engineering & Management*, 2(4), 23. https://doi.org/10.20897/jisem.201723

Published: August 30, 2017

ABSTRACT

Software maintenance is the activity in which changes occur in the artifacts of a software after delivery, with the purpose of keeping it available, correcting its failures, improving its performance and adapting it to new or modified requirements, according to the needs of Your users. It is an indispensable activity, without which existing systems would quickly become lagging and inefficient for organizations. The objective of this work is to verify the adherence of existing software maintenance models and processes in an organization whose product is used by a large number of users. After identifying and evaluating the problems of the company's current situation, carried out through a case study, actions are suggested to improve its software maintenance process.

Keywords: software maintenance, case study

INTRODUCTION

The software development life cycle consists of a series of phases: planning, analysis and specification of requirements, design, implementation, testing, delivery and deployment, operation and maintenance. Upon completion of its development, the software is delivered to the customer, independent of the application domain, size and complexity, will continue to evolve over time. In the scope of software changes occur when errors are corrected, when there is adaptation to a new environment, when the client requests new improvements or new functions and when the application goes through a reengineering process to provide benefits in a modern context (Sommerville, 2007).

Observing the aforementioned considerations, it becomes evident that software companies understand the need to research and relate more information about software maintenance activity. These need to highlight the main difficulties and challenges of the maintenance activity, the best practices and processes to be adopted, and the existing tools that support this maintenance activity. Second (Sommerville, 2007) points out that the success of the operations of several organizations is directly linked to the proper functioning of the systems. The lack of knowledge in models and approaches that help in the software maintenance process in organizations, makes many systems have characteristics of legacy systems, according to (Mellegard et al., 2016) and (Ahmad et al., 2016). Therefore, it is necessary to ensure the availability of systems and efficiency of care of software maintenance requests.

In this sense, this article presents a case study that aims to map the software maintenance process of a company, and verify the adherence of models, processes and practices of the software maintenance activity.

The case study was prepared by making a survey of the existing software maintenance process in the company. Subsequently, a search was made for the best practices, processes and software maintenance activities that fit the company, within the model used.

The structure of this text includes a literature review, presented in section II. A systematic mapping models in order to identify processes is shown in section III, a case study and operated is explained in section IV, the results of the analysis in Section V and in section VI findings are presented, followed by references.

THEORETICAL BACKGROUND

Software Maintenance is directed to the preservation of existing software, correcting errors and making the changes necessary to maintain reciprocity in software operating environment. Evolution, on the other hand, enhances the functionality and quality of the software. Maintenance is driven by bug reports and change requests. Evolution is driven by new requirements, functional or non-functional (Assessment et al., 2016). Both come in the form of a request for change.

Software Maintenance

The software maintenance activity is characterized by the modification of a software product already delivered to the client, to correct any errors, performance improvements, or any attribute, or to adapt that product to a modified environment. The software maintenance may be defined as a part of the software development cycle (Sommerville, 2011a; Standard, 2006). In fact, it is not uncommon for a software organization to spend 60 to 70 percent of all software maintenance resources (Pressman, 2007).

Types of Software Maintenance

According to (Rezende, 2005) "in general, all software undergoes maintenance, either for simple post-deployment adjustments, or for substantial improvements, by virtue of the legislation and, finally, because it is generating erros". In this sense, Sommerville (2011b) recalls that there are three different types of software maintenance: maintenance to repair defects in the software, maintenance to adapt the software to a different operating environment and maintenance to add functionality. A fourth category of service is presented by some authors (Pressman, 2009). These four types of software maintenance, are structured in accordance with ISO / IEC 14764 (Standard, 2006) and are described below:

- Correction Maintenance:
  - Corrective maintenance: Corrective maintenance refers to modifications required by actual errors in a software product.
  - Preventive Maintenance: They seek to identify previously possible sources of problems in the software and to correct them in advance.
- Improvements Maintenance:
  - Adaptive Maintenance: refer to tailor the software to your external environment, including changes to implement new system interface requirements, new system requirements, or new hardware requirements.
  - Perfective Maintenance: aim to add new functionality features to the software, usually due to user requests.

SYSTEMATIC MAPPING

The systematic mapping, in this work, aims to find and analyze relevant and recognized primary works in the area of software maintenance. In this sense, the purpose of this section is to identify which models, practices and/or existing software maintenance processes, in addition, we sought to identify the main tools used.

This study was based on the guidelines proposed by (Petersen et al., 2008) and was conducted by 4 steps - Definition of the research questions, String of searches, inclusion and exclusion criteria and analysis of the results.

Definition of Research Questions

We sought to answer the research questions that served as a guide in the mapping of this research, so that it was possible to conclude and explain about software maintenance in companies. The previously defined questions were:

(Q1) What are the main challenges in software maintenance?
(Q2) What models, processes or practices are adopted in software maintenance?
(Q3) What tools are there to support software maintenance?

Table 1. Search strings

Digital Library	**String**
IEEXplore[1]	((*"Abstract":maintenance* software *in companies*) AND *"Document Title":maintenance* software)
ACM Digital Library[2]	*acmdlTitle:(+maintenance+software)* AND *recordAbstract:(+maintenance +software+ companies)*
Springer[3]	*Maintenance* AND software AND *companies*

[1] http://ieeexplore.ieee.org/Xplore/home.jsp
[2] http://dl.acm.org/
[3] http://www.springer.com

Search String

To search the digital libraries, an English search string was created, formatted according to the rules of each digital library. According to (Petersen et al., 2008) primary studies are identified by using search strings in the database or manually in periodicals. **Table 1** presents the elaborated strings.

Inclusion and Exclusion Criteria

The selected studies were analyzed in order to identify those relevant to the research subject. According to (Kitchenham and Charters, 2007) the initial searches return a great amount of studies that are not relevant, do not answer the questions or even have no relation with the topic in question. Thus, the selection stage of the primary studies was divided into: Definition of Inclusion and Exclusion Criteria for primary studies and definition of the selection process.

In this study, inclusion criteria were defined as:

1. Primary studies published in conferences and periodicals that report on Software Maintenance;
2. Articles that show similar studies, only the most recent (as of 2009) should be included;
3. Articles published between 2009 and 2016;
4. Studies that present reports of experience in the software factory, or research of an experimental or theoretical nature, as long as it presents examples of application, description of experiments or real cases of use of approaches to support Software Maintenance activities.

A search was carried out in the ISO database, in order to identify standards that support the software maintenance process, the ISO / IEC 14764 standard was identified, which contains the activities and tasks necessary to modify an existing software, preserving its integrity, and all activities and tasks to be performed in a software will be the responsibility of the maintenance team (Standard, 2006).

The exclusion criteria of this study were:

1. Articles that have no relationship with Maintenance of software in companies will be excluded;
2. Studies that are not included in the context of Software Projects, Software Industry or Software Engineering;
3. Studies that clearly do not address research questions;
4. Studies that do not meet the period from 2009 to 2016.

Initially the searches returned 41 primary studies that reported the maintenance of software in companies, so the titles, abstracts and keywords of all articles were read and each one generated a list of selected studies. After the analysis of the list, the reading of the abstracts, introduction and conclusion and the application of the inclusion and exclusion criteria were carried out again, which resulted in a subset of 18 studies considering the 3 digital libraries.

Analysis of Results

This section presents the results of the analysis performed in the 18 studies, answering the research questions defined in Section A. After presenting the general data of the analysis, the answers to the research questions based on the analysis that was carried out in the systematic mapping.

(Q1) What are the main challenges in maintaining software in organizations?

According to the concluded survey, the authors (Li et al., 2010) cite the high costs as a major challenge in software maintenance. The authors (Stojanov et al., 2014) affirm that the estimation of software maintenance effort in companies is one of the main challenges, since software maintenance consumes most daily hours of work for all programmers. In addition, most software maintenance activities are related to the resolution of user requests, on the other hand the authors (Tsunoda and Ono, 2014) and (Tsunoda et al., 2015) state that work efficiency for maintenance and software support is something to be studied for companies. Still in this context the authors (Van

Der Schuur et al., 2011) state that the prioritization of software maintenance tasks as well as the implementation of processes in this activity is totally essential for software maintenance tasks.

The lack of knowledge in the corrective maintenance activity of software in companies is cited by the authors (Alaranta and Betz, 2011), and the authors (Mellegard et al., 2016) and (Ahmad et al., 2016) cite the lack of models and approaches that help the development of these software in the many of them have the characteristics of legacy systems.

The evaluation of the quality of software products, measurement of defects and approaches of identification of individual versions in software maintenance is reported by (Singh and Sangwan, 2014) as one of the challenges, since all three allow software professionals to evaluate what works and what which does not work on software identifying improvements and ease of maintenance.

For the authors (Poklemba et al., 2009), the main challenge in software maintenance is the high cost that the company has in the moment of a new development to carry out maintenance on its products.

Change the structure of the analysts participating in the software maintenance and practice of new improvements that maintenance staff is crucial in view of the authors (Ohba et al., 2007), because from this new structure will be possible to create an indicator to time the transfer of software maintenance responsibilities within the company and to consider the delivery time of system maintenance responsibilities.

The authors (Stojanov et al., 2014)consider that the evaluation of maintenance processes is a major challenge for companies, as well as improve the planning activities is essential to increase the efficiency of services provided to customers and product quality Software.

(Q2) What are the best models, processes and / or practices to be adopted in software maintenance?

For the authors (Stojanov et al., 2013) the best practice to be adopted is a model to estimate the average number of working hours for the known frequency of software maintenance requests. The authors (Poklemba et al., 2009) and (Dantas et al., 2013) believe that the best practice to adopt is to implement a process that explores what should be the correct way of software design, what are the most common mistakes, and how to win the maintenance of low cost in this software. And propose an evaluation and selection of software maintenance processes that assist in the planning of maintenance activities is approach that software companies should adopt according to the authors (Stojanov et al., 2014) and (Marques-Neto et al., 2013).

The application of agile methodology is a practice to facilitate software maintenance, as these agile methods can help speed up the process and improve code quality, bringing user satisfaction and motivation as well as communication among team members. Maintenance authors according to (Ahmad et al., 2016). Regarding the work of the software maintenance team, the authors (Van Der Schuur et al., 2011) argue that applying a history of software operations will promote consensus among employees on the priority of maintenance tasks to be developed and how many bugs need to be corrected, thereby facilitating communication between teams and prioritizing delivery dates and activities and the authors (Ohba et al., 2007) state that the creation of a structuring model for new software maintenance team, thus analyzing the time to deliver maintenance improvements and determining the optimal time to do so is essential For the adaptation and productivity of this maintenance team. In this same segment in software maintenance team improvement, the authors (Alaranta and Betz, 2011) suggest that the implementation of a theory of knowledge coordination in development groups can solve problems in the software maintenance process of companies.

Implement a framework of metrics software process-oriented aspects to predict the bugs and identify the software maintenance index is a proposal to improve according to the authors (Singh and Sangwan, 2014).

The application and development of UML diagrams in the development of software (Unified Modeling Language) according to authors (Dantas et al., 2013) and (Marques-Neto et al., 2013), would facilitate future software maintenance tasks, encouraging maintainers to maintain diagrams updated and thus help companies to analyze and verify how systems are being maintained, making it easier to reach the prospects of their end users.

The current standard that supports planning, management and execution of software maintenance activities is ISO / IEC 14764: 2006, this standard defines that a potential means of containing software maintenance costs is the use of CASE tools.

(Q3) What tools are there to support software maintenance?

With the aim of raising the critical reading of the articles presented in this paper, question 3 was elaborated to describe tools found in those articles that support processes or improvements in software maintenance in companies.

Software maintenance activities require the help of tools. The vision for CASE is an interrelated set of tools to support all aspects of software development and maintenance.

According to (Ahmad et al., 2016), Kanban has been a tool that supports software maintenance because it offers several benefits for maintenance work, such as bringing visibility to maintenance tasks, protecting teams from excessive commitment, helping to prioritize tasks , synchronization of work with other teams, still state that

Table 2. Elements identified versus current scenario

Element identified in systematic mapping	Company's current scenario
The prioritization of software maintenance tasks, implementation of processes in this activity [17].	Nonexistent. Absence of a process of maintenance of Software; Task Overload
Adoption of models and approaches that support the maintenance of software in the company [18] [4] and [5].	Nonexistent.
Greater consumption of programmers' hours in software maintenance [14].	Existing. It causes a very large cost for software maintenance activity.
Most software maintenance activities are linked to user requests [15].	Existing. Most software maintenance requests are identified by the clients / users of the systems.
The evaluation of the quality of software products, measurement of defects and approaches of identification of individual versions in software maintenance [19]	Nonexistent. Today the maintenance and performed when reported to the staff.
High cost of the company at the time of a new development to perform maintenance on their products [20].	Existing.
Evaluation and selection of software maintenance processes that assist in the planning of maintenance activities, as well as documentation [23]	Nonexistent. Earlier maintenance record too shallow. Insufficient documentation.

when using Kanban, team action for urgent work is more spontaneous, and the work can be pulled according to high priorities.

The systems used for maintenance management are commonly referenced as CMMS - Computerized Maintenance Management Systems (Wireman, 2008). In Brazil, they are simply called the Maintenance System. The CMMS are tools for companies to track equipment and inventory items, to detail when and how work orders are to be executed, to link all labor costs, materials and tools (Peters, 2015).

CASE STUDY

According to (Fonseca et al., 2002) the research allows an approximation and an understanding of the reality to investigate, as a permanently unfinished process. It proceeds through successive approximations of reality, providing subsidies for intervention in the real.

The method used for the development of this work was the case study, since it is an empirical investigation that allows the study of a contemporary phenomenon within its real life context, especially when the boundaries between the phenomenon and the context are not clearly defined (Yin et al., 2011).

The research focused on identifying how the organization performs software maintenance. The product investigated is an application with more than 7000 users allocated to 400 customers. These users are scattered from north to south of Brazil.

Company Characterization and Current Scenario

The organization studied is a medium-sized company focused on mobility systems for industries, wholesale and retail, located in the interior of the state of São Paulo. Founded in 1991, it is staffed by a team of forty-five software engineering professionals, and is looking for improvements in its software maintenance processes.

An analysis was performed in the current company scenario, and identified that software maintenance is performed only by a single sector, called the coding sector, there is no defined model for this activity. This sector does not only meet software maintenance demands, but also performs other activities relevant to the development area. In **Table 2** it is possible to analyze the challenges, processes raised in the literature and the comparison with the current scenario of software maintenance of the company.

After mapping the current scenario and analyzing all the information collected, the following problems were selected and enumerated, within the case studied:

A. Lack of a process for activities of the software maintenance team;
B. Lack of visibility of software maintenance tasks and activities;
C. Absence of planning, criteria and documentation of activities;
D. High cost in relation to hours applied in this activity;
E. Overloading tasks;
F. Failure to communicate with the user;
G. Delivery time.

The lack of a process in the software maintenance activity is one of the biggest problems (A), according to the company analyzed. The lack of a standard procedure to follow in relation to software maintenance in the company

Table 3. Identified elements

ID	Name
E1	Knowledge management
E2	Communication
E3	Process (maintenance request, after implantation)
E4	Activity Management
E5	Cost management
E6	Document Management
E7	Ticket Tools
E8	Development tools
E9	Management Tools

is one of the main challenges because the company can not identify these demands. There is an inherent concern of delivering requests to customers, as many need faster answers and solutions. Cases of emergency maintenance, when they occur, are prioritized to the detriment of others.

The second problem raised (B), results in the time wasted of the maintenance team, since there is no routine and visibility of the tasks in progress in software maintenance, nor a survey of the activities, they are considered only with resolution of Called. From the Team Coordinator perspective, it becomes difficult to get a broad view of what is being done and several bottlenecks go unnoticed.

The lack of planning, criteria and documentation of the maintenance area is another problem for the company (C). This lack of planning and criteria causes the activity to be performed informally, without considering the specific importance of each request made by the user. There is no documentation with records history of software maintenance, it is perceived that lack of documentation is something that, secondly, the company could assist in the process.

High cost in relation to the hours applied in this activity (D), for the internal control of the sector, today the professional that takes care of software maintenance requests, loses a lot of time in this activity, often due to the lack of knowledge in performing maintenance on the product.

Overloading tasks (E), in the current context the development task needs to merge with maintenance, contributing to the decrease of performance of the maintainer. This worsens when tests on a new feature are being done in parallel to some fix requested by the client.

Failure to communicate with users (F), there is a gap between users and the process, ie, the user can not see the flow of their demand, preventing better monitoring. The professionals in charge of maintenance also reported that the client does not always fully understand what he is requesting, often generating discussions between the company and the client, generating a communication problem between user and developer.

The case studied has identified that in many cases the customer believes that a certain maintenance will suffice to adapt the software to its new business need, when in fact it is not enough. This difficulty of understanding the software by the client end up generating wear situations in the relationship between company and client.

Delivery time is always a problem when there is no process to conduct maintenance. This was another factor verified, since not always the maintenance with a scheduled and informed date to the customer, can be delivered in the agreed term, usually due to rework in maintenance already done, or changes of priorities of delivery according to the customers.

In relation to the analysis of the current model of the company, it is realized that the implementation of an improvement for the software maintenance process is necessary. The next section will investigate the adherence identified in the literature compared to the needs of the company.

Analysis of the case study and adherence to the elements identified in the systematic mapping

After the case study and the systematic mapping, the following elements were identified. Table 3 shows the elements identified with the case study carried out in the company.

After defining the elements, the relationship was established with the actors (Organization, Developer, Client / Users). This adherence of the elements identified in the case study will establish the responsibilities with the actors to meet the ideal model to be implemented in the company.

It is possible to verify the relationship of the elements with the actors through Table 4.

Table 4. Elements with authors

ID	Organization	Developer	Customer / Users
E1	You should be concerned about your employees' knowledge of the product, and ensure that turnover does not compromise the product.	Should ensure the knowledge necessary to encode the maintenance / updates in the product.	Must ensure that the customer knows what you are asking.
E2	Alignment between contract, client and developers.	Alignment between developers to analyze customer contracts.	Alignment between developers and customers. Common means of request.
E3	Define the whole software maintenance process in the company.	Knowledge of the process applied by the company and execution of the same.	Keep up the current process your request.
E4	It should ensure the management of all activities related to software maintenance.	Must be aware of all activities.	Not applicable
E5	The estimated costs related to hours worked in software maintenance / correction should be performed.	Have the necessary knowledge to evaluate the estimated hours to perform certain maintenance.	You should receive the estimated hours / costs of your request.
E6	It should be ensured that all documentation related to the development and maintenance is carried out.	All changes must be recorded in the documentation.	Not applicable
E7	One should ensure a tool for managing HelpDesk calls.	Must have knowledge to analyze calls and report each step.	You should see the entire flow of your request.
E8	Ensure the help of development tools that facilitate the code for later maintenance.	Knowledge of using and supporting this tool for development.	Not applicable
E9	Ensure the assistance of management tools such as CMMS for software maintenance activities.	Knowledge of using and supporting these tools for their activities.	Not applicable

ANALYSIS OF RESULTS

After cross-identification of the elements, shown in Table 3, this section presents the adherence of the models found in the literature through the systematic mapping in the elements shown in Table 2. This work does not contemplate the implantation, that is, they are evaluated if the models / process and practices that are adherent, but are not mitigated techniques of implementation of these.

ISO / IEC 14764: 2006 complies with the following elements E1, E2, E3, E4, E5 and E6. The element E1 is partially attended, because for the implantation of this model the team must know the products and the types of maintenances that must be realized in these. The E2 element is met by the standard, since the maintenance process of software in a company is initiated through a request, another factor that attends the communication is the fact that in the implementation stage of the execution of the process the maintenance team must perform a Communicate with the customer and maintain feedback. The E3 Element is one of Norma's main goals, as it relies on all software maintenance processes, making it a complete flow for service in that industry. Through the standard it is also possible to manage maintenance activities, where all activities have inputs, controls and outputs, so this standard will meet element E4.

The controls provide guidance to ensure that maintenance activity produces correct outputs. In relation to the costs raised through Element 5, the standard makes a control for cost estimation before executing the maintenance process, that is, since it is only an estimate, it partially meets this element. Documentation management is one of the main activities within each stage of the standard because every change in the system or development requirements must be recorded and maintained a history of such documentation.

The elements E7, E8 and E9 are not met by the standard, since the Norma does not offer tools to aid in software development, HelpDesk and management. According to reports in the literature it is possible to raise the use of tools that assist companies in the development, management and support. According to (Tsunoda and Ono, 2014) and (Poklemba et al., 2009), the aid of a tool for system maintenance and support is essential for a construction of a dataset so that it can analyze through the maintenance phase planning reports and project support from software companies.

CONCLUSIONS

This paper presents the case study of an organization, whose problems in software maintenance are aggravated by the lack of a process for such activities and tasks within a business context. It was evidenced that such problems directly interfere in the software maintenance activity, generating lack of planning, high cost, overloading of tasks, and failure to communicate with the users of these systems. In addition, a systematic mapping was carried out to show through the literature reports processes and activities that support the maintenance of software in companies.

With this study, it was possible to perceive the flaws in the software maintenance process of the company, since it also made it possible to visualize what needs to be improved so that this maintenance is carried out successfully and that the product is delivered to the highest quality possible.

After the analysis presented in Section V (results analysis), we conclude that no process, a practice identified in the literature through systematic mapping, has fully addressed the problem reported in the case study, and it is necessary to create a model or strategy that can help Company in this area.

FUTURE WORKS

It is suggested as future work the elaboration of a strategy of improvement in the Company, as well as an elaboration with the employees and managers, of a plan for non-invasive implantation. The carrying out of further case studies with the purpose of further contributing to the enrichment of the software maintenance process.

REFERENCES

Ahmad, M.O., Kuvaja, P., Oivo, M. and Markkula, J. (2016). Transition of Software Maintenance Teams from Scrum to Kanban. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016–March(i), pp. 5427–5436. https://doi.org/10.1109/HICSS.2016.670

Alaranta, M. and Betz, S. (2011). Knowledge Problems in Corrective Software Maintenance - A Case Study. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 3746–3755. https://doi.org/10.1109/HICSS.2012.406

Assessment, M.P., Prentner, W. and Sneed, H.M. (2016). Analyzing Data on Software Evolution Processes. , pp. 1–10. https://doi.org/10.1109/IWSM-Mensura.2016.18

Dantas, F., Garcia, A., Whittle, J. and Araújo, J. (2013). Enhancing Design Models with Composition Properties: A Software Maintenance Study. *AOSD 2013 - Proceedings of the 2013 ACM on Aspect-Oriented Software Development*, pp. 49–60. https://doi.org/10.1145/2451436.2451444

Fonseca, J.J.S., Danton, G., Freitas, C.C., … Wollmann, J. (2002). Metodologia Do Trabalho Científico. *São Carlos: Serviço de Biblioteca E Informação …*, pp. 1–48.

Kitchenham, B. and Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering. *Engineering*, 2, p. 1051. https://doi.org/10.1145/1134285.1134500

Li, J., Stålhane, T., Kristiansen, J.M.W. and Conradi, R. (2010). Cost Drivers of Software Corrective Maitenance: An Empirical Study in Two Companies. In *IEEE International Conference on Software Maintenance, ICSM*. https://doi.org/10.1109/ICSM.2010.5609538

Marques-Neto, H., Aparecido, G.J. and Valente, M.T. (2013). A Quantitative Approach for Evaluating Software Maintenance Services. *Proceedings of the ACM Symposium on Applied Computing*, pp. 1068–1073. https://doi.org/10.1145/2480362.2480565

Mellegard, N., Ferwerda, A., Lind, K., Heldal, R. and Chaudron, M.R. V. (2016). Impact of Introducing Domain-Specific Modelling in Software Maintenance: An Industrial Case Study. *IEEE Transactions on Software Engineering*, 42(3), pp. 248–263. https://doi.org/10.1109/TSE.2015.2479221

Ohba, M., Nakamura, Y., Oomiya, N., Yamamoto, H. and Maruyama, Y. (2007). Development of the Simulation for Taking Over Process in the Software Maintenance. *40th International Conference on Computers and Industrial Engineering: Soft Computing Techniques for Advanced Manufacturing and Service Systems, CIE40 2010*, pp. 0–5.

Peters, R.W. (2015). Your CMMS-EAM System. In *Reliable Maintenance Planning, Estimating, and Scheduling*. pp. 99–143. https://doi.org/10.1016/B978-0-12-397042-8.00008-5

Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. *EASE'08 Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 68–77. https://doi.org/10.1142/S0218194007003112

Poklemba, T., Sivy, I. and Havlice, Z. (2009). Maintenance Software Processes for Web 2.0 Based Learning Management Systems. *Emerging eLearning Technologies and Applications (ICETA), 2011 9th International Conference on*, pp. 167–170. https://doi.org/10.1109/ICETA.2011.6112608

Pressman, R.S. (2007). Engenharia de Software - Uma Abordagem Profissional. *Development*, pp. 42–60. https://doi.org/9788563308337

Pressman, R.S. (2009). *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. https://doi.org/10.1017/CBO9781107415324.004

Rezende, D.A. (2005). *Engenharia de Software E Sistemas de Informaça o*. São Paulo: Brasport.

Van Der Schuur, H., Jansen, S. and Brinkkemper, S. (2011). Sending out a Software Operation Summary: Leveraging Software Operation Knowledge for Prioritization of Maintenance Tasks. *Proceedings - Joint Conference of the 21st International Workshop on Software Measurement, IWSM 2011 and the 6th International Conference on Software Process and Product Measurement, MENSURA 2011*, (section VII), pp. 160–169. https://doi.org/10.1109/IWSM-MENSURA.2011.14

Singh, P.K. and Sangwan, O.P. (2014). A Process Metrics Based Framework for Aspect Oriented Software to Predict Software Bugs and Maintenance. *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, pp. 831–836. https://doi.org/10.1109/CONFLUENCE.2014.6949323

Sommerville, I. (2007). *Software Engineering*. Harlow, UK: Pearson Education Limited.

Sommerville, I. (2011a). *Engenharia de Software*. Brazil: Pearson.

Sommerville, I. (2011b). *Engenharia de Software*, São Paulo: Pearson Brasil.

Standard, I. (2006). *INTERNATIONAL STANDARD ISO / IEC Software Engineering — Software Life*. 2nd ed., Vol. 2006.

Stojanov, Z., Brtka, V. and Dobrilovic, D. (2014). Evaluating Software Maintenance Processes in Small Software Company Based on Fuzzy Screening. , pp. 67–72. https://doi.org/10.1109/SACI.2014.6840037

Stojanov, Z., Dobrilovic, D., Stojanov, J. and Jevtic, V. (2013). Context Dependent Maintenance Effort Estimation: Case Study in a Small Software Company. *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 461–466. https://doi.org/10.1109/SACI.2013.6609019

Tsunoda, M., Monden, A., Matsumoto, K., Ohiwa, S. and Oshino, T. (2015). Benchmarking Software Maintenance Based on Working Time. *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pp. 20–27. https://doi.org/10.1109/ACIT-CSI.2015.13

Tsunoda, M. and Ono, K. (2014). Pitfalls of Analyzing a Cross-Company Dataset of Software Maintenance and Support. *2014 IEEE/ACIS 15th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2014 - Proceedings*. https://doi.org/10.1109/SNPD.2014.6888729

Wireman, T. (2008). *Maintenance Work Management Processes*. 1st ed. Industrial Press.

Yin, R.K., Grassi, D., Ph, D., … Yang, J.-I. (2011). Estudo de Caso Planejamento E Métodos. *Journal of Food Biochemistry*, 35(3), pp. 715–734. https://doi.org/10.1111/j.1745-4514.2010.00412.x