GAIA Estimation: A Framework based on Maturity Levels and Services for Assessment and Application of the Software Estimation Process

Bruno Omena Mesquita and Rodolfo Miranda de Barros
Computer Science Department
State University of Londrina
Londrina, Brazil
bruno.omena@gmail.com, rodolfo@uel.br

ABSTRACT

With the current growth of software development, plan and manage well the development practices become vital for effective coordination of a software project. Within this context, the estimation process presents as basis for achieving a good planning and control of these projects. So, more accurate estimates mean a better planning, and with this the project will take less risks. On the other hand, bad or inaccurate estimates increase the risks which the planning takes on and this may lead to large losses for the project or until its cancellation. However, there are few sources which provide guidelines on how to perform and manage the estimation process, further hindering their use. Then, in order to fill this gap, this work proposes a Framework called GAIA Estimation aiming support the management of the software estimation process through maturity levels and services.Thus establishing a gradually and incrementally way to implement and assess the software estimation practices inside the organizations.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Management—*cost estimation, maturity model*

General Terms

Framework

Keywords

Estimation Process, Services, Maturity Levels, Assessment Questionnaire

1. INTRODUCTION

One of the big challenges of software engineering is accurately estimate the effort, cost and schedule required to develop a software [16]. Accurate estimates are important in many areas of development, e.g. they are the main input

for planning and project control, for all decisions concerning the budget, for bidding, among others [13].

According to the Chaos Report [8], unrealistic estimates is one of the most frequent causes of failure in software projects. The root of the problem is that many projects do not meet the estimated deadlines and break the budget. During the years several studies has been done to try remedy this problem. Therewith many methodologies and techniques have been developed, refined and combined for this purpose [11].

And even with this amount of studies still does not exist one unanimous model, because each of the techniques and methodologies meets a certain kind of problem or are based on empirical research [11]. Allied with others factor like the difficulty of understanding of the techniques and apply them, bias and lack of a standardized procedure to perform the estimates, several of those techniques and methodologies end up staying unused by industry.

This also occurs because many of those techniques and methodologies do not bring a way which shows how to implement them in practice and how integrate them to the development environment. Moreover, most of those techniques and methodologies do not consider the maturity of the organization which aspires deploy them and this may cause losses to these organizations due the lack of preparation to perform the required tasks.

Thus, in this paper, we propose a framework to help the organizations to apply and manage effectively the techniques and methodologies needed for deploy a software estimation process inside these organizations. This framework is compound by services, maturity levels and one assessment questionnaire of the estimates practices. Through it we present a way to implement and assess the estimation practices of the organizations, aiming offer an adaptable form for accomplish this task.

This work is divided as follows: section 2 brings a brief theory review of the estimation process, models based on maturity levels and related work, respectively. In section 3, the proposed framework is presented, section 4 contains a case study of the application of the framework and finally section 5 contains the conclusions and future work.

2. THEORY

2.1 Estimation Process

Estimate the software development is a continuous process and should be performed throughout the whole project life cycle [1, 17]. This process usually consists of some phases as: estimate size, effort, cost, schedule, resources, assess risks, verify and validate the estimates, track and re-estimate, measure and improve the process. The common organization of these steps can be seen in Figure 1.
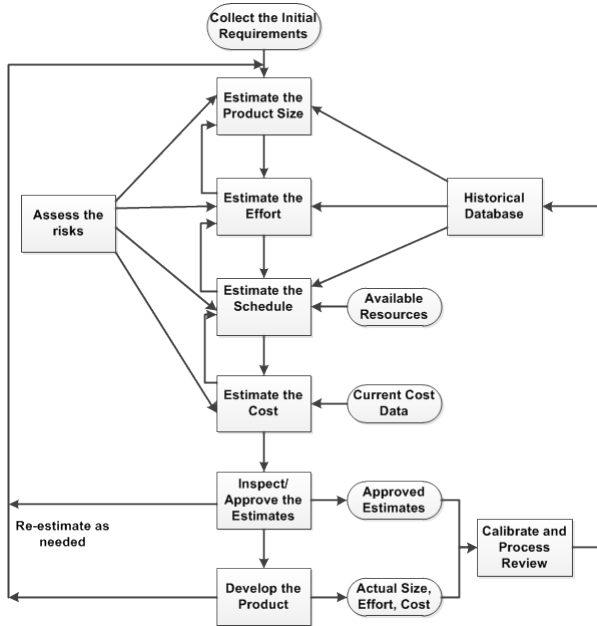

Figure 1: Project Estimation Process. Adapted from [1].

In an estimation process inputs such as project scope, priorities and constraints are considered. Data from past projects are also used as input to make estimates and calibrate models [15]. The step of re-estimate is primordial to update the estimates when new project artefacts are available or when any restriction is added, changed or deleted from the project.

2.2 Maturity Models

Maturity models seek to establish levels of processes evolution, called maturity levels which characterize stages of improvement in process implementation within the organization [19]. These maturity levels, in turn, indicate the company profile and ways to improve the process in question.

Maturity Models see being used with successfully for establishment and assessment of several processes [5, 6, 18, 19, 9]. The advantage of using maturity levels is which the process can be implemented in a gradual and incremental form. Suiting of the reality of each organization and is not required to meet all levels of these models whether the organizations objectives were achieved with the implementation of a certain level.

2.3 Related Work

The software estimation area has received much attention in recent decades, all are seeking develop better techniques and models to fulfill this task. And as the failure in estimate is one of the major reasons of software projects do not reach their goals [8]. Many models and frameworks have been and are being proposed to solve this problem.

One of the most known and documented of these methods is the Constructive Cost Model (COCOMO), proposed by Barry Boehm [3]. It allows organizations to estimate their cost, effort and schedule for a new software. The initial version of COCOMO (also called COCOMO 81) was developed by Barry Boehm and published in the book Software Engineering Economics in 1981. This model uses basic rules of linear regression to predict the effort, schedule and cost based on historical data.

In the first version of COCOMO, Barry Boehm examined 63 software projects in order to find the influence of the effort and development time on these projects. Boehm found out which software size is the main factor which influence the effort. Others influencing factors were the type of project which will be developed, the skills of the developers and the performance characteristics.

However, in the 90s, the COCOMO encountered difficulties in estimates the costs of new projects because of the new paradigms, like the new life cycle non-sequential, agile models, reuse, and object-oriented approaches [4]. So to fix this deficiency, the COCOMO II was released by Boehm et al. [4]. The basic concepts of COCOMO and COCOMO II are pretty the same. The COCOMO II was developed in the 90s and calibrated using a data set of 161 projects.

Over the years, many models and frameworks were developed and each year many others arise, so it is difficult to know which is more efficient and which to use for each environment. Thinking about it, Menzies et al. proposed a tool for assess estimation models using data mining, this tool is called COSEEKMO [14]. This evaluation is done by generating numerous effort models of a specific data set using a variety of techniques (e.g. Local calibration, linear regression, and an algorithm called a wrapper). Each technique is chosen by its potential to improve the prediction model and reduce deviations. The effort models found by COSEEKMO are assessed through rejection rules which slaughter the weaker models.

In 2010, Li and Keung, developed a framework for estimate the costs of systems of service-oriented architecture (SOA) using the divide and conquer approach [12]. Within this framework services are classified into three primitives and a combined type according to the development process. The estimate of each primitive type is seen as a sub problem. The cost and effort are calculated gradually according with reverse integration of the services division.

Another framework was presented by Ahmed et. al [2], to make estimates especially in the early stages of the project. In their framework they developed a system which generates probabilistic values for size and these are used as a source for predicting the effort. The information used is based

in conceptual models UML (Unified Modeling Language), generally created at the beginning of the software life cycle. The framework deals with the uncertainty in predicting the size and effort providing an estimate as a probability density function instead of an exact value.

3. GAIA ESTIMATION FRAMEWORK

As can be seen in the previous sections, the use of models and frameworks to make estimates is well discussed in the literature. But few of these studies support the estimation process, and those which do it is only partially. There is not a explicit form for gradual implementation contained in the models / frameworks, which makes difficult the use of these in environments little suited to make estimates. Furthermore, these frameworks do not address the external factors of the estimates, such as the historical database, inspection, measurement and assess the risks and all of these factors have strong influence on the estimates [1, 4, 13].

Compared with other, for example, we can mention the COCOMO, it is a good and widespread method to estimate, but requires trained and experienced people in estimation, plus a good database of past projects to calibrate its parameters effectively. And these requirements are hardly met with success by organizations with low maturity in estimating.

Have techniques as presented by Li and Keung [12] or by Ahmed et .al [2] at first moment can present as good choices but both require further knowledge in other areas, which may not be easily accessible in organizations with low experience. Techniques which require less technical knowledge and a better understanding of the organization operation are more appropriate for such cases, like expert judgement. However, this techniques are very dependent of expert people of the organization and if they are not available the estimate may be compromised withal they are very susceptible to bias.

Considering this, the GAIA Estimation framework presents an adaptive and gradual way to increase the accuracy and reliability of the estimates through maturity levels and services. Where it will indicate which of the most appropriate technique for each organization depending on their maturity in estimation and how and when make the transition / addition of other techniques and structures. Besides, it shows how to implement the techniques and models in practice, fact the most of techniques and frameworks do not bother to make it clear.

The Framework is composed of five maturity levels and an assessment questionnaire. Its purpose is assist the implementation and assessment of the software estimation practices inside the development organizations. The Framework uses as basis the estimation process illustrated in the Figure 1, since the use of a standardized process will serve like a guide to perform the estimates, besides it will mitigate the communication issues, rework and ambiguities helping organizations which do not have practice in estimate. Moreover, this process contains basically all steps needed to make a good estimation [1, 17].

3.1 Maturity Levels

The maturity levels are formed by services which aim to help the execution of each step of the process. The obtaining of services occurred, in turn, by fragmentation of the process of Figure 1 and the ten steps listed by Galorath and Evans [7] for the execution of the estimation process. The organization of services within the maturity levels and the establishment of them follow the standard of the CMMI [18] and the activities contained in the process of Figure 1. As we do not found in the literature any approach which utilizes maturity levels for this purpose, we chose to follow the CMMI standard, because it is one of most successful and widespread maturity model in the world. The Figure 2 shows the organization of the Maturity Levels and their Services.

Level 0	Level 1	Level 2	Level 3	Level 4
	Historical Database ◇	Level 1 +	Level 2 +	Level 3 +
	Heuristic Approach to Estimate ◇	Formal Sizing ◇	Parametric Models ◇	Statistical Analysis of Estimates ◇
		Derivation of Simple Models ◇	Risk Assess ◇	Lessons Learned and Process Review ◇
		Comparison process ◇	Verification and Validation ◇	
		Measurement and Analysis ◇		

Figure 2: GAIA Estimation Maturity Levels and their Services.

As can be seen in Figure 2, the maturity model contains five levels and eleven services, starting at level 0 and ending in level 4. Each level has a special focus on the process which can be entirely or partly implemented, as needed. Each of these levels deploy their own services and the services of the previous levels, as presented by Figure 2. The choice of using services as components of maturity levels follows the proposal of Gaffo and Barros [6].The focus and the characteristics of each level are described as follow:

Level 0 (Not Implemented): The level 0 represents organizations which do not perform estimates in its projects or perform in a completely intuitive way. This level is necessary for any organization can be allocated at least some maturity level. This level does not have services.

Level 1 (Known): This level provides two services : Historical Database and Heuristic Approach to Estimate, aiming to help organizations with low maturity in estimating. This services are very important to create a solid basis for support the next levels. Here the estimation process is already known, but is not executed with all its components, and the amount of projects information in the organization is small.

Level 2 (Performed): At level 2 the estimation process is already known and implemented, with most of its steps, within the organization. In this level the organization should have a better knowledge of their own software develop process, so the organization will have more familiarity and easiness to implement services which depend on other areas. The services of this level are: Formal Sizing, Derivation of Simple Models, Comparison Process and Measurement and

Analysis.

Level 3 (Defined): Here the process is executed completely and is standardized within the organization. The organization already has the basis to implement all steps of the process and manage it. At this level the boundaries of the process are clear and there is a good basis of data collected from past projects. The estimation process must be part of the software develop process. The services of this level are: Parametric Models, Risk Assess and Verification and Validation.

Level 4 (Improvement): In this level the process is reviewed periodically for identify possible areas for improvement as well as bottlenecks in implementation. The organization is already mature enough and able to control the execution of the process statistically, knowing the areas of higher productivity, error rates, variance and so on. The services of this level are: Statistical Analysis of Estimates and Lessons Learned and Process Review.

3.2 Services
The services of the maturity model aims to support the implementation of one or more stages of the estimation process, thus enabling them to achieve their goals. The evolution of each service aims to add value to the process, making it more consistent and reliable over the levels. That depends on the maturity of the organization to perform certain level. As explained in the previously the service area, i.e. its structure, was based on the work of Gaffo and Barros [6].

Each service has five areas which organize information relevant to its execution, and these can be customized according to the need of the project, client and / or organization. The following is given a brief description of each of the five information areas of the services.

Tools and Techniques: Comprise tools and methodologies which enable and help apply and incorporate the role of the service.

Templates: Consist of models of documents with the purpose of standardizing the records, identify and determine the artefacts generated and consumed during their deployment.

Indicators: Define metrics to measure the evolution of the service and encourage continuous improvement.

Workflows: Standard flow to manage the implementation of the activities of the service.

Vocabulary: Terminologies, nomenclatures and acronyms of common activities of each service.

Figure 3 shows the basic structure of the service verification and validation.

3.3 Assessment Questionnaire
The assessment questionnaire aims to assess the current implementation of estimation process, inside the levels and services defined in the maturity model. It determines the
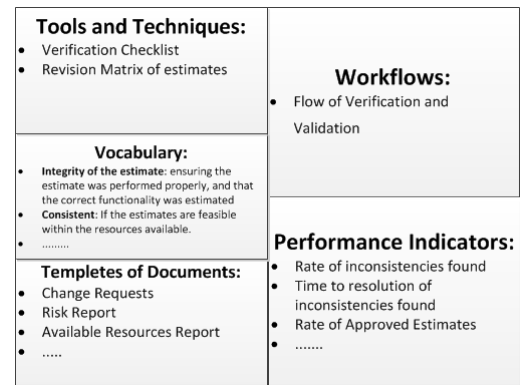

Figure 3: Structure of Service of Verification and Validation.

rank values to the services, the criteria for determining the maturity level of the organization and provides an evaluation process. This part is based on ISO / IEC 15504 - Part 2: Process Assessment [10].

One level of maturity is reached when services of the current level are fully or largely achieved and the services of previous levels are fully achieved. The questionnaire contains a set of questions for each attribute specified in the maturity model. The answer to each question can be simply "yes" or "no". If the answer is something between "yes" and "no" like "partially" the answer must be "no". Because the degree of partially can be variable and this may be lead to erroneous interpretations and wrong assumptions.

In the questionnaire, if 0-15% of the answers are "yes", the service was not achieved, 16-50% of the answers are "yes", the service is partially achieved, 51-85% of the answers are "yes", the service is largely achieved, and 86-100% of the answers are "yes", then the service is fully achieved. If a question cannot be answered, since it is not applicable to the context, this issue is not taken into account. These thresholds are proposed on ISO / IEC 15504 and all questions are of equal importance. The questionnaire has 80 questions distributed as follow: 0 questions for level 0, 18 questions for level 1, 28 questions for level 2, 20 questions for level 3 and 14 questions for level 4. Table 1 brings the example of the questions of the questionnaire to assess the service of Historical Database.

4. STUDY CASE
As a study case, the assessment questionnaire was applied in two software development companies, one from academia and one from the private sector. Organization A called GAIA, develops software aimed to Web, it is composed by students of undergraduate and postgraduate course in computer science from the State University of Londrina. Organization B is a privately held company which develops software aimed to the public sector, they have a staff of 31 employees, for reasons of confidentiality the name of organization B will not be published.

In this study case we applied the questionnaire for that, in first moment, assess the maturity of organizations in

Table 1: Questions to assess the service of Historical Database

Service: Historical Database
1. Is there any mechanism for storing data within the organization?
1.1. Is it based on database?
1.2. Are there mechanisms for maintaining the database?
2. Is there a process of acquiring information?
2.1. Is this process standardized?
2.2. Are there forms for data acquisition?
3. Is there a process for data collection?
3.1. Is this process standardized?
3.2. Are the types of data defined?
3.3. **Are those responsible for the collection identified?**

relation to their estimates practices. A level is considered completed whether the services which compose are, at least, largely achieved and the services of the lower levels are fully achieved. It is not recommended the deployment of services which belong to two ahead levels of the current level, because there may be no technical basis for its implementation effectively. The questionnaire was answered in both organizations by the project manager.

4.1 Results

The results of the questionnaire are summarized in Table 2, Table 3 and Figure 4. The tables 2 and 3 show the result of application of assessment questionnaire. The table 2 presents the results on the rank reached by each service and table 3 presents the result by percentage of answers "yes". On table 3 in parentheses are the number of questions answered "yes" of the total questions of each service. And the Figure 4 shows one graphic comparing the results of the two organizations, in this graphic on the left side are the thresholds of the questionnaire and on the right side the acronym of each service with the level which it belong in parentheses.

Table 2: Results of the Application of Assessment Questionnaire

Levels	Services	Organizations	
		A	B
Level 1	Historical Database	Fully achieved	Partially
	Heuristic Approach to Estimate	Fully achieved	Largely
Level 2	Formal Sizing	Fully achieved	Largely
	Derivation of Simple Models	Largely	Partially
	Comparison Process	Not achieved	Not achieved
	Measurement and Analysis	Largely	Not achieved
Level 3	Parametric Models	Not achieved	Partially
	Assess Risk	Largely	Not achieved
	Verification and Validation	Not achieved	Not achieved
Level 4	Statistical Analysis of Estimates	Not achieved	Not achieved
	Lessons Learned and Process Review	Partially	Not achieved

4.2 Analysis

Analysing the results contained in Table 2 and Table 3, it can be concluded which Organization A is at level 1, since it fully implements all services on the level 1 of the model and

Table 3: Results of the Application of Assessment Questionnaire, by percentage of "yes".

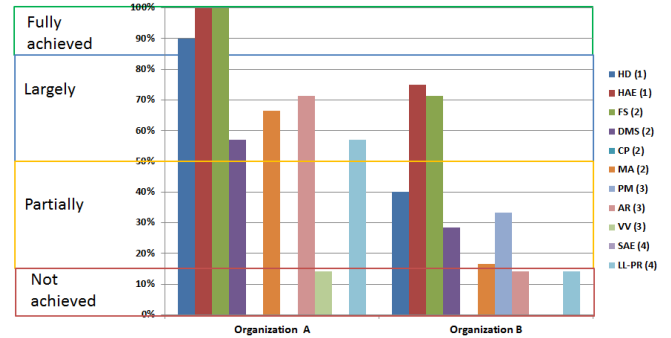| Levels	Services	Organizations	
		A	B
Level 1	Historical Database	90 % (9/10)	60% (6/10)
	Heuristic Approach to Estimate	100% (8/8)	75% (6/8)
Level 2	Formal Sizing	100% (7/7)	71,42% (5/7)
	Derivation of Simple Models	57,14% (4/7)	28,57% (2/7)
	Comparison Process	0% (0/8)	0% (0/8)
	Measurement and Analysis	66,6% (4/6)	16,6% (2/6)
Level 3	Parametric Models	0% (0/6)	33,3% (2/6)
	Assess Risk	71,42% (5/7)	14,28% (2/7)
	Verification and Validation	14,28% (1/7)	0% (0/7)
Level 4	Statistical Analysis of Estimates	0% (0/7)	0% (0/7)
	Lessons Learned and Process Review	57,14% (4/7)	14,28 (1/7)

Figure 4: Comparison of positive answers between Organization A and B.

despite the Organization A implements almost all services at level 2 fully or largely it does not implement the service of Comparison Process. At level 3 implements largely a service, but does not implement other two requirements, therefore, did not reach this level. The Organization B is at level 0 of the model because it does not implement, at least largely, all services from level 1.

Looking into the results, can be seen which organization A is more structured than organization B about their estimation practices. The organization A has deployed key services like Historical database and Formal sizing and these services are very important for evolution of the framework and consecutively to get more accurate estimates since they are used by almost entire estimation process. However, the Organization B does not have a easy way to evolution its estimation practices. Because as the results show, the organization B is in level 0, since it just implement some services partially of several levels and no set of services are largely or fully implemented at same level. This shows the immaturity of the organization B, because it implements partially many parts without know how to use them in together or what prioritize first and this probably will cause inaccurate results.

Also was observed with the outcome of the interview in organization B, which it implements the service partially of parametric models. But as shown in Table 1 service Historical Database is not largely implemented. This can impair service tasks of parametric models because there is

no basis for this calibration, then there is not sufficient and structured information of past projects, which is a factor essential for the effectiveness of this kind of technique. Thus was considered the Organization B is not mature enough to implement this service, which can generate too much effort to run it, inconsistent and inaccurate results.

5. CONCLUSIONS

As explained in this paper, the process of software estimation is a great help to organize and apply techniques to get accurately estimates in software projects, thereby contributing to its success. But there are few sources which guide how to apply this process in practice. Thus, was considered the proposed model achieved his goal to filling this gap, establishing a clear, effective and gradual way to implementation and assessment, via maturity levels, services and a assessment questionnaire.

By the study case, we conclude that the framework can assess the current estimation practices in an organization and position it within the levels of the proposed maturity model.Besides be able to identify practices that should not be implemented because there is not enough maturity of the organization to perform this task and identify potential areas for improvements.

The study was limited in the first instance to position the organizations within the model and verify if their estimation practices are consistent with their reality. But we need further investigation about deployment and maintenance of a service, considering the effort of its implementation. So as future work we intend besides positioning organizations within the model also deploy the services and observe their behaviour. The data about the behaviour of services deployment also will serve as feedback for improving the model.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] R. Agarwal, M. K. Yogesh, S. Mallick, R. M. Bharadwaj, and D. Anantwar. Estimating software projects. *ACM SIGSOFT Software Engineering Notes*, 26(4):60–67, 2001.

[2] M. A. Ahmed, I. Ahmad, and J. S. AlGhamdi. Probabilistic size proxy for software effort prediction: A framework. *Information and Software Technology*, 55(2):241 – 251, 2013.

[3] B. W. Boehm. *Software Engineering Economics*. Prentice Hall, Englewood Cliffs, 1 edition, 1981.

[4] B. W. Boehm, Clark, Horowitz, Brown, Reifer, Chulani, R. Madachy, and B. Steece. *Software Cost Estimation with Cocomo II with Cdrom*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[5] M. Diaz-Ley, F. Garcia, and M. Piattini. Mis-pyme software measurement capability maturity model supporting the definition of software measurement programs and capability determination. *Advances in Engineering Software*, 41(10-11):1223–1237, 2010.

[6] F. Gaffo and R. de Barros. Gaia risks - a service-based framework to manage project risks. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–10, 2012.

[7] D. D. Galorath and M. W. Evans. *Software Sizing, Estimation, and Risk Management: When Performance is Measured Performance Improve*. Auerbach Publications, New York, 2006.

[8] S. Group. *The CHAOS Manifesto: Think Big, Act Small*. Standish Group International, 2013.

[9] F. E. A. HORITA and R. M. BARROS. Gaia human resources: An approach to integrate itil and maturity levels focused on improve the human resource management on software development. In *International Conference on Computer Applications in Industry and Engineering*, 2012.

[10] ISO/IEC. *ISO/IEC 15504-2:2003. Process assessment - part 2: performing an assessment*. 2003.

[11] M. Jorgensen and M. Shepperd. A systematic review of software development cost estimation studies. *Software Engineering, IEEE Transactions on*, 33(1):33–53, 2007.

[12] Z. Li and J. Keung. Software cost estimation framework for service-oriented architecture systems using divide-and-conquer approach. In *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, pages 47–54, 2010.

[13] S. MaConnell. *Software Estimation: Demystifying the Black Art*. Microsoft Press, Washington, USA, 2006.

[14] T. Menzies, Z. Chen, J. Hihn, and K. Lum. Selecting best practices for effort estimation. *Software Engineering, IEEE Transactions on*, 32(11):883–895, 2006.

[15] N. Mittas and L. Angelis. Visual comparison of software cost estimation models by regression error characteristic analysis. *Journal of Systems and Software*, 83(4):621 – 637, 2010.

[16] K. Molokken and M. Jorgensen. A review of software surveys on software effort estimation. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 223–230, 2003.

[17] C. E. L. Peixoto, J. L. N. Audy, and R. Prikladnicki. The importance of the use of an estimation process. In *Proceedings of the 2010 ICSE Workshop on Software Development Governance*, SDG '10, pages 13–17, New York, NY, USA, 2010. ACM.

[18] SEI. *Capability Maturity Model Integration (CMMI-DEV v1.3): Improving processes for developing better products and services*. Software Engineering Institute, Massachusetts, 1 edition.

[19] SOFTEX. *Guia Geral MPS de Software (MR-MPS-SW)*. SOFTEX, Brasilia, 2012.