

**Technical Report
ES-XXX/08
Experimental Software Engineering Group**

A System Dynamics Model based on Cause and Effect Diagram to Observe Object-Oriented Software Decay

Marco Antônio Pereira Araújo
Guilherme Horta Travassos

Systems Engineering and Computer Science Program

Federal University of Rio de Janeiro - Brazil

COPPE/UFRJ

<http://www.cos.ufrj.br>

May/2008

1 Introduction

Due to the ever growing necessities of evolution in Software Systems, it is necessary to advance towards the understanding on how these systems suffer changes, so as to be more easily modified to accommodate these needs, throughout successive cycles of evolutive maintenance. In this fashion, an environment for experimental studies on software evolution is underway. This environment explores simulation techniques based on System Dynamics (Madachy, 2008) (Barros, 2001) models, to observe the decay of object oriented software. The Laws of Software Evolution (LSE) (Lehman, 1980) (Lehman and Ramil, 2003) have influence the building of the model. These Laws describe how a system behaves through its versions, regarding its decaying. The application of System Dynamics techniques is justified by the fact that evolving systems present a dynamic behavior, non-static, which needs to be considered. However, the complexity of identifying and defining the process to regulate the behavior, justify the use of semi-quantitative analysis. The semi-quantitative analysis (Camiletti and Ferracioli, 2002) offers the possibility of data analysis through trends of pre-determined software characteristics, providing for the observation of the behavior of evolving systems, as proposed in the environment presented in this work.

2 Building a System Dynamics Model to Observe Software Decay

Most of the scientific researches begin with a literature review, usually executed by using an ad-hoc approach. However, if this review is not complete and fair, it won't be scientifically valid. That's the reason why we should consider the using of a systematic review. A systematic review of the literature is one of the existing means of identifying, evaluating and interpreting all the pertaining research to a particular research question (Biolchini et al., 2005). Besides that, there are more specific reasons that could justify the systematic review usage (Kitchenham, 2004):

- To summarize some existing evidence on a determined theory or technology, for instance;
- To identify open points for the research in question, allowing for a definition of areas where more investigation must be carried out;
- To provide the basis for new research activities.

In the context of this work, the main objective on the execution of the systematic review is to provide a formal basis for the study of evolutive maintenance and decay of object oriented software. In the view of it, it has been elaborated a review protocol to guide the execution of the literature review, which is going to be presented in the up coming sections.

2.1 Systematic Review

To undertake a systematic review whose purpose is to investigate the influences between software characteristics (Size, Periodicity, Complexity, Effort, Modularity, Reliability and Maintainability) identified in (Araújo et al., 2005). The analysis of the influences between software characteristics must be executed for each one of the phases of an object oriented software development process, (normally encompassing the phases of Requirements Specification, High Level Design, Low Level Design and Coding).

In the context of this work, the software characteristics are related to the Law of Software Evolution - LSE and interpreted as follow (Araújo et al., 2005):

- Size: the amount of artifacts¹ produced in each phase of the proposed software development process;
- Periodicity: the interval of time elapsed between each produced version of each artifact;
- Complexity: the elements that can measure the structural complexity of the artifact;
- Effort: the amount of interventions in the artifacts (number of inclusions, modifications and exclusions in each artifact), the amount of people and allocated resources, spent time and average productivity of the team, by version of each artifact;
- Modularity: the coupling and cohesion characteristics in artifacts as, for instance, coupling between Use Cases in Requirements Specification, between classes in High and Low Level Design and cohesion in methods in Coding;
- Reliability: the amount of identified defects by artifact in each version of it, besides system's availability;
- Maintainability: the spent time in the identification of defects and the spent time in their removal.

Table 1 presents suggestions for possible metrics for each one of the software characteristics at different phases of the development process in question, evidenced by how each characteristic can be observed in each of the phases. The metrics that give support to each one of the characteristics described previously in the different development process phases were based, generically, in the works of Pfleeger (2001) and Pressman (2001). More specific metrics related to the context of the object-oriented paradigm were based on Chidamber and Kemerer (1994), Lorenz and Kidd (1994), Travassos et al. (2001) and Travassos (2003). Further details can be found in (Araújo et al., 2005).

Table 1 - Metrics associated by Software Characteristic in each phase of the Process

	Size	Periodicity	Complexity	Effort	Reliability	Maintainability
Requirements Specification	<ul style="list-style-type: none"> • # Function Points • # Use Case Points • # Requirements 	<ul style="list-style-type: none"> • Interval between Versions 	<ul style="list-style-type: none"> • # Use Cases 	<ul style="list-style-type: none"> • # Requirements Handled • # Use Cases Handled • # People • Allocated Resources • Spent Time • Average Productivity of the Team 	<ul style="list-style-type: none"> • # Detected Defects • # Corrected Defects 	<ul style="list-style-type: none"> • Spent Time in the Diagnostic of Defects • Spent Time in the Removal of Defects
High Level Design	<ul style="list-style-type: none"> • # Classes • # Methods per Class 	<ul style="list-style-type: none"> • Interval between Versions 	<ul style="list-style-type: none"> • # Class Diagrams • # Sequence Diagrams 	<ul style="list-style-type: none"> • # Class Diagrams Handled • # Sequence Diagrams 	<ul style="list-style-type: none"> • # Detected Defects • # Corrected Defects 	<ul style="list-style-type: none"> • Spent Time in the Diagnostic of Defects • Spent Time in the

¹ An artifact can be an input for a process activity or a product generated by one process activity.

	Size	Periodicity	Complexity	Effort	Reliability	Maintainability
			<ul style="list-style-type: none"> • # State Diagrams • # Package Diagrams • # Activity Diagrams • Depth of Inheritance per Class • # Children per Class 	<ul style="list-style-type: none"> Handled • # State Diagrams Handled • # Package Diagrams Handled • # Activity Diagrams Handled • # People • Allocated Resources • Spent Time • Average Productivity of the Team 		Removal of Defects
Low Level Design	<ul style="list-style-type: none"> • # Key Classes • # Support Classes • # Methods per Class • # Subsystems 	<ul style="list-style-type: none"> • Interval between Versions 	<ul style="list-style-type: none"> • # Class Diagrams • # Sequence Diagrams • Depth of Inheritance per Class • Coupling between Objects • Response for a Class • Lack of Cohesion in Methods • # Children per Class 	<ul style="list-style-type: none"> • # Class Diagrams Handled • # Sequence Diagrams Handled • # People • Allocated Resources • Spent Time • Average Productivity of the Team 	<ul style="list-style-type: none"> • # Detected Defects • # Corrected Defects 	<ul style="list-style-type: none"> • Spent Time in the Diagnostic of Defects • Spent Time in the Removal of Defects
Coding	<ul style="list-style-type: none"> • # Lines of Source Code • # Methods per Class 	<ul style="list-style-type: none"> • Interval between Versions 	<ul style="list-style-type: none"> • Depth of Inheritance per Class • Coupling between Objects • Response for a Class • Lack of Cohesion in Methods • # Children per Class • Cyclomatic Complexity per Method 	<ul style="list-style-type: none"> • # Lines of Source Code Handled • # People • Allocated Resources • Spent Time • Average Productivity of the Team 	<ul style="list-style-type: none"> • # Detected Defects • # Corrected Defects • System Availability 	<ul style="list-style-type: none"> • Spent Time in the Diagnostic of Defects • Spent Time in the Removal of Defects

System Dynamics techniques can be applied so as to understand and to influence the way in which the elements of a system vary throughout the time (Madachy, 2005). These techniques utilize control strategies with feedback loops to organize the available information regarding a system, making up models that can be simulated on a computer (Forrester, 1991). Developed models with System Dynamics techniques can be represented through Cause and Effect Diagrams which are the simplest mechanisms for representing System Dynamics models, or Stock and Flow Diagrams which present a higher level of detail, thus forcing a refinement of definition of system structure (Albin, 1997).

The described environment in Araújo et al (2005), Araújo and Travassos (2005a) and Araújo and Travassos (2006) has used the System Dynamics models through Cause and Effect Diagrams, aiming to observe the behavior of evolving systems. This environment is described in terms of software characteristics (Size, Effort, Periodicity, Complexity, Maintainability, Modularity, and Reliability), generalizing then the behavior for different levels of abstraction in the software development. These levels of abstraction

refer to Requirement Specification, High and Low Level Design and Coding phases, normally present in object-oriented software development processes. In this respect, the model can be instantiated for these different levels of abstraction through the substitution for software characteristics with the specific metrics for each one of these phases.

Based on the previous scenario, the following questions represent the basis for identifying the influences between some of these software characteristics, depending on the software development process phase. The meta model to these systematic reviews is showed in Appendix A.

1. Question 01: Evaluation of the influence between Size and Complexity (Appendix B)
2. Question 02: Evaluation of the influence between Size and Reliability (Appendix C)
3. Question 03: Evaluation of the influence between Complexity and Effort (Appendix D)
4. Question 04: Evaluation of the influence between Effort and Reliability (Appendix E)
5. Question 05: Evaluation of the influence between Complexity and Maintainability (Appendix F)
6. Question 06: Evaluation of the influence between Effort and Maintainability (Appendix G)
7. Question 07: Evaluation of the influence between Effort and Periodicity (Appendix H)
8. Question 08: Evaluation of the influence between Periodicity and Maintainability (Appendix I)
9. Question 09: Evaluation of the influence between Size and Effort (Appendix J)
10. Question 10: Evaluation of the influence between Size and Maintainability (Appendix K)
11. Question 11: Evaluation of the influence between Periodicity and Size (Appendix L)
12. Question 12: Evaluation of the influence between Complexity and Reliability (Appendix M)
13. Question 13: Evaluation of the influence between Periodicity and Complexity (Appendix N)
14. Question 14: Evaluation of the influence between Maintainability and Reliability (Appendix O)

15.Question 15: Evaluation of the influence between Periodicity and Reliability
(Appendix P)

The Table 2 summarizes the influence relationships between the software characteristics that should be investigated for each phase of the object-oriented software development process, based on systematic reviews.

Table 2. Influences between Software Characteristics

Question	Direction	Intensity	References
Q01	Size → Complexity	Yes	(Misic and Tesic,1998) (Ebert, 1996) (Basili and Perricone, 1984)
Q02	Size → Reliability	Yes	(Malaiya and Denton, 2000) (Selby, 1990) (Basili and Perricone, 1984)
Q03	Complexity → Effort	Yes	(Misic and Tesic,1998)
Q04	Reliability → Effort	Yes	(Bianchi et al., 2001)
Q05	Complexity → Maintainability	Yes	(Aggarwal, 2006) (Koten and Gray, 2005) (Lindell and Hagglund, 2004) (McCabe, 1976)
Q06	Effort → Maintainability	Yes	(Nikora and Munson, 2003)
Q07	Periodicity → Effort	Yes	(Mockus et al., 2003)
Q08	No evidence	No	None
Q09	Size → Effort	Yes	(Premraj et al., 2005) (Boehm, 1984) (Boehm, 1981) (Bailey and Basili, 1981) (Walston and Felix, 1977)
Q10	No evidence	No	None
Q11	Periodicity → Size	Yes	(Premraj et al., 2005)
Q12	Complexity → Reliability	Yes	(Basili et al., 1995) (McCabe, 1976)
Q13	Periodicity → Complexity	Yes	(Munson, 1996)
Q14	Maintainability → Reliability	Yes	(Schneidewind, 1999)
Q15	Periodicity → Reliability	Yes	(Schneidewind, 1999)

2.2 System Dynamic Model

Thus, the System Dynamics is appropriated for the objectives of behavior simulation of LSE, proposed in this work. Figure 1 presents this model through Cause and Effect Diagram, based on the review literature showed in Table 2.

This Cause and Effect Diagram represents the basis for experimental studies to be accomplished in the context of the described environment in this work. The logical

formulations built up for the LSE do not establish a relationship between these Laws and absolute values of their characteristics, but a description of their behavior through these characteristics trends (Araújo et al., 2005). This diagram is, at first glance, enough to the initial construction of simulation models through System Dynamics.

This argument leads to definition of the type of statistical analysis of data that will be utilized in the experimental studies. Once the model considers the increasing or decreasing trends of the variables, instead of their absolute values, a quantitative analysis is not adequate in this context. So the semi-quantitative analysis involves a description of situations where the direction of a change in a part of a system is known, but not the size of the effect of this change in other parts (Camiletti and Ferracioli, 2002). The analysis of these effects asks for a comprehension of the direction of the causal relationship (increase or decrease) but not of the numerical values knowledge. The semi-quantitative analysis explores the fact that either the quantitative or qualitative analysis does not capture all the important aspects of a system. Consequently, the construction of models under a semi-quantitative manner demands that the comprehension of the behavior of a system is based on causal relationships among the variables that describe itself.

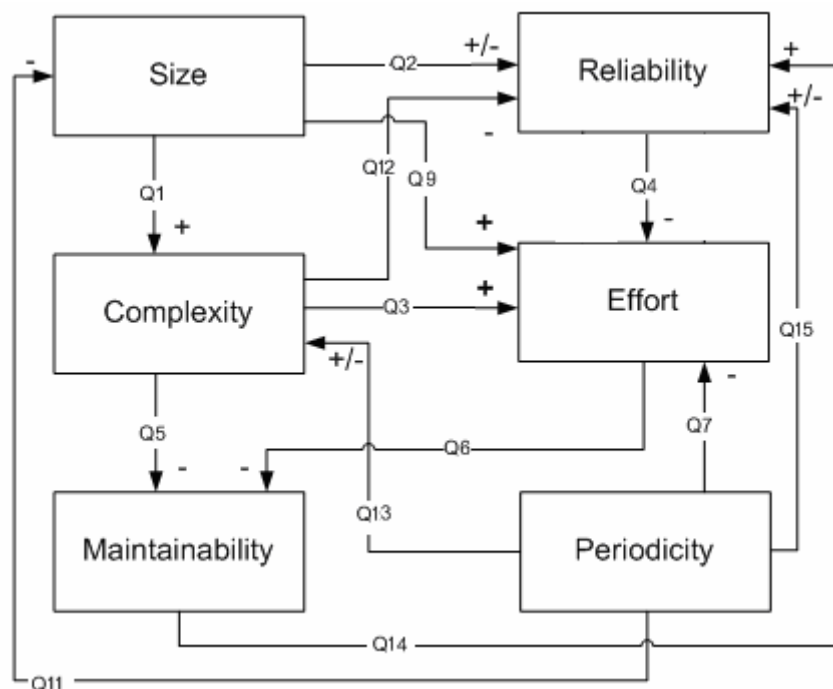


Figure 1. Cause-effect model

Through this model, it has been observed the cause and effect relationship among the considered variables, aiming to verifying the relationship among them. In the

observation of real systems, these variables can be substituted with collected values of available metrics for each phase of the considered software development process, characterizing an *in-virtuo* experiment². In case of simulation of future version of a system, it can be simulated software characteristics as well as the metrics themselves, characterizing an *in-silico* experiment³ (Travassos and Barros, 2003).

This model also has the objective of being the basis for another, more generic, representing the behavior of Laws of Software Evolution. Since each LSE has been described through a logical formulation involving its interest characteristics, and also of the influences among the Laws (Araújo et al., 2005), the model also serves as a subsidy for the observation of the behavior of the LSE in function of the simulation of their software characteristics.

2.3 Differences in relation to the model previously built

The Cause and Effect Diagram initially presented in Araujo and Travassos (2006b), showed in Figure 2, was modified based on in this more detailed study using Systematic Reviews. These studies contributed to a better understanding of software decay process and contributed to the construction of a model of system dynamics more refined.

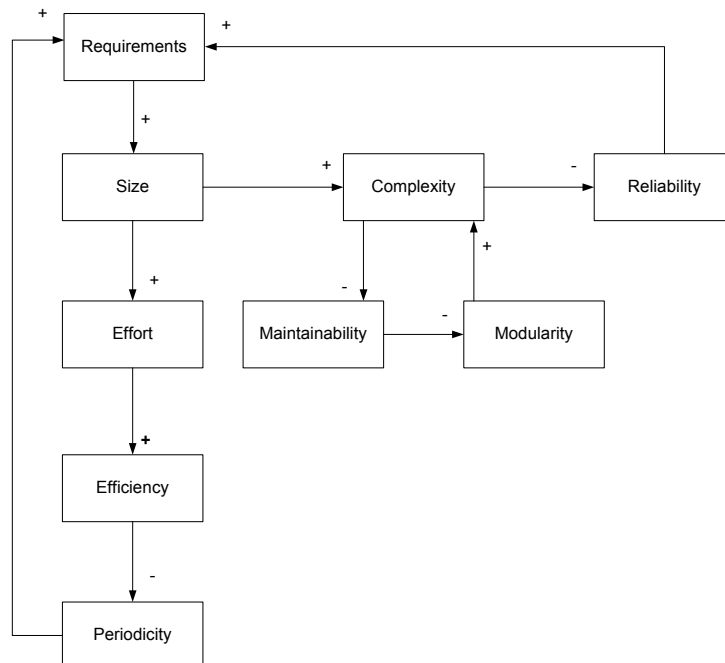


Figure 2. Initial Cause and Effect Diagram (Araujo and Travassos, 2006b)

² *In virtuo* experiments: such experiments involve the interaction among participants and a computerized model of reality.

³ *In silico* experiments: studies characterized by both the subjects and the real world being described as computer models.

Through studies based on technical literature review, some relationships appeared, others were eliminated, and still others changed their sense of influence. There is still that the software characteristics of modularity and efficiency been removed. Modularity can not be observed from models and difficult to be measured by product metrics. Efficiency was incorporated to effort, since the interpretation given to efficiency, in terms of team productivity, is treated in literature as effort.

3 Final Considerations

Therefore, it is expected to evaluate the applicability of Laws of Software Evolution in the context of object-oriented software development process through an environment that allows the studying of decay causes and their consequences in the software development process. The accomplishment of experimental studies may offer some knowledge regarding the feasibility of building software decay models using simulation techniques based on System Dynamics that will allow simulating the behavior of evolving software systems. The cause and effect relationships among the software characteristics need evidences or results from experimental studies that subsidize them. In this fashion, it has been applied a construction approach based on experimentation (Mafra and Travassos, 2006), emphasizing planning and execution of secondary studies through systematic review (Kitchenham, 2004), so as to search for these evidences in the specialized technical literature (Biolchini et al., 2005). A systematic review of the literature is one of the existing means of identifying, evaluating and interpreting all the pertaining research to a particular research question.

References

Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R. Application of Artificial Neural Network for Predicting Maintainability using Object Oriented Metrics. TRANSACTIONS ON ENGINEERING, COMPUTING AND TECHNOLOGY VOLUME 15 OCTOBER 2006.

Albin, S., 1997. "Building an System Dynamics Model Part 1: Conceptualization", Technical Report D-4597, MIT System Dynamics Group, Cambridge, MA.

Araújo, M.A.P., Travassos, G.H., 2005a, "Aplicação das Leis de Evolução de Software em Manutenção Evolutiva", Revista RTInfo – Edição Especial sobre Novas Abordagens para Manutenção de Software.

Araújo, M.A.P., Travassos, G.H., 2006, "An Environment to Observe Object-Oriented Software Evolution", Jornadas Iberoamericana de Ingeniería de Software e Ingeniería del Conocimiento - JIISIC'06, Puebla, México.

Araújo, M.A.P., Travassos, G.H., 2006b, "Building Models through Experimentation to Observe Object-Oriented Software Decay", Experimental Software Engineering Latin American Workshop – ESELAW'06.

Araújo, M.A.P., Travassos, G.H., Kitchenham, B., 2005, "Evolutive Maintenance: Observing Object-Oriented Software Decay", Technical Report ES-672/05 -COPPE/UFRJ.

Bailey, J. W., Basili, V. A Meta-Model for Software Development Resource Expenditures. IEEE, 1981.

Barros, M.O., 2001. "Gerenciamento de Projetos Baseados em Cenários: Uma Abordagem de Modelagem Dinâmica e Simulação". Doctorate Thesis, COPPE/UFRJ.

Basili, V., Briand, L., Melo, W. A Validation of Object Oriented Design Metrics as Quality Indicators. Technical Report, Univ. of Maryland, Dep. of Computer Science, College Park, MD, 20742 USA. April 1995.

Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. *Communications of the ACM*, **1984**, 27, 42 – 52.

Bianchi, A. Caivano, D., Lanubile, F., Visaggio, G. Evaluating Software Degradation through Entropy. IEEE, 2001.

Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H., 2005. "Systematic Review in Software Engineering: Relevance and Utility". Technical Report. PESC - COPPE/UFRJ. Brazil.

Boehm, B. Software Engineering Economics. IEEE Transactions on Software Engineering, 1984.

Boehm, B. Software Engineering Economics. Prentice Hall, 1981.

Camiletti, G.G., Ferracioli, L., 2002, "The Use of Semiquantitative Computational Modelling in the Study of Predator-Prey System". X International Organization for Science and Technology Education, 2002, Foz do Iguaçu.

Ebert, C. Evaluation and application of complexity-based criticality models. *International Software Metrics Symposium, Proceedings*, **1996**, 174 – 184.

Forrester, J.W., 1991, "System Dynamics and the Lessons of 35 Years", Technical Report D-4224-4, MIT System Dynamics Group, Cambridge, MA.

Kitchenham, B., "Procedures for Performing Systematic Reviews", Joint Technical Report, Keele University, 2004.

Koten, C., Gray, A. An Application of Bayesian Network for Predicting Object Oriented Software Maintainability. The Information Science Discussion Paper Series. University of Otago, 2005.

Lehman, M.M., 1980, "Programs, Life Cycle and the Laws of Software Evolution", Proc. IEEE, IEEE.

Lehman, M.M., Ramil, J.F., 2003, "Software Evolution – Background, Theory, Practice", Information Processing Letters, vol. 88, pp. 33 – 44.

Lindell, J., Hagglund, M. Maintainability Metrics for Object Oriented Systems. Software Quality, 2004.

Madachy, R.J. Software Process Dynamics. John Wiley & Sons, 2008.

Mafra, S.N., Travassos, G.H., 2006. "Estudos Primários e Secundários Apoiando a Busca por Evidência em Engenharia de Software". Technical Report ES-687/06, COPPE/UFRJ, Brasil.

Malaiya, Y.K. & Denton, J. Module size distribution and defect density. *Proceedings of the International Symposium on Software Reliability Engineering, ISSRE*, **2000**, 62 – 71.

McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.

Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study. *Journal of Systems and Software*, **1998**, 41, 133 – 143.

Mockus, A. Weiss, D., Zhang, P. Understanding and Predicting Effort in Software Projects. ICSE'03.

Munson, J.C. Software faults, software failures and software reliability modeling. Information and Software Technology, 1996.

Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS'03.

Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).

Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. *IEEE Transactions on Software Engineering*, 1999.

Selby, R.W. Empirically based analysis of failures in software systems. *IEEE Transactions on Reliability*, **1990**, 39, 444 – 454.

Travassos, G.H., Barros, M.O., 2003, “Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering”, Proc. of the WSESE03, Fraunhofer IRB Verlag, Rome.

Walston, C. E., Felix, C.P. A Method of Programming Measurement and Estimation. *IBM System Journal*, 1977.

Appendix A

Meta Model

The meta model presents the structure to be used in the conduct of several systematic reviews, observing the influence between the different software characteristics.

Question Formulation

P1: Is there any influence between the software characteristics <A> and in the <X> phase of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics <A> and .
- **Intervention:** influence between the software characteristics <A> and .
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics <A> and .
- **Outcome measure:** number of works that identify the influence between the software characteristics <A> and .
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics <A> and , discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics <A> and , that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics <A> and in the <X> phase of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics <A> and .
- **Intervention:** direction of the influence between the software characteristics <A> and .
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics <A> and .
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics <A> and .
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics <A> and , that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics <A> and in the <X> phase of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics <A> and .
- **Intervention:** intensity/rate of the influence between the software characteristics <A> and .
- **Control:** not defined.

- **Effect:** to characterize the intensity/rate of the influence between the software characteristics <A> and .
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics <A> and .
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics <A> and , that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

The basic source of information will be represented by some digital libraries including conferences and journals listed below. It will also be considered the search for proceedings of conferences whose themes are concerned with software maintenance.

General key words

- Software Characteristic
- Metric
- Relation, relationship, correlation, dependency, influence, effect (specific for P1)
- Direction, Primary study, experimental study, empirical study (specific for P1.1)
- Intensity, rate (specific for P1.1.1)

Idiom of the Studies

English.

Identification of Sources

Source Search Methods:

The sources will be accessed via web. In the context of this review, the manual search won't be initially considered.

Source List:

- IEEE Digital Library
- ACM Digital Library
- Web of Science
- EI COMPENDIX
- INSPEC

Types of Papers

Theoretical, Concept Evidence, Experimental Studies.

Criteria of Inclusion and Exclusion of Papers

- The papers must be available on the web;
- The papers must be written in English;
- The papers must consider studies of relationship between software characteristics or metrics;
- The papers must consider software applications (E-type Systems). Thus, we won't be taking into consideration applications related to basic software (operational systems, compilers, protocols);
- The papers must deal with the influence between software characteristics; (for P1)
- The papers must deal with the direction of the influences between software characteristics; (for P1.1)
- The papers must deal with the intensity/rate of influences between software characteristics.(for P1.1.1)

Process of Selection of Preliminary Studies

A researcher will apply the search strategy to identify the papers in potential. The identified papers will be selected by the same researcher through reading and verification of the inclusion and exclusion criteria established, what include the extraction of data. Once it has been done, a second researcher will evaluate the results, looking for a consensus on the papers selection.

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

Results Summary

For P1 question (Influence between the software characteristics <A> and):

Paper	Yes	No	Not conclusive

For P1.1 question (Direction of the influence between the software characteristics <A> and):

Paper	A→B	B→A	Not conclusive

For P1.1.1 question (Intensity/rate of the influence between the software characteristics <A> and):

Paper	Range/Means of the Rate	Not conclusive

Evaluation of the Quality of the Primary Studies

Each paper must be evaluated according to the questions in table below, attributing a quality score per paper.

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?					
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?					
2. Did the studies carry out a sensitivity or residual analysis?					
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?					
2.2. Was the result of the sensitivity or					

residual analysis used to remove abnormal data points if necessary?					
3. Were accuracy statistics based on the raw data scale?					
4. How good was the study comparison method?					
5. Is it clear what projects were used to construct each model?					
6. Is it clear how accuracy was measured?					
7. Is it clear what cross-validation method was used?					
8. Were all model construction methods fully defined (tools and methods used)?					
Total primary study using scores					

Top-level questions without sub-questions are answered Yes/No, corresponding to scores 1 and 0, respectively (Question 3). Whenever a top-level question has sub-questions (Question 1 and 2), scores will be attributed to each sub-question such that the overall score for the top-level question will range between 1 and 0. For example, Question 1 has two sub-questions, thus each “Yes”, and “No” for a sub-question contributes with scores of 0.5, and 0 respectively.

For Question 4 the following criteria will be used: Less than 2 projects: Poor quality (score = 0); Between 2 and 5 projects: Fair quality (score = 0.33); Between 5 and 10 projects: Good quality (score = 0.67); More than 10 projects: Excellent quality (score = 1).

Questions 5 to 8 range from 0 to 4, representing very poor and excellent quality, respectively.

The researcher will evaluate each paper against each criterion. Weighted scores will be attributed to the primary studies and will be presented in the table, and will indicate that, according to the scoring scheme, the papers with the highest and lowest quality scores. In the case of divergence between the articles, this shall be considered the highest score.

Appendix B

Question 01: Evaluation of the influence between Size and Complexity

P1: Is there any influence between the software characteristics Size and Complexity of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Size and Complexity.
- **Intervention:** influence between the software characteristics Size and Complexity.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Size and Complexity.
- **Outcome measure:** number of works that identify the influence between the software characteristics Size and Complexity.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Size and Complexity, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Size and Complexity, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Size and Complexity of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Size and Complexity.
- **Intervention:** direction of the influence between the software characteristics Size and Complexity.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Size and Complexity.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Size and Complexity.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Size and Complexity, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Size and Complexity of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Size and Complexity.
- **Intervention:** intensity/rate of the influence between the software characteristics Size and Complexity.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Size and Complexity.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Size and Complexity.
- **Population:** selected works in question **P1.1**.

- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Size and Complexity, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Size: size, Function Points, Use Case Points, Requirement

Complexity: complexity or Use Case

High Level Design (or Design or Analysis)

Size: size or Classes or Methods per Class

Complexity: complexity or Class Diagrams or Sequence Diagrams or State Diagrams or Package Diagrams or Activity Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Children per Class or NOC or Number of Children)

Low Level Design (or Design or Detailed Design)

Size: size or Key Classes or Support Classes or Methods per Class or Subsystems

Complexity: complexity or Class Diagrams or Sequence Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Coupling between Objects or CBO or Response for a Class or RFC or Lack of Cohesion in Methods or LCOM or Children per Class or NOC or Number of Children)

Coding (or Codification or Programming or Building or Construction or Implementation)

Size: size or LOC or lines of code or source lines of code or methods per classes

Complexity: complexity or Depth of Inheritance per Class or Coupling between Objects or Response for a Class or Lack of Cohesion in Methods or Children per Class or Cyclomatic Complexity per Method)

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or "Function Points" or "Use Case Points" or "Requirement")

and

(complexity or "use case")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition"

or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or Classes or "Methods per Class")
and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "State Diagrams" or "Package Diagrams" or "Activity Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Children per Class" or NOC or "Number of Children")

and

(Design or "High Level Design" or Analysis)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or "Key Classes" or "Support Classes" or "Methods per Class" or "Subsystems")
and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Coupling between Objects" or CBO or "Response for a Class" or RFC or "Lack of Cohesion in Methods" or LCOM or "Children per Class" or NOC or "Number of Children")

and

(Design or "Low Level Design" or "Detailed Design")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or

Linkage)

and

(size or LOC or “lines of code” or “source lines of code” or “methods per classes”)

and

(complexity or “Depth of Inheritance per Class” or “Coupling between Objects” or “Response for a Class” or “Lack of Cohesion in Methods” or “Children per Class” or “Cyclomatic Complexity per Method”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or “software measure” or “software measurement”)

Results Summary

For P1 question (Influence between the software characteristics Size and Complexity):

Paper	Yes	No	Not conclusive
Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998 , 41, 133 - 143	X		
Ebert, C. Evaluation and application of complexity-based criticality models <i>International Software Metrics Symposium, Proceedings</i> , 1996 , 174 - 184	X		
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM</i> , 1984 , 27, 42 - 52	X		

For P1.1 question (Direction of the influence between the software characteristics Size and Complexity):

Paper	A→B	B→A	Not conclusive
-------	-----	-----	----------------

Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998 , 41, 133 - 143	X		
Ebert, C. Evaluation and application of complexity-based criticality models <i>International Software Metrics Symposium, Proceedings</i> , 1996 , 174 - 184	X		
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM</i> , 1984 , 27, 42 - 52	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Size and Complexity):

Paper	Range/Means of the Rate	Not conclusive
Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998 , 41, 133 - 143	X	
Ebert, C. Evaluation and application of complexity-based criticality models <i>International Software Metrics Symposium, Proceedings</i> , 1996 , 174 - 184		X
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM</i> , 1984 , 27, 42 - 52	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes	Yes	Yes		

1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0.5	0	0		
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5	1	1		
2. Did the studies carry out a sensitivity or residual analysis?	No	No	No		
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0	0	0		
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0	0		
3. Were accuracy statistics based on the raw data scale?	1	0	0		
4. How good was the study comparison method?	0.67	0	0		
5. Is it clear what projects were used to construct each model?	1	1	1		
6. Is it clear how accuracy was measured?	1	0	1		
7. Is it clear what cross-validation method was used?	1	0	0		
8. Were all model construction methods fully defined (tools and methods used)?	0	0	0		
Total primary study using scores	5.67	2	3		

Appendix C

Question 02: Evaluation of the influence between Size and Reliability

P1: Is there any influence between the software characteristics Size and Reliability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Size and Reliability.
- **Intervention:** influence between the software characteristics Size and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Size and Reliability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Size and Reliability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Size and Reliability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Size and Reliability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Size and Reliability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Size and Reliability.

- **Intervention:** direction of the influence between the software characteristics Size and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Size and Reliability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Size and Reliability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Size and Reliability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Size and Reliability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Size and Reliability.
- **Intervention:** intensity/rate of the influence between the software characteristics Size and Reliability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Size and Reliability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Size and Reliability.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Size and Reliability, that is, in which intensity/rate one software characteristic influences the other one, in addition

to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Size: size, Function Points, Use Case Points, Requirement

Reliability: reliability, Detected Defects, Corrected Defects

High Level Design (or Design or Analysis)

Size: size or Classes or Methods per Class

Reliability: reliability, Detected Defects, Corrected Defects

Low Level Design (or Design or Detailed Design)

Size: size or Key Classes or Support Classes or Methods per Class or Subsystems

Reliability: reliability, Detected Defects, Corrected Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Size: size or LOC or lines of code or source lines of code or methods per classes

Reliability: reliability, Detected Defects, Corrected Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list

- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or "Function Points" or "Use Case Points" or "Requirement")

and

(reliability or "Detected Defects" or "Corrected Defects")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or Classes or "Methods per Class")

and

(reliability or “Detected Defects” or “Corrected Defects”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or “Key Classes” or “Support Classes” or “Methods per Class” or “Subsystems”)

and

(reliability or “Detected Defects” or “Corrected Defects”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or LOC or “lines of code” or “source lines of code” or “methods per classes”)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Size and Reliability):

Paper	Yes	No	Not conclusive
Malaiya, Y.K. & Denton, J. Module size distribution and defect density <i>Proceedings of the International Symposium on Software Reliability Engineering, ISSRE, 2000, 62 - 71</i>	X		
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM, 1984, 27, 42 - 52</i>	X		
Selby, R.W. Empirically based analysis of failures in software systems <i>IEEE Transactions on Reliability, 1990, 39, 444 - 454</i>		X	

For P1.1 question (Direction of the influence between the software characteristics Size and Reliability):

Paper	A→B	B→A	Not conclusive
Malaiya, Y.K. & Denton, J. Module size distribution and defect density <i>Proceedings of the International Symposium on Software Reliability Engineering, ISSRE, 2000, 62 - 71</i>	X		
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM, 1984, 27, 42 - 52</i>	X		
Selby, R.W. Empirically based analysis of failures in software systems <i>IEEE Transactions on Reliability, 1990, 39, 444 - 454</i>			X

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Size and Reliability):

Paper	Range/Means of the Rate	Not conclusive
Malaiya, Y.K. & Denton, J. Module size distribution and defect density <i>Proceedings of the International Symposium on Software Reliability Engineering, ISSRE, 2000</i> , 62 - 71	X	
Basili, V.R. & Perricone, B.T. SOFTWARE ERRORS AND COMPLEXITY: AN EMPIRICAL INVESTIGATION. <i>Communications of the ACM, 1984</i> , 27, 42 - 52	X	
Selby, R.W. Empirically based analysis of failures in software systems <i>IEEE Transactions on Reliability, 1990</i> , 39, 444 - 454		X

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes	Yes	Yes		
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0	0	0		
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	1	1	1		
2. Did the studies carry out a sensitivity or residual analysis?	No	No	No		
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0	0	0		
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0	0		

3. Were accuracy statistics based on the raw data scale?	0	0	0		
4. How good was the study comparison method?	2	0	0		
5. Is it clear what projects were used to construct each model?	1	1	1		
6. Is it clear how accuracy was measured?	1	1	1		
7. Is it clear what cross-validation method was used?	0	0	0		
8. Were all model construction methods fully defined (tools and methods used)?	0	0	0		
Total primary study using scores	5	3	3		

Appendix D

Question 03: Evaluation of the influence between Complexity and Effort

P1: Is there any influence between the software characteristics Complexity and Effort of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Complexity and Effort.
- **Intervention:** influence between the software characteristics Complexity and Effort.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Complexity and Effort.
- **Outcome measure:** number of works that identify the influence between the software characteristics Complexity and Effort.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Complexity and Effort, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Complexity and Effort, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Complexity and Effort of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Complexity and Effort.
- **Intervention:** direction of the influence between the software characteristics Complexity and Effort.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Complexity and Effort.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Complexity and Effort.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Complexity and Effort, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Complexity and Effort of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Complexity and Effort.
- **Intervention:** intensity/rate of the influence between the software characteristics Complexity and Effort.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Complexity and Effort.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Complexity and Effort.
- **Population:** selected works in question **P1.1**.

- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Complexity and Effort, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Complexity: complexity, Use Case

Effort: effort or Requirements handled or Use Cases handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

High Level Design (or Design or Analysis)

Complexity: complexity or Class Diagrams or Sequence Diagrams or State Diagrams or Package Diagrams or Activity Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Children per Class or NOC or Number of Children)

Effort: effort or Class diagrams handled or Sequence diagrams handled or State Diagrams handled or Package Diagrams handled or Activity Diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Low Level Design (or Design or Detailed Design)

Complexity: complexity or Class Diagrams or Sequence Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Coupling between Objects or CBO or Response for a Class or RFC or Lack of Cohesion in Methods or LCOM or Children per Class or NOC or Number of Children)

Effort: effort or Class diagrams handled or Sequence diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Coding (or Codification or Programming or Building or Construction or Implementation)

Complexity: complexity or Depth of Inheritance per Class or Coupling between Objects or Response for a Class or Lack of Cohesion in Methods or Children per Class or Cyclomatic Complexity per Method)

Effort: effort or LOC handled or lines of code handled or source lines of code handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “use case”)

and

(effort or “Requirements handled” or “Use Cases handled” or efficiency or

“People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(“Requirement Specification” or “Requirement Elicitation” or “Requirement Definition” or Analysis or “User Requirement” or “Requisite”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “Class Diagrams” or “Sequence Diagrams” or “State Diagrams” or “Package Diagrams” or “Activity Diagrams” or “Depth of Inheritance per Class” or “Depth of Inheritance Tree” or DIT or “Children per Class” or NOC or “Number of Children”)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or “State Diagrams handled” or “Package Diagrams handled” or “Activity Diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “Class Diagrams” or “Sequence Diagrams” or “Depth of Inheritance per Class” or “Depth of Inheritance Tree” or DIT or “Coupling between Objects” or CBO or “Response for a Class” or RFC or “Lack of

Cohesion in Methods” or LCOM or “Children per Class” or NOC or “Number of Children”)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “Depth of Inheritance per Class” or “Coupling between Objects” or “Response for a Class” or “Lack of Cohesion in Methods” or “Children per Class” or “Cyclomatic Complexity per Method”)

and

(effort or “LOC handled” or “lines of code handled” or “source lines of code handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Codification or Programming or Building or Construction or Implementation or Coding)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Complexity and Effort):

Paper	Yes	No	Not conclusive
Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998, 41, 133 - 143	X		

For P1.1 question (Direction of the influence between the software characteristics Complexity and Effort):

Paper	A→B	B→A	Not conclusive
Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998, 41, 133 - 143	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Complexity and Effort):

Paper	Range/Means of the Rate	Not conclusive
Misic, V.B. & Tesic, D.N. Estimation of effort and complexity: An object-oriented case study <i>Journal of Systems and Software</i> , 1998, 41, 133 - 143	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0.5				
1.2. Was the result of the investigation	0.5				

used appropriately to transform the data and select appropriated data points?					
2. Did the studies carry out a sensitivity or residual analysis?	No				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.67				
5. Is it clear what projects were used to construct each model?	1				
6. Is it clear how accuracy was measured?	1				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	0				
Total primary study using scores	5.67				

Appendix E

Question 04: Evaluation of the influence between Effort and Reliability

P1: Is there any influence between the software characteristics Effort and Reliability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Effort and Reliability.
- **Intervention:** influence between the software characteristics Effort and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Effort and Reliability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Effort and Reliability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Effort and Reliability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Effort and Reliability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Effort and Reliability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Effort and Reliability.

- **Intervention:** direction of the influence between the software characteristics Effort and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Effort and Reliability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Effort and Reliability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Effort and Reliability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Effort and Reliability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Effort and Reliability.
- **Intervention:** intensity/rate of the influence between the software characteristics Effort and Reliability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Effort and Reliability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Effort and Reliability.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Effort and Reliability, that is, in which intensity/rate one software characteristic influences the other one, in addition

to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Effort: effort or Requirements handled or Use Cases handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Reliability: reliability or Detected Defects or Corrected Defects

High Level Design (or Design or Analysis)

Effort: effort or Class diagrams handled or Sequence diagrams handled or State Diagrams handled or Package Diagrams handled or Activity Diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Reliability: reliability or Detected Defects or Corrected Defects

Low Level Design (or Design or Detailed Design)

Effort: effort or Class diagrams handled or Sequence diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Reliability: reliability or Detected Defects or Corrected Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Effort: effort or LOC handled or lines of code handled or source lines of code handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Reliability: reliability or Detected Defects or Corrected Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source

- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "Requirements handled" or "Use Cases handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(reliability or "Detected Defects" or "Corrected Defects")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or “State Diagrams handled” or “Package Diagrams handled” or “Activity Diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(reliability or “Detected Defects” or “Corrected Defects”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(reliability or “Detected Defects” or “Corrected Defects”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "LOC handled" or "lines of code handled" or "source lines of code handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Effort and Reliability):

Paper	Yes	No	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS'03.	X		
Bianchi, A. Caivano, D., Lanubile, F., Visaggio, G. Evaluating Software Degradation through Entropy. IEEE, 2001.	X		

For P1.1 question (Direction of the influence between the software characteristics Effort and Reliability):

Paper	A→B	B→A	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS'03.	X		
Bianchi, A. Caivano, D., Lanubile, F., Visaggio, G. Evaluating Software		X	

Degradation through Entropy. IEEE, 2001.			
--	--	--	--

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Effort and Reliability):

Paper	Range/Means of the Rate	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS'03.	X	
Bianchi, A. Caivano, D., Lanubile, F., Visaggio, G. Evaluating Software Degradation through Entropy. IEEE, 2001.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes	Yes			
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0	0.5			
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5	0.5			
2. Did the studies carry out a sensitivity or residual analysis?	Yes	No			
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0.5	0			
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0			
3. Were accuracy statistics based on the raw data scale?	1	1			

4. How good was the study comparison method?	0.33	0.33			
5. Is it clear what projects were used to construct each model?	1	1			
6. Is it clear how accuracy was measured?	2	1			
7. Is it clear what cross-validation method was used?	1	1			
8. Were all model construction methods fully defined (tools and methods used)?	1	1			
Total primary study using scores	7.33	6.33			

Appendix F

Question 05: Evaluation of the influence between Complexity and Maintainability

P1: Is there any influence between the software characteristics Complexity and Maintainability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Complexity and Maintainability.
- **Intervention:** influence between the software characteristics Complexity and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Complexity and Maintainability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Complexity and Maintainability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Complexity and Maintainability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Complexity and Maintainability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Complexity and Maintainability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Complexity and Maintainability.
- **Intervention:** direction of the influence between the software characteristics Complexity and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Complexity and Maintainability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Complexity and Maintainability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Complexity and Maintainability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Complexity and Maintainability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Complexity and Maintainability.
- **Intervention:** intensity/rate of the influence between the software characteristics Complexity and Maintainability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Complexity and Maintainability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Complexity and Maintainability.

- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Complexity and Maintainability, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Complexity: complexity or Use Case

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

High Level Design (or Design or Analysis)

Complexity: complexity or Class Diagrams or Sequence Diagrams or State Diagrams or Package Diagrams or Activity Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Children per Class or NOC or Number of Children)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Low Level Design (or Design or Detailed Design)

Complexity: complexity or Class Diagrams or Sequence Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Coupling between Objects or CBO or Response for a Class or RFC or Lack of Cohesion in Methods or LCOM or Children per Class or NOC or Number of Children)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Complexity: complexity or Depth of Inheritance per Class or Coupling between Objects or Response for a Class or Lack of Cohesion in Methods or Children per Class or Cyclomatic Complexity per Method)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “use case”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(“Requirement Specification” or “Requirement Elicitation” or “Requirement Definition” or Analysis or “User Requirement” or “Requisite”)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "State Diagrams" or "Package Diagrams" or "Activity Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Children per Class" or NOC or "Number of Children")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

(Design or "High Level Design" or Analysis)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Coupling between Objects" or CBO or "Response for a Class" or RFC or "Lack of Cohesion in Methods" or LCOM or "Children per Class" or NOC or "Number of Children")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

(Design or "Low Level Design" or "Detailed Design")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or

Linkage)

and

(complexity or “Depth of Inheritance per Class” or “Coupling between Objects” or “Response for a Class” or “Lack of Cohesion in Methods” or “Children per Class” or “Cyclomatic Complexity per Method”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Complexity and Maintainability):

Paper	Yes	No	Not conclusive
Lindell, J., Hagglund, M. Maintainability Metrics for Object Oriented Systems. Software Quality, 2004.	X		
Koten, C., Gray, A. An Application of Bayesian Network for Predicting Object Oriented Software Maintainability. The Information Science Discussion Paper Series. University of Otago, 2005.	X		
Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R. Application of Artificial Neural Network for Predicting Maintainability using Object Oriented Metrics. TRANSACTIONS ON ENGINEERING, COMPUTING AND TECHNOLOGY VOLUME 15 OCTOBER 2006.	X		
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X		

For P1.1 question (Direction of the influence between the software characteristics Complexity and Maintainability):

Paper	A→B	B→A	Not conclusive
Lindell, J., Hagglund, M. Maintainability Metrics for Object Oriented Systems. Software Quality, 2004.	X		
Koten, C., Gray, A. An Application of Bayesian Network for Predicting Object Oriented Software Maintainability. The Information Science Discussion Paper Series. University of Otago, 2005.	X		
Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R. Application of Artificial Neural Network for Predicting Maintainability using Object Oriented Metrics. TRANSACTIONS ON ENGINEERING, COMPUTING AND TECHNOLOGY VOLUME 15 OCTOBER 2006.	X		
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Complexity and Maintainability):

Paper	Range/Means of the Rate	Not conclusive
Lindell, J., Hagglund, M. Maintainability Metrics for Object Oriented Systems. Software Quality, 2004.		X
Koten, C., Gray, A. An Application of Bayesian Network for Predicting Object Oriented Software Maintainability. The Information Science Discussion Paper Series. University of Otago, 2005.	X	
Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R. Application of Artificial Neural Network for Predicting Maintainability using Object Oriented Metrics. TRANSACTIONS ON ENGINEERING, COMPUTING AND TECHNOLOGY VOLUME 15 OCTOBER 2006.	X	
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	No	Yes	Yes	No	
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0	0	0	0	
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0	0.5	0.5	0	
2. Did the studies carry out a sensitivity or residual analysis?	No	Yes	Yes	No	
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0	0.5	0.5	0	
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0	0	0	
3. Were accuracy statistics based on the raw data scale?	0	1	1	0	
4. How good was the study comparison method?	0	0.67	0.33	1	
5. Is it clear what projects were used to construct each model?	1	2	1	2	
6. Is it clear how accuracy was measured?	0	1	1	1	
7. Is it clear what cross-validation method was used?	0	1	0	0	
8. Were all model construction methods fully defined (tools and methods used)?	0	1	1	0	
Total primary study using scores	1	7.67	5.33	4	

Appendix G

Question 06: Evaluation of the influence between Effort and Maintainability

P1: Is there any influence between the software characteristics Effort and Maintainability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Effort and Maintainability.
- **Intervention:** influence between the software characteristics Effort and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Effort and Maintainability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Effort and Maintainability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Effort and Maintainability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Effort and Maintainability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Effort and Maintainability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Effort and Maintainability.

- **Intervention:** direction of the influence between the software characteristics Effort and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Effort and Maintainability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Effort and Maintainability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Effort and Maintainability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Effort and Maintainability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Effort and Maintainability.
- **Intervention:** intensity/rate of the influence between the software characteristics Effort and Maintainability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Effort and Maintainability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Effort and Maintainability.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Effort and Maintainability, that is, in

which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Effort: effort or Requirements handled or Use Cases handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

High Level Design (or Design or Analysis)

Effort: effort or Class diagrams handled or Sequence diagrams handled or State Diagrams handled or Package Diagrams handled or Activity Diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Low Level Design (or Design or Detailed Design)

Effort: effort or Class diagrams handled or Sequence diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Effort: effort or LOC handled or lines of code handled or source lines of code handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "Requirements handled" or "Use Cases handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "Class diagrams handled" or "Sequence diagrams handled" or "State Diagrams handled" or "Package Diagrams handled" or "Activity Diagrams handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

(Design or "High Level Design" or Analysis)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "Class diagrams handled" or "Sequence diagrams handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

(Design or "Low Level Design" or "Detailed Design")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "LOC handled" or "lines of code handled" or "source lines of code handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Effort and Maintainability):

Paper	Yes	No	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS’03.	X		

For P1.1 question (Direction of the influence between the software characteristics Effort and Maintainability):

Paper	A→B	B→A	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS’03.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Effort and Maintainability):

Paper	Range/Means of the Rate	Not conclusive
Nikora, A. P., Munson, J.C. Developing Fault Predictors for Evolving Software Systems. Proc. Ninth International Software Metrics Symposium – METRICS '03.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0				
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	Yes				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0.5				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.33				
5. Is it clear what projects were used to construct each model?	1				

COPPE/UFRJ

6. Is it clear how accuracy was measured?	2				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	7.33				

Appendix H

Question 07: Evaluation of the influence between Effort and Periodicity

P1: Is there any influence between the software characteristics Effort and Periodicity of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Effort and Periodicity.
- **Intervention:** influence between the software characteristics Effort and Periodicity.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Effort and Periodicity.
- **Outcome measure:** number of works that identify the influence between the software characteristics Effort and Periodicity.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Effort and Periodicity, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Effort and Periodicity, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Effort and Periodicity of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Effort and Periodicity.

- **Intervention:** direction of the influence between the software characteristics Effort and Periodicity.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Effort and Periodicity.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Effort and Periodicity.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Effort and Periodicity, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Effort and Periodicity of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Effort and Periodicity.
- **Intervention:** intensity/rate of the influence between the software characteristics Effort and Periodicity.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Effort and Periodicity.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Effort and Periodicity.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Effort and Periodicity, that is, in which intensity/rate one software characteristic influences the other one, in addition

to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Effort: effort or Requirements handled or Use Cases handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Periodicity: periodicity or Interval between Versions

High Level Design (or Design or Analysis)

Effort: effort or Class diagrams handled or Sequence diagrams handled or State Diagrams handled or Package Diagrams handled or Activity Diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Periodicity: periodicity or Interval between Versions

Low Level Design (or Design or Detailed Design)

Effort: effort or Class diagrams handled or Sequence diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Periodicity: periodicity or Interval between Versions

Coding (or Codification or Programming or Building or Construction or Implementation)

Effort: effort or LOC handled or lines of code handled or source lines of code handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Periodicity: periodicity or Interval between Versions

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type

- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "Requirements handled" or "Use Cases handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(periodicity or "Interval between Versions")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or “State Diagrams handled” or “Package Diagrams handled” or “Activity Diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(periodicity or “Interval between Versions”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(periodicity or “Interval between Versions”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(effort or "LOC handled" or "lines of code handled" or "source lines of code handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

(periodicity or "Interval between Versions")

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Effort and Periodicity):

Paper	Yes	No	Not conclusive
Mockus, A. Weiss, D., Zhang, P. Understanding and Predicting Effort in Software Projects. ICSE'03.	X		

For P1.1 question (Direction of the influence between the software characteristics Effort and Periodicity):

Paper	A→B	B→A	Not conclusive
Mockus, A. Weiss, D., Zhang, P. Understanding and Predicting Effort in Software Projects. ICSE'03.		X	

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Effort and Periodicity):

Paper	Range/Means of the Rate	Not conclusive
Mockus, A. Weiss, D., Zhang, P.	X	

Understanding and Predicting Effort in Software Projects. ICSE'03.		
--	--	--

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0				
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	No				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0				
5. Is it clear what projects were used to construct each model?	1				
6. Is it clear how accuracy was measured?	2				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	6.5				

Appendix I

Question 08: Evaluation of the influence between Periodicity and Maintainability

P1: Is there any influence between the software characteristics Periodicity and Maintainability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Periodicity and Maintainability.
- **Intervention:** influence between the software characteristics Periodicity and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Periodicity and Maintainability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Periodicity and Maintainability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Periodicity and Maintainability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Periodicity and Maintainability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Periodicity and Maintainability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Periodicity and Maintainability.

- **Intervention:** direction of the influence between the software characteristics Periodicity and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Periodicity and Maintainability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Periodicity and Maintainability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Periodicity and Maintainability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Periodicity and Maintainability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Periodicity and Maintainability.
- **Intervention:** intensity/rate of the influence between the software characteristics Periodicity and Maintainability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Periodicity and Maintainability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Periodicity and Maintainability.
- **Population:** selected works in question **P1.1**.

- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Periodicity and Maintainability, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Periodicity: periodicity or Interval between Versions

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

High Level Design (or Design or Analysis)

Periodicity: periodicity or Interval between Versions

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Low Level Design (or Design or Detailed Design)

Periodicity: periodicity or Interval between Versions

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Periodicity: periodicity or Interval between Versions

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type

- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or

Linkage)

and

(periodicity or “Interval between Versions”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or “Interval between Versions”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or “Interval between Versions”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Codification or Programming or Building or Construction or Implementation or

Coding) and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Periodicity and Maintainability):

Paper	Yes	No	Not conclusive

For P1.1 question (Direction of the influence between the software characteristics Periodicity and Maintainability):

Paper	A→B	B→A	Not conclusive

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Periodicity and Maintainability):

Paper	Range/Means of the Rate	Not conclusive

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?					
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data					

points?					
2. Did the studies carry out a sensitivity or residual analysis?					
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?					
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?					
3. Were accuracy statistics based on the raw data scale?					
4. How good was the study comparison method?					
5. Is it clear what projects were used to construct each model?					
6. Is it clear how accuracy was measured?					
7. Is it clear what cross-validation method was used?					
8. Were all model construction methods fully defined (tools and methods used)?					
Total primary study using scores					

Appendix J

Question 09: Evaluation of the influence between Size and Effort

P1: Is there any influence between the software characteristics Size and Effort of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Size and Effort.
- **Intervention:** influence between the software characteristics Size and Effort.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Size and Effort.
- **Outcome measure:** number of works that identify the influence between the software characteristics Size and Effort.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Size and Effort, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Size and Effort, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Size and Effort of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Size and Effort.

- **Intervention:** direction of the influence between the software characteristics Size and Effort.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Size and Effort.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Size and Effort.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Size and Effort, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Size and Effort of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Size and Effort.
- **Intervention:** intensity/rate of the influence between the software characteristics Size and Effort.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Size and Effort.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Size and Effort.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Size and Effort, that is, in which intensity/rate one software characteristic influences the other one, in addition

to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Size: size, Function Points, Use Case Points, Requirement

Effort: effort or Requirements handled or Use Cases handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

High Level Design (or Design or Analysis)

Size: size or Classes or Methods per Class

Effort: effort or Class diagrams handled or Sequence diagrams handled or State Diagrams handled or Package Diagrams handled or Activity Diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Low Level Design (or Design or Detailed Design)

Size: size or Key Classes or Support Classes or Methods per Class or Subsystems

Effort: effort or Class diagrams handled or Sequence diagrams handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Coding (or Codification or Programming or Building or Construction or Implementation)

Size: size or LOC or lines of code or source lines of code or methods per classes

Effort: effort or LOC handled or lines of code handled or source lines of code handled or efficiency or People or Allocated Resources or Spent Time or Average Productivity

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source

- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or "Function Points" or "Use Case Points" or "Requirement")

and

(effort or "Requirements handled" or "Use Cases handled" or efficiency or "People" or "Allocated Resources" or "Spent Time" or "Average Productivity")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or Classes or “Methods per Class”)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or “State Diagrams handled” or “Package Diagrams handled” or “Activity Diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or “Key Classes” or “Support Classes” or “Methods per Class” or “Subsystems”)

and

(effort or “Class diagrams handled” or “Sequence diagrams handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or LOC or “lines of code” or “source lines of code” or “methods per

classes”)

and

(effort or “LOC handled” or “lines of code handled” or “source lines of code handled” or efficiency or “People” or “Allocated Resources” or “Spent Time” or “Average Productivity”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Size and Effort):

Paper	Yes	No	Not conclusive
Walston, C. E., Felix, C.P. A Method of Programming Measurement and Estimation. IBM System Journal, 1977.	X		
Bailey, J. W., Basili, V. A Meta-Model for Software Development Resource Expenditures. IEEE, 1981.	X		
Boehm, B. Software Engineering Economics. Prentice Hall, 1981.	X		
Boehm, B. Software Engineering Economics. IEEE Transactions on Software Engineering, 1984.	X		
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X		

For P1.1 question (Direction of the influence between the software characteristics Size and Effort):

Paper	A→B	B→A	Not conclusive
Walston, C. E., Felix, C.P. A Method of Programming Measurement and Estimation. IBM System Journal, 1977.	X		

Bailey, J. W., Basili, V. A Meta-Model for Software Development Resource Expenditures. IEEE, 1981.	X		
Boehm, B. Software Engineering Economics. Prentice Hall, 1981.	X		
Boehm, B. Software Engineering Economics. IEEE Transactions on Software Engineering, 1984.	X		
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Size and Effort):

Paper	Range/Means of the Rate	Not conclusive
Walston, C. E., Felix, C.P. A Method of Programming Measurement and Estimation. IBM System Journal, 1977.	X	
Bailey, J. W., Basili, V. A Meta-Model for Software Development Resource Expenditures. IEEE, 1981.	X	
Boehm, B. Software Engineering Economics. Prentice Hall, 1981.		X
Boehm, B. Software Engineering Economics. IEEE Transactions on Software Engineering, 1984.		X
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes	Yes	No	No	Yes
1.1. Was the data investigated to identify outliers and to assess	0	0	0	0	0.5

distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5	0.5	0	0	0.5
2. Did the studies carry out a sensitivity or residual analysis?	Yes	Yes	No	No	Yes
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0.5	0.5	0	0	0.5
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0	0	0	0.5
3. Were accuracy statistics based on the raw data scale?	1	1	0	0	1
4. How good was the study comparison method?	0.33	0.33	0	0	0.33
5. Is it clear what projects were used to construct each model?	1	2	1	1	2
6. Is it clear how accuracy was measured?	2	2	0	0	2
7. Is it clear what cross-validation method was used?	1	1	0	0	1
8. Were all model construction methods fully defined (tools and methods used)?	1	1	0	0	1
Total primary study using scores	7.33	8.33	1	1	9.88

Appendix K

Question 10: Evaluation of the influence between Size and Maintainability

P1: Is there any influence between the software characteristics Size and Maintainability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Size and Maintainability.
- **Intervention:** influence between the software characteristics Size and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Size and Maintainability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Size and Maintainability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Size and Maintainability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Size and Maintainability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Size and Maintainability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Size and Maintainability.

- **Intervention:** direction of the influence between the software characteristics Size and Maintainability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Size and Maintainability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Size and Maintainability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Size and Maintainability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Size and Maintainability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Size and Maintainability.
- **Intervention:** intensity/rate of the influence between the software characteristics Size and Maintainability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Size and Maintainability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Size and Maintainability.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Size and Maintainability, that is, in

which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Size: size, Function Points, Use Case Points, Requirement

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

High Level Design (or Design or Analysis)

Size: size or Classes or Methods per Class

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Low Level Design (or Design or Detailed Design)

Size: size or Key Classes or Support Classes or Methods per Class or Subsystems

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Size: size or LOC or lines of code or source lines of code or methods per classes

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category

- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or "Function Points" or "Use Case Points" or "Requirement")

and

(maintainability or "Defects" or "Diagnostic of Defects" or "Removal of Defects")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition"

or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or Classes or “Methods per Class”)
and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or “Key Classes” or “Support Classes” or “Methods per Class” or “Subsystems”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(size or LOC or “lines of code” or “source lines of code” or “methods per classes”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Size and Maintainability):

Paper	Yes	No	Not conclusive

For P1.1 question (Direction of the influence between the software characteristics Size and Maintainability):

Paper	A→B	B→A	Not conclusive

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Size and Maintainability):

Paper	Range/Means of the Rate	Not conclusive

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?					
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?					
2. Did the studies carry out a sensitivity or residual analysis?					

2.1. Were the resulting estimation models subject to sensitivity or residual analysis?					
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?					
3. Were accuracy statistics based on the raw data scale?					
4. How good was the study comparison method?					
5. Is it clear what projects were used to construct each model?					
6. Is it clear how accuracy was measured?					
7. Is it clear what cross-validation method was used?					
8. Were all model construction methods fully defined (tools and methods used)?					
Total primary study using scores					

Appendix L

Question 11: Evaluation of the influence between Periodicity and Size

P1: Is there any influence between the software characteristics Periodicity and Size of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Periodicity and Size.
- **Intervention:** influence between the software characteristics Periodicity and Size.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Periodicity and Size.
- **Outcome measure:** number of works that identify the influence between the software characteristics Periodicity and Size.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Periodicity and Size, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Periodicity and Size, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Periodicity and Size of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Periodicity and Size.

- **Intervention:** direction of the influence between the software characteristics Periodicity and Size.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Periodicity and Size.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Periodicity and Size.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Periodicity and Size, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Periodicity and Size of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Periodicity and Size.
- **Intervention:** intensity/rate of the influence between the software characteristics Periodicity and Size.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Periodicity and Size.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Periodicity and Size.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Periodicity and Size, that is, in which intensity/rate one software characteristic influences the other one, in addition

to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Periodicity: periodicity or Interval between Versions

Size: size, Function Points, Use Case Points, Requirement

High Level Design (or Design or Analysis)

Periodicity: periodicity or Interval between Versions

Size: size or Classes or Methods per Class

Low Level Design (or Design or Detailed Design)

Periodicity: periodicity or Interval between Versions

Size: size or Key Classes or Support Classes or Methods per Class or Subsystems

Coding (or Codification or Programming or Building or Construction or Implementation)

Periodicity: periodicity or Interval between Versions

Size: size or LOC or lines of code or source lines of code or methods per classes

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics

- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(size or "Function Points" or "Use Case Points" or "Requirement")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(size or Classes or "Methods per Class")

COPPE/UFRJ

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or “Interval between Versions”)

and

(size or “Key Classes” or “Support Classes” or “Methods per Class” or “Subsystems”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or “Interval between Versions”)

and

(size or LOC or “lines of code” or “source lines of code” or “methods per classes”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Periodicity and Size):

Paper	Yes	No	Not conclusive
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X		

For P1.1 question (Direction of the influence between the software characteristics Periodicity and Size):

Paper	A→B	B→A	Not conclusive
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Periodicity and Size):

Paper	Range/Means of the Rate	Not conclusive
Premraj, R., Shepperd, M., Kitchenham, B., Forselius, P. An Empirical Analysis of Software Productivity over Time. 11th IEEE International Software Metrics Symposium (METRICS 2005).	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to	0.5				

identify outliers and to assess distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	Yes				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0.5				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0.5				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.33				
5. Is it clear what projects were used to construct each model?	2				
6. Is it clear how accuracy was measured?	2				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	9.88				

Appendix M

Question 12: Evaluation of the influence between Complexity and Reliability

P1: Is there any influence between the software characteristics Complexity and Reliability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Complexity and Reliability.
- **Intervention:** influence between the software characteristics Complexity and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Complexity and Reliability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Complexity and Reliability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Complexity and Reliability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Complexity and Reliability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Complexity and Reliability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Complexity and Reliability.
- **Intervention:** direction of the influence between the software characteristics Complexity and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Complexity and Reliability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Complexity and Reliability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Complexity and Reliability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Complexity and Reliability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Complexity and Reliability.
- **Intervention:** intensity/rate of the influence between the software characteristics Complexity and Reliability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Complexity and Reliability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Complexity and Reliability.

- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Complexity and Reliability, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Complexity: complexity or Use Case

Reliability: reliability or Detected Defects or Corrected Defects

High Level Design (or Design or Analysis)

Complexity: complexity or Class Diagrams or Sequence Diagrams or State Diagrams or Package Diagrams or Activity Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Children per Class or NOC or Number of Children)

Reliability: reliability or Detected Defects or Corrected Defects

Low Level Design (or Design or Detailed Design)

Complexity: complexity or Class Diagrams or Sequence Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Coupling between Objects or CBO or Response for a Class or RFC or Lack of Cohesion in Methods or LCOM or Children per Class or NOC or Number of Children)

Reliability: reliability or Detected Defects or Corrected Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Complexity: complexity or Depth of Inheritance per Class or Coupling between Objects or Response for a Class or Lack of Cohesion in Methods or Children per Class or Cyclomatic Complexity per Method)

Reliability: reliability or Detected Defects or Corrected Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or “use case”)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(“Requirement Specification” or “Requirement Elicitation” or “Requirement Definition” or Analysis or “User Requirement” or “Requisite”)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "State Diagrams" or "Package Diagrams" or "Activity Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Children per Class" or NOC or "Number of Children")

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

(Design or "High Level Design" or Analysis)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Coupling between Objects" or CBO or "Response for a Class" or RFC or "Lack of Cohesion in Methods" or LCOM or "Children per Class" or NOC or "Number of Children")

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

(Design or "Low Level Design" or "Detailed Design")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(complexity or "Depth of Inheritance per Class" or "Coupling between Objects" or "Response for a Class" or "Lack of Cohesion in Methods" or "Children per Class" or "Cyclomatic Complexity per Method")

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

(Codification or Programming or Building or Construction or Implementation or Coding)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Complexity and Reliability):

Paper	Yes	No	Not conclusive
Basili, V., Briand, L., Melo, W. A Validation of Object Oriented Design Metrics as Quality Indicators. Technical Report, Univ. of Maryland, Dep. of Computer Science, College Park, MD, 20742 USA. April 1995.	X		
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X		

For P1.1 question (Direction of the influence between the software characteristics Complexity and Reliability):

Paper	A→B	B→A	Not conclusive
Basili, V., Briand, L., Melo, W. A Validation of Object Oriented Design Metrics as Quality Indicators. Technical Report, Univ. of Maryland, Dep. of Computer Science, College Park, MD, 20742 USA. April 1995.	X		
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Complexity and Reliability):

Paper	Range/Means of the Rate	Not conclusive
Basili, V., Briand, L., Melo, W. A Validation of Object Oriented Design Metrics as Quality Indicators. Technical Report, Univ. of Maryland, Dep. of Computer Science, College Park, MD, 20742 USA. April 1995.	X	
McCabe, T. J. A Complexity Measure. IEEE Transactions on Software Engineering, 1976.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes	No			
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0	0			
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5	0			
2. Did the studies carry out a sensitivity or residual analysis?	No	No			
2.1. Were the resulting estimation models subject to sensitivity or residual	0	0			

analysis?					
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0	0			
3. Were accuracy statistics based on the raw data scale?	1	0			
4. How good was the study comparison method?	0.33	1			
5. Is it clear what projects were used to construct each model?	2	2			
6. Is it clear how accuracy was measured?	2	1			
7. Is it clear what cross-validation method was used?	1	0			
8. Were all model construction methods fully defined (tools and methods used)?	1	0			
Total primary study using scores	7.88	4			

Appendix N

Question 13: Evaluation of the influence between Periodicity and Complexity

P1: Is there any influence between the software characteristics Periodicity and Complexity of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Periodicity and Complexity.
- **Intervention:** influence between the software characteristics Periodicity and Complexity.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Periodicity and Complexity.
- **Outcome measure:** number of works that identify the influence between the software characteristics Periodicity and Complexity.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Periodicity and Complexity, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Periodicity and Complexity, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Periodicity and Complexity of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Periodicity and Complexity.

- **Intervention:** direction of the influence between the software characteristics Periodicity and Complexity.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Periodicity and Complexity.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Periodicity and Complexity.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Periodicity and Complexity, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Periodicity and Complexity of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Periodicity and Complexity.
- **Intervention:** intensity/rate of the influence between the software characteristics Periodicity and Complexity.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Periodicity and Complexity.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Periodicity and Complexity.
- **Population:** selected works in question **P1.1**.

- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Periodicity and Complexity, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Periodicity: periodicity or Interval between Versions

Complexity: complexity or Use Case

High Level Design (or Design or Analysis)

Periodicity: periodicity or Interval between Versions

Complexity: complexity or Class Diagrams or Sequence Diagrams or State Diagrams or Package Diagrams or Activity Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Children per Class or NOC or Number of Children)

Low Level Design (or Design or Detailed Design)

Periodicity: periodicity or Interval between Versions

Complexity: complexity or Class Diagrams or Sequence Diagrams or Depth of Inheritance per Class or Depth of Inheritance Tree or DIT or Coupling between Objects or CBO or Response for a Class or RFC or Lack of Cohesion in Methods or LCOM or Children per Class or NOC or Number of Children)

Coding (or Codification or Programming or Building or Construction or Implementation)

Periodicity: periodicity or Interval between Versions

Complexity: complexity or Depth of Inheritance per Class or Coupling between Objects or Response for a Class or Lack of Cohesion in Methods or Children per Class or Cyclomatic Complexity per Method)

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title

- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(complexity or "use case")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "State Diagrams" or "Package Diagrams" or "Activity Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Children per Class" or NOC or "Number of Children")

and

(Design or "High Level Design" or Analysis)

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or "Interval between Versions")

and

(complexity or "Class Diagrams" or "Sequence Diagrams" or "Depth of Inheritance per Class" or "Depth of Inheritance Tree" or DIT or "Coupling between Objects" or CBO or "Response for a Class" or RFC or "Lack of Cohesion in Methods" or LCOM or "Children per Class" or NOC or "Number of Children")

and

(Design or "Low Level Design" or "Detailed Design")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(periodicity or “Interval between Versions”)

and

(complexity or “Depth of Inheritance per Class” or “Coupling between Objects” or “Response for a Class” or “Lack of Cohesion in Methods” or “Children per Class” or “Cyclomatic Complexity per Method”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or “software measure” or “software measurement”)

Results Summary

For P1 question (Influence between the software characteristics Periodicity and Complexity):

Paper	Yes	No	Not conclusive
Munson, J.C. Software faults, software failures and software reliability modeling. Information and Software Tecnology, 1996.	X		

For P1.1 question (Direction of the influence between the software characteristics Periodicity and Complexity):

Paper	A→B	B→A	Not conclusive
Munson, J.C. Software faults, software failures and software reliability modeling. Information and Software Tecnology, 1996.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Periodicity and Complexity):

Paper	Range/Means of the Rate	Not conclusive
Munson, J.C. Software faults, software failures and software reliability modeling. Information and Software Tecnology, 1996.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0				
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	Yes				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0.5				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.33				
5. Is it clear what projects were used to construct each model?	1				
6. Is it clear how accuracy was measured?	2				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	7.33				

Appendix O

Question 14: Evaluation of the influence between Maintainability and Reliability

P1: Is there any influence between the software characteristics Maintainability and Reliability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Maintainability and Reliability.
- **Intervention:** influence between the software characteristics Maintainability and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Maintainability and Reliability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Maintainability and Reliability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Maintainability and Reliability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Maintainability and Reliability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question**:

P1.1: What is the direction of the influence between the software characteristics Maintainability and Reliability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Maintainability and Reliability.
- **Intervention:** direction of the influence between the software characteristics Maintainability and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Maintainability and Reliability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Maintainability and Reliability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Maintainability and Reliability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Maintainability and Reliability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Maintainability and Reliability.
- **Intervention:** intensity/rate of the influence between the software characteristics Maintainability and Reliability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Maintainability and Reliability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Maintainability and Reliability.

- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Maintainability and Reliability, that is, in which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Reliability: reliability or Detected Defects or Corrected Defects

High Level Design (or Design or Analysis)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Reliability: reliability or Detected Defects or Corrected Defects

Low Level Design (or Design or Detailed Design)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Reliability: reliability or Detected Defects or Corrected Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Maintainability: maintainability or Defects or Diagnostic of Defects or Removal of Defects

Reliability: reliability or Detected Defects or Corrected Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source

- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics
- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(“Requirement Specification” or “Requirement Elicitation” or “Requirement Definition” or Analysis or “User Requirement” or “Requisite”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(maintainability or “Defects” or “Diagnostic of Defects” or “Removal of Defects”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Maintainability and Reliability):

Paper	Yes	No	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X		

For P1.1 question (Direction of the influence between the software characteristics Maintainability and Reliability):

Paper	A→B	B→A	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Maintainability and Reliability):

Paper	Range/Means of the Rate	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess distributional properties before analysis?	0				
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	No				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.33				
5. Is it clear what projects were used to construct each model?	1				
6. Is it clear how accuracy was measured?	1				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	5.88				

Appendix P

Question 15: Evaluation of the influence between Periodicity and Reliability

P1: Is there any influence between the software characteristics Periodicity and Reliability of the object-oriented software development process?

- **Problem:** to find research works that identify the influence between the software characteristics Periodicity and Reliability.
- **Intervention:** influence between the software characteristics Periodicity and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the influence between the software characteristics Periodicity and Reliability.
- **Outcome measure:** number of works that identify the influence between the software characteristics Periodicity and Reliability.
- **Population:** results of primary studies regarding object oriented software projects that describe the influence between the software characteristics Periodicity and Reliability, discussed in scientific papers.
- **Application:** fundamental to evidence the influence between the software characteristics Periodicity and Reliability, that is, which software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause-effect model based on the Laws of Software Evolution.

As from the results obtained in **P1** question, we search for responses for a **second question:**

P1.1: What is the direction of the influence between the software characteristics Periodicity and Reliability of the object-oriented software development process?

- **Problem:** to find the works that identify the direction of the influence between the software characteristics Periodicity and Reliability.

- **Intervention:** direction of the influence between the software characteristics Periodicity and Reliability.
- **Control:** not defined.
- **Effect:** the characterization of the direction of the influence between the software characteristics Periodicity and Reliability.
- **Outcome measure:** number of works that identify what is the direction of the influence between the software characteristics Periodicity and Reliability.
- **Population:** selected works in question **P1**.
- **Application:** fundamental to evaluate the direction of the influence between software characteristics Periodicity and Reliability, that is, in which direction one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support the cause effect model to software evolution based on the Laws of Software Evolution.

As from the results obtained in **P1.1** question, we search for responses for a **third question**:

P1.1.1: Which is the intensity/rate of the influence between the software characteristics Periodicity and Reliability of the object-oriented software development process?

- **Problem:** to find works that identify the intensity/rate of the influence between the software characteristics Periodicity and Reliability.
- **Intervention:** intensity/rate of the influence between the software characteristics Periodicity and Reliability.
- **Control:** not defined.
- **Effect:** to characterize the intensity/rate of the influence between the software characteristics Periodicity and Reliability.
- **Outcome measure:** number of works that identify the intensity/rate of the influence between software characteristics Periodicity and Reliability.
- **Population:** selected works in question **P1.1**.
- **Application:** fundamental to evaluate the intensity/rate of the influence between the software characteristics Periodicity and Reliability, that is, in

which intensity/rate one software characteristic influences the other one, in addition to subsidizing the relationship between these characteristics so as to support systems dynamics models to software evolution.

Selection of Sources

Specific Key Words

Requirement Specification (or Requirement Elicitation or Requirement Definition or Analysis or User Requirement or Requisite)

Periodicity: periodicity or Interval between Versions

Reliability: reliability or Detected Defects or Corrected Defects

High Level Design (or Design or Analysis)

Periodicity: periodicity or Interval between Versions

Reliability: reliability or Detected Defects or Corrected Defects

Low Level Design (or Design or Detailed Design)

Periodicity: periodicity or Interval between Versions

Reliability: reliability or Detected Defects or Corrected Defects

Coding (or Codification or Programming or Building or Construction or Implementation)

Periodicity: periodicity or Interval between Versions

Reliability: reliability or Detected Defects or Corrected Defects

Strategy of Information Extraction:

For each selected study, after the execution of the selection process, the researcher will extract the following data:

- Paper title
- Authors
- Source
- Paper type
- Category
- Context and application technology
- Software characteristics and metrics list
- Description of influences between software characteristics or metrics

- Description of the direction of influence between software characteristics or metrics
- Description of the intensity/rate of influence between software characteristics or metrics

Search

Due to the fact that the search environment is wide, it becomes necessary to restrict the search scope. This restriction varies according to a search string utilized and considers where the key words are looked up (all the text or its abstract).

For P1 question:

String Associated to Requirements Specification phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

(periodicity or "Interval between Versions")

and

("Requirement Specification" or "Requirement Elicitation" or "Requirement Definition" or Analysis or "User Requirement" or "Requisite")

and

("Software Characteristic" or "Software Metric" or "Software Development Project" or "software project" or "software measure" or "software measurement")

String Associated to High Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or "Detected Defects" or "Corrected Defects" or "System Availability")

and

COPPE/UFRJ

(periodicity or “Interval between Versions”)

and

(Design or “High Level Design” or Analysis)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Low Level Design phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(periodicity or “Interval between Versions”)

and

(Design or “Low Level Design” or “Detailed Design”)

and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

String Associated to Coding phase

(Relationship or Relation or Correlation or Influence or Dependence or Effect or Linkage)

and

(reliability or “Detected Defects” or “Corrected Defects” or “System Availability”)

and

(periodicity or “Interval between Versions”)

and

(Codification or Programming or Building or Construction or Implementation or Coding) and

(“Software Characteristic” or “Software Metric” or “Software Development Project” or “software project” or "software measure" or "software measurement")

Results Summary

For P1 question (Influence between the software characteristics Periodicity and Reliability):

Paper	Yes	No	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X		

For P1.1 question (Direction of the influence between the software characteristics Periodicity and Reliability):

Paper	A→B	B→A	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X		

For P1.1.1 question (Intensity/rate of the influence between the software characteristics Periodicity and Reliability):

Paper	Range/Means of the Rate	Not conclusive
Schneidewind, N. Measuring and Evaluating Maintenance Process using Reliability, Risk and Test Metrics. IEEE Transactions on Software Engineering, 1999.	X	

Evaluation of the Quality of the Primary Studies

Question	Quality Score per Paper				
	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
1. Is the data analysis appropriated?	Yes				
1.1. Was the data investigated to identify outliers and to assess	0				

distributional properties before analysis?					
1.2. Was the result of the investigation used appropriately to transform the data and select appropriated data points?	0.5				
2. Did the studies carry out a sensitivity or residual analysis?	No				
2.1. Were the resulting estimation models subject to sensitivity or residual analysis?	0				
2.2. Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary?	0				
3. Were accuracy statistics based on the raw data scale?	1				
4. How good was the study comparison method?	0.33				
5. Is it clear what projects were used to construct each model?	1				
6. Is it clear how accuracy was measured?	1				
7. Is it clear what cross-validation method was used?	1				
8. Were all model construction methods fully defined (tools and methods used)?	1				
Total primary study using scores	5.88				