S&G JOURNAL
ISSN: 1980-5160

A MODEL TO SUPPORT SOFTWARE QUALITY MANAGEMENT BASED ON SYSTEM DYNAMICS

Germano Fenner
germanofenner@gmail.com
Ceará Federal University - UFC, Fortaleza, CE, Brazil.

Alberto Sampaio Lima
albertosampaio@ufc.br
Ceará Federal University – UFC, Quixadá, CE, Brasil.

José Maria da Silva Monteiro Filho
jmmfilho@gmail.com
Ceará Federal University - UFC, Fortaleza, CE, Brazil.

Lincoln Souza Rocha
lincoln@dc.ufc.br
Ceará Federal University - UFC, Fortaleza, CE, Brazil.

José Antão Beltrão Moura
antao@dsc.ufcg.edu.br
Campina Grande Federal University - UFCG, Campina Grande, PB, Brazil.

José Neuman de Souza
neuman.souza@gmail.com
Ceará Federal University - UFC, Fortaleza, CE, Brazil.

ABSTRACT

Software development consists of a complex process, with numerous approaches, which suggest management proposals in well-defined phases, aiming at a production that guarantees products with the quality desired by its customers. Based on the various existing technology options and the growing demand in different areas of society, companies must adopt models that allow them to evaluate the process of software development and production. This paper presents a model to evaluate the software quality of organizations' business scenarios, based on systems dynamics. The model allows simulations of scenarios, evaluating the percentage of software development capacity, contributing to the improvement of management processes. During the research validation phase, a case study was carried out in three companies, and the results obtained were promising in terms of preference, usefulness and effectiveness.

Keywords: IT service management; software quality management; system dynamics; business-oriented IT management.

PROPPI / DOT
DOI: 10.20985/1980-5160.2021.v16n1.1625

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

1. INTRODUCTION

Globalisation and the massive expansion of internet access, associated with the growth of Information Technology (IT) areas in companies, have generated a continuous and growing demand for a specific service: software production. More and more, businesses depend on IT services involving software. Thus, there is a concern in managing and maintaining these services effectively (OGC, 2007). The managers of successful organizations recognize that IT is as significant for the business as any other part of the organization (Fernandes and Abreu, 2012).

IT services whose results can be measured (Lahtela et al., 2010) are required in mobile applications, in the work tools, as well as in the integration of the various types of equipment that can connect to the Internet. Software is the essential component of an IT service. Without effective software support, business processes (ABPMP, 2013) do not happen. When an event is triggered, whether in the form of a meeting appointment reminder, a control alert or simply obtaining electronic information, there is the use of programming code, where, depending on the degree of criticality, the slightest error can generate devastating consequences for the organisation. A software problem can compromise the image of an organisation and put its credibility in check. The prediction of process behaviours has been studied and evaluated with the proposition of several techniques (de Mello, 2010).

The current global scenario has indicated an increasing use of computer systems (software) integrated into the daily life of all sectors of society, reaching a large and diverse number of users. The search is for a software product with quality (Guerra and Colombo, 2009).

Critical points related to the software include processing capacity, accuracy and precision, among others. It should be considered that there is a wide variety of software segments that cover commercial applications, military simulations, educational products, among other types. When analyzing many items and considerations that involves the complexity of making software, it is possible to make the following questions:

- Which business aspects should be considered in software quality management?
- How to treat software quality in service management?
- What is the influence/contribution of the quality models in the business vision?
- What is the need of the evaluator/manager regarding the effectiveness of management support tools?

There is a need to analyze different business scenarios involving software quality so that managers can make effective decisions about what should be tested to meet the needs of stakeholders. The present research presents a new model to simulate and quantitatively estimate the evaluation of software quality, supporting management and decision making. The contribution of this work consists of the possibility of performing simulations for quality evaluation based on information from real business scenarios. The information generated is useful for decision making by quality and service managers.

The contributions generated by this research are practical (model design and implementation) and empirical (model validation). The following specific contributions can also be listed:

- Proposal of a business-oriented and innovative model that enables the realization of simulations to support decision making in software quality management;
- Alignment between business objectives and information generated by the model in the simulations.

This article consists of six sections. The first presents the introduction, whilst the second section presents the theoretical foundation that guided the research and shows some related works. In the third section, the proposed model is detailed. The methodology is presented in the fourth section. The case study carried out and the analysis of the results are discussed in the fifth section. The sixth section presents the research conclusions and future work.

2. THEORETICAL FOUNDATION AND RELATED WORK

In 1977 the first steps towards software quality were taken. McCall's model evaluated software in three levels: the first level considered the external view of the software, as seen by users; the second level evaluated the internal view of the software, as seen by the developer; and the third level considered the metrics used to provide a scale and methods for measurement. The Hewlett-Pachard model, on the other hand, based on McCall's model, considered two more quality attributes, functionality, a set of program features and capabilities, the generality of functions, in this case, overall system security, and usability, which took into consideration human factors, overall aesthetics, consistency, documentation.

This research was guided by the principles of Business-driven IT management (BDIM) (Bartolini and Stefanelli, 2011), involving the application of a set of models, practices, techniques and tools to map and quantitatively assess interdependencies between business performance and IT solutions. Theory and practice in BDIM are discussed from a decision-making perspective in Bartolini and Stefanelli (2011) and Lima et al. (2018). BDIM research is also presented in Sousa et al. (2018) and Nóbrega et al. (2014). The guides COBIT - Control Objectives for Information and related Technology (ISACA, 2014) and ITIL - IT Infrastructure Library (OGC, 2007) design guidelines for the classification and prioritization of investments that should be made in services. The Project Management Body of Knowledge - PMBOK Guide (PMI, 2012) guides the management of projects involving IT services.

Software quality

The demand for software solutions has grown considerably in recent years. Increasingly, there is a growth in the complexity of the solutions offered and greater demand in terms of product quality. According to Bartolini and Stefanelli (2011), quality reflects the degree to which a set of characteristics inherent to an object satisfies its requirements. The higher this degree, the greater the customer satisfaction. The concept of quality was initially applied in the industry. As time went by, other areas adopted the terminology, such as transportation (ISO 14.001 and 28.000), food (ISO 22.000), safety and health (ISO 45.001), among others.

In the IT area, efforts at the international level to ensure the quality have resulted in the creation of standards such as ISO 20,000 for IT services, the series of ISO 27,000, which deals with information security, in addition to several guidelines such as OGC (2007) and ISACA (2014), which address aspects related to the IT Corporate Governance and ABNT (2015), which deals with the aspects related to the IT infrastructure services. According to Guerra and Colombo (2009), software quality constitutes an area whose demand is growing significantly, because users increasingly demand efficiency and effectiveness, among other important quality characteristics for a product as special as the software. In parallel with the market demand, there is a national and international movement towards establishing standards in Software Engineering, as is the case of ISO - International Organization for Standardization, in the Software Engineering Subcommittee, and ABNT - Brazilian Association of Technical Standards.

In Brazil, since 1993, government, universities and industry initiatives have emerged in product quality and software processes. Since its creation, the Brazilian Association of Technical Norms (ABNT) has been expanding actions related to software quality. An example of this was the creation of the software subcommittee (ABNT, 2016), responsible for elaborating standards to generate Brazilian standards related to software quality.

If on one hand, the industry has been continuously improving its products and aligning criteria with the most rigorous standards in use in the world, the IT governance area needs to follow this evolution to offer support and conditions for an effective alignment between IT and business.

When considering the various ISO standards related to the subject from a software perspective, it is possible to identify that the subject of quality is quite extensive. The ISO/IEC 9126 standard refers to a set of attributes that have an impact on the software's capacity to maintain its performance level within the established conditions for a given period. This standard is divided into four parts:

1. ISO 9126-1:2001 - Emphasis on usage guidelines and software product quality characteristics;

2. ISO 9126-2:2003 - Refers to external metrics to determine the implementation rate of the functions defined in the requirements specification and the control of the occurrence of failures;

3. ISO 9126-3:2003 - Refers to internal metrics, verifies if the functions are adequate to what was specified, determines the number of expected failures, estimates the response time of the application and verifies if changes are properly recorded;

4. ISO 9126-4:2004 - Refers to in-use quality metrics based on effectiveness, productivity, security and user satisfaction.

The ISO/IEC 9126 standard was replaced in 2011 by ISO/IEC 25010 (ISO, 2016), which defines the quality characteristics that all software must-have. This replacement intended to bring even more benefits, such as compatibility and security. In addition to the set of ISO standards, the software industry has methodologies to help in the quality assurance process.

The Capability Maturity Model Integration (CMMI) is an international model of specific best practices for the development of software products (da Silva, 2016). As presented in Figure 1, the CMMI model is organized into five levels: 1 - Initial, 2 - Managed, 3 - Defined, 4 - Managed and 5 - Optimized. CMMI is based on the best practices for product development and maintenance. Companies that adopt this practice have greater chances of producing more stable, reliable and secure software since the model ensures that the company follows strict software development process standards.

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

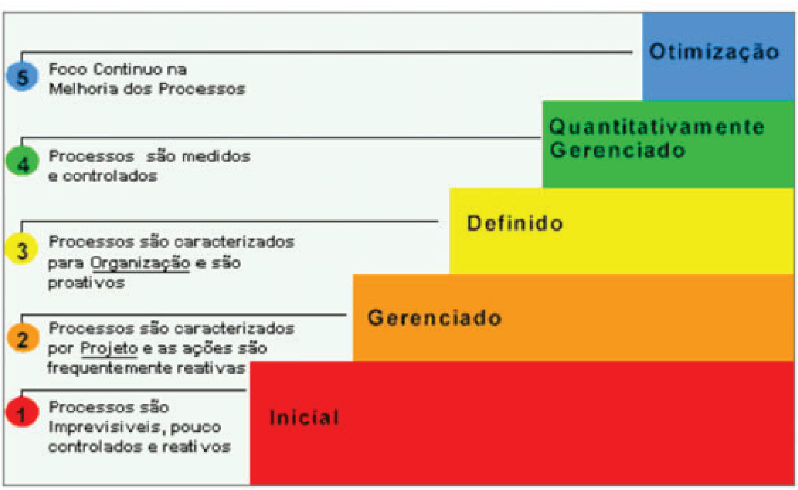
From the existence of CMMI at an international level, it was created in Brazil the model called Brazilian Software Process Improvement - MPS.BR (PROMOVE, 2016). It is a Brazilian program proposed by the SOFTEX organization, to develop and disseminate best practices in software development.

As shown in Figure 2, the MPS.BR model is organized into seven levels of maturity. Contrarily to the CMMI, the MPS.BR uses letters, from G to A, to determine the maturity level of the organization. These levels serve as a roadmap for continuous improvement.

As time went by, software production ceased to be activity-based only on intuition or on the experience of developers, having been the object of numerous studies. The aim is to offer models that enable production management and ensure the quality desired by its consumers. Despite the growing and acknowledged complexity involved in software development, some companies seem to be unaware that they need to create or even improve their internal processes (Mecenas and de Oliveira, 2005).

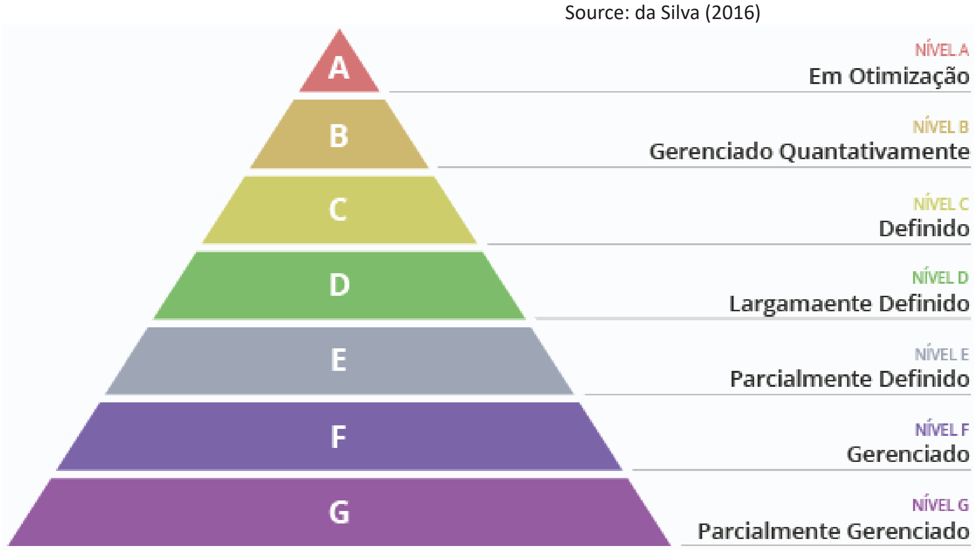
System dynamics

System dynamics (SD) theory, an aspect of systems theory, was created to model and analyse the complex behaviour of systems (Sterman, 2000). The method is based on the recognition that the structure of any system (often circular and closed), sometimes with delays between the relationships of its components, is as important in determining its behaviour as the individual components themselves. The technique



1. Initial - Processes are unpredictable, poorly controlled and reactive
2. Managed - Processes are characterised by Project and actions are often reactive
3. Defined - Processes are characterised for Organisation and are proactive
4. Quantitatively managed - Processes are measured and controlled
5. Optimisation - Continuous focus on process improvement

Figure 1. CMMI Model Stages
Source: da Silva (2016)



Level A: in optimisation
Level B: Quantitatively managed
Level C: defined
Level D: Broadly defined
Level E: Partially defined
Level F: Managed
Level G: Partially managed

Figure 2. MPS.BR Stages
Source: PROMOVE (2016)

allows modelling and analysing behaviour in complex scenarios of social systems. DS had more specific application in the industrial context in its first applications. However, the technique has been used for the development of models of the most diverse types of systems: social, economic and environmental, where a holistic view has a relevant role and the feedback links are fundamental to understand the complex interrelationships (Rodrigues, 1996).

DS is based on a direct structure of flows and stocks, designed to model systems with numerous variables and with delayed feedbacks between variables. A model for a process of demands received by an IT sector is presented in Figure 3. DS foresees the use of two types of models: influence diagrams or causal-link diagrams (qualitative models) and computer-based models or stock-flow diagrams (quantitative models). The former is considered the core of DS. Many studies adopting the DS methodology have created only influence diagrams. These diagrams are sufficient for the identification of the various feedback loops (reinforcement and balancing) and the factors influencing the flow of the modelled system. However, only through computer simulation (quantitative models) can one have a vision of the real dynamic effects of these feedback loops and a greater understanding of the system under analysis. In this way, the possibilities afforded using possible simulations subsidize the exploration of experimentation with the model (Sterman, 2000; Rodrigues,1996).

Importantly, SD models present a dynamic view of the performance of business systems (Sterman, 2000). The models are of the "activity on the arrow" diagram type rather than "activity in the box" diagrams like other notations and are especially useful in developing dynamic life cycle models that focus on the overall performance of systems and the impact of changing key variables that affect the overall performance. The models seek the business view (Morecroft, 2008) and are most commonly used when modelling an entire organisation or a line of business, and infrequently used for low-level workflow models.

Related work

A model that uses system dynamics is not static, since it presents the movement and reflects how the change in variables affects a process (Bezerra et al., 2014).

The work of Oliveira (2010) suggests a model quantify the value of an IT service, while the work of Lima et al. (2012) sought to quantify the percentage of quality of an IT service, considering that the same influences its value. In Bezerra et al. (2014) it is proposed a decision-making model for IT outsourcing. In Wan et al. (2011) it was presented a model that evaluates the dynamic capabilities and benefits management to assess IT resources management in public companies that need outsourcing. In Fernandes (2003) it was presented a proposal for incorporating the system dynamics methodology with the strategic planning steps, using the BSC model. The use of the system dynamics technique in problems related to software processes was proposed in Macachy (2007) and Abdel-Hamid (1989).

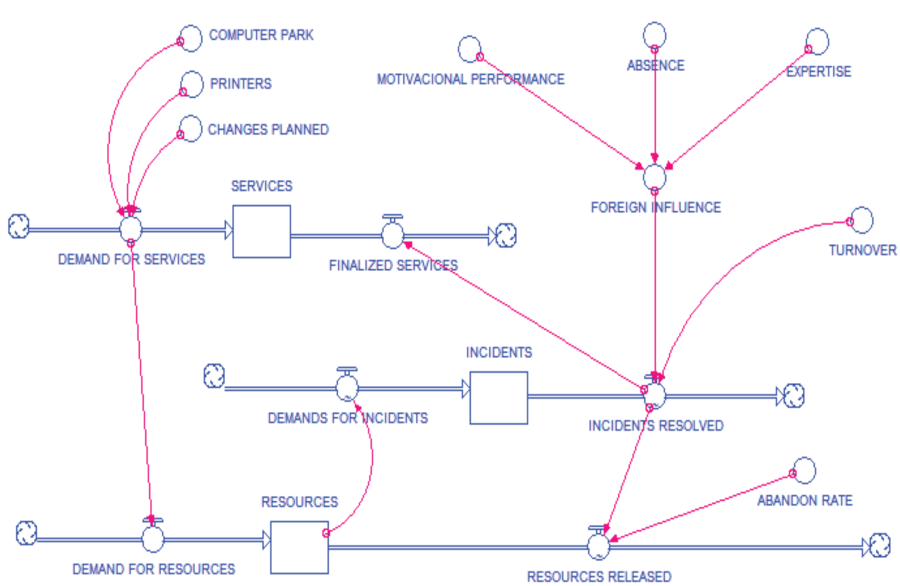


Figure 3. modelling with DS for an IT sector

Source: The author(s)

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

The proposed model

The quality of the software product can be evaluated in several ways. The assessments will vary according to the size of the company, business criticality and type of segment of the organization. There are different visions of software quality and its metrics. These metrics may vary according to the stage of the software life cycle.

The proposed model considers interactive activities based on the vision of the ITIL OGC (2007) guide. These activities can be executed in a continuous improvement cycle, similar to the PDCA (Plan-Do-Check-Act) cycle (Lima et al., 2012). The model uses ISO/IEC 25010 (ISO, 2016) as a reference. Product quality categorises product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, safety, compatibility, maintainability and portability). Each characteristic is composed of a set of related sub-characteristics (ISO, 2016). Figure 4 shows the evaluation characteristics and sub-characteristics of ISO/IEC 25010.

The proposed model considers each of the sub-characteristics indicated in Figure 4. The modelling of the scenario with the use of the principles of system dynamics, through the specialized software tool (Isee Systems, 2016), used in this research is presented in Figure 5.

The proposed model is generic and can be used by any company that needs to apply quality criteria to evaluate its software product having as reference the ISO 25010 standard. The model considers the most generic aspects of evaluation. Very specific characteristics of each company were not considered in this proposal.

Data entry

The user (manager) must provide various information so that the model can perform the simulations. Figure 6 shows the initial screen where the evaluator/manager provides the data, settings required for simulation.

In this stage, the manager must supply the largest amount of data possible so that the simulation reaches a higher degree of accuracy. The information must be configured in the following characteristics:

Figure 4. Features and subfeatures of ISO 25.10

Source: ISO (2016)

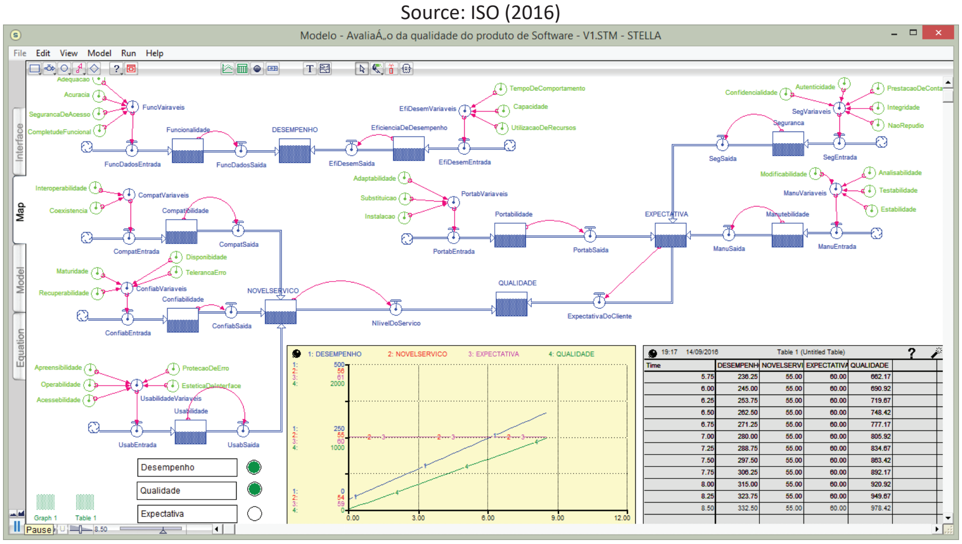


Figure 5. Modelling the sub-characteristics of ISO 25010

Source: The author(s)

Functionality

The model allows the evaluator to configure information considering the software functionality characteristic (Figure 7). This functionality addresses issues of adequacy, accuracy, access safety and functional completeness.

Performance efficiency

In addition to the functionality characteristics, the manager must provide information on the performance efficiency characteristic of the software (Figure 8). This characteristic considers points such as behaviour time, resource utilization and capacity.

The configuration parameter inputs presented show the process in the system dynamics model, having an interface that facilitates data entry, as presented in Figure 9.

Reliability and Compatibility

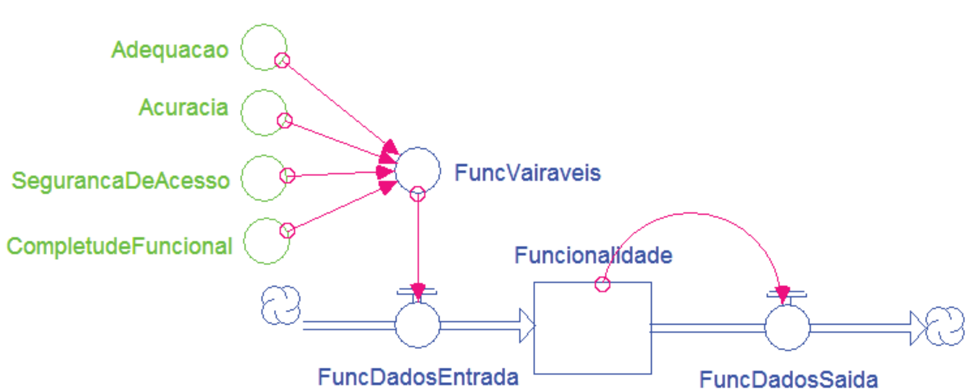
Software quality may vary considerably depending on: the compatibility feature, which assesses coexistence, interoperability; and reliability, which assesses the maturity, availability, error tolerance and recoverability features (Figure 10).

Usability and Security

In this step, the manager must inform parameters related to the characteristics of usability and security. The usability addresses sub-characteristics related to apprehensibility, operability, accessibility, protection from user error and aesthetics of the user interface, while security addresses the aspects of confidentiality, integrity, non-repudiation, accountability and authenticity (Figure 11).

Figure 6. Information configuration screen

Source: The author(s)

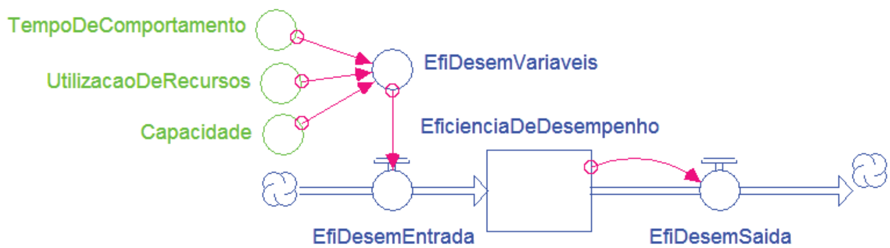


PT - EN
Adequacao - Adequacy
Acuracia - Accuracy
SegurancaDeAcesso - Security of Access
CompletudeFuncional - Functional Completeness
FuncVariaveis - FuncVariables
FuncDadosEntrada - FuncEntryData
Funcionalidade - Functionality
FuncDadosSaida - FuncOutputData

Figure 7. Configurations of functionalities features

Source: The author(s)

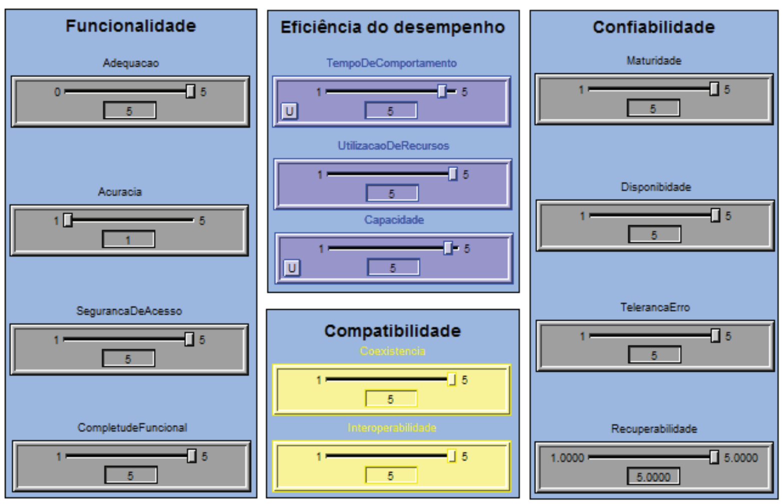
S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625



TempoDeComportamento - Time of Behaviour
UtilizacaoDeRecursos - Resource Usage
Capacidade - Capacity
EfiDesemVariaveis - EffiDevVariables
EficienciaDeDesempenho - Performance efficiency
EfiDesemEntrada - EffiDevInput
EfiDesemSaida - EffiDevOutput

Figure 8. Configurations of the performance efficiency characteristics

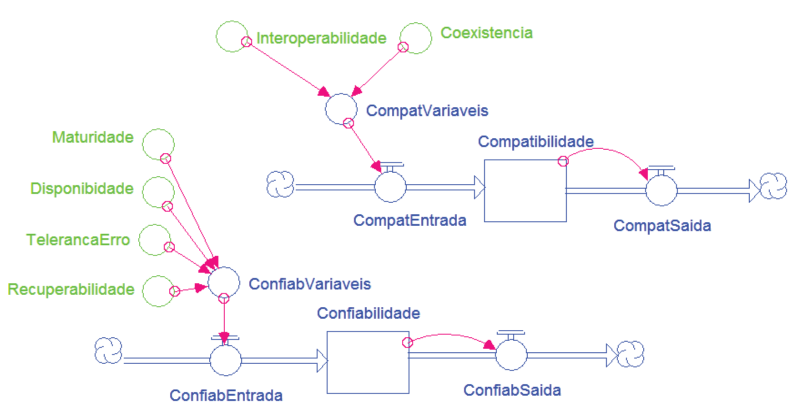
Source: The author(s)



Functionality	Performance Efficiency	Reliability
Suitability	Behaviour time	Maturity
Accuracy	Capacity	Availability
Access Security	Compatibility	Error Tolerance
Functional Completeness	Coexistence	Recoverability
	Interoperability	

Figure 9. Interface with the configuration parameters

Source: The author(s)



PT - EN
Maturidade - Maturity
Disponibilidade - Availability
ToleranciaErro - Error Tolerance
Recuperabilidade - Recoverability
Interoperabilidade - Interoperability
Coexistencia - Coexistence
CompatVariaveis - Compatible Variables
CompatEntrada - CompatInput
Compatibilidade - Compatible
ComaptSaida - CompatOutput
ConfiabVariaveis - Reliable Variables
Confiabilidade - Reliability
ConfiabEntrada - ReliInput
ConfiabSaida - ReliOutput

Figure 10. Configuration of the characteristics of Reliability and Compatibility

Source: The author(s)

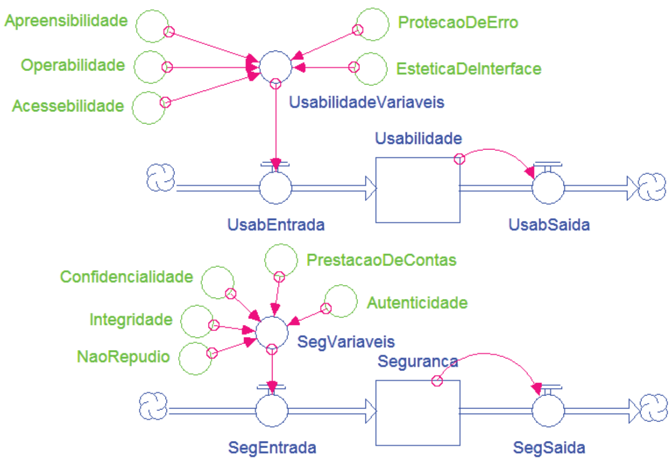
Maintainability and Portability

Finally, the manager will provide the last information for a software evaluation, which are the maintainability characteristics, which evaluate the aspects related to analyzability, stability, modifiability, testability, portability and, finally, the portability characteristics, which will consider the adaptability, installation and replacement aspects (Figure 12), thus completing the evaluation items recommended by ISO/IEC 25.010.

If the scenario evaluated does not contemplate the need for all the information, for example, the company's tests do not consider sub-characteristics such as coexistence, interoperability, non-repudiation and accountability, it will be enough for it to leave zero for these items and the system

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

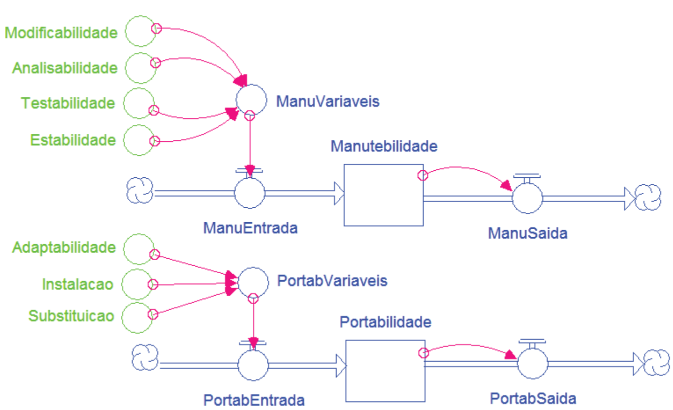
will disregard the data referring to these parameters to generate the results.



<u>PT – EN</u>
Apreensibilidade – Apprehensibility
Operabilidade – Operability
Acessebilidade – Accessibility
ProtecaoDeErro – Error Protection
EsteticaDeInterface - Interface aesthetics
UsabilidadeVariaveis - Usability Variables
Usabilidade – Usability
UsabEntrada – UsabInput
UsabSaida – UsabOutput
Confidencialidade – Confidentiality
Integridade – Integrity
NaoRepudio - no repudiation
PrestacaoDeContas – accountability
Autenticidade – Authenticity
SegVariaveis – Safe Variables
Segurança – Safety
SegEntrada – Safe Inputs
SegSaida – Safe Outputs

Figure 11. Configuration of usability and security characteristics

Source: The author(s)



Modificabilidade - Modifiability
Analisabilidade - Analysability
Testabilidade - Testability
Estabilidade – Stability
ManuVariaveis – Maintenance Variables
Manutebilidade – Maintainability
ManuEntrada – MaintInput
ManuSaida – MaintOutput
Adaptabilidade – Adaptability
Instalacao – Installation
Substituicao – Replacement
PortabVariaveis – Portable Variables
Portabilidade – Portability
PortabEntrada – PortabInput
PortabSaida - PortabOutput

Figure 12. Configuration of maintainability and portability

Source: The author(s) himself

3. METHODOLOGICAL ISSUES

Aiming at the evaluation of the quality monitoring model, a multiple case study was carried out in three companies specialised in software creation, development and deployment.

- The research carried out involved the following phases:
- Literature review;
- Development of the model;
- Meeting with the managers of the companies;
- Data collection;
- Use of the model with different scenarios in each company;
- Presentation and discussion of results;
- Tabulation of results;
- Validation of the model with the managers/evaluators;
- Dissemination of results.

The present research involved a triangulation between literature review, observation and the conduct of a case study Yin (2005). Throughout the process of verifying and validating the model, different calibrations were used as part of the design of the experiments, allowing an ideal environment to be created, real environments to be reproduced and alternative environments to be tested. The model improvement was defined from validation (external correctness) and verification (internal correctness) strategies, with calibration performed by running simulations.

The verification and validation process should be concerned with creating sufficient confidence in a simulation model so that its results are accepted by users and stakeholders (Radzicki, 2020). As for the topic of validity in simulation models with system dynamics (Barlas, 1989), it can be stated that there are no fully valid models because they

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

are all something smaller than the object or system being modelled. A priori, it is practically impossible to prove the correctness of a simulation model that aims to generate unexpected behaviour that has never been observed before (Rebs et al., 2019). Behaviour validation was performed to determine whether the behaviour patterns generated by the model would be close enough to the main patterns exhibited by the real system. During learning validation, there was an evaluation of the usefulness of the model, aimed at provoking reflection and broadening the understanding of the problem.

4. CASE STUDY AND ANALYSIS OF RESULTS

For reasons of information secrecy, the companies evaluated in the case study will be named Alpha, Beta and Omega. Each one of these companies selected a product to submit to the model.

Software quality assessment, seeking a common consensus (for instance, which metrics need to be considered) is still a challenging problem, especially when people are unaware that there are globally tested models available in the market. Many of these evaluations are based on qualitative criteria, which makes measurement difficult because they do not have numerical accuracy.

The proposed model aimed to be a tool to support decision making regarding software improvements, aiming to reach an acceptable level of quality. The proposed model has input parameters that are divided into two groups, the first one being the input parameters of the product and the second one the input parameters of the product.

The proposed model has input parameters divided into eight groups of characteristics, or categories: functionality, performance efficiency, compatibility, reliability, usability, safety, maintainability and portability.

For the model to be used, the manager will need to provide several parameters that characterize his software quality scenario to be achieved. There are parameters for internal and external demands, qualitative criteria (level of knowledge, motivation of resources) and subjective criteria such as abandonment rate, people turnover and absenteeism.

Figure 6 shows the parameterization screen of the simulation system that automated the model. Users gave scores through concepts that ranged from 0 to 5 with the following parameterizations: 0 - no evaluation, 1 - very bad, 2 - bad, 3 - average, 4 - good, 5 - very good.

In case one of the parameters is not part of the evaluated scenario, such as, for instance, user error protection, it is enough to leave zero in the information, and it will be disregarded in the evaluation process.

The proposed model generates results that seek to simulate the expected result of software products in terms of quality concerning the input data.

Case 1 – Alpha Company

The Alpha company has a staff of over 150 employees working specifically on the development of commercial systems, 10 of which are formed exclusively by the software testing and quality assessment team.

Regarding the ISO/IEC 25.010 standard, Alpha considers the criteria presented in Table 1 for its assessment.

The information in Table 1 was reported by the evaluator/manager of the area, who considered Alpha's current scenario for the reporting of information. By reporting these data to the model, the result generated with the current scenario is presented in the graph of Figure 13.

Considering Alpha's current scenario, there is a software quality deficiency in the characteristics of Efficiency and Reliability.

Alpha must, as a priority, work on aspects related to time behaviour, resource usage, capacity, maturity, availability, error tolerance and recoverability. This will make it reach a balance in all the quality characteristics of the software product, according to the guidelines of the ISO/IEC 25.010 standard.

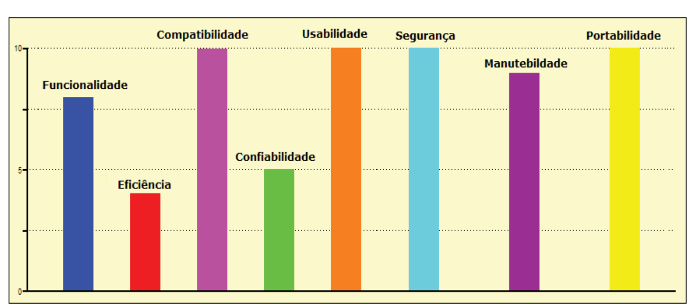


Figure 13. Evaluation scenario of company Alpha

Functionality - Efficiency - Compatibility - Reliability - Usability - Security - Maintainability – Portability

Source: The author(s)

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

Table 1. Scenario of Company Alpha

Parameters		Initial information from the evaluator	
IT Resources		150	
Quantity for testing		10	
Product Submitted		Accounting system	
Evaluation Criteria following ISO/IEC 25.010			
Functionality			
Appropriateness	5	Security of access	1
Accuracy	1	Functional completeness	1
Performance efficiency			
Time behaviour	2	Capacity	1
Resource utilisation	1		
Compatibility			
Co-existence	5	Interoperability	5
Reliability			
Maturity	1	Error tolerance	2
Availability	2	Recoverability	0
Usability			
Understandability	5	Accessibility	5
Operability	5	User error protection	5
User interface aesthetics	5		
Security			
Confidentiality	5	Non-repudiation	5
Integrity	5	Accountability	5
Authenticity	5		
Maintainability			
Analyzability	5	Modifiability	5
Stability	5	Testability	5
Portability			
Adaptability	5	Installation	5
Replacement	5	Security of access	

Source: The author(s)

Case 2 – Beta Company

Beta serves customers in the logistics and distribution segment. It has a staff of 80 employees, with 8 of these working specifically with software quality.

The information in Table 2 was reported by the evaluator/manager of the area, who considered the current scenario of the evaluated product of Beta. When these data were reported to the model, the result shown in Figure 14 was obtained.

If, on the one hand, the quality of the software product of Beta presents an excellent performance in the criteria of functionality, efficiency, compatibility, reliability and security, there is a deficiency in terms of usability, manageability and portability. Accessibility, user error protection, user interface aesthetics, stability testability and adaptability are some of the sub-characteristics that need to be improved to decrease this considerable difference between the software quality characteristics.

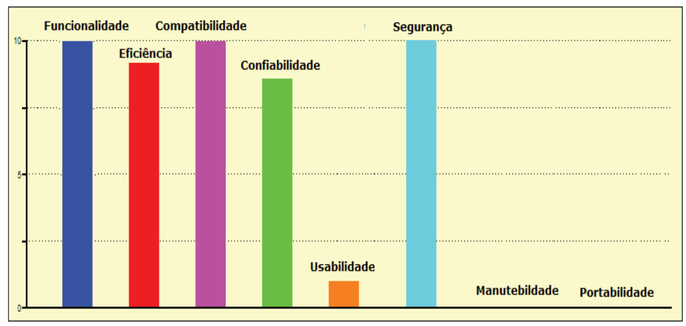


Figure 14. Evaluation scenario of the Beta company

Functionality - Efficiency - Compatibility - Reliability - Usability - Security - Maintainability – Portability

Source: The author(s)

Table 2. Beta Company Scenario

Parameters		Initial information from the evaluator	
IT Resources		80	
Quantity for testing		08	
Product Submitted		Accounting system	
Evaluation Criteria following ISO/IEC 25.010			
Functionality			
Appropriateness	5	Security of access	5
Accuracy	0	Functional completeness	5
Performance efficiency			
Time behaviour	4	Capacity	3
Resource utilisation	5		
Compatibility			
Co-existence	5	Interoperability	5
Reliability			
Maturity	5	Error tolerance	4
Availability	4	Recoverability	4
Usability			
Understandability	1	Accessibility	1
Operability	1	User error protection	2
User interface aesthetics	2		

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

Security			
Confidentiality	5	Non-repudiation	5
Integrity	5	Accountability	5
Authenticity	5		
Maintainability			
Analyzability	0	Modifiability	0
Stability	0	Testability	0
Portability			
Adaptability	0	Installation	0
Replacement	0		

Source: The author(s)

Case 3 – Ômega Company

The company Ômega has a medium size, with a board superior to 200 employees, of which 20 are allocated in the IT area, being 3 from the tests and quality area. The company has its software and outsourced ones. Its ERP system is outsourced. Some products, such as the customer service software, evaluated in this research, are of in-house development.

The information in Table 3 was reported by the evaluator/manager of the area, who considered the current scenario for software product quality evaluation. Unlike the companies Alpha and Beta, it is important to point out that IT is not the end activity of the company Ômega, a fact which was considered in the analysis of the results. By informing the company's data in the model, the result of the current scenario of product quality.

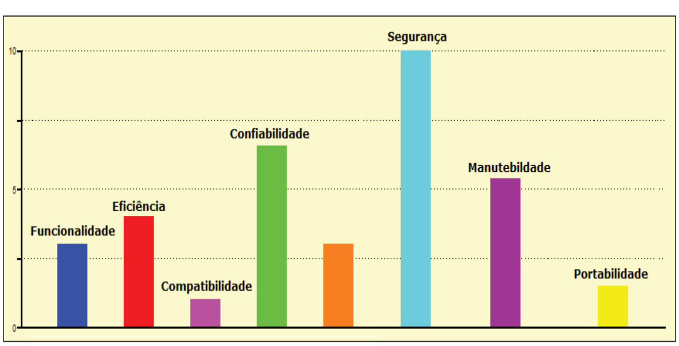


Figure 15. Evaluation scenario of the company Ômega

Functionality - Efficiency - Compatibility - Reliability - Usability - Security - Maintainability – Portability

Source: The author(s)

Table 3. Scenario of the Company Ômega

Parameters		Informação inicial do avaliador	
IT Resources		80	
Quantity for testing		08	
Product submitted		Accounting system	
Evaluation Criteria according to ISO/IEC 25.010			
Functionality			
Adequacy	5	Security of access	0
Accuracy	0	Functional completeness	0
Performance Efficiency			
Time behaviour	4	Capacity	3
Resource utilisation	5		
Compatibility			
Co-existence	5	Interoperability	5
Reliability			
Maturity	5	Error tolerance	4
Availability	4	Recoverability	4
Usability			
Understandability	1	Accessibility	1
Operability	1	User error protection	2
User interface aesthetics	2		
Security			
Confidentiality	5	Non-repudiation	5
Integrity	5	Accountability	5
Authenticity	5		
Maintainability			
Analyzability	2	Modifiability	2
Stability	3	Testability	2
Portability			
Adaptability	2	Installation	0
Replacement	1		

Source: The author(s)

As indicated in Figure 15, except for the characteristic safety, the company Ômega must work in all the other seven quality characteristics for software product oriented by the standard ISO/IEC 25.010.

In this scenario, it is indicated the revision of the software process. In this case, the adoption of models such as the Capability Maturity Model - Integration (CMMI) and the Brazilian Software Process Improvement (MPS.BR) may considerably influence the results in the next simulation.

5. CONCLUSION

The present research involved the triangulation between literature review, observation and case study in three real companies. A simulation model based on system dynamics was proposed as a solution to deal with the complexity inherent to the software product quality assessment process, aiming at improving the understanding of the problems involved and generating reflection points that may increase the management effectiveness. The model presented allowed the simulation of scenarios, aiming at supporting the evaluation of software products in companies. To test the model, a case study applied to the companies Alpha, Beta and Ômega was carried out.

During the case study, it was considered the work demand, alterations, tests and behaviour of the system in the last 6 months. The objective was to evaluate the product's characteristics, having as metrics reference the ISO/IEC 25.010 norm. The research team had access to records and reports that showed the number of calls related to product maintenance. In the behaviour validation, it was evaluated if the behaviour patterns generated by the model were close enough to the main patterns exhibited by the real system. The following aspects were evaluated: behaviour replication, visual adequacy and sensitivity of the output attribute to changes in the input variables.

The sensitivity tests of the input variables and the calibration of the proposed model were performed in three distinct moments: a) during the execution of the structural tests, to correct equations; b) during the behavioural tests, to eliminate parameters of little relevance and correctly calibrating the model and; c) during the execution of the learning tests, to generate uncertainty in the inputs and analyzing risks. The results obtained with the behavioural tests were presented to the users and subsidised the model refinement.

For the use, maintenance and evaluation of the model's effectiveness, activities were performed in the final phase of this research, through model use sessions with managers. The synthesis of strategies used for model verification and validation, executed during this research, is presented in Table 4.

The model validation process included interviews with ten managers working with software services and quality and, additionally, 11 managers from IT areas. The exercise included the presentation of the model and its results. The three hypotheses analysed were related to the usefulness, preference and effectiveness of the proposed model. The appearance validity exercise (Runerson and Host, 2009) conducted refuted the negative hypotheses. To perform the hypothesis testing, statistical inference (Berger, 1987; Casella, 2002) was used by running a Berger (1987) binomial statistical test at 5% significance, as shown in Table 5.

As a first threat to the validity of the research, restrictions from the budgetary and logistical point of view are pointed out. About construct validity, there are always doubts as to whether the variables are well understood by managers and it may not be possible to obtain simulation results that correspond to reality. It was possible to observe during the experiment involving the simulations performed, that companies prioritize aspects and characteristics that are more related to functionality, efficiency and safety.

Table 4. Strategies for verification and validation

Verification of model implementation	Assess whether the model implementation was error-free	Strategy adopted
Dimensional Consistency	Variable sizes are consistent and units are correct	Model testing and software implementation
Syntax validation	Behaviour - model management estimates are free of syntax errors	Software Testing and Implementation
Semantic validation	Behaviour - model management estimates are free of semantic errors	Software Testing and Implementation
Conceptual validation of the model	Evaluate model structure	Model testing and software implementation
Structure confirmation	Model estimates correspond to actual process relationships	Literature review, observation and appearance validity Runerson (2009)
Parameter confirmation	Evaluating model parameters against process knowledge	Literature review, observation and appearance validity Runerson (2009)
Extreme conditions	Evaluating model behaviour under extreme conditions	Literature review, observation and appearance validity Runerson (2009)
Sensitivity of the behaviour	Identification of process parameters where the model is highly sensitive	Literature review, observation and appearance validity Runerson (2009)

Source: The author(s)

S&G Journal
Volume 16, Number 1, 2021, pp. 19-33
DOI: 10.20985/1980-5160.2021.v16n1.1625

Table 5. Hypotheses to Test Model Appearance Validity.

Hypotheses	% who believe	Is there sufficient statistical evidence to accept the hypothesis?
Preference: managers preferred the model presented over the current form used.	90	Yes
Utility: Managers considered the model useful.	100	Yes
Effectiveness: Managers considered the model to be effective in supporting decision making	92	Yes

Source: The author(s)

Among the research limitations, we can highlight that there was difficult access to companies and professionals who knew the ISO/IEC 25.010 standard or who, at least, already had some kind of knowledge about it. It was noticed during the observation phase a somewhat vague knowledge of some of those involved regarding the development process models CMMI and MPS.BR. The perception of users and assessors regarding the assessment metrics also stands out. Some managers/evaluators, when asked about which metrics could be applied to assess the quality of the software product, demonstrated uncertainty, due to the limitation of their tests to the retrieval of records from the database and the accuracy of calculations.

As future work, it is intended to extend the model to perform a suggestion of points, characteristics and sub-characteristics that can be used for quality increment, aiming at greater user satisfaction.

REFERENCES

Office of Government Commerce – OCG, (2007), ITIL Vol. 3 (Information Technology Infrastructure Library)/Service Operation, London: TSO.

COBIT 5, (2014), "Modelo corporativo para governança e gestão de TI da organização", edição em português, São Paulo, Chapter with the permission of ISACA.

Guerra, A. C.; Colombo, R. M. T. (2009), "Qualidade de produto de software", *PBQP Software*, Rio de Janeiro.

Bartolini C.; Stefanelli, C. 2011), "Business-driven IT management, proceedings of IFIP/IEEE international symposium on integrated network management (IM), Vol. 1, pp. 963–969.

ABNT NBR ISO/IEC 9000:2015. (2015), Rio de Janeiro: ABNT. Subcomitê de software da ABNT (SC-21:10), http://docplayer.com.br/12433555-Subcomite-de-software-da-abnt-sc-21-10.html (acesso em set. 2016).

da Silva, M. C., (2016), "CMMI para iniciantes", http://www.linhadecodigo.com.br/artigo/1401/cmmi-para-iniciantes.aspx (acesso em set. 2016).

PROMOVE, "Como usar o MPS.Br", (2016), http://promovesolucoes.com/mps (acesso em set. 2016).

Mecenas, I.; de Oliveira, V. (2005), "Qualidade em software - uma metodologia para homologação de sistemas", Editora Alta Books, Rio de Janeiro.

Sterman, J. (2000), "Business dynamics: systems thinking and modeling for a complex world", Irvin-McGraw-Hill, Chicago, IL.

Rodrigues, A.; Bowers, J. (1996), "The role of system dynamics in project management", *International Journal of Project Management*, Vol. 14, pp. 213-220.

BPM CBOK Versão 3.0. (2013), 1ª ed. em português. Association of Business Process Management Professional, ABPMP, São Paulo.

Fernandes, A. A.; Abreu, V. F. (2012), *Implantando a governança de TI – da estratégia **à** gestão dos processos e serviços*, 3ª edição, Brasport, Rio de Janeiro:.

ISO/IEC 25010 (2011). https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en (acesso em set. 2016).

Lahtela, A.; Jantti, M.; Kaukola, J. (2010), "Implementing an ITIL - based IT service management measurement system", *Proceedings of fourth international conference on digital society*, St. Maarten, pp. 249-254.

Bezerra, T. R.; Moura, J. A. B.; Lima, A. S. (2014), "A system dynamics model to support strategic decision making on IT outsourcing: a case study at a state revenue agency in Brazil", Proceedings of NOMS 2014 2014 IEEE/IFIP Network Operations and Management Symposium, Krakow. 2014 IEEE Network Operations and Management Symposium (NOMS). pp. 1-4.

de Oliveira, J. A. N. (2010), Um modelo formal para avaliar o valor de negócio e sua aplicação no contexto de gestão e governança de TI, Tese de Doutorado, Universidade Federal de Campina Grande, PB.

Lima, A. S.; Sauvé, J.; de Souza, J. N. (2012), "Capturing the quality and business value of IT services using a business-driven model", IEEE transactions on network and service management.

Fernandes, A. C. (2003), "Scorecard Dinâmico - Em direção **à** integração da dinâmica de sistemas com o balanced scorecard", Tese, Universidade Federal do Rio de Janeiro, UFRJ, RJ.

Morecroft, J. (2008), "Strategic modeling and business dynamics – a feedback systems approach", Wiley, New Delhi, India.

Wan, J.; Zhang, H.; Wan, D. (2011), "Evaluation on information technology service management process with AHP", *Technology and Investment*, Vol. 2, No. 1, pp. 38–46.

Project Management Institute - PMI. (2012), "A guide to the project management body of knowledge – PMBOK guide", Pennsylvania, USA.

Runerson, P., Host, M. (2009), "Guidelines for conducting and reporting case study research in software engineering", Vol. 14, pp.31-164, Springer: Empiric Software Eng., DOI 10.1007.

Lima, A. S.; de Souza, J. N.; Moura, J. A. B.; da Silva, I. P., (2018), "A consensus-based multicriteria group decision model for information technology management committees", *IEEE Transactions on Engineering Management*, Vol. 65, pp. 1-17.

de Mello, R. F., (2010), "Sistemas dinâmicos e t**écnicas** inteligentes para a predição de comportamento de processos: uma abordagem para otimização de escalonamento em grades computacionais", 129p, Tese de Livre Docência, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, USP, São Carlos, SP

Isee Systems, (2016), "Stella software**"**. Disponível em https://www.iseesystems.com/ store/products/stella-architect.aspx (acesso em 17 ago. 2016).

Barlas, Y. (1989), "Multiple tests for validation of system dynamics type of simulation models", *European Journal of Operational Research*, Vol. 42, No. 1, pp. 59-87.

Macachy, R. J. (2007), "Software process dynamics", John Wiley & Sons.

Abdel-Hamid, T. K. (1989), "The dynamics of software project staffing: a system dynamic based simulation approach", *IEEE Transactions on Software Engineering*, Vol. 15, No. 2, pp. 109-119.

Yin, R. K. (2005), "Estudo de caso: planejamento e m**étodos**", Bookman, São Paulo, SP.

Berger, J. O.; Delampady, M. (1987), "Testing precise hypotheses", *Statistical Science 2*.

Casella, G.; Berger, R. L. (2002), "*Statistical inference*", 2nd ed., Duxbury Advanced Series, California.

Sousa, A. B.; Lima, A. S.; de Souz,a N.; Moura, A.; da Silva, A. C. (2018), "Business risk-based redundancy points identification in synchronous digital hierarchy optical networks", *IEEE Latin America Transactions*, Vol. 16, No. 8, pp. 2254-2260.

Nóbrega, D. A.; Fenner, G.; Lima, A. S. (2014), "A risk management methodology proposal for information technology projects", *IEEE Latin America Transactions*, Vol. 12, No. 5, pp. 643-656.

Rebs, T.; Brandenburg, M.; Seuring, S. (2019), "System dynamics modeling for sustainable supply chain management: a literature review and systems thinking approach", *Journal of Cleaner Production*, Vol. 208, pp. 1265-1280.

Radzicki, M. J. (2020), "System dynamics and its contribution to economics and economic modeling, system dynamics: theory and applications", Springe.

Received: 22nd Abr 2020

Approved: 9th Mar 2021

DOI: 10.20985/1980-5160.2021.v16n1.1625

How to cite: Fenner. G., Lima, A.S., Monteiro Filho, J.M.S., Rocha, L.S., Moura, J.A.B., Souza, J.S. (2021). A model to support software quality management based on system dynamics. Revista S&G 16, 1, 19-33. https://revistasg.emnuvens.com.br/sg/article/view/1625