METHODOLOGICAL GUIDELINES FOR SQA IN DEVELOPMENT PROCESS

An Approach Based on the SPICE Model

Anna Grimán, María Pérez, Luis Mendoza
Processes and Systems Department – LISI
Universidad Simón Bolívar
Caracas – Venezuela

Keywords: Software Quality, SPICE model, Software Development Methodology, Methodological guidelines.

Abstract: As far as international standards for promoting Software Process Quality are concerned, one of the most popular and accepted is ISO 15504 (or SPICE model). On the other hand, since a development methodology must guide the main activities in software development, it is necessary that this one fulfils some Quality Base Practices to guarantee a high-level product. The purpose of this research is analyzing a set of five methodologies widely used by developers, to identify its adjustment with respect to the aforementioned standard. This analysis allowed us: (1) determining the degree of alignment of these methodologies with respect to the SPICE model, and (2) proposing a synthesis of methodological guidelines, based on the best practices obtained from these methodologies, that supports the characteristics contained in the studied standard.

1 INTRODUCTION

In the world of Information Systems, one of the solutions being currently offered to develop quality products are the assessment models, both software product as well as software process. These models are highly useful because they deliver internationally accepted measurements. This research is limited to one of these dimensions: the process. To this objective, the SPICE model has been used. According to ISO/IEC (2004), Software Process Improvement and Capability dEtermination (SPICE) is a model designed for the assessment of software processes, which is included into the ISO documents and has evolved to a draft of the ISO 15504 Standard (Pressman, 2005). Pérez et al. (2001) have adopted the SPICE Quality Model, and have modified it to include efficiency and effectiveness aspects of development process.

With the emergence of standards and models, and the increasing commitment of organizations to quality, development methodologies have been trying to incorporate elements of Quality Control and Assurance into their processes. This has been evident for developers who use modern methodologies with a popularity that has exponentially grown from their inception.

However, these currently accepted methodologies do not always support the quality level proposed by this SPICE model for the software development process. This leaves a gap that has to be increasingly filled due to the certification needs faced by the software developing companies desiring to compete globally.

This paper presents the analysis of five methodologies according to base practices and quality characteristics of the SPICE model (adapted), to determine their degree of compliance for each methodology studied. At the same time, we identified their quality-supporting strengths according to the model within a set of methodological guidelines which can be incorporated into any existing methodology to improve quality level of the development process.

First, SPICE model is described followed by the method used for the methodology analysis. Then a brief description of each methodology under study is presented. Next, Assessment criteria and feature scoring for each methodology is described. Then, analysis of the scores with a summary of the guidelines derived from this study is presented.

Grimán A., Pérez M. and Mendoza L. (2006).
METHODOLOGICAL GUIDELINES FOR SQA IN DEVELOPMENT PROCESS - An Approach Based on the SPICE Model.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 269-275
DOI: 10.5220/0002462102690275

Finally, a report on the evaluation, the conclusions and recommendations for future research are described.

2 QUALITY MODEL BASIS

A wide range of models is available for quality assessment of the software development process, such as: Personal Software Process (PSP) (Humphey, 1997), CMM (Baltzer et el., 1993), BOOTSTRAP (Engelbart and Engelbart, 1990) and Software Process Improvement and Capability dEtermination –SPICE- (ISO/IEC, 2004). This research is based on an international standard: ISO 15504 (or SPICE).

SPICE provides a framework for the assessment of software processes. This framework can be used by organizations involved in planning, management, monitoring, control, and improvement of acquisition, supply, development, functioning, assessment and support of software (ISO/IEC, 2004).

Each instance of the process is characterized by a set of five (5) levels of process capability, each one being an aggregation of the enough practice assessments belonging to each specific level. Proper practice assessments are the basis for the assessment system.

However, SPICE model does not consider the characteristics inherent to the development of Software Systems, such as process efficiency and effectiveness. For this reason, the Quality Model used for this research is based on a model that integrates the Systemic Quality approach (Callaos and Callaos, 1996) with the features present in the SPICE process model. The model used here has a complex structure defined by level, where each higher level is made up of lower level elements (Pérez et al., 2001). This structure is described below.

Level 0: Life Cycles. As with the SPICE process model, three Life Cycles are considered. The inter-relationship between these cycles guarantees the quality of the Information Systems development process. These are: Primary Life Cycle (it is made up of two categories: Customer–Supplier and Engineering); Support Life Cycle (only contains the Support category); Organizational Life Cycle (it is composed of the Management and Organizational categories).

Level 1: Category. This model covers five categories of process, in accordance with SPICE. These are given Table 1.

Level 2: Processes. Each category has a set of characteristic processes that define the key areas to be met to achieve, ensure, maintain and control quality. Each process has an identifier associated with it that distinguishes it unequivocally. Table 1 shows the processes associated with each category.

Level 3: Principles. Each process has a Principle (P) associated, which is defined as an abstract and generic feature of the organization and serves as an indicator to determine the levels of quality in the development of Information Systems.

Level 4: Base Practices. A set of Base Practices (BP) is defined as a set of guidelines to be implemented by the organization in order to attain a principle. It should be noted that it was necessary to reasonably increase the number of BP present in the SPICE (ISO/IEC 2004) processes model, in order to maintain a balance in the dimension of the Systemic Quality (Callaos and Callaos, 1996). For more details about this model and how it has been evaluated through several case studies, see (Pérez et al., 2001).

Table 1: Processes for each category of model.

Category	Processes
Customer-Supplier (CUS)	CUS.1 System or Product Acquisition Process, CUS.2 Supply Process, CUS.3 Requirements Bidding Process, CUS.4 Operation
Engineering (ENG)	ENG.1 Development, ENG.2 Software and Systems Maintenance
Support (SUP)	SUP.1 Documentation, SUP.2 Configuration Management, SUP.3 Quality Assurance Process, SUP.4 Verification, SUP.5 Validation, SUP.6 Joint Review, SUP.7 Auditing, SUP.8 Problem Solving Process
Management (MAN)	MAN.1 Management, MAN.2 Project Management , MAN.3 Quality Management , MAN.4 Risk Management
Organizational (ORG)	ORG.1 Organizational Alignment , OGR.2 Change Management, ORG.3 Process Set-up Process, ORG.4 Process Evaluation Process, ORG.5 Improvement, ORG.6 Infrastructure Process , ORG.7 Measurement Process ,O RG.8 Re-use Process

3 RESEARCH APPROACH

In this research, the combination of two DESMET´s evaluation methods (Kitchenham, 1996) was used to identify the methodological issues presented in each development methodology. First, a Feature Analysis-Screen Mode was carried out. The evaluation is performed by a single person based on documentation only. It is the best approach for screening a large number of methods/tools and is often used as a first stage in a more complex evaluation exercise reducing a large number of possible methods/tool to a short-list of one or two candidate method/tools that can be evaluated in more detail. In this case the evaluator is responsible for: identifying the candidate methods/tools; devising the assessment criteria (i.e. the features to be assessed and the judgment scales) with or without specific help from potential tools users; compiling information about the method/tools; scoring each feature for each method/tool; analysing the scores; presenting a report on the evaluation.

Second, once obtained a set of results (methodological guidelines), a Qualitative Effects Analysis was followed to validate them. It provides a subjective assessment of the quantitative effect of methods and tools, based on a knowledge base of expert opinion about generic methods and techniques. The researcher requests an assessment of the effects of individual methods and/or the combined impact of several methods. This is quite a useful approach because the information held in a database containing expert opinion can be updated as and when the results of objective studies become available (Kitchenham, 1996).

Due to format restrictions, only the main results obtained after applying these methods are presented in next sections.

4 CANDIDATES DEVELOPMENT METHODOLOGIES

A software development methodology is a set of steps and procedures to be followed to develop a Software System (Whitten, Bentley and Dittman, 2004). The development methodologies studied and analyzed in this research are presented below.

Microsoft Operation Framework (MOF) is a collection of best practices, principles, and models (Microsoft, 2002). MOF is based on operations such as software development with a life cycle consisting of different phases working concurrently. MOF is composed of three core models: process, team and risk.

Microsoft Solution Framework (MSF) provides guidance for planning, building, and deploying a single project life cycle (Microsoft, 2000). People, processes and continuous risk management are key elements, besides technology, to achieve successful IT projects. MSF uses three models, through which it implements its principles: Risk Management, Team Management, and Process Management.

Rational Unified Process (RUP) is a Software Engineering process that provides guidance for the allocation of tasks and responsibilities within an organization. (Kruchten, 2003). It uses an iterative and incremental method based on the review of requirements and risks, as well as design, implementation, and validation, followed by a new review of these elements until finally the end product is obtained. The phases on which RUP is based for this iterative development are as follows: Inception, Elaboration, Construction and Transition.

Extreme Programming (XP) is a discipline of software development. In comparison to the heavy methods, XP is an agile method covering a set of techniques and common sense principles at extreme levels (Kendall and Kendall, 2002). It is based on the definition of four variables applicable to any software project: Cost, Time, Quality, and Scope. XP puts particular emphasis on small development teams that could increase if necessary.

Unified Process (UP) is based on a program engineering relying on principles approximated to software development (Jacobson et al., 1999). It consists of iterative principles, requirements and architecture base development. UP's life cycle consists of four consecutive phases: Inception, Elaboration, Construction and Transition. There is a fifth phase, Production, which supplements those previous ones.

5 ASSESSMENT CRITERIA

This section describes the criteria used to analyze each one of the methodologies under study.

In this case, the base practices included in each category associated to a development process proposed by SPICE (CUS, ENG and MAN) represent a methodological element (activity, technique, deliverable, etc.) to be satisfied by the candidate methodology.

SPICE SUP and ORG categories were not considered, because they are beyond the scope of the development methodologies. These categories contain processes establishing corporative business goals and developing process goods (values), products and resources, and support processes, whereas development methodologies are focused on the development process and manage this latter at the operational level.

The application of criteria established previously follows the algorithm mentioned below:

1. For each Process in each one of the aforementioned Categories is there any element (activity/artefact/role) in the methodology, which satisfies each one of the Base Practices (BP) associated at the lowest model level?
2. If there is a methodology that satisfies BP, the one adding most value to the process will be selected.
3. For each proposal of any methodological element, its selection is justified illustrating the criterion on which this selection was based.

In this sense, 89 criteria were proposed based on BPs, for examples:

C1: Methodology should identify clearly the existing resources and entities interacting with the system and/or software product, to determine its scope.

C2: Methodology should establish quality goals for the product and the processes that can be assessed throughout the project, preferably in a qualitative fashion, based on the status and the client-inherent quality requirements.

6 SCORING EACH FEATURE FOR EACH METHODOLOGY

Tables 2 shows a sample of the analysis applied to each methodology and each BP. It is worth mentioning that a total of 89 analyses were made, one for each criterion (BP).

Once this criteria analysis process was applied, the best practices were obtained (that will be transformed into guidelines), as well as a summary for each one of the analyzed methodologies with respect to the SPICE process they satisfy. Table 3 shows the relationship among them.

Table 2: Sample of the analysis applied to ENG category.

Category: Engineering (ENG)	Process: ENG.1 Development Process
Criterion: using the scale and scope of the software system or product to be developed as a basic definition of the activities and tasks required to develop the product or system in an effective, efficient and profitable fashion	Methodology analysis: MSF: Vision Document UP: Provisional Plan and Use-Case Model RUP: Vision Document. Supplementary Specifications. Project Plan
Elements selected as guidelines: **Vision document**, because it is useful to determine the scope of the project and therefore effectively define the activities needed as a function of the requirements. The Vision Document also contains all the system specification offering an overview of the basic requirements and characteristics of the project. **Provisional Plan**, because it can be used to clarify the requirements related to the initial goals, including scope determination. **Use-Case Model**, because with the captured requirements it is possible to determine an initial scope that will make it possible to establish activities. **Project Plan**, because it defines all basic activities and tasks to develop the system. **Supplementary Specifications**, because they specify the elements supplementing the contents of the Vision Document, related to the definition of activities and tasks.	

Table 3: Relationship obtained for the analyzed methodologies.

Candidate Method.	Quality Criteria Supported
MSF	CUS1, CUS2, CUS3, CUS4, ENG1, MAN1, MAN2, MAN4
MOF	CUS1, CUS2, CUS3, CUS4, ENG1, ENG2, MAN1,MAN2, MAN3, MAN4
RUP	CUS1, CUS2, CUS3, CUS4, ENG1, ENG2, MAN1, MAN2, MAN3, MAN4
UP	CUS1, CUS2, CUS3, CUS4, ENG1, ENG2, MAN1, MAN2, MAN3, MAN4
XP	CUS1, CUS3, CUS4, ENG1, ENG2, MAN1, MAN2, MAN3

Table 4: Examples of Activities by Discipline.

Discipline	Activity that satisfy the category	Artefact	Responsible
Analysis	Customer-Supplier	Functional specifications, Use-Case Model, List of Characteristics (potential requirements), Vision Document, Supplementary Specifications, Requirement Matrix, approved Development Plan, Product Acceptance Plan.	System Analyst, Architect, Use-Case Specifier, Use-Case Engineer.
	Engineering	Team Model, Organization Pattern, Project Plan including tasks network.	Project Leader
	Management	Metrics Plan (Metrics of the analysis model, the design model, the source code, and maintenance; and quality measurements such as: correction, maintenance easiness, integrity and use easiness); Test Model, Quality Assurance Plan.	Project Leader
Design	Customer-Supplier	Solution Design Model, Process Model, Use-Case Model, Exploratory Prototype, Product Acceptance Plan, Project Plan, Supplementary Specification, Vision Document, Business Case, Quality Assurance Plan, Test Plan.	System Analyst, Architect, System Integrator, Use-Case Engineer, Use-Case Specifier
	Engineering	Intermediate deliverables of the development components, Improvement Plan, Iteration Plan, Small Versions.	Component Engineer
	Management	Corporate Knowledge Basis of Change, Project Plan.	Process Engineer

7 ANALYSING THE SCORES

As a final result of the analysis conducted, the strengths identified to support each one of the base practices proposed by SPICE have been summarized. These strengths have been grouped into a set of guidelines that can easily be introduced in order to improve the current practices in any organization (Table 4 shows a sample of guidelines proposed).

To this objective, they have been organized according to the development disciplines they correspond to, presenting for each one the related SPICE categories, the suggested activity, the artefact they produce, and the role proposed as responsible.

A total of 34 artefacts were obtained for Analysis discipline, 27 artefacts were obtained for Design discipline, 35 artefacts were obtained for Implementation discipline; and 7 artefacts were obtained for Maintenance discipline. Figure 1 shows the percentage corresponding to each methodology based on their analysis.

According to Figure 1, most of the artefacts related to the Analysis discipline are represented by the RUP methodology, reaching 35% with 12 artefacts; UP methodology follows with 27% corresponding to 9 artefacts; the next methodology is MSF with 21%, 7 artefacts; the fourth is MOF methodology (12%, 4 artefacts), and finally, the proposal of own artefacts referenced by the different authors. Based on this analysis, it can be concluded that Rational Unified Process (RUP) is the methodology that is best identified with the SPICE elements in the analysis, with respect to the activities in the analysis.

For Design disciplines, Figure 1 shows a small difference with respect to the artefacts proposed by UP an RUP, since the percentage represented by each one is 22% and 25%, respectively, being RUP the methodology that is most identified with the elements of the SPICE Quality Model. It also shows the presence of the artefacts proposed by XP methodology with 4%, MSF with 11%, and MOF and the own artefact proposals taken from different references with 19%.

For Implementation discipline, according to Figure 1, the highest value, 40%, corresponds to UP, meaning that this was the methodology which most identified with the SPICE elements regarding the activities performed during implementation. The second methodology is RUP with 23%; then come the own proposals of artefacts taken from different references, 17%, MOF with 14%, and finally MSF with 6%.

Finally, respect to Maintenance, Figure 1 shows the absence of the methodologies presented in previous results such as MOF and MSF, which means that they did not identified with the SPICE elements regarding the activities corresponding to Maintenance. It can also be observed that RUP is the methodology with the highest value (57%), which represents more than a half of the artefacts

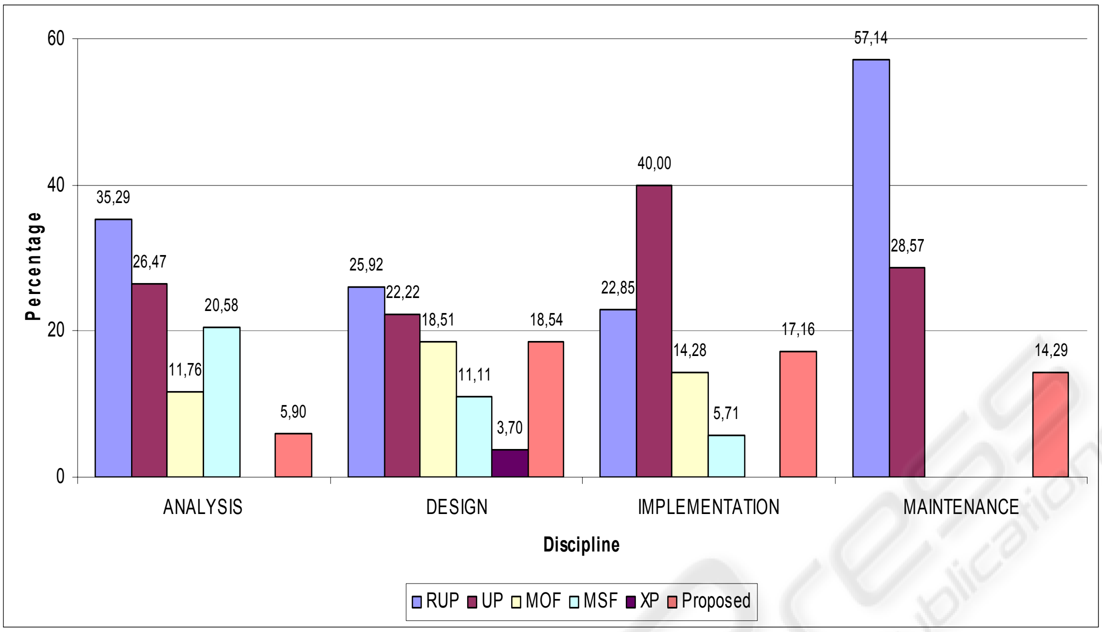

Figure 1: Artefacts per methodology in each phase.

proposed for this phase and a significant identification with the SPICE elements it contains.

8 REPORT OF THE EVALUATION

The SPICE model presents a more or less uniform distribution of practices related to the disciplines of Analysis, Design, and Implementation, which have practices associated to CUS, ENG, and MAN categories, whereas for Maintenance the practices observed are associated only to ENG.

RUP participation in almost all disciplines is significant, meaning that this methodology is the most aligned with the SPICE Quality model, followed by its predecessor, UP. However, it is worth mentioning that MSF has a moderate participation in the Analysis discipline.

According to our proposal was necessary in all cases to satisfy the standard completely; this means that these methodologies are not still fully mature as far as quality is concerned. The methodological guidelines presented can be analyzed according to their strengths and weaknesses: it takes into account the process efficiency and effectiveness; it makes it possible to ensure the systemic quality of the development process; and, since it is open, can be adapted to the requirements of any organization; a special emphasis is observed in risk and change management; the customer facilitates user involvement; it establishes disciplines and activities, and therefore can be incorporated into most of the methodologies based on the classic development cycle.

However, some of the activities defined could be divided into other activities to reduce their complexity, and many artefacts obtained are indicative of an object-oriented development, since most methodologies studied here are focused on this type of development. Finally, when each one of the proposed activities is developed, it is necessary to take into account the standards established by the organization where they are applied, including the definition of standards for the documents produced by Methodological Adaptation. In addition, techniques and tools incorporated into each activity could be fed back by the organization performing it, besides being implicit in each one of the proposed artefacts.

9 CONCLUSIONS

The methodological guidelines presented here are based on the analysis of different methodologies (RUP, UP, MOF, MSF, and XP), among which, according to the results obtained, Rational Unified Process (RUP) is the most comprehensive regarding the categories proposed by the SPICE process quality standard. However, with these results it is not possible to state that any of these methodologies is 100% aligned with the ISO 15504 quality standard.

As final result the importance of applying a methodological orientation in the development process can be established, since it helps to assure quality, including activities, artefacts, and roles associated to each process phase, to fill the loopholes of the existing methodologies and develop a strategy of competitive advantages and international certification.

REFERENCES

Baltzer, A., Daryanani, S., and Díaz S. (1993) Modelo de Capacidad de Madurez (CMM) del SEI, Madrid, España: Addison Wesley

Callaos, N. and Callaos, B. (1996) A systemic “systems methodology”. Proceedings of the 6th International Conference on Systems Research Informatics and Cybernetics, Alemania.

Engelbart, D., and Engelbart, C. (1990) Bootstrapping and the Handbook Cycle, Telematics and Informatics (7:1), January, pp. 27-32.

Humphey, W. (1997) Introduction to the Personal Software Process, Cambridge, Massachusetts: Addison Wesley Longman, Inc.

International Organization for Standardization (2004) Software Process Assessment, TR 15504, on-line, WG 10: Software Process Assessment, http://www.sqi.gu.edu.au/spice/.

Jacoboson, I.; Booch, J. and Rumbaugh, J. (1999) El proceso de desarrollo de software, España.

Kendall, J. and Kendall, K. (2005) System Analysis and Design. 6th ed. Prentice-Hall

Kitchenham, B. (1996) Evaluating Software Engineering methods an Tools. ACM SIGSOFT. Software Engineering Notes, 21, 1

Kruchten, P. (2003) The Rational Unified Process, USA.

Microsoft (2000) Microsoft Solution Framework (MSF), Retrieved form http://www.microsoft.com/msf

Microsoft (2002) Microsoft Operation Framework (MOF),. Retrieved from http://www.microsoft.com/mof

Pérez, M.; Rojas, T.; Mendoza, L. and Grimán, A. (2001) Systemic Quality Model for System Development Process: Case Study. Proceedings of Americas Conference on Information Systems, Boston.

Pressman, R. (2005) Ingeniería del Software: un enfoque práctico, España.

Whitten, J.; Lonnie, D.; Bentley, L and Dittman, K. (2004) Systems Analysis and Design Methods, (Kitchenham, 1996).Sixth Edition Irwin McGrawHill NY.