

THE INSPECTION TECHNIQUE APPLIED FOR THE VALIDATION OF A SPECIFIC DEVELOPMENT PROCESS FOR SCIENTIFIC SOFTWARE

J. O. Gomes¹, G. F. Moita¹, A. B. Moreira¹

¹Intelligent Systems Laboratory, Federal Center for Technological Education of Minas Gerais, Av. Amazonas 7675, Nova Gameleira, 30.510-000, Belo Horizonte - MG, Brazil (corresponding - gray@dppg.cefetmg.br)

Abstract. *The increasing demand for software has generated greater need for software development processes. The literature, on the other hand, does not present processes for the development of scientific software, and, in this way, in previous studies, a Specific Process for Scientific Software (SPSS) was proposed. It was conceived based on the Humphrey's Methodology, which defines that for the acquisition of a process it is necessary the execution of eight steps. Having in mind that the six first steps have already been concluded, the current work considers the execution of the seventh and eighth steps of the Methodology, which are the validation of the initial process and its later improvement. It is important to highlight the significance of these steps because, for a process to ensure its efficiency, it must be tested and validated. So, firstly, a webapp to automate and model the SPSS was developed, in order to help in its implementation. In the continuation of the study, SPSS was applied in real environments of scientific software development, to obtain results which allow them to be analyzed through a validation technique, by inspection, named VProcInsp. At the end, in spite of some inconsistencies, it was concluded that the SPSS had indicated that it can really contribute in the development of academic scientific software. The results are shown and discussed in this the article.*

Keywords: *Software Processes, Process Validation, Scientific Software.*

1. INTRODUCTION

With the increasing evolution of its power of storage and processing, and diminishing costs, computers are more and more present in the society nowadays, which, in turn, is seen each day more dependent of these machines and their software. The latter have presented a significant increasing in its internal complexity, which results in a bigger error incidence and, consequently, and also a fall in its quality [8].

To manage the larger internal complexity of the software, software engineering techniques began to be used to allow a better control upon the development process, and hence, to avoid its quality can be undermined. These techniques – known as Software Development Processes – when well used, allow software developments of high reliability and quality. A

software development process is a set of activities, partially ordered, with a purpose of obtaining a product of software. The question surrounding the software development process is mainly the stability, control and organization that it can supply to the product development activity.

Even though there are, in the literature, many researches about Development Processes, none reaches all kinds of projects, because usually each project has its own and different needs and, as a consequence, there is the necessity of use of a "almost-customized" process. Because of this, there is not a perfect process model for all the applications.

Within this subject, a topic that deserves careful attention is the study of the development process for scientific software of academic nature. To do this, an investigation about specific processes for academic environment was made and it could be verified that the existent processes have their main focus on the production of conventional software [8]. Consequently, from this findings, a new process - the so-called SPSS (Specific Development Process for Scientific Software) - was proposed, which is shown in the Section 2 of this paper. This process was conceived and defined from the six first steps of the Humphrey's Methodology [5]. In this work the two remain steps of the Methodology are carried out.

The seventh step consists in the validation of the initial process and the eighth and last step is the correction and improvement of the process. The Validation of the Initial Process is an important matter and can answer to a natural worry of the projects management – the guarantee that the development process guide the participants in an efficient and correct way during the creation of the software. Excessive bureaucracy or an ambiguous orientation can confuse, instead of guiding, the development in the product creation cycle.

In this context, this article has the purpose of showing the results from the SPSS validation, through an inspection technique, that aims to investigate if the new process provides effectiveness to the scientific software development of academic nature.

The following section shows some general concepts about software development processes, and specially the SPSS, which motivated this research. Section 3 shows the processes validation and in Section 4 the validation of the SPSS is discussed. Section 5 points out the results of the validation. Finally, the Section 6 shows the final considerations about the validation of the SPSS.

2. SOFTWARE DEVELOPMENT PROCESSES

In every software development, meeting the requirements, accomplishing deadlines, adjusting real costs and resources and complying with many other specificities demanded to guarantee a good quality software, are not simple tasks to be accomplished. It is specially necessary an extremely detailed monitoring of the tasks which evolves its entire construction, since the creation until its installation at the final client. From this, the Software Engineering come up with a specific technique to guarantee quality and trustworthiness in the execution of the tasks, named Software Development Processes: a set of systematized activities, partially ordered, which make an effort to acquire the software product.

With the time, it has been proved that, when the development process is focused, it can guarantee the quality of the software since the beginning of its manufacturing, because it can be controlled step by step and its quality can be tested before it reaches the client.

The software processes are the basis for the managing control of software projects and it establishes the context in which the technical methods are applied, the working products (models, documents, data, reports, forms) are produced, the marks are established, the quality is assured and the changes are adequately managed. Pressman [9] still highlights that the process is the main point that keeps the union of the technology layers. Figure 1 shows the Software Engineering layers.

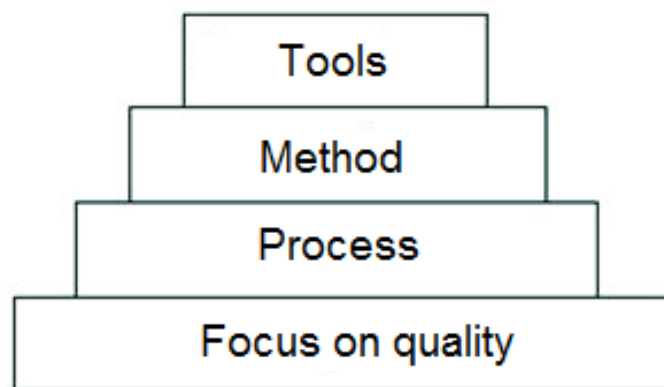


Figure 1. Software Engineering Layers [9]

Even with the known benefits obtained with the use of the countless existing software development processes, the application and introduction of these techniques are still complex tasks, highly dependent on the environment in which they are placed in. Furthermore, there are quite a few processes that do not fulfill the expectations in all the extents and there is not a model that can be regarded suitable process for all the applications.

In this context, an issue that deserves special attention is the study of the scientific software development process of academic nature, detailed ahead in the study of the SPSS (Specific Development Process for Scientific Software).

2.1. SPSS

The SPSS considers the term scientific software as “software developed by researchers in their scientific research projects”. A vast amount of these projects are of scientific nature or, in other words, research projects of scientific initiation, master’s degree, doctoral and postdoctoral stages, and others, which rely upon software creation to help in their researches. [10].

Motivated by the lack of processes for the construction of scientific software of academic nature, the SPSS was thought as a possible alternative. It was initiated by Purri [10] and advanced with the work of Pereira Jr. [8], both during their M.Sc. studies at the Computing and Mathematics Modeling Program at the Federal Center for Technological Education

of Minas Gerais - CEFET-MG, Brazil. As already mentioned, Humphrey's Methodology was used for the definition of the process.

The initial approach was strongly influenced by the Unified Process (UP) and by the Extreme Programming (XP) methodologies. The life cycle used at the SPSS was defined as iterative and gradual. In each of the iterations, the following phases are contemplated: Planning, Development, Testing and Deployment, and consequently the creation of its respective artifacts, as shown in the spiral depicted in the Figure 2 below.

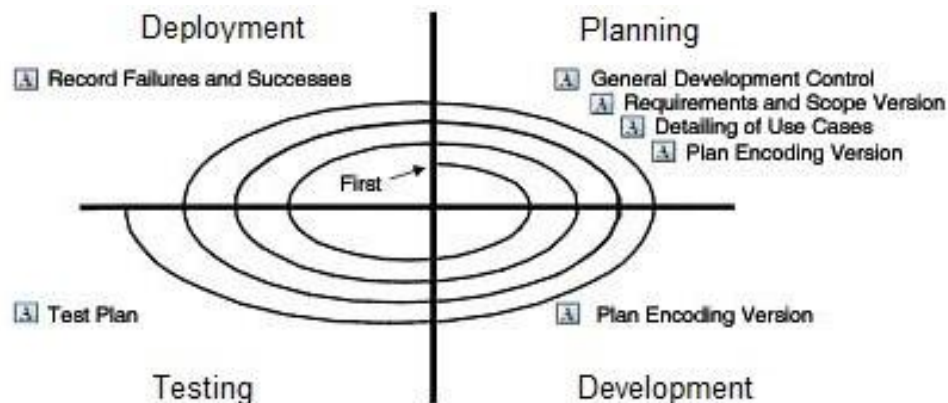


Figure 2. Life cycle proposed by the SPSS.

The four phases of the development are described below:

- **Planning** – In this phase, each iteration must be planned to be small, in order to maximize the development control. All the iterations must be planned to be simple, but it does not allow to forgetting any important detail to the development.
- **Development** – Based on the fast and reduced planning made previously, a version of the iteration in development is created, observing the defined codification standards.
- **Testing** – After the codification is completed, each version must pass the integration and unit tests. The unit tests are suitable to guarantee that this version works according to the plan. After the approval of the unit tests, the integration process is run, so that the compatibility of the new code with the existing version is guaranteed.
- **Deployment** – After passing the integration and unit tests, the new version must be incorporated into the stable project, so that the development of the application can be proceed.

An artifact is a product (or more activities) within the context of development of software or a system. In the SPSS, it is defined as self-explained document and is used to customize the information related to the software development. six artifacts make up the SPSS: one for the general control of the process and five related to the specific area of development:

- **General Development Control** – Known as “umbrella” artifact, it is specific for the general control of the software development. In this document, the registration of the

information regarding the development, the projected modules, the project architecture and the changes made in each artifact in charge of having control of the development process as a whole are recorded. This artifact is created only once during the whole development of a specific project.

- **Version Requirements and Scope** – Used and/ or modified in the Planning Phase, which is the first stage of the SPSS development cycle. Here, the developer must register what has been planned for that iteration. Each iteration of the life cycle must generate a Requirement Document and a Version Scope.
- **Detailing of the Use Cases** – Based on the information collected and documented in the General Development Control and Version Requirements and Scope, this artifact can be modified even in the Planning Phase. It includes the use case diagrams and their specifications. Each iteration must, ideally, produce a complete artifact of Detailing of the Use Cases.
- **Version Codification Planning** – Can be modified in the Planning Phase and must be consulted and, if it is necessary, modified in the Development Phase. It is particularly important for the success of the development, because it defines all the details regarding the development of the source code itself, including the classes diagrams. Ideally, each iteration must yield a complete Version Codification Planning artifact.
- **Tests Planning** – The generated source code serves as a basis for the application of the tests which will be documented in the Tests Planning artifact. In this artifact, the developer has the possibility to document the whole test process that has been run. As described in the SPSS life cycle, unit tests must be made and afterwards integration tests, to guarantee to the researcher that the built version is working.
- **Failures and Successes Recording** – Used at the final phase of the life cycle of the SPSS. After a complete iteration, the researcher should have faced difficulties and possible failures and, also, should have achieved successes in its development. This artifact allows the developer to register the failures and successes obtained in the development of each version. A Failure and Successes Recording document must be generated for each iteration of the life cycle.

These artifacts constitute an understructure of the development and the researcher/developer has the option of omitting the optional information in these documents, what makes the SPSS a highly customizable process. However, some parts of the artifacts are mandatory, with the intention of guarantying the minimum documentation needed for the control of the development process.

To continue with the development of the SPSS, the present research is mainly concentrated in the implementation of the seventh step of the Humphrey's Methodology, with the intention of completing and publishing of the process. This step tries to assure the process "correctness" or, in other words, to verify if SPSS produces positive results during the conception of a scientific software of academic nature. With the purpose of achieve this, firstly, the SPSS process was automated through the creation of the webapp *PESC-V.1.0*. This tool is intended to model, systematize and enable the centralization of the artifacts produced by the process during the development of a determined project. With the automation, the

understanding, as well as the adaptation and the management of the development process, become easier, enabling, in a simple way, the analysis of the data generated with the execution of the SPSS during the validation process.

3. PROCESSES VALIDATION

For a software process to guarantee the efficacy of the product, it must be validated with the intention of identifying possible errors during its modeling, because the errors could turn the software development into a chaotic process. The validation can accomplish many objectives, and, among them, to guarantee the consistency of the formal process, because the execution of the process must be compatible with the process model described. And this, consequently, increases the reliability of the results of any analysis executed with the formal model. The process validation can also be used as a certification tool, discovering differences between the desired behavior and the real one. Finally, the validation can reveal when a process may need to evolve in order to accommodate activities and requirements of a new project [1].

However, to validate a development process is not a simple task. One of the problems is that the validation is conceptually complex, because it refers to a large scale of questions, many times subjective ones. The few researches in this area are complex and dependent of the context in which they were applied. Moreover, it was not found any issued documentation proving or demonstrating the applicability of the validation techniques in a relevant quantity of real environments. Cook and Wolf [1], and Moor and Delugach [6] adopt, as a principle for the processes validation, the comparison between model and practice, but they make it in different ways. Moor and Delugach [6] adopt a formalism based on the Graphs Theory, where model and practice are represented and compared.

The method adopted by Cook and Wolf [1] is to measure the differences between the model and the practice is the String Edition, where the number of insertions, exclusions and symbolic substitutions (*tokens*), needed to transform a string (sequence of events of the process/practice) in another are analyzed. The ideas of Cook and Wolf [1] and Moor and Delugach [6] are capable to produce interesting results to validate a process.

However, the use of the concepts of the Graphs Theory or String Edition can make the process validation of an agile development impracticable, because they preach flexibility and easiness of use. Huesh [4] utilizes the concept based on simulation and suggests the process validation through a game of projects management. The game is represented by a scenery of a software plant containing schedule and resources related to the project in development, and it has a determined process associated to the development. In this scenery, it is possible to make the simulation and verify how the process treats a particular situation [7].

Ferreira and Moita [2] also suggested an approach to compare the formal model and the execution of the process, but they adopted concepts which based themselves in the software inspection. The method suggested has as the main motivation to simplify and to give flexibility to the comparison of the formal model and its practice. This approach wants to be: (i) Generic; (ii) Simple; (iii) Extensible. Through the application of the technique to a case study [2], it could be demonstrated that there are evidences to these three characteristics. In

addition, this technique confirms that it can be applied in different processes with a low cost for the involved personnel regarding to any specific knowledge and bureaucracy in its use.

New processes as the SPSS are created with the intention of supplying quality to the software development, without the application of a large amount of human or financial resources. If the process validation technique does not follow this intention, it can block its use. From the work of Gomes et al. [3], the procedure that can be better applied to the SPSS validation is the proposal by Ferreira and Moita [2]. The description of this technique is presented in the following section.

3.1. Inspection Technique applied for Development Process Validation

In Ferreira and Moita [2], the software development process inspection validation technique (VProcInsp) is performed by means of the following tasks: (1) Determine the software artifact which must be evaluated; (2) Identify the profiles and roles of the people who must be involved in this inspection; (3) Define the evaluation technique used to identify the faults in the artifact to be evaluated; (4) Specify the process compound by the activities to be executed.

Below, all the tasks needed to the application of the technique are described:

- *Determination of the Artifacts:* In the VProcInsp, the artifact can be considered: documentation, the electronic communication, the version control software, compilers, and other software information [2].
- *Definition of the Role and the Profile of the Validation Participant Team:* The VProcInsp is made of four roles with well-defined characteristics of acting: (1) **Moderator** – Independent person, individual and impartial who coordinates the planning process, preparing, the inspection behavior, and meetings (2) **Author** – a person or a team in charge of the formal model of the software development process. The responsibility of the author is to explain and to introduce the process documentation (3) **Inspector** – responsible to identify, analyze and compare the process formal model, with the process executed in the practice. The inspector does not have the function to find errors in the artifacts, but to analyze if what was proposed by the process is being done. (4) **Analyst** – a person or a group of people involved in the software development process, who are doing use of the process under validation.
- *Use of the Inspection Technique:* The main purpose of this technique is to validate the artifacts by a simple and methodic way and, this way, verify if the process is achieving what was proposed. According to Ferreira and Moita [2], the most appropriated technique is the Checklist. They highlight that the idea is not create a fixed checklist, but to show a template with directives to allow the moderator to create the checklist in accordance with the needs of each development process. The inspection is used to compare products and the flow of the process actually executed with the products and the flow of the formal model.

Figure 3 shows the model proposed by the VProcInsp and the description of this model is made subsequently.

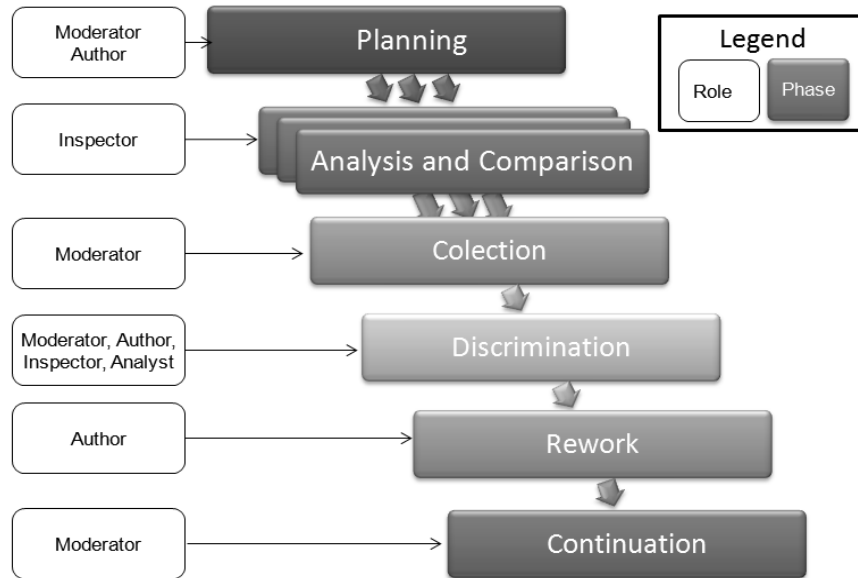


Figure 3. Inspection Model proposed by the VProcInsp [2].

In the **Planning** phase, a professional who will exercise the role of inspection moderator is identified. The Planning is similar to a Projects Management, because in this activity is defined the team, terms, costs, and other functions concerned to the management. The main activity of the Planning is the configuration of the checklist to the validation inspection. To configure it, the Moderator should use the documentation or formal model of the process under validation with the support of the process author.

In the phase of **Analysis and Comparison**, the inspectors individually analyze the artifacts and compare them with the process formal model. They are guided by the checklist, identifying discrepancy between the two sets of documents. It is important to highlight that most of the times, the inspectors will have to investigate all the information produced during the development process. During the execution of the phase, the inspector must register and classify the discrepancies in a report.

In the **Collection** step, the Moderator has access to all the lists of discrepancies through the Discrepancies Reports produced by the Inspectors in the activity of Analysis and Comparison. The Moderation can select discrepancies of these lists and discharge or classify as replication, if there is more than one discrepancy representing the same damage. When the discrepancy is discharged, it is withdrawn from the next activities of the VProcInsp.

The Moderator, the Author and the Inspectors have participation in the **Discrimination** phase. During this phase, the discrepancies are treated as discussion topics. Each participant can add his or her comments related to each of the discrepancies, which stays available as a discussion topic while the Moderator and the Author do not decide if the item really represents a discrepancy or not. It is important to stress that the solution for the discrepancies are not discussed in the Discrimination. This is a task of the Author of the process in the phase of **Reworking**, where the Author corrects the discrepancies detected during the inspection.

Finally, the Continuation consists in reviewing the material corrected by the authors for the Moderator, who makes an analysis of inspection as a whole and revalues the quality of the inspected item. He has the autonomy to decide if a new inspection must occur or not.

4. THE SPSS VALIDATION

The SPSS validation questions if the initial process grants effectiveness to a scientific software development process of academic nature. With the purpose of selecting researchers to use the SPSS in the creation of their software, the process was presented to investigators in the *Intelligent Systems Laboratory* of CEFET-MG and to some researchers from other Brazilian universities. Five researchers place themselves at full disposal to use the SPSS as the development tool of the construction of their academic nature software.

The initial thought was to select studies in which the software to be developed had different profiles, in such an approach so that they could exercise, in diverse ways, the SPSS process and the artifacts. With this in mind, the following projects were selected: two M.Sc. developments, from the Computing and Mathematics Modeling Program of CEFET-MG and from the Computer Science Program of UFMG, respectively; one study from the Positivo University; and, two under graduation projects from the Computer Science Course, from Faculdades Integradas de Caratinga, and the other from Automation and Control Engineering, also from UFMG. A brief summary of each project is given below, to show the diversity of software developed. It is important to mention that the students are from different research areas and with complete different purposes and backgrounds.

- An Investor Psychology Study through an Automaton Cell (Investor's Psychology): This work points to a new field which opposes the efficient market hypothesis, called behavioral finances. It shows, through empirical researches, that the investor's behavior, many times, escapes from the rationality. This hypothesis is studied by a automaton cell model, with the purpose of analyzing how and how much an investment psychology, from investors, can influence in the stock market stability.
- Vehicles Routing Problem with Cross-Docking (Vehicles Routing): The problem with Vehicle Routing Problem with Cross-Docking (VRPCD) integrates characteristics from Cross-Docking strategy, with the classical problem of Vehicles Routing Problem. VRPCD considers a scenery where a group of customers and suppliers (nodes), geographically distributed, and with their respective demand of delivery and collection, must be attended by a team of vehicles which depart from a distribution center, called Cross-Dock. For this, a software was developed, whose objective was to determine the minimum number of vehicles, and the best route for each one, to visit all the nodes just once, and, therefore minimizing the transports cost.
- Digital images transfer using J2ME and Bluetooth in mobile devices (Images Transfer): This work proposes the implementation of an application in JME using JSR of the Bluetooth, that allows the users to send their images directly to other cell phones, which necessarily need to support Bluetooth Technology, in a fast way and without the need to use a computer.

- The Use of Artificial Neural Networks for Physical Conditioning Verification of a Professional Soccer Player (Physical Conditioning): This work has the purpose to use an artificial neural network (ANN) to verify the physical conditioning of a professional soccer player, avoiding muscular exhaustion. The ANN training was made with data of the players of Cruzeiro Esporte Clube, (a well-known Brazilian soccer team), so that it could be used to provide the real performance of the player in that circumstances.
- Development of a Tool to a System for Diagnostic of Alarms (Alarms Diagnostic System): In this work, the development of a tool to help alarms identification, when they are badly configured, and for the diagnostic of alarms systems from its log files, is presented. In the developed software, the alarm identification occurs in two ways: by statistical analysis recommended by EEMUA 191, the norm which represents the main practices for alarm systems projects, or using data mining techniques, which identify rules of association among alarms. It is emphasized that the result of the mining of data is presented in the form of a complex network, with which is possible to identify the alarms redundancy, in an intuitive way.

In continuation, the SPSS validation was performed with application of the VProcInsp technique in the artifacts produced after the use of the SPSS by the researchers. With regard to the configuration of the validation process using the VProcInsp, it is important to underline that: the moderator also acted as the process author; five checklists concerning each of the artifacts of the SPSS were done; two inspectors were in charge of the application of the checklists; and, the analyst was preserved as the own consumer of the SPSS. The validation occurred in all phases proposed by the SPSS, guaranteeing that the formal process would be validated as a whole.

At this point it should be said that a more complete description of the customization of the validation process used for the SPSS, and its analysis, will be postpone to be presented in a future publication.

5. PRESENTATION OF THE RESULTS

Table 1 below shows a summary of the discrepancies found during the execution of the process validation technique VProcInsp, separated by artifacts of the SPSS. It is important to mention that the technique validated the items which appear in the SPSS formal process. However, if the project needs something else to the development of the software which does not appear in the process, the lack of this item was not taken into consideration in this table. Due to this, the “Evaluation and Suggestions” option of the SPSS webapp tool allows the researcher to relate any extra need to software development. With this possibility, the process validation team could verify a possible requirement of the researcher that the process would not be beholding.

Table 1. Summary of the discrepancies found during the SPSS validation

| Artifact / Project | Investor Psychology | Vehicle Routing | Image Transfer | Physical Conditioning | Diagnostic Alarm Systems |
|----------------------------------|---|---|---|---|---|
| General Development Control | Item Description Details of Architecture. | Item Description Details of Architecture | | | Item Description Details of Architecture |
| Requirements and Scope Version | Item Description Requirements and Scope Version Ambiguity: "Scope Release" and Expected Features | Item Description Requirements and Scope Version Ambiguity: "Scope Release" and Expected Features | Item Description Requirements and Scope Version Description of the item "Special Requirements" is not clear. Ambiguity: "Scope Release" and Expected Features | Item Description Requirements and Scope Version Ambiguity: "Scope Release" and Expected Features (should be optional) | Item Description Scope Version Item Description viability Item Description Requirements and Scope Version |
| Use case detailing | Description of "extension points" | Description of "extension points" | Description of "extension points" | Description of "extension points" | Description of "extension points" |
| Plan Version Coding | No information was found for all items of the classes details | No information was found for all items of the classes details | No information was found for all items of the classes details | No information was found for all items of the classes details | No information was found for all items of the classes details |
| Test Planning | There were no documented interface test, source code, performance and integration. | There were no documented interface test, source code, performance and integration. | There were no documented interface test, source code, performance and integration. | There were no documented interface test, and integration. | There were no documented interface test, source code, and integration. performance |
| Successes and Failures of Record | - | - | - | - | - |

The validation of the process took place through the artifacts generated during the software development or, in other words, all the activities described in the five artifacts that composes the SPSS were validated.

The process of validation began from the General Development Control artifact, in which it was detected that the three researchers did not understand how the process treated the item Software Architecture. To solve this point, it is suggested that the description of this item should be rewritten, to include examples in order to eliminate ambiguities.

With regard to the artifact Version Requirements and Scope, it was pointed out, in all the developed projects, that the items Version Scope and Expected Functionalities were causing ambiguity and, additionally, they were complicating the understanding of the researchers. It was also cited by all the researchers that the item Requirements and Version Scope is not described in a clear and concise way. The developer of the project "Alarms Diagnostic System" indicated that he did not understand the description in the item Special Requirements, which did not allowed its production. The item Viability was not filled by the researcher of the "Vehicles Routing", who claimed that he did not understand its description. From these analyses, it is suggested that the items previously cited be restructured in the Version Requirements and Scope.

As for the artifact Detailing of the Use Cases, the item Extension Points was not understood by any of the researchers and could not be used. Another item that was not comprehended by the responsible for the "Alarms Diagnostic System" was the Special Requirements. Hence, it is recommended that these two items be revisited, in such a way to facilitate its understanding by the researchers.

In the artifact Version Codification Planning, all the developers claim that, during the filling of the item Class Detailing, the topic Attributes, Operations and Responsibilities should not be compulsory, because some classes might not contemplate all these items.

The artifact Tests Plan was the most questioned during the validation by the researchers. It was pointed out that none of the projects contemplate, in a documented form, the items Source-Code Testing, Integration Testing, Interface Testing and Performance Testing. It is important to highlight that the items related to source-code, interface and integration tests are set as mandatory by the SPSS, which may be considered an inconsistency with the simplicity and flexibility advocated by the process. In a general, the researchers related the lack of information about the techniques to be used by the tests indicated in the artifact Tests planning, making it difficult, in such a way, its production.

Finally, in the validation regarding the artifact of Failures and Successes Recording, no inconsistencies were found, indicating that all the items of this artifact are relevant in the development process.

At this point, the validation regarding the SPSS formal process could be regarded as completed or, in other words, all the items of the process were checked and tested during the development of the case studies.

With the purpose of aggregating more quality to the process validation, it was also analyzed the item Evaluation and Suggestions of the SPSS webapp, to examine the expectations of the researcher during the development of his or her software.

Some researchers related that the process does not mention anything with respect to the modeling and the diagrams of the database, which is extremely important to the software development, mainly considering any possible change in the software. These researchers also called attention to include an option to insert more UML diagrams in the process, because some of them do need to use extra diagrams in their developments.

Another point raised was regarding the artifact Tests Planning. The researchers reinforced that, as each one should know his own research work, the tests could be created according to their particular needs, so that the existing compulsory status could be set as optional, and still, in this way, work for the description of the techniques to be used to support the researcher in their use.

As a final comment, it should be said that before the SPSS process be released to the general application (and evaluation) by the scientific society, all these adjustments must be made in the formal document of the process, as well as into the tool that implements it. It is important to highlight that the validation technique used obtained good results and it was very simple to be applied during the process of validation of the SPSS artifacts.

6. FINAL CONSIDERATIONS

In the present work, the SPSS (Specific Development Process for Scientific Software) was validated. The validation was carried out using the validation inspection technique VProcInsp and was applied to five projects that develop scientific software. Once the selected projects were very distinct and related to different fields of applications, it was possible to widely “exercise” the process. Other interesting observation is diverse formation of each researcher involved in this practice, because it was observed the difficulty of some of them with relation to Software Engineering terms; they had to appeal to the literature and, thus, needed some added effort to the development of the software. This fact will be further exploited in a future paper to be published. However, even with such a problem, the SPSS could be used, which confirms its simplicity.

Although the SPSS has shown some inconsistencies after the validation process, it is important to underline that, even though, with the adjustments of Section 5, the process offers strong evidences that it can contribute with the academic society in the development of scientific software. It is also important to draw attention to the improvement and evolution of the SPSS that can be obtained from the feedbacks from people who use it, in order to provide more and more consistency and credibility to the process.

Finally, it must be said that the case studies used during the validation process are not enough to assure that the process is fully validated. However, they can be sufficient to give evidences of the characteristics which enable its use by the scientific community.

7. REFERENCES

- [1] Cook, J. E, Wolf, A. L., “Software validation: quantitatively measuring the correspondence of a process to a model”. *ACM Transactions on Software Engineering and Methodology*. 8 (2), 147-176, 1999.
- [2] Ferreira, B, Moita, G. F., “Inspection technique for the validation of software development processes”. *IOP Conference Series: Materials Science and Engineering (MSE)*. 10, p. 012008, 2010.
- [3] Gomes, J. O., Moita, G. F., Moreira, A. B., “Comparative analysis of different approaches for the Validation of Software Processes”. In: *XXXII CILAMCE Iberian Latin-American Congress on Computational Methods in Engineering*, Ouro Preto, Brazil, 2011.
- [4] Huesh, N. L, Shen, W. H., Yang, Z. W., Yang, D.L., "Applying UML and software simulation for process definition, verification, and validation". *Information and Software Technology*, 50 (9-10), 897-911, 2008.
- [5] Humphrey, W. S., “A Discipline for Software Engineering”. Addison-Wesley. Reading-MA, 1995.
- [6] Moor, A., Delugach H., “Software Process Validation: Comparing Process and Practice

Models”. In: *Contributions to ICCS 2005*, Kassel, Germany. 533- 540, 2005a.

- [7] Navarro, E. O., Hoek, A. V. D., “Software process modeling for an educational software engineering simulation game”. In: *The 5th International Workshop on Software Process Simulation and Modeling. PROSIM*, 311- 325, 2005.
- [8] Pereira Jr, M., Moita, G. F., “Proposta para concepção de um processo de desenvolvimento para software científico”. In: *VIII Simpósio de Mecânica Computacional*, Belo Horizonte, Brazil, 2008.
- [9] Pressman, R. S., “Software Engineering: A Practitioner's Approach”. 7th ed. Science Engineering & Math, 2009.
- [10] Purri, M. M. S., “Estudo e Propostas Iniciais para a Definição de um Processo de Desenvolvimento para Software Científico”. M.Sc. Dissertation. Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG. Belo Horizonte, Brazil, 2006.