# Bringing together space systems engineering and software engineering processes based on standards and best practices

Miriam B. Alves[a]; Martha A. D. Abdala[b]; Rovedy Busquim e Silva[b]

[a]Software Engineer, Instituto de Aeronautica e Espaco –IAE (São José dos Campos), BR.
[b] Instituto de Aeronautica e Espaco –IAE/CTA, São José dos Campos-SP, Brazil.

**Abstract.** The growing complexity of the current space systems results an increasing responsibility for the software embedded in them. This is particularly significant when the systems are employed for space critical missions. Usually the software has rigid real time requirements to fulfil which demands high reliability and a disciplined development process. This paper relates the effort of defining a set of software development processes for the on-board computer flight control software (SOAB), a component of the Brazilian Satellite Launcher (VLS), developed by the Instituto de Aeronautica e Espaco – IAE. To achieve the strict requirements for space missions, the SOAB development team's degree of maturity and technological proficiency had to harmonize with a well defined set of software development processes integrated into the systems engineering. Furthermore, these processes definition had to consider international space systems engineering standards and standards of quality established by IAE. Best practices in software engineering were considered as well.

**Keywords.** SOAB, software development process, systems engineering, standards

## 1 Introduction

Engineering in space systems must be a team activity where the various individuals involved are aware of the important relationship between specialties and their roles in the development as an organizational process. Successful accomplishment of engineering objectives requires not only a combination of technical specialties and expertise, but also principles and best practices to harmonize the systems engineering activities and the software development process.

[1] Software Engineer, Instituto de Aeronautica e Espaco (Sao Jose dos Campos). Praca Marechal Eduardo Gomes, 50, Vila das Acacias, Sao Jose dos Campos, SP, CEP: 12228-904, Brazil. Tel: +55 (12) 3947 4969; Fax: +55 (12) 3947 5019; Email: miriamalves@iae.cta.br;

This is particularly significant while developing critical systems where the embedded software is required to perform critical functions, mostly in real time. The on-board computer flight control software (SOAB) for the Brazilian Satellite Launcher (VLS) is included in this category of critical software. The flight control software takes critical responsibility for the control of the launcher – except for the launcher destruction – from a few minutes preceding the lift-off until the satellite has been deployed into the Earth's orbit. The software is also responsible for the checkout of various launcher systems, including inertial platforms, autopilot chain and sequencing chain. Beside that, each flight has different characteristics and the software has to be prepared to incorporate new issues like a new type of inertial measurement unit. Consequently, this requires well-organized processes of development and maintenance that are directly or indirectly responsible for the specification, acceptance testing, and execution of flight software for the VLS.

On one hand as SOAB is part of the VLS space project, its development process is intrinsically related with systems engineering activities, management and product quality assurance of the organization. On the other hand, the way of developing software has to be consistent with software engineering methodologies. The final outputs of the working process are operational prototypes made available for extended operational validation. The results in terms of technology exploitation experience and novel implemented solutions become an asset for reuse for future space projects.

This paper presents the result of structuring and defining a set of software development processes based on the Brazilian Standard for Quality Management Systems [1] which has been adopted by IAE as a reference for its Quality Management System and also on ESA ECSS Architecture for Space Projects [3, 4, 5], particularly the E-40 family [6,7]. The processes definition also considered the software system life cycle, the development environment, best practices and techniques.

This paper is organized in six sections. Section 1 is the introduction, followed by section 2 that presents the importance of the adoption of space standards. Section 3 presents the relationship between the software development process and the systems engineering process in the organization. Section 4 describes how a set of processes was customized and Section 5 presents the visual representation chosen to symbolize these set of processes. Finally, the conclusion and future prospects are summarized.

## 2 The role of space standards

During the structuring and definition of the set of processes to be used for the SOAB development, the standard ECSS E-40- Part 1B [6] and the Unified Software Development Process – UP [9] were considered as standard process and framework respectively, due to their specific intended use for software development. They served as a guide in recognized methods for engineering tasks and managing of a software system throughout its life-cycle, addressing the tasks performed by software engineers.

The Brazilian Standard [1], due to its more general role in quality management, was used more as a guide for product quality and it was also a helpful guide to specify the requirements for the deliverable or result, especially in software product acquisition. Furthermore, this standard provided a good orientation for the necessary engineering activities to carry on a system development, while ensuring that the system is properly designed with quality. The guidance of this standard could not only be applied to a new system but also as an incremental enhancement to an existing system. By addressing system requirements, the content of this standard helped to describe an integrated approach to system development.

In order to adopt the Brazilian Standard ABNT NBR 15100, E-40 Part 1B and the Unified Software Development Process [9] in a consistent way, the technical efforts were aimed in:

a) Defining a software development process totally integrated with the systems engineering activities and the space system life cycle and that could be implemented in an existent environment, considering the restrictions of the infrastructure and the development team;

b) Defining a methodology of development, associated with techniques, that should take in account not only the environment and the related conditions in which the software system would be deployed, but also functional and performance requirements and quality factors, like usability and safety;

c) Defining processes for requirements, design, codification, testing, delivery, maintenance and acquisition, which are necessary to provide life cycle support for the software system;

d) Establishing a manner to represent the set of processes in such way that they could be easily understood and employed by the development team;

e) Publishing the set of processes in the same development environment (Computer Aided Software Environment - CASE) in order to make them directly accessible by the team in their working environment.

## 3 Real-time space software and the systems engineering process

Integration of systems and software engineering processes in space systems development activities is an ongoing challenge. The ability to deal with a high level of complexity in a flexible way makes software a critical and growing part of space segments and ground segments products. Software engineering can be found at several parts of space systems, embracing high level system functionality and also basic functions implemented in firmware.

According to Blanchard [2], systems engineering and analysis, when joined with new and emerging technologies may expose unpredicted opportunities for creating new and better systems and products. However, there used to be a lack of integrity between systems engineering and software engineering teams during the SOAB development.

This situation had caused several problems for the software development team since the software system specification was usually delivered by the systems engineering team without any participation of the software team in its elaboration. Some decisions were made that would strongly affect the software and not always in a smooth way. Aware that this situation had to change, a new software development approach was elaborated and adopted by the software team, with the approval of the systems engineering staff and the quality group.

According to this new software development approach and based on [6], the requirement engineering process, in which the software requirements and specifications are defined, has a special role in the software engineering activities considering that these activities are purely intellectual and the outputs are normally documents. Even the code can be taken as a special form of electronic document. At this point resides the importance of adopting the ECSS-E-40 Part 2B [7], which focus is on the requirements for the structure and content of the documentation produced.

The requirement engineering process includes the activity of System Analysis and Architecture, which results in a Software System Specification that is delivered to the software team to be reviewed in the System Requirements Review (SRR). This review has the participation of both teams: systems engineering and software development teams. The consolidation of the Software System Specification generates the requirement baseline, which will serve as a base for the SOAB requirements during all its development.

Subsequently, the software development team will start the requirements engineering analysis activity in which the requirements and specifications of the software are defined. This activity is part of the set of processes mentioned in the next section and usually takes a large and underestimated time in the whole development. Aware of that, the software quality team gives special attention when planning the development schedule. Because of this activity, a software technical specification is elaborated and reviewed jointly with the systems engineering team in the Preliminary Design Review (PDR). In this review, the systems engineering team will verify if all the requirements with respect to the requirements baseline are captured in the technical specification and if the software architecture is well identified. This is the opportunity to make sure the software will totally fulfil the systems requirements, considering all the environmental definitions and restrictions.

From this point on, the software team, based on a set of processes, will carry on the software development, which includes two more internal formal reviews apart from those reviews made jointly with the system engineering team: Detailed Design Review (DDR) and Critical Design Review (CDR). The CDR is conducted at the end of the design and the completeness of the software validation activities with respect to the technical specification is reviewed and verified.

Once the CDR is realized, the subsequent planned activities are carried out. The accomplishment of these activities will lead the software team to the next formal review, the Qualification Review (QR), which is conducted in cooperation with the systems engineering team. At this point, the system and software engineering processes

come together again to verify if the software meets all its specified requirements in the approved requirement baseline, before its integration into the system. The consistency and completeness of all software documentation are also reviewed and verified.

The next formal meeting with both teams is in the Acceptance Review (AR), which aim is to prepare the software to be integrated with the space system, where a summary of tests reports and the software user manual are reviewed. The consistency and completeness of the documentation are verified again. At this point, the system becomes internally validated and it is delivered to the systems engineering team in order to start the Software Acceptance activity.

The Software Acceptance and System Acceptance activities, as part of the systems engineering process, will integrate the software into the system and perform all the required operational tests. These tests are executed to guarantee that the software is working correctly and accurately in the system environment as well as with other system components.

## 4 Structuring a set of software development processes

As part of IAE's Quality Management System implementation [1], it was highly recommended that all of its divisions that develop VLS components should document the manner or method of realizing their products, outlined as operational procedures. However, in order to support the software development approach detailed in the previous section, and to align this approach with international space standards, an organized and documented way of software development should also be set down. To meet both necessities, a set of software development processes were defined in a high level, considering jointly the IAE's Quality Management System recommendations, the ECSS-E-40 family and the established best practices of the Unified Process.

To meet the Brazilian Standard [1] recommendations about the product realization, the operational procedures were written in a level of detail that could be understood by other individuals besides software engineers. However, the software engineering team would need a more detailed description of this set of processes to allow them to work properly and uniformly, employing best practices in a consistent way. The result was a set of software development process, or operational procedures, in which the activities were split up in a more comprehensive and detailed level to be employed by the software engineers. The CASE environment facilities were fundamental to help the graphic representation of this set of processes and their respective sublevels as well.

The first step considered in order to structure the set of software development processes was the tailoring of ECSS-E-40 standards. The result of the tailoring activity was a document, which served as a guide for future on board flight control software development.

Although ECSS-E40 Part 1B covers most of the activities of a software development, there are some other important activities not totally covered by this

standard. In this case, the Brazilian Standard [1] and the best practices of the UP served as a complement to base on the processes activities.

There was a leading culture of how to develop software among the members of the software team that reflected the way they had been working for years. This was a big issue to be treated adequately as well. The team's adherence and trust in adopting the set of processes was fundamental for their implantation.

A set of six software development processes was initially defined:

1. Requests Analysis;
2. Software System Development;
3. Software System Delivery;
4. Software System Acquisition;
5. Software System Acceptance;
6. Software System Maintenance.

The Requests Analysis process reflects the way the software team handles all the requests of services and it was mainly based on the organization workflow. These requests may be for a new software system development, maintenance of an existing space software system or for providing some technological assistance (consulting, training, etc.).

The main body of the Software System Development process was strongly based on ECSS-E-40 Part 1B standard. This process was the hardest one to be defined because it reproduces the principal effort of the software group. Besides ECSS-E-40 Part 1B, its definition had to bring together software development methodologies, software life cycle and best practices of software development.

The Software System Delivery process was delineated to complement the Software System Development process and it defines the means the software is delivered to the system engineering team, after being internally validated.

As being a system itself, SOAB or any ground software system segment may have a few of its components or, in some cases, the entire system, acquired from suppliers. Considering that IAE's Quality Management System follows the NBR 15100 recommendations and they are very rigorous regarding how the acquisitions and the suppliers control should be, a Software System Acquisition process was defined to fulfil these recommendations. The Brazilian standard NBR 15100 was adopted since the ECSS-E-40 Part 1B does not give recommendations about the acquisition process, as its primary focus is on software development.

The Software System Acceptance process was outlined to complement the Software System Acquisition process and it details how the software should be received from the supplier by the software team. In this case, besides other issues, the software team has to be aware of the integration tests that have to be executed in order to accept the system properly.

The last listed process is the Software System Maintenance that was identified to fix and maintain the several SOAB configurations to keep them operational for different VLS flights.

It is worth to mention that all the activities represented in this set of processes have another level of detail that was implemented as hyperlinks in the CASE environment, where not only the activity itself is better described via another graphical representation (UML activity diagram), but also the role and responsibilities for each activity

## 5 A visual representation and integration of the defined process in a CASE tool

Once the set of processes was defined, it was necessary to consider how to implement it and achieve maximum efficiency in following it. A process can be defined and represented as a set of interrelated activities, which transforms inputs into outputs [8]. In order to have an easy and more comprehensive view of the set of processes, they were drawn in a CASE tool (IBM© Rational Suite) using an adaptation of the UML activity diagrams. Each process in the set was represented by one activity diagram, where each activity in a process was mapped into an activity of the UML activity diagram. Activities in the processes were also detailed in another activity diagram, when necessary.

The choice of incorporating the visual representation of the process in the CASE environment helped to bring together the development environment, the process of development itself and the best practices of the Rational Unified Process (RUP), allowing to apply the very same techniques to disciplines other than software that also use models, especially systems engineering, promoting more integration among them.

The CASE environment allowed the project managers to use the IBM© RUPBuilder to select, configure, create custom views and publish the defined set of processes for their projects. They could start from the pre-established set of processes and make further choices based on their project's unique needs. Additionally, they could publish the processes in Web sites

The way of publishing a customized process was managed from the user interface, which gave the project managers the means to describe their process selecting components from the defined set of processes to compose their customized process. They could also create different process views of the selected process for different members of the software team.

## 6 Conclusion and future prospects

This paper presented the effort of structuring and defining a set of software development processes for the on-board computer flight control software (SOAB), part of the Brazilian Satellite Launcher (VLS), developed by the Instituto de Aeronautica e Espaco (IAE). This set of processes was based on international space systems engineering standards and also standards of quality established by IAE. The result is a

disciplined approach for software development, integrated into systems engineering, that increases the chance of releasing a software system more reliable that meets the strict requirements for space missions. Furthermore, this approach brought a better integration between the systems engineering team and the SOAB development team, in a desired degree of maturity. In order to have this set of processes incorporated into the CASE environment, a visual representation of the processes was prepared, facilitating their use and integration into the resulting work products.

As future prospects, it is still necessary to identify and implement a mechanism of continuous improvement of the set of processes, based on the experiences acquired by the software team in employing them. The suggestions of the systems engineering team colleted mainly throughout the formal reviews have to be considered as well. This mechanism should take advantage of the facilities of the CASE environment in order to guarantee that the suggestions and feedbacks of the software team members, systems engineering team and quality group are considered in time. The continuous improvement of the processes ought to be a real and tangible goal to turn the development of critical space systems with embedded software less risky and more successful.

## 7 References

[1] ABNT Sistema da Qualidade Aeroespacial – Modelo para a garantia da qualidade em projetos, desenvolvimento, produção, instalação e serviços associados. ABNT NBR 15100:2004.

[2] Blanchard BS, Fabrycky, WJ. Systems Engineering and Analysis. Prentice Hall, New Jersey, 1998. 3rd edn.

[3] ECSS-E-10 Space Engineering - Systems Engineering Part 1B: Requirements and process. ECSS-E-10 Part 1B, 18 Nov 2004.

[4] ECSS-E-10 Space Engineering - Systems Engineering Part 6A: Functional and technical specifications. ECSS-E-10 Part 6A, 09 Jan 2004.

[5] ECSS-E-10 Space Engineering - Systems Engineering Part 7A: Product data exchange. ECSS-E-10 Part 7A, 25 August 2004.

[6] ECSS-E-40 Space engineering – Software Part 1: Principles and requirements. ECSS-E-40 Part 1B, 28 November 2003.

[7] ECSS-E-40 Space Engineering - Software Part 2: Document requirements definitions (DRDs). ECSS-E-40 Part 2B, 31 March 2005.

[8] ISO/IEC Standard for information technology –Software life cycle process. ISO/IEC 12207:1995.

[9] Jacobson I, Booch, G, Rumbaugh, J. The Unified Software Development Process. Addison Wesley Logman, Inc., 1999.

[10] Vorthman Jr, RG, Stephen MH. Towards a Rational Approach to Standards for Integrated Ocean Observing Systems (IOOS) Development. In: Proceedings of the Oceans 2006 MTS/IEEE Conference. Boston, MA. Sept 18-21, 2006.

# Bringing together space systems engineering and software engineering processes based on standards and best practices

Miriam B. Alves[a]; Martha A. D. Abdala[b]; Rovedy Busquim e Silva[b]

[a]Software Engineer, Instituto de Aeronautica e Espaco –IAE (São José dos Campos), BR.
[b] Instituto de Aeronautica e Espaco –IAE/CTA, São José dos Campos-SP, Brazil.

**Abstract.** The growing complexity of the current space systems results an increasing responsibility for the software embedded in them. This is particularly significant when the systems are employed for space critical missions. Usually the software has rigid real time requirements to fulfil which demands high reliability and a disciplined development process. This paper relates the effort of defining a set of software development processes for the on-board computer flight control software (SOAB), a component of the Brazilian Satellite Launcher (VLS), developed by the Instituto de Aeronautica e Espaco – IAE. To achieve the strict requirements for space missions, the SOAB development team's degree of maturity and technological proficiency had to harmonize with a well defined set of software development processes integrated into the systems engineering. Furthermore, these processes definition had to consider international space systems engineering standards and standards of quality established by IAE. Best practices in software engineering were considered as well.

**Keywords.** SOAB, software development process, systems engineering, standards

## 1 Introduction

Engineering in space systems must be a team activity where the various individuals involved are aware of the important relationship between specialties and their roles in the development as an organizational process. Successful accomplishment of engineering objectives requires not only a combination of technical specialties and expertise, but also principles and best practices to harmonize the systems engineering activities and the software development process.

[1] Software Engineer, Instituto de Aeronautica e Espaco (Sao Jose dos Campos). Praca Marechal Eduardo Gomes, 50, Vila das Acacias, Sao Jose dos Campos, SP, CEP: 12228-904, Brazil. Tel: +55 (12) 3947 4969; Fax: +55 (12) 3947 5019; Email: miriamalves@iae.cta.br;

This is particularly significant while developing critical systems where the embedded software is required to perform critical functions, mostly in real time. The on-board computer flight control software (SOAB) for the Brazilian Satellite Launcher (VLS) is included in this category of critical software. The flight control software takes critical responsibility for the control of the launcher – except for the launcher destruction – from a few minutes preceding the lift-off until the satellite has been deployed into the Earth's orbit. The software is also responsible for the checkout of various launcher systems, including inertial platforms, autopilot chain and sequencing chain. Beside that, each flight has different characteristics and the software has to be prepared to incorporate new issues like a new type of inertial measurement unit. Consequently, this requires well-organized processes of development and maintenance that are directly or indirectly responsible for the specification, acceptance testing, and execution of flight software for the VLS.

On one hand as SOAB is part of the VLS space project, its development process is intrinsically related with systems engineering activities, management and product quality assurance of the organization. On the other hand, the way of developing software has to be consistent with software engineering methodologies. The final outputs of the working process are operational prototypes made available for extended operational validation. The results in terms of technology exploitation experience and novel implemented solutions become an asset for reuse for future space projects.

This paper presents the result of structuring and defining a set of software development processes based on the Brazilian Standard for Quality Management Systems [1] which has been adopted by IAE as a reference for its Quality Management System and also on ESA ECSS Architecture for Space Projects [3, 4, 5], particularly the E-40 family [6,7]. The processes definition also considered the software system life cycle, the development environment, best practices and techniques.

This paper is organized in six sections. Section 1 is the introduction, followed by section 2 that presents the importance of the adoption of space standards. Section 3 presents the relationship between the software development process and the systems engineering process in the organization. Section 4 describes how a set of processes was structured and Section 5 presents the visual representation chosen to symbolize these set of processes. Finally, the conclusion and future prospects are summarized.

## 2 The role of space standards

During the structuring and definition of the set of processes to be used for the SOAB development, the standard ECSS E-40 Part 1B [6] and the Unified Software Development Process – UP [9] were considered as standard process and framework respectively, due to their specific intended use for software development. They served as a guide in recognized methods for engineering tasks and managing of a software system throughout its life-cycle, addressing the tasks performed by software engineers.

The Brazilian Standard [1], due to its more general role in quality management, was used more as a guide for product quality and it was also a helpful guide to specify the requirements for the deliverable or result, especially in software product acquisition. Furthermore, this standard provided a good orientation for the necessary engineering activities to carry on a system development, while ensuring that the system is properly designed with quality. The guidance of this standard could not only be applied to a new system but also as an incremental enhancement to an existing system. By addressing system requirements, the content of this standard helped to describe an integrated approach to system development.

In order to adopt the Brazilian Standard ABNT NBR 15100, E-40 Part 1B and the Unified Software Development Process [9] in a consistent way, the technical efforts were aimed in:

a) Defining software development processes totally integrated into the systems engineering activities and the space system life cycle that could be implemented in an existent environment, considering the restrictions of the infrastructure and the development team;

b) Defining a methodology of development, associated with techniques, that should take in account not only the environment and the related conditions in which the software system would be deployed, but also functional and performance requirements and quality factors, like usability and safety;

c) Defining processes for requirements, design, codification, testing, delivery, maintenance and acquisition, which are necessary to provide life cycle support for the software system;

d) Establishing a manner to represent the set of processes in such way that they could be easily understood and employed by the development team;

e) Publishing the set of processes in the same development environment (Computer Aided Software Environment - CASE) in order to make them directly accessible by the team in their working environment.

## 3 Real-time space software and the systems engineering process

Integration of systems and software engineering processes in space systems development activities is an ongoing challenge. The ability to deal with a high level of complexity in a flexible way makes software a critical and growing part of space segments and ground segments products. Software engineering can be found at several parts of space systems, embracing high level system functionality and also basic functions implemented in firmware.

According to Blanchard [2], systems engineering and analysis, when joined with new and emerging technologies may expose unpredicted opportunities for creating new and better systems and products. However, there used to be a lack of integrity between systems engineering and software engineering teams during the SOAB development.

This situation had caused several problems for the software development team since the software system specification was usually delivered by the systems engineering team without any participation of the software team in its elaboration. Some decisions were made that would strongly affect the software development and not always in a smooth way. Aware that this situation had to change, a new software development approach was elaborated and adopted by the software team, with the approval of the systems engineering staff and the quality group.

According to this new software development approach and based on [6], the requirement engineering process, in which the software requirements and specifications are defined, has a special role in the software engineering activities considering that these activities are purely intellectual and the outputs are normally documents. Even the code can be taken as a special form of electronic document. At this point resides the importance of adopting the ECSS-E-40 Part 2B [7], which focus is on the requirements for the structure and content of the documentation produced.

The requirement engineering process [6] includes the activity of System Analysis and Architecture, which results in a Software System Specification that is delivered to the software team to be reviewed in the System Requirements Review (SRR). This review has the participation of both teams: systems engineering and software development teams. The consolidation of the Software System Specification generates the requirement baseline, which will serve as a base for the SOAB requirements during all its development.

Subsequently, the software development team will start the requirements engineering analysis activity in which the requirements and specifications of the software are defined. This activity is part of the set of processes mentioned in the next section and usually takes a large and underestimated time in the whole development. Aware of that, the software quality team gives special attention when planning the development schedule. As a result of this activity, a software technical specification is elaborated and reviewed jointly with the systems engineering team in the Preliminary Design Review (PDR). In this review, the systems engineering team will verify if all the requirements with respect to the requirements baseline are captured in the technical specification and if the software architecture is well identified. This is the opportunity to make sure the software will totally fulfil the systems requirements, considering all the environmental definitions and restrictions.

From this point on the software team, based on a set of processes (section 4), will carry on the software development, which includes two more internal formal reviews apart from those reviews made jointly with the system engineering team: Detailed Design Review (DDR) and Critical Design Review (CDR). The CDR is conducted at the end of the design and the completeness of the software validation activities with respect to the technical specification is reviewed and verified.

Once the CDR is realized, the subsequent planned activities are carried out. The accomplishment of these activities will lead the software team to the next formal review, the Qualification Review (QR), which is conducted in cooperation with the systems engineering team. At this point, the system and software engineering processes

come together again to verify if the software meets all its specified requirements in the approved requirement baseline, before its integration into the system. The consistency and completeness of all software documentation are also reviewed and verified.

The next formal meeting with both teams is in the Acceptance Review (AR), which aim is to prepare the software to be integrated into the space system, where a summary of tests reports and the software user manual are reviewed. The consistency and completeness of the documentation are verified again. At this point, the system becomes internally validated and it is delivered to the systems engineering team in order to start the Software Acceptance activity.

The Software Acceptance and System Acceptance activities [6], as part of the systems engineering process, will integrate the software into the system and perform all the required operational tests. These tests are executed to guarantee that the software is working correctly and accurately in the system environment as well as with other system components.

## 4 Structuring a set of software development processes

As part of IAE's Quality Management System implementation [1], it was highly recommended that all of its divisions that develop VLS components should document the manner or method of realizing their products, outlined as operational procedures. However, in order to support the software development approach detailed in the previous section, and to align this approach with international space standards, an organized and documented way of software development should also be set down. To meet both necessities, a set of software development processes were defined in a high level, considering jointly the IAE's Quality Management System recommendations, the ECSS-E-40 family and the established best practices of the Unified Process.

To meet the Brazilian Standard [1] recommendations about the product realization, the operational procedures were written in a level of detail that could be understood by other individuals besides software engineers. However, the software engineering team would need a more detailed description of this set of processes to allow them to work properly and uniformly, employing best practices in a consistent way. The result was a set of software development process, or operational procedures, in which the activities were split up in a more comprehensive and detailed level to be employed by the software engineers. The CASE environment facilities were fundamental to help the graphic representation of this set of processes and their respective sublevels as well.

The first step considered in order to structure the set of software development processes was the tailoring of ECSS-E-40 standards. The result of the tailoring activity was a document, which served as a guide for future on board flight control software development.

Although ECSS-E40 Part 1B covers most of the activities of a software development, there are some other important activities not totally covered by this

standard. In this case, the Brazilian Standard [1] and the best practices of the UP served as a complement to base on the processes activities.

There was a leading culture of how to develop software among the members of the software team that reflected the way they had been working for years. This was a big issue to be treated adequately as well. The team's adherence and trust in adopting the set of processes was fundamental for their implantation.

A set of six software development processes was initially defined:

1. Requests Analysis;
2. Software System Development;
3. Software System Delivery;
4. Software System Acquisition;
5. Software System Acceptance;
6. Software System Maintenance.

The Requests Analysis process reflects the way the software team handles all the requests of services and it was mainly based on the organization workflow. These requests may be for a new software system development, maintenance of an existing space software system or for providing some technological assistance (consulting, training, etc.).

The main body of the Software System Development process was strongly based on ECSS-E-40 Part 1B standard. This process was the hardest one to be defined because it reproduces the principal effort of the software group. Besides ECSS-E-40 Part 1B, its definition had to bring together software development methodologies, software life cycle and best practices of software development.

The Software System Delivery process was delineated to complement the Software System Development process and it defines the means the software is delivered to the system engineering team, after being internally validated.

As being a system itself, SOAB or any ground software system segment may have a few of its components or, in some cases, the entire system, acquired from suppliers. Considering that IAE's Quality Management System follows the NBR 15100 recommendations and they are very rigorous regarding how the acquisitions and the suppliers control should be, a Software System Acquisition process was defined to fulfil these recommendations. The Brazilian standard NBR 15100 was adopted since the ECSS-E-40 Part 1B does not give recommendations about the acquisition process, as its primary focus is on software development.

The Software System Acceptance process was outlined to complement the Software System Acquisition process and it details how the software should be received from the supplier by the software team. In this case, besides other issues, the software team has to be aware of the integration tests that have to be executed in order to accept the system properly.

The last listed process is the Software System Maintenance that was identified to fix and maintain the several SOAB configurations to keep them operational for different VLS flights.

It is worth to mention that all the activities represented in this set of processes have another level of detail that was implemented as hyperlinks in the CASE environment, where not only the activity itself is better described via another graphical representation (UML activity diagram), but also the role and responsibilities for each activity

## 5 A visual representation and integration of the defined process in a CASE tool

Once the set of processes was defined, it was necessary to consider how to implement it and achieve maximum efficiency in following it. A process can be defined and represented as a set of interrelated activities, which transforms inputs into outputs [8]. In order to have an easy and more comprehensive view of the set of processes, they were drawn in a CASE tool (IBM© Rational Suite) using an adaptation of the UML activity diagrams. Each process in the set was represented by one activity diagram, where each activity in a process was mapped into an activity of the UML activity diagram. Activities in the processes were also detailed in another activity diagram, when necessary.

The choice of incorporating the visual representation of the process in the CASE environment helped to bring together the development environment, the process of development itself and the best practices of the Unified Process (UP), allowing to apply the very same techniques to disciplines other than software that also use models, especially systems engineering, promoting more integration among them.

The CASE environment allowed the project managers to use the IBM© RUPBuilder to select, configure, create custom views and publish the defined set of processes for their projects. They could start from the pre-established set of processes and make further choices based on their project's unique needs. Additionally, they could publish the processes in Web sites.

The way of publishing a customized process was managed from the user interface, which gave the project managers the means to describe their process selecting components from the defined set of processes to compose their own customized process. They could also create different process views of the customized process for different members of the software team.

## 6 Conclusion and future prospects

This paper presented the effort of structuring and defining a set of software development processes for the on-board computer flight control software (SOAB), part of the Brazilian Satellite Launcher (VLS), developed by the Instituto de Aeronautica e Espaco (IAE). This set of processes was based on international space systems engineering standards and also standards of quality established by IAE. The result was

a disciplined approach for software development, integrated into systems engineering, that increases the chance of releasing a software system more reliable, which meets the strict requirements for space missions. Furthermore, this approach brought a better integration between the systems engineering team and the SOAB development team, in a desired degree of maturity. In order to have this set of processes incorporated into the CASE environment, a visual representation of the processes was prepared, facilitating their use and integration into the resulting work products.

As future prospects, it is still necessary to identify and implement a mechanism of continuous improvement of the set of processes, based on the experiences acquired by the software team in employing them. The suggestions of the systems engineering team colleted mainly throughout the formal reviews have to be considered as well. This mechanism should take advantage of the facilities of the CASE environment in order to guarantee that the suggestions and feedbacks of the software team members, systems engineering team and quality group are considered in time. The continuous improvement of the processes ought to be a real and tangible goal to turn the development of critical space systems with embedded software less risky and more successful.

## 7 References

[1] ABNT Sistema da Qualidade Aeroespacial – Modelo para a garantia da qualidade em projetos, desenvolvimento, produção, instalação e serviços associados. ABNT NBR 15100:2004.

[2] Blanchard BS, Fabrycky, WJ. Systems Engineering and Analysis. Prentice Hall, New Jersey, 1998. 3rd edn.

[3] ECSS-E-10 Space Engineering - Systems Engineering Part 1B: Requirements and process. ECSS-E-10 Part 1B, 18 Nov 2004.

[4] ECSS-E-10 Space Engineering - Systems Engineering Part 6A: Functional and technical specifications. ECSS-E-10 Part 6A, 09 Jan 2004.

[5] ECSS-E-10 Space Engineering - Systems Engineering Part 7A: Product data exchange. ECSS-E-10 Part 7A, 25 August 2004.

[6] ECSS-E-40 Space engineering – Software Part 1: Principles and requirements. ECSS-E-40 Part 1B, 28 November 2003.

[7] ECSS-E-40 Space Engineering - Software Part 2: Document requirements definitions (DRDs). ECSS-E-40 Part 2B, 31 March 2005.

[8] ISO/IEC Standard for information technology –Software life cycle process. ISO/IEC 12207:1995.

[9] Jacobson I, Booch, G, Rumbaugh, J. The Unified Software Development Process. Addison Wesley Logman, Inc., 1999.

[10] Vorthman Jr, RG, Stephen MH. Towards a Rational Approach to Standards for Integrated Ocean Observing Systems (IOOS) Development. In: Proceedings of the Oceans 2006 MTS/IEEE Conference. Boston, MA. Sept 18-21, 2006.