A Brazilian Software Industry Experience in Using ECSS for Space Application Software Development

Fátima Mattiello-Francisco[a,1], Valdivino Santiago[a], Ana Maria Ambrósio[a], Leise Jogaib[b] and Ricardo Costa[b]

[a]National Institute for Space Research – INPE, São José dos Campos, SP, BR.
[b]DBA Engenharia de Sistemas LTDA, Rio de Janeiro, RJ, BR.

Abstract. This paper presents the tailoring of ECSS software product assurance requirements aiming at the development of scientific satellite payload embedded software by a Brazilian software supplier. The software item, named SWPDC, developed by DBA Engenharia de Sistemas LTDA within Software Factory context, is part of an ongoing research project, named Quality of Space Application Embedded Software - QSEE, developed by National Institute for Space Research – INPE, with FINEP financial support. Among other aspects, QSEE project allowed to evaluate the adherence of a Software Factory processes to INPE's embedded software development process requirements. Although not familiar with space domain, the high maturity level of such supplier, CMMI-3 formally evaluated, facilitates the Software Factory to comply with the requirements imposed by the custumer. Following the software verification and validation processes recommended by ECSS standards, an Independent Verification and Validation - IVV approach was used by INPE in order to delegate the software acceptance activities to a third party team. ECSS standard tailored form contributions along the execution of the project and the benefits provided to the supplier in terms of process improvements are also presented herein.

Keywords. Software quality, software development processes, space mission lifecycle,

1 Software for Space Systems

In space systems, a software product is part of a network comprising several systems. Space systems include manned and unmanned spacecraft, launchers, payloads, experiments and their associated ground equipment and facilities. Such software includes firmware components. Space projects are generally expensive and demand considerable amount of time to be completed.

[1] Senior Software Engineer in Space Applications, Space Tecnologies Enginerring, National Institute for Space Research - INPE, Av. dos Astronautas, 1758, Sao Jose dos Campos, SP, Brazil; Tel: +55 (12) 3945 7124; Fax: +55 (12) 3945 7100; Email: fatima@dss.inpe.br; http://www.inpe.br/atifs/fatima/

The insertion of industry into the space programs environment as subsystem supplier has demanded improvements in the space agency's project management processes in order to successfully accomplish the projects. In different areas of application, one can see both industrial and governmental initiatives towards standardization and the use of best practices for project management. European Cooperation for Space Standardization – ECSS is a great result of cooperative efforts among the European Space Agency - ESA, National Space Agencies and European space industry association.

Typical space mission lifecycle is the basis of the model used by ECSS to manage the development of the different space mission subsystems, which includes space application software. Nowadays, space agencies have dedicated special attention to the software project management. Such concern is expressed in ECSS volumes for management, quality assurance and software engineering.

This article addresses the Software Product Assurance volumes [ECSS-Q-80A and B] only. The fundamental principle of this standard is to facilitate the customer-supplier relationship assumed for all software developments, at all levels. They have contributed to these objectives by providing a set of requirements to be met throughout the system lifetime which involves the development and maintenance of space application software. Such requirements deal with quality management and framework, lifecycle activities and process definition, and quality characteristics of the software product [3].

Once the objective of software product assurance is to provide the customer with adequate confidence, the standard is usually tailored for a particular contract by defining specific subset requirements.

In order to evaluate Software Factory model as a satellite payload embedded software item supplier, the standard was tailored for the *Quality of Space Application Embedded Software - QSEE* project, developed under the National Institute for Space Research – INPE coordination. INPE, within such context, plays the role of the customer [1].

This paper is organized as follows: section 2 introduces ECSS-Q-80 structure and tailoring guidelines; section 3 presents DBA's Software Factory model and the software development processes followed by the supplier; section 4 discusses the standard tailored form for this particular software development item, the satellite payload embedded software – SWPDC, a case study of QSEE project. Finally, section 5 concludes with contributions resulting from standard tailored form to the project execution and the benefits provided to the supplier in terms of process improvements.

2 ECSS-Q-80 Structure andTailoring

According to the structure of this ECSS Software Product Assurance standard, the requirements are grouped in the following three categories: (i) management requirements and software product assurance framework; (ii) software lifecycle activities and processes requirements; (iii) software products quality requirements, including both executable code and related products such as documentation and test data. Each requirement has a corresponding *Required Output* identified that,

among others, is intended to assist the customer in selecting applicable requirements during the tailoring process.

Tailoring for software development constraints takes into account the special characteristics of the software being developed, such as the software type (database, real-time) and the system target (embedded processor, web, host system), and the development environment as well. Those issues are subject of Space System Software Engineering Process as defined in ECSS-E-40. Together, the two standards specify all processes for space software development [2].

In order to carry out the tailoring of the software product assurance process under the scope of QSEE project without addressing ECSS-E-40, two relevant aspects must be pointed out. First, the software product is a satellite payload embedded software, therefore, a critical item. Second, although the supplier is not familiar with the technology and software development environment imposed by the customer, the Software Factory maturity CMMI-3 provides the customer with confidence in terms of its solid software development structure based on well established processes. Thus, such tailoring took into account the software development processes and the software lifecycle adopted by the supplier.

3 DBA´s Software Factory Model

The Software Factory concept uses industrial manufacturing fundamentals, such as standardized components, specialized skill sets, parallel processes and a predictable and scalable quality consistency. It can reach a higher level of application assembly even when assembling new or horizontal solutions. Software Factories have gained recent popularity as a cost-efficient way to reduce the time spent on the software development. Conceptually, Software Factories represent a methodology that seeks to incorporate pre-built standard functionalities into software which is typically disaggregated by domain.

The macro-flow diagram on the left of Figure 1 depicts the software development lifecycle typically performed by DBA. The related sub-processes represented by each of the four columns on the right side of Figure 1 show the adherence to SW-CMMI key process areas: Process Control, Change Management, Configuration Management and Quality Assurance.

In order to be produced by DBA´s Software Factory (FSW), a Software PROJECT shall be characterized and a particular project team be allocated to sort out the requirements and software development project management activities. Figure 2 presents the relationship among the PROJECT and FSW teams.

Prior to implementing such project management structure, the following steps and documentation should be forthcoming:

1. Customer provides an initial view of schedule and scope of the proposed job.
2. DBA gives an initial assessment of the effort required to meet those requirements (including an estimate number of Function Points required), and whether it can provide such services.
3. Customer will then further specify the elements of an application development service order, along with the following items: (a) All available documentation of the project, architecture, patterns, interfaces, etc.; (b) Hardware and

software configurations for the development and production environment; (c) Restrictions that should be observed during the software development.

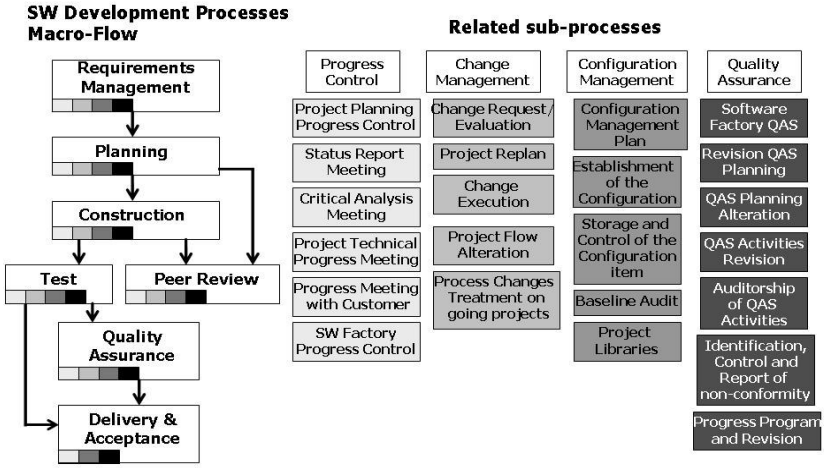

Figure 1. DBA Software Development Processes and related Sub-processes

4. Service Level parameters: DBA and Customer jointly establish the artefacts that are deployed at Analysis and Design and that describe the technical specifications necessary for FSW programmers to implement and test the code. Possible artefacts are: Use Case Specification; Information flow for System X - Template to describe the whole system information flow (to detail information flow for each Use Case).

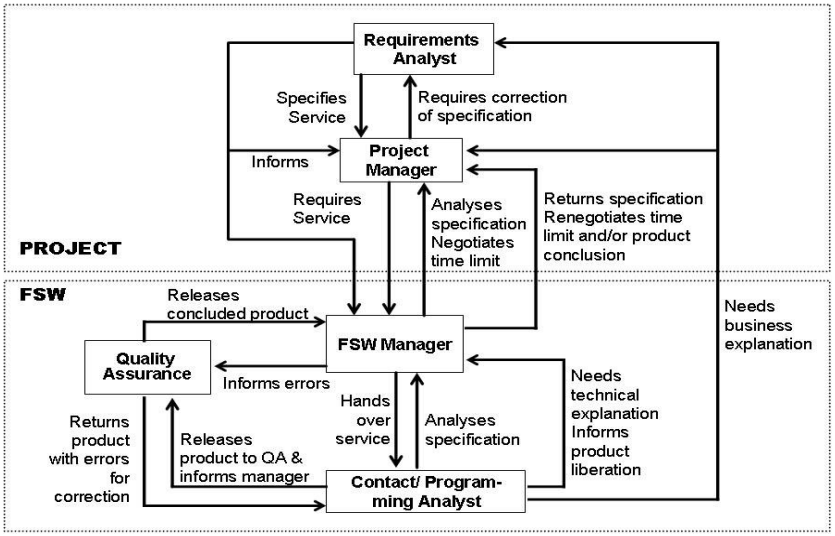

Figure 2. PROJECT and FSW relationship

5. The Project Designer will then issue a production order (OP) along with a set of artefacts to be provided (Use Case, Classes and Sequence diagrams).
6. The Project Test Plan including testing guidelines that should be followed for FSW developed product acceptance
7. Test Design **-** describes test cases for each technical specification.

Since SWPDC software supply by the FSW is subject of space domain application technology transfer to the Brazilian software industry, the PROJECT team was formed by one senior DBA Project Manager and two Software Analysts. The latter have been on-job trained in similar embedded application at INPE laboratory during six months before the SWPDC was effectively started.

4 Tailored Requirements

Software product assurance plays a mandatory role within the overall software engineering process. The complexity of software development and maintenance requires discipline to build quality into the product from the very beginning.

According to ECSS-Q-80B, the software product assurance requirements are grouped in three sets of activities: (i) the assurance programme implementation, (ii) the assurance of the development and maintenance process and (iii) the assurance of the quality of the software product.

Table 1. Requirements related to software product assurence programme implementation

Requirement	Description	Tailored Form	Tool	Document
Contractual Aspects	Supplier and Customer define a contract	Established when project QSEE was approved by governmental financial support (FINEP)	-	Agreements on Project Proposal
Software Product Assurance Planning and Control	Supplier provides a plan complying requirements and being approved by customer	SPA items included in DBA Software Development Plan document, reviewed and approved in SSR	Compliance matrix	SDPlan
Software Product Assurance Reporting	Supplier provides mechanisms for assessment of the current quality of the product	Reviews data package and tool allowing the customer to follow each Production Order (OP) into the FSW	Reports from a proprietary tool (SAF)	SDPlan
Non-conformance	Software Reviewer Board and baseline established by supplier/ customer	Formal Reviews point out the discrepancies (RIDs) and project control meetings	-	RB
Software Problem	Supplier defines and implements procedures for logging, analysis and corrections of software problems	Software Problems identified in the FSW have well established internal procedure. RNCs are problems identified on acceptance testing.	DBA FSW work-flow involves QA team	SDPlan

The tailoring process was carried out following these three groups in a supplementary way by means of careful analysis of their requirements. Table 1, Table 2 and Table 3 summarize, as examples, some of the applicable requirements analyzed and their tailored form for SWPDC product assurance. The first two columns of each table contain the ECSS-Q-80 requirement and its description, respectively. The column entitled *Tailored Form* describes the way the recommended requirement has been tailored in this project. Whenever a facility is provided to support the requirement, the *Tool* column introduces it. The *Document* column lists the customized documents in which such requirement is complied with. Table 1 lists some requirements corresponding to the group (i).

Requirements Baseline (RB) is the main document provided by customer. It imposes six formal reviews: Software Specification Review (SSR), Preliminary Design Review (PDR), Detailed Design Review (DDR), Critical Design Review (CDR), Qualification Review (QR), and Acceptance Review (AR). RB also defines the documents to be provided by the supplier: Software Development Plan (SDPlan), Software Test Plan (STPlan), Software Technical Specification (STSpec), Software Design Document (SDD), Software Test Specification (TestSpec) and Test Report (TRep). The following documents are required from the independent team: Independent Verification and Validation Plan (IVVPlan), Independent Verification and Validation Test Specification (IVVTSpec), Independent Verification and Validation Report (IVVRep). The Formal Reviews are documented in Technical Review Report (TRRep) which includes identified discrepancies (RIDs). During the acceptance phase, Non-conformance report (RNC) is delivered by IVV team to the supplier with a copy to the customer.

In respect to ECSS requirements presented in Table 1, a brief analysis about their correspondence with DBA Software Development processes and related sub-processes (Figure 1) shows that such requirements have met the Progress Control and Quality Assurance sub-processes.

Table 2 lists some requirements corresponding to group (ii). Since ECSS-Q-80B subdivides the software assurance process requirements in three categories, that organization was also adopted in that table.

ECSS requirements related to software lifecycle are met by two DBA Software Development processes: Requirements Management and Planning. And by related sub-processes: Progress Control and Change Management. The process assurance requirements applicable to all software engineering processes are met by Peer Review, Quality Assurance and Delivery & Acceptance processes. And by related sub-processes: Configuration Management and Quality Assurance. Whereas the process assurance requirements related to individual software engineering activities are met by Construction and Test processes. And by related sub-processes: Change Management.

Table 3 lists some requirements concerning group (iii). The correspondence between the requirements on Table 3 and DBA processes presented in the Figure 1 macro-workflow is consequence of the software development lifecycle phases. Thus, the first requirement row meets the Requirements Management and Planning processes. Second requirement meets the Construction process. And the last two rows meet the Test, Peer Review and Delivery & Acceptance processes.

Table 2. Requirements related to the software process assurence

Requirement	Description	Tailored Form	Tool	Document
Software Development Lifecycle				
Life cycle definition and review	Software life cycle defined by supplier and reviewed by customer	Software development lifecycle followed by DBA FSW has been approved by the customer on SSR	DBA Work-flow	SDPlan
Milestones	A series of technical meetings or reviews shall be defined	Reviews have been established on the RB by customer according to space mission life cycle.	-	RB and SDPlan
Applicable to all Software Engineering processes				
Documentation of Processes	Software project plans cover all software activities	Development Plan and Test Plan provided by supplier and by the IVV team	-	SDPlan, STPlan, IVVPlan
Handling of Critical Software	Apply measures to assure software confidence	IVV team designs model-based testing for automatic test cases generation.	CoFI Conda do	IVV Plan IVVTSpec IVVRep
Software Configuration Management	Supplier shall use a configuration management tool.	The system used by the FSW process has been approved by customer	VSS	SDPlan
Verification	Verification plan describes facilities, training and skills to carry out the verification activities	FSW comply the reviews imposed in RB adding pair review (internal verification technique). IVV Plan is provided by the independent team.	-	RB, SDPlan, IVVPlan
Related to Individual Software Engineering activities				
Software Requirement Analysis	Requirements specification shall be provided by customer as input.	Technical Specification document was elaborated by supplier taking RB as input customer provided	-	RB and Protocol Spec TSpec
Architectural Design	Use of a design methodology and design standards appropriated to the software type.	Customer required UML artefacts on Software Design, usually adopted in the FSW, and provided SDD document template.	Interprise Architecture	SDD
Software Delivery and Acceptance	Customer judges whether the product is acceptable, following previous agreement criteria.	The acceptance process defined in RB focus on testing at instrument level, subsystem and system. Model-based testing is used in the acceptance process.	CoFI methodology MME Conda do	IVV Plan IVVTSpec IVVRep

Almost every ECSS-Q-80 requirement has been analyzed during tailoring process. The exceptions are: Supplier Selection & Control and Procurement processes because they are related to activities out of FSW scope. It is worth mentioning that the Assessment and Improvement Processes, Process Metrics and Product Metrics are internally executed by FSW to support the processes presented in Figure 1.

Table 3. Requiments related to the software product quality assurance

Requirement	Description	Tailored Form	Tool	Doc.
Technical Specification	Software requirements shall be documented in a Software Technical Specification	Complete, detailed and unambiguous requirements are provided by supplier taking into account RB as input.	Traceability matrix	RB and STSpec
Design and Related Documentation	Software design with minimum hierarchy dependency and interfaces among software components	SDD produced by FSW contains the solution to the requirements of the STSpec. Use Case artefacts from UML are recommended.	-	SDD
Test and Validation Documentation	Detailed test planning (test cases, procedures, results) shall be consistent with test strategy.	A set of document was defined by customer in order to cover the two testing level strategy (internal to FSW and IVV)	-	STPlan, TestSpec IVVPlan, IVVTSpec
Reports and Analysis	Reports of all assurance, verification and validation activities	Two report documents were required to cover the two testing level strategy (internal to FSW and IVV). Also a RIDs and RNCs..	-	TRep IVVRep. TRRep

5 Conclusions

The tailored form contributed to simplify the embedded software technology transfer process from INPE to DBA. Specific requirements concerning independent verification and validation carried out by a third party team were defined because the full validation of the software product on the target computer was not feasible within the FSW context. This team participation on the reviews allowed for early understanding of the software operational behavior which contributed to the applicability of model-based testing techniques as part of the acceptance process.

Although not familiar with space domain, the supplier maturity level, CMMI-3 formally evaluated, facilitates FSW to comply with the requirements imposed by the customer. The project-oriented approach adopted by DBA to deal with the well stabilized processes of its FSW minimized the difficulties inherent to adding new project knowledge domain to FSW environment.

6 References

[1] Quality of Space Application Embedded Software (QSEE). Available at: <http://www.cea.inpe.br/~qsee>. Accessed on: Feb. 23th 2007.
[2] Space Engineering - Software, ECSS-E-40 standard, May 2005.
[3] Space Product Assurance – Software, ECSS-Q-80A and B standard, October 2003.