

## **Annex A: ECSS-E40 Tailoring EXERCISE 2 –CAD TOOL**

### **1 Introduction**

ECSS-E40 (Space Engineering – Software) has replaced the PSS-05 for the development of new space software, that is software involved in the production of space systems. ECSS-E40 has the same goal as its PSS-05 predecessor, which is to assist developers in applying good software practices during the development. Compared to PSS-05, however, ECSS-E40 allows for more flexibility in that:

- The standard encompasses a set of software processes without prescribing any specific life cycle.
- Each software process terminates with reviews that directly tie with those of a satellite development, so that the former explicitly contribute to the progress of the latter.
- Each software process releases descriptive information, not necessarily a set of documents with prescribed table of contents. The contractor may place and organise the required information in whatever form they may choose to. The contractor is able to apply their specific development methodology, as long as that satisfied the ECSS-E40 process requirements.
- The ECSS-E40 standard requirements must be tailored and adjusted to the specific needs, the costs and risks of the project.

This annex specifically addresses the last item of the above list. The latest version ECSS-E40B is made up of several sections, of which only section 5 expresses requirements.

### **2 Description of the tailoring**

#### **2.1 Project characteristics**

[Role/Importance of software in the system]

The system is made of the only software tool running on a standard PC, with a graphical man machine interface. 50 users will use it. The specification are not complex, but does not exist yet, the design is standard for a software company used to that kind of tools. The tool is installed by a Windows "Install" procedure

[Estimation of the risky areas]

The risk is that the users complain about bugs in the tool and compromise the customer's business.

#### **2.2 Roles**

The customer is the mechanical company. The Supplier is the Contractor. The Users are 50 people over Europe. The maintainer is the Contractor, during the warranty period as far as this contract is concerned. The Operator is the User.

#### **2.3 Processes involved**

[Justification for the main tailoring according to the project features/needs]

The following software processes are part of this project:

- The software requirement and architecture engineering process, which describe the supplier specification and architectural design of the software,

- The software design and implementation engineering process, where the software is detailed designed, coded, unit tested and integrated,
- The software validation where the software is validated with respect to the Technical Specification and to the customer requirements (Requirements baseline),
- The software delivery and acceptance process where the software is delivered, installed in its target environment and formally accepted by the customer,
- The software maintenance process, when the software is maintained.

The following software processes are not part of this project:

- The system engineering process, which describe the customer and system needs for the software, because the system is made of a single component: the software tool. The system requirements and the software requirements can be merged.
- The software operation process, when the software is operated, because the software is dead simple to operate
- The software management process, where the life cycle and the development plan are defined. However, the RFQ will require to have a draft development plan in the proposal, in order to evaluate the software engineering maturity of the bidders

There is no SRR because no System requirement engineering process

There is a PDR because the Customer wants to make sure that the Supplier has well understood what he wants.

There is no CDR because the Supplier does not want to have visibility nor control on the implementation.

The QR and the AR are merged in a single Acceptance of the tool by the Customer.

The maintenance will be part of another contract, but needs to be prepared in this contract with the maintenance plan.

Due to the low budget, the verification activities are waived. However, because the tool needs a strong validation (user satisfaction), the traceability from software requirements to validation tests is required, so as to ensure a test coverage of 100%.

### 3 Software process mapping to work packages

The software development processes introduced in ECSS-E40B are mapped on the work packages and activities of the statement of work in the following way.

<b>ECSS-E40B Software activities</b>	<b>Reference in the statement of work</b>
System requirements, Hardware/Software partitioning, Requirement Baseline	
Software SRR	
Software requirements, Top level architecture, Technical Specification	
Software PDR	
Design, code, unit tests, integration tests	
Software CDR	
Validation against the TS	
Validation against the RB	
Software QR	
Delivery and Acceptance	
Software AR	
Software operation	
Software maintenance	
Project management	
Verification activities	

#### 4 List of ECSS-E40 applicable requirements

ECSS-E-40 Draft 3 part 1B from 19 June 2003	APPLICABILITY
<b>5.2 System Engineering Processes related to software</b>	
5.2.2.1 System requirements specification	N/A
5.2.2.2 System and functional criticality analysis	
5.2.2.3 MMI software mock up requirements	
5.2.2.4 MMI general requirements and guidelines	
5.2.3.1a System design	
5.2.3.1b System design to system requirements conformance	
5.2.3.1c System requirements to system design traceability	
5.2.3.2a Software-hardware interface requirements	
5.2.3.2b Traceability to system partitioning	
5.2.3.2c System partition with definition of items (HW, SW, human operation)	
5.2.3.2d System configuration items list	
5.2.4.2 qualification engineering requirements	
5.2.4.3 Software validation requirements at system level	
5.2.4.4 requirement baseline verification	
5.2.4.5 SRR	
5.2.5.1 Identification of observability requirements	
5.2.5.2 Control and data interfaces for system level integration	
5.2.5.3 Data medium requirements for integration	
5.2.5.4 System database specification (content and use)	
5.2.5.5 Identification of development constraints (to support the software integration into the system)	
5.2.5.6 Definition of constraints for software to be reused	
5.2.5.7 Identification of customer's input for software integration into the system	
5.2.5.8 Identification of customer's output for software integration into the system	
5.2.5.9 Planning of supplier support to system integration	
5.2.6.1 Phasing and management / operational plan	
5.2.6.2 System requirements definition for software operations	
5.2.7.1 Software maintenance requirements	
5.2.7.2 Definition of inflight capabilities for flight software	
<b>5.3 Software Management Process</b>	
5.3.2.1 Definition of software life cycle	Draft development plan in the proposal
5.3.2.2 Software life cycle identification	
5.3.2.3 Identification of inputs and outputs associated to each phases	
5.3.2.4 Identification of documentation relevant to each	
5.3.2.5 Identification of interface between the development and the maintenance processes	
5.3.2.6 Requirements baseline at the SRR	
5.3.2.7 Software technical specification phase	
5.3.2.8 PDR (Preliminary Design Review)	
5.3.2.9 DDR (detailed design review for flight software)	
5.3.2.10 CDR (Critical Design Review)	
5.3.2.11 software verification and validation process	
5.3.2.12 QR (Qualification Review)	

5.3.2.13 AR (Acceptance Review)	Draft development plan in the proposal
5.3.2.14 Validation activities phasing wrt AR	
5.3.2.14 Software procurement process implementation	
5.3.3.2 Support to software reviews	
5.3.3.3 Technical reviews	
5.3.4.1 Interface definition	
5.3.4.2 Interface management procedures	
5.3.5.1 Technical budget and margin philosophy	
5.3.5.2 Technical budget and margin status at each milestone	
<b>5.4 Software requirements &amp; architecture engineering process</b>	
5.2.4.1 Establishment and documentation of software requirements / software requirements specification:	Yes
5.4.2.1-a software requirements – functional and performance	Yes
5.4.2.1-b software requirements – software product quality requirements	No
5.4.2.1-c software requirements – security specifications	No
5.4.2.1-d software requirements – human factors / ergonomics specifications	Yes
5.4.2.1-e software requirements – data definition and DataBase requirements	Yes
5.4.2.1-f software requirements – Interfaces external to the software item	Yes
5.4.2.2 Definition of functional and performance requirements for inflight modification	No
5.4.2.3 Identification of requirements unique identifier	Yes
5.4.2.4 Definition of a software logical model	Yes if UML
5.4.2.5 Definition of a behavioural view	No
5.4.2.6a MMI specifications for software	As you want!
5.4.2.6b Report on evaluation on MMI specifications using a software mock up	
5.4.2.6c end users participation in the MMI mock up evaluation	
5.4.3.1 Transformation of software requirements into a software architecture	Yes
5.4.3.2 Software design description	Yes
5.4.3.3 Software design documentation	Yes
5.4.3.4 Software architectural design contents	No
5.4.3.5 Software design method	No
5.4.3.6a Selection of a computational model for real time software	No
5.4.3.6b Selection of a computational model for real time software	No
5.4.3.7 Description of software dynamic behaviour	No
5.4.3.8 Development and documentation of the software interfaces	Yes
5.4.3.9 Definition of methods and tools for software to be reused	N/A
5.4.3.10 Evaluation of potential reuse of software	
5.4.3.11 Evaluation of reuse of predeveloped software	
5.4.3.12 Analysis of potential reusability	
5.4.3.13 Definition and documentation of the software integration requirements and plan	No
5.4.3.8 Conducting a PDR (Preliminary Design Review)	Yes
<b>5.5 Software design &amp; implementation engineering process</b>	
5.5.2.1 Detailed design of each software components	No
5.5.2.2 Development and documentation of the software interface detailed design	No
5.5.2.3 Production of software items physical model	No
5.5.2.4 Utilization of method for software static design	No
5.5.2.5a Description of the software dynamic aspects of physical model for	No

real-time software	
<b>5.5.2.5b</b> Description of the software dynamic aspects of physical model for real-time software	No
<b>5.5.2.6</b> Utilization of description techniques for the software behaviour	No
<b>5.5.2.7</b> Determination of design methods consistency for real-time software	No
<b>5.5.2.8</b> Development and documentation of the software user manual	Yes
<b>5.5.2.9</b> Definition and documentation of the software unit test requirements and plan	No
<b>5.5.2.10</b> Updating of the software integration requirements and plan	No
<b>5.5.2.11</b> Conducting a DDR (Detailed Design Review) for flight software	No
<b>5.5.3.1</b> Development and documentation of the software units, test procedures and test data	Yes (code only)
<b>5.5.3.2</b> Software unit testing	No
<b>5.5.3.3</b> software user manual updating	Yes
<b>5.5.3.4</b> Updating of the software integration test requirements and plan	No
<b>5.5.4.1</b> Software integration test plan development	No
<b>5.5.4.2</b> Software units and software components integration and testing	No
<b>5.5.4.3</b> Software user manual updating	Yes
<b>5.6 Software validation process</b>	
<b>5.6.2.1</b> Determination of the validation effort	Yes
<b>5.6.2.2</b> Establishment of a validation process	Yes
<b>5.6.2.3</b> Selection of an ISVV organization	No
<b>5.6.2.4</b> Development and documentation of a validation plan	Yes
<b>5.6.3</b> Validation with respect to the technical specification	
<b>5.6.3.1</b> Development and documentation of a software validation testing specification (SVTS) wrt TS	No
<b>5.6.3.2</b> Conducting the validation wrt TS	No
<b>5.6.3.3</b> Updating the software user manual	No
<b>5.6.3.4</b> Test readiness review	No
<b>5.6.3.5</b> Conducting a CDR (Critical Design Review)	No
<b>5.6.4</b> Validation with respect to the requirements baseline	
<b>5.6.4.1</b> Development and documentation of a software validation testing specification (SVTS) wrt RB	Yes
<b>5.6.4.2</b> Conducting the validation wrt RB	Yes
<b>5.6.4.3</b> Updating the software user manual	Yes
<b>5.6.4.4</b> Test readiness review	No
<b>5.6.4.5</b> Conducting a QR (Qualification Review)	Merged with AR
<b>5.7 Software delivery and acceptance</b>	
<b>5.7.2</b> Software delivery and installation	
<b>5.7.2.1</b> Preparation of the software product	Yes
<b>5.7.2.2</b> Supplier's provision of training and support	No
<b>5.7.2.3</b> Installation planning	No
<b>5.7.2.4</b> Installation activities reporting	No
<b>5.7.3</b> Software acceptance	
<b>5.7.3.1</b> Acceptance test planning	Yes
<b>5.7.3.2</b> Acceptance test execution	Yes
<b>5.7.3.3</b> Executable code generation and installation	Yes
<b>5.7.3.4a</b> Supplier's support to customer's acceptance	Yes
<b>5.7.3.4b</b> Link with Q	No
<b>5.7.3.4c</b> Acceptance testing documentation	Yes
<b>5.7.3.5</b> Evaluation of acceptance testing	Yes
<b>5.7.3.6</b> Conducting an AR (Acceptance Review)	Yes

<b>5.8 Software verification process</b>	
5.8.2.1 Determination of the verification effort for the project	No
5.8.2.2 Establishment of the verification process, methods and tools	
5.8.2.3 Selection of the organization responsible for conducting the verification	
5.8.2.4 Development and documentation of a verification plan covering the software verification activities	
<b>5.8.3 Verification activities</b>	
5.8.3.1 Verification of software requirements	No
5.9.3.2 Verification of software architectural design	No
5.8.3.3 Verification of software detailed design	No
5.8.3.4 Verification of code	No
5.8.3.5 Verification of software integration	No
5.8.3.6 Verification of software documentation	No
5.8.3.7 Verification of test specifications	No
5.8.3.8 verification of software validation with respect to TS and RB	No
5.8.3.9 Evaluation of validation: complementary system level validation	No
5.8.3.10 problem and non conformance handling	No
<b>5.8.3.11 Schedulability analysis for real-time software</b>	
5.8.3.11a /as support for verification of software requirements & architectural design	N/A
5.8.3.11b /as support for verification of software detailed design	
5.8.3.11c /as support for verification of software coding and testing	
<b>5.8.3.12 Technical budgets management</b>	
5.8.3.12a / as support for verification of software requirements & architectural design / sizing (memory) and timing (CPU load) estimation	N/A
5.8.3.12b / as support for verification of software detailed design/ sizing (memory) and timing (CPU utilization in WCET) estimation refinement	
5.8.3.12c/ as support for verification of software coding and testing / sizing (memory) and timing (CPU utilization in WCET) calculation	
<b>5.8.3.13 Behaviour modelling verification</b>	
5.8.3.13a/ as support for verification of software requirements & architectural design / verification of the behavioural view of the logical model	N/A
5.8.3.13b/ as support for verification of software detailed design / modelling the software behaviour and verifying by means of the techniques used for its description	
5.8.3.14 Verification of design: feasibility of testing / availability of appropriate verification points, assertions, capability of fault injection	
<b>5.9 Software Operation Process</b>	
5.9.2.1 Operational plans and standards development	N/A
5.9.2.2 Problem handling procedures definition	
5.9.2.3 Operational testing definition	
5.9.3.1 Operational testing execution	
5.9.3.2 software operational requirements demonstration	
5.9.4 software operation	
5.9.5.1 user's assistance	
5.9.5.2 handling of user's requests	
5.9.5.3 provisions of work-around solutions	
<b>5.10 Software Maintenance Process</b>	
5.10.2.1 Software maintenance process planning	Yes
5.10.2.2 Software maintenance process: procedures, methods and standards	Yes
5.10.2.3 problem reporting and handling	Yes

5.10.2.4 Implementation of configuration management process	Yes
5.10.3.1 Problem analysis	N/A
5.10.3.2 Problem verification	
5.10.3.3 Development of options for modifications	
5.10.3.4 Documentation of problem, analysis and implementation	
5.10.3.5 Customer approval of selected modifications options	
5.10.4.1 Analysis and documentation of product modification	
5.10.4.2 Documentation of software product changes	
5.10.4.3 Invoking of software engineering process for modification implementation	
5.10.5 Conducting maintenance review	
5.10.6.1 Applicability of this standard to software migration	
5.10.6.2 Migration planning and execution	
5.10.6.3 Contribution to the migration plan	
5.10.6.4 Preparation for migration	
5.10.6.5 Notification of transition to migrated system	
5.10.6.6 Post-operation review	
5.10.6.7 Maintenance and accessibility of data of former system	
5.10.7.1 Retirement planning	
5.10.7.2 Notification to the operator of retirement	
5.10.7.3 identification of requirements for software retirement	
5.10.7.4 maintenance and accessibility to data of the retired product	

## 5 Documentation

The Supplier must provide all the expected outputs required by the Software Standards sub clauses, organized in the identified Documentation Files:

- Requirements Baseline
- Technical Specification
- Design Definition File
- Design Justification File
- Management File
- Maintenance File
- Product Assurance File

Each file contains a set of documents. The ECSS software standards include some DRDs, describing the most important software documents. The standardised DRD list is a subset of the exhaustive list of documents to be produced in order to cover all the work output required by the standards. For the other documents, some unofficial DRDs are available.

The expected output of the requirements resulting of this tailoring can be placed in the following standardised DRDs:

Document item	Acronym in DRD	Folder	Applicability
(Software) System Specification	SSS	RB	No
Software Interface Requirements Document	-		

Software Requirements Specification	SRS	TS	<b>Yes</b>	
Software Interface Control Document	-			
Software Design Document (including Software Components Design)	SDD	DDF	No	
Software Source Code	-			
Software Configuration File	SCF		No	
Software Release Document	SRD		<b>Yes</b>	
Training material			No	
Software User Manual	SUM		<b>Yes</b>	
Software Reuse File	SRF			No
Software Verification Plan	SverP		No	
Software Validation Plan	SValP		<b>Yes</b>	
ISVV Plan	-		No	
Software Units/Integration Test Plan	SUITP		No	
Software Validation Testing Specification wrt TS (Analyses & Inspection) verification report wrt TS	SVTS -		No	
Software Validation Testing Specification wrt RB (Analyses & Inspection) verification report wrt RB	SVTS -		<b>Yes</b> <b>Yes</b>	
Software Traceability Matrices	-		No	
Software Acceptance Test Plan			<b>Yes</b>	
Software Requirements Verification Report	-		No	
Software Arch. Design and Interface Verification Report	-		No	
Software Detailed Design verification Report	-		No	
Software Code Verification Report	-	DJF	No	
Software Documentation Verification Report	-		No	
Software Integration Verification Report	-		No	
Software Unit/Integration Test Report	-		No	
Software Validation Test Report wrt TS	-		No	
Software Validation Test Report wrt RB	-		<b>Yes</b>	
Validation Evaluation Report wrt TS	-		No	
Validation Evaluation Report wrt RB			Yes	
Software Design & Test Evaluation Report	-		No	
Acceptance Test Report			Yes	
Installation Plan			No	
Installation Report			No	
Software Budget Report	-		No	
Software Acceptance Data Package	-		?	
Schedulability Analyses	-		No	
Software Behavior Verification	-		No	
Testing Feasibility Report	-		No	
Problems and Nonconformance Report	-		No	
Milestones Report	-		No	
Software Maintenance Plan	-		MF	<b>Yes</b>
PR & NCR - Modification analysis report -Problem analysis report				No
Migration Plan	-			No
Software Operational Plan	-		OP	
Software Development Plan	SDP		Proposal	
Software Product Assurance Plan	SPAP	PAF	No	
Software Criticality Analysis	-		No	
Software Product Assurance Report	-		No	

## 6 Intellectual Property Rights

The software to be delivered as part of the contract is to be considered as:

General case software [IPRs to Contractor, license to ESA]	X
Operational software [IPRs to ESA]	
Co-funded development [IPRs to Contractor with adaptation]	
Public domain software [IPRs to nobody]	
Open source software [IPRs to Contractor + imposed pre-defined license, e.g. GNU]	

The software to be delivered as part of the contract is expected to make use of:

COTS	
Supplier proprietary software	
Open Source software	