

Example of SOW Annex : ECSS-E40 Tailoring

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 baselined version ECSS-E40 part 1B (from 28 November 2003) is made up of several sections, of which only section 5 expresses requirements.

2 Description of the tailoring

2.1 Project characteristics

A software project or product can be characterized as a function of technical, operational and management factors that are “key drivers” for the tailoring process of the ECSS-E-40B Standard.

Technical factors	Low	Medium	High
Novelty of the domain of application			
Complexity of the software, as measured by the number of interface or similar metrics [External interfaces]			
Amount of software that has to be produced [estimated]			
Reusability required of the software being developed			
Interface to system development projects			
Degree of use of COTS or existing software			
Maturity of the COTS			
Completeness or stability of the user requirements			
Operational factors			
Type of application (platform, payload, experiment)			
Number of potential users of the software			
Criticality of the software as measured by the consequences of its failure			
Expected lifetime (until retirement's notification) of the			

software			
Number of sites where the software is used			
Maintenance constraints			
Management factors			
Amount of effort required to develop the software			
Amount of time required to develop the software			
Budget requirements for implementing and operating the software			
Schedule requirements for delivering the software (In months)			
Number of people required to develop, operate and maintain the software			

2.2 Project risks

Hereafter, a table giving a rough estimation of the magnitude of the most encountered risk in software development.

Risk description	Risk magnitude		
	LOW	MEDIUM	HIGH
Complex specification			
Tricky design			
Reliability critical			
Safety critical			
Long term use			
supplier's background experience and maturity			

2.3 Roles

- The Customer is
- The Supplier is the
- The User is
- The Maintainer is
- The Operator is

2.4 Processes involved

The following software processes are part of this project (e.g.):

- The system engineering processes related to software,
- The software requirement engineering process, which describes the software requirements specification and architectural design,
- The software design engineering process, where the software is detailed designed, coded and tested at unit and software component levels (unit and integration testing)

- The software validation process where the software is validated, at first against the supplier specification (technical specification), then against the system requirements document (Requirement Baseline),
- The software verification since a particular effort has to be done to verify the intermediate work outputs of the final software product
- The software delivery and acceptance process,
- The software management process as the bidder will have to include in its proposal a draft development plan (including organization breakdown structure, work breakdown structure, life cycle, development methods and tools, reused software products, documentation to be produced, risk management, milestones, deliveries).

The following software processes are not part of this project (e.g.):

- The software operation process as no helpdesk is necessary to operate this software.
- The software maintenance process as this contract covers only the development activities to be performed (except during the warranty period during which some corrective maintenance activities can be requested by the customer)

3 Software process mapping to work packages

The software development processes introduced in ECSS-E40-Part 1 B are mapped on the work packages and activities of the Statement of Work in the following way.

ECSS-E40B Software activities	Reference in the Statement of Work
System requirements, Hardware/Software partitioning, Requirement Baseline	
Software SRR	
Software requirements, Top level architecture, Technical Specification	
Software PDR	
Detailed design, code, unit tests, integration tests,	
Software DDR	
Software CDR	
Validation against the TS Validation against the RB	
Software QR	
Delivery	
Software AR	
Software operation	
Software maintenance	
Project management	
Verification activities	

4 List of ECSS-E40 applicable requirements

ECSS-E-40 part 1B from 28 November 2003	Appli cabil ity	Notes
5.2 System Engineering Processes related to software		
5.2.2.1 System requirements specification	Y	
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	Y	
5.2.3.1b System design to system requirements conformance	Y	
5.2.3.1c System requirements to system design traceability	Y	
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 (verification & validation process requirements)		
5.2.4.3 Software validation requirements at system level		
5.2.4.4 requirement baseline verification		
5.2.4.5 SRR	Y	
5.2.5.1 Identification of observability requirements		
5.2.5.2 Control and data interfaces for system level integration	Y	
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	Y	
5.2.7.1 Software maintenance requirements	Y	
5.2.7.2 Definition of inflight capabilities for flight software		
5.3 Software Management Process		
5.3.2.1 Definition of software life cycle	Y	
5.3.2.2 Software life cycle identification	Y	
5.3.2.3 Identification of inputs and outputs associated to each phases	Y	
5.3.2.4 Identification of documentation relevant to each	Y	
5.3.2.5 Identification of interface between the development and the maintenance processes	Y	
5.3.2.6 Requirements baseline at the SRR	Y (if 5.2)	

ECSS-E-40 part 1B from 28 November 2003	Appli cabilit y	Notes
5.3.2.7 Software technical specification phase	Y	
5.3.2.8 PDR (Preliminary Design Review)	Y	
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	Y	
5.3.2.12 QR (Qualification Review)		
5.3.2.13 AR (Acceptance Review)	Y	
5.3.2.14 Validation activities phasing wrt AR		
5.3.2.15 Software procurement process implementation		
5.3.3.2 Support to software reviews	Y	
5.3.3.3 Technical reviews	Y	
5.3.4.1 Interface definition	Y	
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		
5.4 Software requirements & architecture engineering process		
5.2.4.1 Establishment and documentation of software requirements / software requirements specification:	Y	
5.4.2.1-a software requirements – functional and performance	Y	
5.4.2.1-b software requirements – software product quality requirements		
5.4.2.1-c software requirements – security specifications		
5.4.2.1-d software requirements – human factors / ergonomics specifications		
5.4.2.1-e software requirements – data definition and DataBase requirements	Y	
5.4.2.1-f software requirements – Interfaces external to the software item	Y	
5.4.2.2 Definition of functional and performance requirements for inflight modification		
5.4.2.3 Identification of requirements unique identifier	Y	
5.4.2.4 Definition of a software logical model		
5.4.2.5 Definition of a behavioural view		
5.4.2.6a MMI specifications for software		
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		
5.4.3.2 Software design description		
5.4.3.3 Software design documentation		
5.4.3.4 Software architectural design contents		
5.4.3.5 Software design method		
5.4.3.6a Selection of a computational model for real time software		
5.4.3.6b Selection of a computational model for real time software		

ECSS-E-40 part 1B from 28 November 2003	Applicability	Notes
5.4.3.7 Description of software dynamic behaviour		
5.4.3.8 Development and documentation of the software interfaces		
5.4.3.9 Definition of methods and tools for software to be reused		
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		
5.4.3.14 Conducting a PDR (Preliminary Design Review)	Y	
5.5 Software design & implementation engineering process		
5.5.2.1 Detailed design of each software components		
5.5.2.2 Development and documentation of the software interface detailed design		
5.5.2.3 Production of software items physical model		
5.5.2.4 Utilization of method for software static design		
5.5.2.5a Description of the software dynamic aspects of physical model for real-time software		
5.5.2.5b Description of the software dynamic aspects of physical model for real-time software		
5.5.2.6 Utilization of description techniques for the software behaviour		
5.5.2.7 Determination of design methods consistency for real-time software		
5.5.2.8 Development and documentation of the software user manual		
5.5.2.9 Definition and documentation of the software unit test requirements and plan		
5.5.2.10 Updating of the software integration requirements and plan		
5.5.2.11 Conducting a DDR (Detailed Design Review) for flight software		
5.5.3.1 Development and documentation of the software units, test procedures and test data	Y	
5.5.3.2 Software unit testing		
5.5.3.3 software user manual updating		
5.5.3.4 Updating of the software integration test requirements and plan		
5.5.4.1 Software integration test plan development		
5.5.4.2 Software units and software components integration and testing		
5.5.4.3 Software user manual updating		
5.6 Software validation process		
5.6.2.1 Determination of the validation effort		
5.6.2.2a Establishment of a validation process		
5.6.2.2b Establishment of a validation process		
5.6.2.3a Selection of an ISVV organization		
5.6.2.4 Development and documentation of a validation plan		
5.6.3 Validation with respect to the technical specification		

ECSS-E-40 part 1B from 28 November 2003	Appli cabilit y	Notes
5.6.3.1 Development and documentation of a software validation testing specification (SVTS) wrt TS		
5.6.3.2 Conducting the validation wrt TS		
5.6.3.3 Updating the software user manual		
5.6.3.4 Test readiness review		
5.6.3.5 Conducting a CDR (Critical Design Review)		
5.6.4 Validation with respect to the requirement baseline		
5.6.4.1 Development and documentation of a software validation testing specification (SVTS) wrt RB	Y	
5.6.4.2 Conducting the validation wrt RB	Y	
5.6.4.3 Updating the software user manual		
5.6.4.4 Test readiness review		
5.6.4.5 Conducting a QR (Qualification Review)		
5.7 Software delivery and acceptance		
5.7.2 Software delivery and installation		
5.7.2.1 Preparation of the software product	Y	
5.7.2.2 Supplier's provision of training and support		
5.7.2.3 Installation planning		
5.7.2.4 Installation activities reporting		
5.7.3 Software acceptance		
5.7.3.1 Acceptance test planning	Y	
5.7.3.2 Acceptance test execution	Y	
5.7.3.3 Executable code generation and installation	Y	
5.7.3.4a Supplier's support to customer's acceptance	Y	
5.7.3.4b Links with Q		
5.7.3.4c Acceptance testing documentation	Y	
5.7.3.5 Evaluation of acceptance testing	Y	
5.7.3.6 Conducting an AR (Acceptance Review)	Y	
5.8 Software verification process		
5.8.2.1 Determination of the verification effort for the project		
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		
5.8.3 Verification activities		
5.8.3.1 Verification of software requirements		
5.8.3.2 Verification of software architectural design		
5.8.3.3 Verification of software detailed design		
5.8.3.4 Verification of code		
5.8.3.5 Verification of software integration		
5.8.3.6 Verification of software documentation		
5.8.3.7 Verification of test specifications	Y	
5.8.3.8 verification of software validation with respect to TS and RB		
5.8.3.9 Evaluation of validation: complementary system level validation		
5.8.3.10 problem and non conformance handling	Y	
5.8.3.11 Schedulability analysis for real-time software		

ECSS-E-40 part 1B from 28 November 2003	Applicability	Notes
5.8.3.11a /as support for verification of software requirements & architectural design		
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		
5.8.3.12 Technical budget management		
5.8.3.12a / as support for verification of software requirements & architectural design / sizing (memory) and timing (CPU load) estimation		
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		
5.8.3.13 Behaviour modelling verification		
5.8.3.13a / as support for verification of software requirements & architectural design / verification of the behavioural view of the logical model		
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		
5.9 Software Operation Process		
5.9.2.1 Operational plans and standards development		
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	Y	
5.9.5.1 user's assistance	Y	
5.9.5.2 handling of user's requests	Y	
5.9.5.3 provisions of work-around solutions	Y	
5.10 Software Maintenance Process		
5.10.2.1 Software maintenance process planning		
5.10.2.2 Software maintenance process: procedures, methods and standards		
5.10.2.3 problem reporting and handling		
5.10.2.4 Implementation of configuration management process		
5.10.3.1 Problem analysis		
5.10.3.2 Problem verification	Y	
5.10.3.3 Development of options for modifications	Y	
5.10.3.4 Documentation of problem, analysis and implementation		
5.10.3.5 Customer approval of selected modifications options	Y	
5.10.4.1 Analysis and documentation of product modification		

ECSS-E-40 part 1B from 28 November 2003	Applicability	Notes
5.10.4.2 Documentation of software product changes	Y	
5.10.4.3 Invoking of software engineering process for modification implementation	Y	
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 ECSS software standards are completed with some DRDs, describing the most important software documents. The 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.

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

Document item	Acronym in DRD	Folder	Applicability
(Software) System Specification	SSS	RB	
Software Interface Requirements Document	-		
Software Requirements Specification	SRS	TS	
Software Interface Control Document	-		
Software Design Document (including Software Components Design)	SDD	DDF	
Software Source Code	-		
Software Configuration File	SCF		
Software Release Document	SRD		
Training material			
Software User Manual	SUM		
Software Reuse File	SRF	DJF	
Software Verification Plan	SverP		
Software Validation Plan	SVaIP		
ISVV Plan	-		
Software Units/Integration Test Plan	SUITP		
Software Validation Testing Specification wrt TS	SVTS		
(Analyses & Inspection) verification report wrt TS	-		
Software Validation Testing Specification wrt RB	SVTS		
(Analyses & Inspection) verification report wrt RB	-		
Software Traceability Matrices	-		
Software Acceptance Test Plan			
Software Requirements Verification Report	-		
Software Arch. Design and Interface Verification Report	-		
Software Detailed Design verification Report	-		
Software Code Verification Report	-		
Software Documentation Verification Report	-		
Software Integration Verification Report	-		
Software Unit/Integration Test Report	-		
Software Validation Test Report wrt TS	-		
Software Validation Test Report wrt RB	-		
Validation Evaluation Report wrt TS	-		
Validation Evaluation Report wrt RB	-		
Software Design & Test Evaluation Report	-		
Acceptance Test Report			
Installation Plan			
Installation Report			
Software Budget Report	-		

Software Acceptance Data Package	-		
Schedulability Analyses	-		
Software Behavior Verification	-		
Testing Feasibility Report	-		
Problems and Nonconformance Report	-		
Milestones Report	-		
Software Maintenance Plan	-	MF	
PR & NCR - Modification analysis report -Problem analysis report			
Migration Plan	-		
Retirement Plan			
Software Operational Plan	-	OP	
Software Development Plan	SDP		
Software Product Assurance Plan	SPAP		
Software Criticality Analysis	-	PAF	
Software Product Assurance Report	-		

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]	
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	