TECHNICAL REPORT

RAPPORT TECHNIQUE

TECHNISCHER BERICHT

CEN/TR 17603-40

June 2022

ICS 49.140; 35.080

This document is a preview generated by EVS

English version

Space engineering - Software engineering handbook

Ingénierie spatiale - Guide d'ingénierie logiciel

Raumfahrttechnik - Handbuch zur Softwareentwicklung

This Technical Report was approved by CEN on 20 April 2022. It has been drawn up by the Technical Committee CEN/CLC/JTC 5.

CEN and CENELEC members are the national standards bodies and national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.





CEN-CENELEC Management Centre:
Rue de la Science 23, B-1040 Brussels

© 2022 CEN/CENELEC All rights of exploitation in any form and by any means reserved worldwide for CEN national Members and for CENELEC Members.

Ref. No. CEN/TR 17603-40:2022 E

Table of contents

European Foreword **6**

Introduction **7**

1 Scope **8**

2 References **10**

3 Terms, definitions and abbreviated terms **12**

3.1 Terms from other documents 12
3.2 Terms specific to the present document 12
3.3 Abbreviated terms 12

4 Introduction to space software **15**

4.1 Getting started 15
4.1.1 Space projects 15
4.1.2 Space standards: The ECSS System 15
4.1.3 Key characteristics of the ECSS System 16
4.1.4 Establishing ECSS Standards for a space project 16
4.1.5 Software / ECSS Standards relevant for Software 17
4.1.6 Why are standards a MUST for the software development process? 19
4.1.7 Executing a space software project 20
4.1.8 Disciplines in Space Software Projects 21
4.2 Getting compliant 22
4.2.1 The ECSS-E-ST-40C roles 22
4.2.2 Compliance with the ECSS-E-ST-40C 26
4.2.3 Characterization of space software leading to various interpretations/applications of the standard 28
4.2.4 Software criticality categories 31
4.2.5 Tailoring 33
4.2.6 Contractual and Organizational Special Arrangements 35

5 Guidelines **41**

5.1 Introduction 41
5.2 Software related system requirement process 41
5.2.1 Overview 41
5.2.2 Software related system requirements analysis 44
5.2.3 Software related system verification 45
5.2.4 Software related system integration and control 46
5.2.5 System requirement review 53
5.3 Software management process 54
5.3.1 Overview 54
5.3.2 Software life cycle management 54
5.3.3 Software project and technical reviews 55
5.3.4 Software project reviews description 55
5.3.5 Software technical reviews description 56
5.3.6 Review phasing 63
5.3.7 Interface management 64
5.3.8 Technical budget and margin management 65
5.3.9 Compliance to this Standard 65
5.4 Software requirements and architecture engineering process 65
5.4.1 Overview 65
5.4.2 Software requirement analysis 65

This document is a preview generated by EVS

5.4.3 Software architectural design 67
5.4.4 Conducting a preliminary design review 76
5.5 Software design and implementation engineering process 76
5.5.1 Overview 76
5.5.2 Design of software items 76
5.5.3 Coding and testing 78
5.5.4 Integration 81
5.6 Software validation process 83
5.6.1 Overview 83
5.6.2 Validation process implementation 84
5.6.3 Validation activities with respect to the technical specification 91
5.6.4 Validation activities with respect to the requirement baseline 91
5.7 Software delivery and acceptance process 93
5.7.1 Overview 93
5.7.2 Software delivery and installation 93
5.7.3 Software acceptance 94
5.8 Software verification process 97
5.8.1 Overview 97
5.8.2 Verification process implementation 97
5.8.3 Verification activities 100
5.9 Software operation process 104
5.9.1 Overview 104
5.9.2 Process implementation 107
5.9.3 Operational testing 107
5.9.4 Software operation support 107
5.9.5 User support 107
5.10 Software maintenance process 108
5.10.1 Overview 108
5.10.2 Process implementation 108
5.10.3 Problem and modification analysis 115
5.10.4 Modification implementation 115
5.10.5 Conducting maintenance review 115
5.10.6 Software migration 115
5.10.7 Software retirement 115

6 Selected topics 116

6.1 Use Cases and Scenarios 116
6.1.1 Relation to the Standard 116
6.1.2 Introduction to use cases 116
6.1.3 Identification of use cases 117
6.1.4 Formalization of each use case 117
6.1.5 Definition and guidelines 119
6.2 Life cycle 120
6.2.1 Relation to the Standard 120
6.2.2 Introduction 120
6.2.3 Existing life-cycle models 122
6.2.4 Choosing a Software life-cycle 131
6.3 Model based Engineering 134
6.3.1 Relation to the Standard 134
6.3.2 Definition and guidelines 135
6.4 Testing Methods and Techniques 138
6.4.1 Relation to the Standard 138
6.4.2 Introduction 138
6.4.3 Definitions 138
6.4.4 Test objectives 138
6.4.5 Testing strategies and approaches 141
6.4.6 Real Time Testing 145
6.5 Autocode 146
6.5.1 Relation to the Standard 146

This document is a preview generated by EVS

6.5.2 Introduction 146
6.5.3 Subsystem and software relationship around autocode 147
6.5.4 From subsystem model to autocoded model 150

7 Real-time software 152

7.1 Relation to the Standard 152
7.2 Software technical budget and margin philosophy definition 152
7.2.1 Introduction 152
7.2.2 Load and real-time 153
7.2.3 Memory capacity 156
7.2.4 Numerical Accuracy 156
7.2.5 Interface timing budget 156
7.3 Technical budget and margins computation 157
7.3.1 Load and real-time 157
7.3.2 Memory margins 158
7.3.3 Numerical accuracy budget management 158
7.3.4 Interface timing budget management 159
7.4 Selection of a computational model for real-time software 159
7.4.1 Introduction 159
7.4.2 Recommended Terminology 160
7.4.3 Computational model 164
7.5 Schedulability analysis for real-time software 169
7.5.1 Overview 169
7.5.2 Schedulability Analysis 169

Annex A Documentation Requirement List 179

Annex B Generic Techniques 183

B.1 Formal Methods 183
B.2 Functional Decomposition and Structured Analysis 184
B.2.1 Introduction 184
B.2.2 Data flow 185
B.2.3 Control Flow 186
B.3 Object-Oriented Analysis (OOA) 187
B.4 Architecture and Design 187
B.5 Data Description Languages 188
B.6 State machine modelling languages 189
B.7 ITIL ® 190

Annex C (normative) "Software Maintenance Plan (SMP) – DRD" 192

C.1 DRD identification 192
C.1.1 Requirement identification and source document 192
C.1.2 Purpose and objective 192
C.2 Expected response 192
C.2.1 Scope and content 192

Figures

Figure 4-1: ECSS relations for discipline software 17
Figure 4-2: Role relationships 26
Figure 4-3 : Delivery of warranty and support between companies 36
Figure 4-4 : Closely Coupled Build and Service Support Contract 36
Figure 5-1: System database 48
Figure 5-2: Constraints between life cycles 50

This document is a preview generated by EVS

Figure 5-3: Software requirement reviews 53
Figure 5-4: Phasing between system reviews and flight software reviews 64
Figure 5-5: Phasing between ground segment reviews and ground software reviews 64
Figure 5-6 : Reuse in case of reference architecture 75
Figure 5-7 : Example ITIL processes 106
Figure 6-1: The autocoding process 149
Figure 7-1: Mitigation of theoretical worst case with operational scenarios. 154
Figure 7-2: An example of a complete task table with all timing figures 175
Figure 7-3 : Maintenance Cycle 191

Tables

Table 5-1: Possible review setup 59
Table 5-2: Example of review objectives and their applicability to each version 60
Table 6-1: Choosing a Software life-cycle 131
Table 6-2: Relation between the testing objectives and the testing strategies 145
Table 7-1: Schedulability Analysis Checklists 176

This document is a preview generated by EVS

European Foreword

This document (CEN/TR 17603-40:2022) has been prepared by Technical Committee CEN/CLC/JTC 5 “Space”, the secretariat of which is held by DIN.

It is highlighted that this technical report does not contain any requirement but only collection of data or descriptions and guidelines about how to organize and perform the work in support of EN 16603-40.

This Technical report (CEN/TR 17603-40:2022) originates from ECSS-E-HB-40A.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association.

This document has been developed to cover specifically space systems and has therefore precedence over any TR covering the same scope but with a wider domain of applicability (e.g.: aerospace).

This document is a preview generated by EVS

Introduction

The ECSS-E-ST-40C Standard defines the principles and requirements applicable to space software engineering. This ECSS-E-HB-40A handbook provides guidance on the use of the ECSS-E-ST-40C.

History of the ECSS-E-40

At the beginning was ESA PSS-05. It was a prescriptive list of requirements ordered all along a waterfall lifecycle. It was necessary to improve it because it was too prescriptive and not flexible enough to apply new technologies such as UML.

ECSS was created in the 90's, and ECSS-E-40A was published in 1999. It was derived from ISO 12207, which is a process model. A process model proposes a set of abstract processes, and the software developer defines its own lifecycle that enters and leaves and re-enters the various processes. The process model was very abstract, with sort of meta-processes that were "invoking" other sub-processes. The new ECSS-E-40A was not prescriptive and very flexible to any kind of lifecycle.

ECSS-E-40A was improved because it was too abstract and it was not clear what had to be done. ECSS-E-40B was worked out in order to downsize the abstraction. The invocation was simplified, some processes were grouped. ECSS-E-40B was sent for public review.

The public review recommended improving further the pragmatic aspects of the standard. Therefore another ECSS-E-40B version was produced where the process model was streamlined.

At a workshop in 2004 on the use of ECSS-E-40B, it was recognised that some of the requirements left room for interpretation, which in turn lead to many discussions in the project reviews (especially when they were overlooked during the Software Development Plan review). Therefore the version C of ECSS-E-ST-40 was produced to improve the usability of the standard, refining and streamlining the open requirements, and somehow coming closer to the ESA PSS-05 spirit.

This document is a preview generated by EVS

1 Scope

This document is a preview generated by EVS

This Handbook provides advice, interpretations, elaborations and software engineering best practices for the implementation of the requirements specified in ECSS-E-ST-40C. The handbook is intended to be applicable to both flight and ground. It has been produced to complement the ECSS-E-ST-40C Standard, in the area where space project experience has reported issues related to the applicability, the interpretation or the feasibility of the Standard. It should be read to clarify the spirit of the Standard, the intention of the authors or the industrial best practices when applying the Standard to a space project.

The Handbook is not a software engineering book addressing the technical description and respective merits of software engineering methods and tools.

ECSS-E-HB-40A covers, in particular, the following:

a. In section 4.1, the description of the context in which the software engineering standard operates, together with the explanation of the importance of following standards to get proper engineering.

b. In section 4.2, elaboration on key concepts that are essential to get compliance with the Standard, such as the roles, the software characteristics, the criticality, the tailoring and the contractual aspects.

c. In section 5, following the table of content of the ECSS-E-ST-40C Standard, discussion on the topics addressed in the Standard, with the view of addressing the issues that have been reported in projects about the interpretation, the application or the feasibility of the requirements. This includes in particular:

    1. Requirement engineering and the relationship between system and software
    2. Implementation of the requirements of ECSS-E-ST-40 when different life-cycle paradigms are applied (e.g., waterfall, incremental, evolutionary, agile) and at different levels of the Customer-Supplier Network
    3. Architecture, design and implementation, including real-time aspects
    4. Unit and integration testing considerations, testing coverage
    5. Validation and acceptance, including software validation facility and ISVV implementation
    6. Verification techniques, requirements and plan
    7. Software operation and maintenance considerations.

d. In section 6 and 7, more information about selected topics addressed in section 5 such as (in section 6) use cases, life cycle, model based engineering, testing, automatic code generation, and (in section 7) technical budget and margin, computational model and schedule analysis.

NOTE In order to improve the readability of the Handbook, the following logic has been selected for sections 5, 6, and 7:

- section 5 follows the table of content of ECSS-E-ST-40C at least up to level 3 and generally up to level 4. For each sub clause of ECSS-E-ST-40C:
  + either information is given fully in section 5,
  + or there is a pointer into section 6 or section 7
  + or the paragraph has been left intentionally empty for consistency with the ECSS-E-ST-40C table of content, in this case, only " –" is mentioned.
- section 6 expands selected parts of section 5 when:
  + either the volume of information was considered too large to stay in section 5,
  + or the topic is addressed in several places of section 5

  In any case, there is a pointer from section 5 to section 6, and section 6 mentions the various places in ECSS-E-ST-40C where the topic is addressed.
- section 7 follows the same principles as section 6, but gathers the topics related to margins and to real-time.

e. In Annex A, as a complement to the ECSS-E-ST-40C Annex A called Document Requirement List [DRL], the documents expected at the Technical Reviews such as SWRR, DDR, TRR and TRB.

f. In Annex B, software engineering techniques appropriate for the implementation of specific ECSS-E-ST-40C clauses and their selection criteria, covering most of the software lifecycle.

g. In Annex C, an example of the Document Requirement Definition of the Software Maintenance Plan.

This document is a preview generated by EVS

2 References

EN Reference	Reference in text	Title
EN 16601-00-01	ECSS-S-ST-00-01C	ECSS system - Glossary of terms
EN 16603-10	ECSS-E-ST-10C	Space engineering – System engineering general requirements
EN 16603-40	ECSS-E-ST-40C	Space engineering - Software
EN 16603-70-01	ECSS-E-ST-70-01C	Space engineering - On board control procedures
	ECSS-E-TM-40-07A	Space engineering - Simulation modelling platform
EN 16601-40	ECSS-M-ST-40C	Space project management - Configuration and information management
EN 16602-80	ECSS-Q-ST-80C	Space product assurance - Software product assurance
	Def Stan 00-54 (March 1999)	Requirements for Safety Related Electronic Hardware in Defence Equipment" Part No: 1: Requirements: Issue 1
	ESA ISVV Guide	ESA Guide for Independent Software Verification and Validation, Version 2.0, December 29, 2008
	IEEE 610.12-1990 (Jan 1990)	IEEE Standard Glossary of Software Engineering Terminology
	IEEE 754-1985 (Aug 1985)	IEEE Standard for Binary Floating-Point Arithmetic
	ISO/IEC 12207:2008	Systems and software engineering – Software life cycle processes
	NASA Study on Flight Software Complexity (May 2009)	Final Report. NASA Study on Flight Software Complexity. Commissioned by the NASA Office of Chief Engineer. Technical Excellence Program Adam West, Program Manager
	RNC-CNES-E-HB-70-501 (September 2008)	Space engineering monitoring and control specification guide, Version 2, September 16, 2008
	NASA Lessons Learned	http://llis.nasa.gov/llis/search/home.jsp
	NASA System Engineering Handbook	NASA/SP-2007-6105 Rev1 December 2007

This document is a preview generated by EVS

EN Reference	Reference in text	Title
	ESA PSS-05-0 Issue 2 (February 1991)	ESA software engineering standards Issue 2
	Flight Computer Initialisation Sequence	Space Avionics Open Interface initiative document: SAVOIR/12-007/FT http://savoir.estec.esa.int

This document is a preview generated by EVS