SLOVENSKI STANDARD
SIST EN 14160:2004

01-maj-2004

Space engineering - Software

Space engineering - Software

Raumfahrttechnik - Software

Ingéniérie spatiale - Logiciel

Ta slovenski standard je istoveten z: **EN 14160:2001**

ICS:

49.140	Vesoljski sistemi in operacije	Space systems and operations

SIST EN 14160:2004 **en**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 14160

December 2001

ICS 49.140

English version

Space engineering - Software

This European Standard was approved by CEN on 11 November 2001.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Ref. No. EN 14160:2001 E

Contents

page

Figures

Foreword

This European Standard has been prepared by CMC.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 2002, and conflicting national standards shall be withdrawn at the latest by June 2002.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

It is based on a previous version[1)] originally prepared by the ECSS Software Engineering Working Group, reviewed by the ECSS Technical Panel and approved by the ECSS Steering Board. The European Cooperation for Space Standardization (ECSS) is a cooperative effort of the European Space Agency, National Space Agencies and European industry associations for the purpose of developing and maintaining common standards.

This Standard is one of the series of space standards intended to be applied together for the management, engineering and product assurance in space projects and applications.

Requirements in this Standard are defined in terms of what shall be accomplished, rather than in terms of how to organize and perform the necessary work. This allows existing organizational structures and methods to be applied where they are effective, and for the structures and methods to evolve as necessary without rewriting the standards.

The formulation of this Standard takes into account the existing ISO 9000 family of documents.

Annex A is normative. Annexes B and C are informative.

This standard includes a Bibliography.

Introduction

This Standard reflects the specific methods used in space system developments, and the requirements for the software engineering process in this context. Together with the requirements found in the other branches of the European Space Standards, this Standard provides a coherent and complete framework for software engineering in a space project.

This Standard is intended to help customers in formulating their requirements and suppliers in preparing their response and implementing the work.

This Standard is not intended to replace textbook material on computer science or technology, and such material has been avoided in this Standard. The readers and users of this Standard are assumed to possess general knowledge of computer science.

[1)] ECSS-E-40A.

1 Scope

This European Standard defines the space software engineering process and its interfaces with the space project management standards (EN 13290) and space product assurance standards (EN 13291) and explains how they apply in the software engineering process.

This Standard is applicable to all the elements of a space system, including the space segment, the launch service segment and the ground segment.

This Standard covers all aspects of space software engineering including requirements definition, design, production, verification and validation, and transfer, operations and maintenance.

The scope of this Standard is the software developed as part of a space project, i.e. "Space system product software". It is not intended to cover software developments out of scope with the system of space standards. An example is the development of commercial software packages, where software is developed for a (large) volume market and not just for a single customer, and the main requirement analysis consists of market analysis, combined with a marketing strategy.

Other classes of software products not covered are: management information systems (e.g. finance, planning), technical information systems (e.g. CAD/CAM, analysis packages) and supporting software products for documentation systems, database systems, spread-sheets. These usually result from the procurement or adaptation of existing commercial products, and not part of the space system development. Such software products are, however, often part of a supporting infrastructure for space systems.

When viewed from the perspective of a specific project context, the requirements defined in this Standard should be tailored to match the genuine requirements of a particular profile and circumstances of a project.

NOTE Tailoring is the process by which individual requirements or specifications, standards and related documents are evaluated and made applicable to a specific project by selection, and in some exceptional cases, modification of existing or addition of new requirements.

2 Normative references

This European Standard incorporates by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this European Standard only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies (including amendments).

EN 13290-1:1999, *Space project management — General requirements — Part 1: Policy and principles.*

EN 13290-2:2001, *Space project management — General requirements — Part 2: Project breakdown structure.*

EN 13290-3:2001, *Space project management — General requirements — Part 3: Project organization.*

EN 13290-4:2001, *Space project management — General requirements — Part 4: Project phasing and planning.*

EN 13290-5:2001, *Space project management — General requirements — Part 5: Configuration management.*

EN 13290-6:2001, *Space project management - General requirements - Part 6: Information/Documentation management.*

EN 13290-7:2001, *Space project management — General requirements — Part 7: Cost and schedule management.*

EN 13291-2:—[2], *Space product assurance — General requirements — Part 2: Quality assurance.*

EN 13292:1999, *Space engineering — Policy and principles.*

EN 13701, *Space systems — Glossary of terms.*

ISO 8402:1994, *Quality management and quality assurance — Vocabulary.*

ISO/IEC 12207:1995, *Information technology — Software life cycle processes.*

ISO 16091:—[2], *Space project management — Integrated logistic support.*

ECSS-E-10, *Space engineering — System engineering.*

ECSS-E-40-01 [2], *Space engineering — Space segment software.*

ECSS-E-40-03 [2], *Space engineering — Ground segment software.*

ECSS-Q-80, *Space product assurance — Software product assurance.*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this European Standard, the terms and definitions given in EN 13701 and the following apply.

3.1.1
(top-level) architecture
highest level(s) structure of the components of a program or system, their interrelationships, and principles and guidelines governing their design and evolution over time

3.1.2
embedded software (deprecated)
NOTE 1 This term is sometimes used to denote that software, at varying levels, is part of a system.

NOTE 2 The term is also sometimes used to emphasize that the software in question has extensive hardware interface requirements or real-time requirements. The term is purely descriptive.

3.1.3
maintainer
organization that performs maintenance activities

[ISO/IEC 12207:1995]

3.1.4
margin philosophy
rationale for margins allocated to the performance parameters and computer resources of a development, and how these margins shall be managed during the execution of the project

3.1.5
migration
porting of a software product to a complete new environment

3.1.6
singular input
individual parameter stress testing

[2] To be published.

3.1.7
software component
general term for a part of a software system

NOTE Components are assembled or decomposed to form new components. In the production phase, components are implemented as modules, tasks or programs, any of which may be configuration items. This use of the term is more general than in ANSI/IEEE parlance, which defines a component as a "basic part of a system or program"; in this Standard components are not "basic" as they can be decomposed.

3.1.8
software item
see 3.1.11, software product

3.1.9
software intensive system
space software product where the dominant part of the constituents are software elements

NOTE In such systems, sub-systems consist mainly of software. For this type of system, the majority of interfaces are software-software interfaces.

3.1.10
software observability
property of a system for which observations of the output variables suffices to determine the initial values of status variables

3.1.11
software product
set of computer programs, procedures and possibly associated documentation and data

3.1.12
software unit
separately compilable piece of code

[ISO/IEC 12207:1995]

NOTE In this Standard no distinction is made between a software unit and a database; both are covered by the same requirements.

3.1.13
stress test
test that evaluates a system or software component at or beyond the limits of its specified requirements

3.1.14
validation
confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled

[ISO 8402:1994]

NOTE The validation process (for software): confirmation that the requirements baseline functions and performances are correctly and completely implemented in the final product.

3.1.15
verification
confirmation by examination and provision of objective evidence that specified requirements have been fulfilled

[ISO 8402:1994]

NOTE The verification process (for software): confirmation that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input.

3.2 Abbreviated terms

The following abbreviated terms are defined and used within this European Standard.

Abbreviation	Meaning
AR	acceptance review NOTE The term SW-AR may be used for clarity to denote ARs that solely involve software products.
CDR	critical design review NOTE The term SW-CDR may be used for clarity to denote CDRs that solely involve software products.
CJF	change justification file
COTS	commercial off-the-shelf software NOTE This term denotes finished software products, that are procured from third parties.
CPU	central processing unit
DDF	design definition file
DJF	design justification file
ICD	interface control document
IRB	interface requirements baseline
IRD	interface requirements document
ISV	independent software validation
ISVV	independent software verification and validation
MMI	man-machine interface
MOTS	modifiable off-the-shelf
MP	maintenance plan
OP	operational plan
ORR	operational requirements review
PDR	preliminary design review NOTE The term SW-PDR may be used for clarity to denote PDRs that solely involve software products.
QR	qualification review NOTE The term SW-QR may be used for clarity to denote QRs that solely involve software products.

RB requirements baseline

SDE software development environment

NOTE Software tools that are supporting the software engineering process.

SRR system requirements review

NOTE The term SW-SRR may be used for clarity to denote SRRs that solely involve software products.

SW software

TS technical specification

4 Space system software engineering

4.1 Introduction

This clause introduces the structure of this Standard and the framework of the space software engineering process that form its basis.

The context of space software engineering is the overall space system engineering process. This clause defines the general relationships between the software engineering processes and the general engineering processes of space systems.

The software engineering standard differs from the other engineering disciplines covered by the European Space Standards in one important aspect: software does not in itself produce heat, have mass or any other physical characteristics. The software engineering activity is a purely intellectual activity and a principle output of the activity is documentation. If the software code itself is considered as a specialized form of electronic documents, all visible outputs are in fact documentation.

It follows that this Standard focuses on requirements for the structure and content of the documentation produced.

Software can be used for the implementation of highly complex functions. The ability to deal with a high level of complexity in a flexible way makes software an essential and increasing part of space segment and ground segment products. In space systems, software engineering is found at all levels ranging from system level functions down to the firmware of a space system part.

Therefore the requirements engineering process, in which the software requirements and specifications are defined, has a special emphasis in this Standard. The software requirements engineering process consumes a large and often underestimated amount of effort in the development of software for space systems.

As a result of the complexity of the functional and performance requirements, special measures and emphasis are required for software verification and validation, especially for space segment software. The functions assigned to software can be critical to the space mission.

The maintenance of software for space systems also poses special problems, because they imply operational lifetimes that far exceed what is expected of general computer software products. For the space segment, this is further complicated by the fact that software in general is the only part of the space segment that undergoes major maintenance and repair, sometimes even re-design, after launch. In extreme cases, the space system mission itself is redesigned, implementing new space segment software after launch. Ground segment software is similarly characterized.

4.2 Organization of this Standard

This Standard is organized in two main parts:

— **General requirements.** These are the core normative requirements for any space system software engineering activity.

— **Special requirements.** These are additional requirements for specific application areas. These requirements are always applicable, but are only active in developments where the addressed disciplines or application areas occur. This separation serves to make the general requirements core compact and clear.

Software documentation summaries are included in annex A for information.

In the preparation of this Standard the ISO/IEC 12207:1995 standard has been used extensively, providing a common internationally recognized framework for the terminology and engineering process description. For completeness, a cross-reference between this Standard and ISO/IEC 12207:1995 is included in annex B.

4.3 Space system software engineering processes

4.3.1 General

The software engineering processes regulated by this Standard are based on the definitions and requirements given in the series of space project management standards (in particular EN 13290-3:2001, EN 13290-4:2001, EN 13290-5:2001 and EN 13290-6:2001), and the general engineering process requirements of EN 13292:1999. These requirements have been used to define the top-level software engineering processes. This general framework defines the processes (that are later treated in detail in the following subclauses) and the top-level interface between the software engineering processes and other space development processes.

The fundamental principle of this Standard is the customer-supplier relationship, assumed for all software developments. The organizational aspects of this are defined in EN 13290-3:2001. The customer is, in the general case, the procurer of two strongly associated products: the hardware and the software for a system, subsystem, set, equipment or assembly (see EN 13292:1999). The concept of the customer-supplier relationship is applied recursively, i.e. the customer may himself be a supplier to a higher level in the space system as shown in Figure 1. The software customer therefore has two important interfaces. The first interface is to his software and hardware suppliers and this includes the functional analysis required for the adequate allocation of function and performance requirements to his suppliers. The other where he is in his role as supplier at a higher level, where he shall ensure higher level system requirements are adequately taken into account.

The customer derives the functional and performance requirements for the hardware and software, based on system engineering principles and methods. The customer also controls the interface between the software and hardware. Software items are defined in the system breakdown at different levels. Nevertheless, it is important to manage the software-software interfaces irrespective of the level at which they occur. The customer's requirements are defined by this initializing process, and provide the starting point for the software engineering.

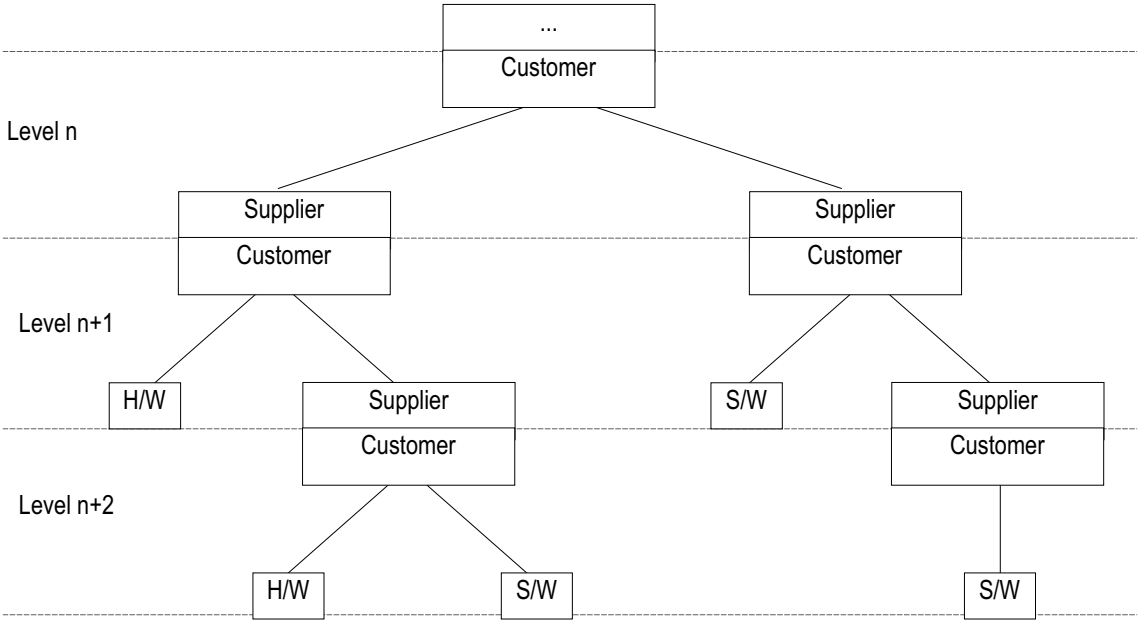

Figure 1 — The recursive customer - supplier model

Reviews are the main interaction points between the customer and supplier. The reviews relevant to the software engineering process are the SRR, PDR, CDR, QR and AR, as defined by EN 13290-4:2001. All reviews are applicable to software. The reviews occur at different levels in the customer-supplier hierarchy and are sequenced according to the overall system level planning. This Standard is designed to be applied at any level, without explicit assumptions of how these reviews shall be integrated with other reviews in the development of a space system. An overview is shown in Figure 2. The commonly designated mission phases (e.g. 0, A, B) are used for the overall mission phases, and play no direct role in the software engineering activities as such. This means that the software engineering processes, together with their reviews and attached milestones as defined in this Standard, are not to be scheduled as the higher-level system mission phases. They should be planned in relation to the immediate higher level development processes.

The notion of engineering processes is fundamental to this Standard, as the processes provide the means to describe the overall constraints and interfaces to the software engineering process at system level, and at the same time, provide the necessary freedom to the supplier to implement the individual activities implied by the processes. The freedom given to the supplier to implement the engineering processes is especially important for software engineering, because of the requirement to organize the work in accordance with a well defined software life cycle[3)]. There is a requirement to accommodate different types of software life cycles, both for reasons of efficient organization of the work and also for reasons related to competitiveness and choice of software engineering technology. Different software life cycle types can be accommodated within the requirements in this Standard. Figure 3 illustrates the constraints imposed. Figure 4 shows examples of variations within these constraints.

[3)] Software life cycles
There is an abundant technical literature on the different types of software life cycles. This Standard does not recommend any specific life cycle. Instead the requirements to be used in the choice of a life cycle for a given software project are to be defined, such that the software life cycle remains consistent with the overall organization and management of the space project organization and management, and that the life cycle outputs remain consistent with the requirements of the space standard EN 13290. Commonly known classes of software life cycles are:

—the Waterfall life cycle,
—the Incremental life cycle, and
—the Evolutionary life cycle.

This Standard has been verified for compatibility with these, but the choice is not limited to these three types.

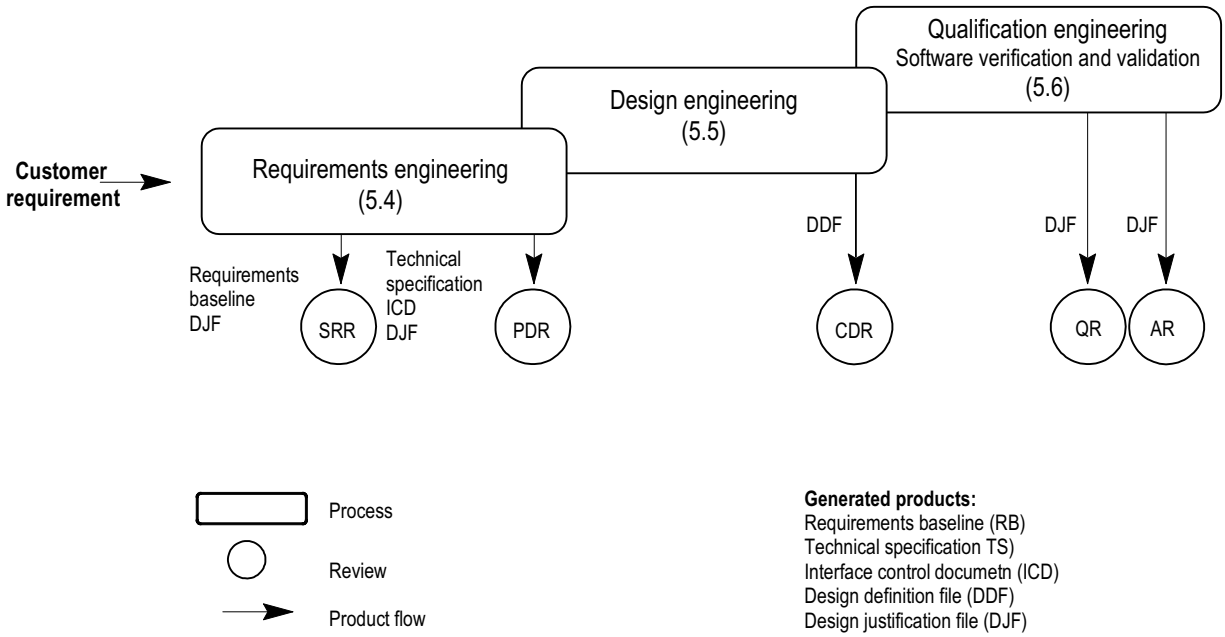

Figure 2 — Overview of the software development processes