INTERNATIONAL STANDARD

ISO/IEC/IEEE 32430

First edition
2021-10

Software engineering — Trial use standard for software non-functional sizing measurements

Ingénierie du logiciel — Norme expérimentale pour la quantification des caractéristiques non fonctionnelles des logiciels

Reference number
ISO/IEC/IEEE 32430:2021(E)

Documents
Document Sharing Hub
© IEEE 2019

Full document details: ISO/IEC/IEEE 32430-2021

COPYRIGHT PROTECTED DOCUMENT

© IEEE 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from IEEE at the address below.

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

© IEEE 2019 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO/IEC documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 32430 was prepared by the Systems and Software Engineering Standards Committee of the IEEE Computer Society (as IEEE Std 2430-2019) and drafted in accordance with its editorial rules. It was adopted, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© IEEE 2019 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO/IEC/IEEE 32430-2021

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

IEEE Std 2430™-2019

IEEE Trial-Use Standard for Software Non-Functional Sizing Measurements

Developed by the

Software & Systems Engineering Standards Committee (C/S2ESC)
of the
IEEE Computer Society

Approved 13 June 2019

IEEE-SA Standards Board

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Abstract: A method for the sizing of nonfunctional software requirements is defined in this standard. It complements ISO/IEC 20926:2009, which defines a method for the sizing of functional user requirements. Non-functional categories for data operations, interface design, technical environment, and architecture software are included in this standard.Steps to determine and calculate the non-functional size are also included. Handling requirements involving both functional and non-functional requirements are explained in this standard, which also covers how to apply non-functional sizing estimates in terms of cost, project duration and quality, and considerations of software performance in terms of productivity and quality. The combination of functional and non-functional size should correspond to the total size necessary to produce the software. The functional size and non-functional size are orthogonal, and both are needed when sizing the software. The complementarity of the functional and the non-functional sizes, to avoid overlaps or gaps between the two size methods, are described in this standard. Calculating the implementation work effort and duration of the non-functional requirements is outside the scope of this standard.

Keywords: IEEE 2430™, IFPUG, non-functional size measurements, non-functional requirements, SNAP

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2019 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 16 October 2019. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

W3C is trademarks or registered trademarks of the W3C®, (registered in numerous countries) World Wide Web Consortium. Marks of W3C are registered and held by its host institutions: Massachusetts Institute of Technology (MIT), European Research Consortium for Information and Mathematics (ERCIM), and Keio University, Japan.

PDF: ISBN 978-1-5044-5987-7 STD23763
Print: ISBN 978-1-5044-5988-4 STDPD23763

IEEE prohibits discrimination, harassment, and bullying.
For more information, visit https://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at https://standards.ieee.org/ipr/disclaimers.html.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. A current IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore at https://ieeexplore.ieee.org/ or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for IEEE standards can be accessed via https://standards.ieee.org/standard/index.html. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore: https://ieeexplore.ieee.org/browse/standards/collection/ieee/. Users are encouraged to periodically check for errata.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Trial-use standards

Publication of this trial-use standard for comment and criticism has been approved by the Institute of Electrical and Electronics Engineers, Inc. Trial-use standards are effective for 36 months from the date of publication. Comments for revision will be accepted for 24 months after publication. Suggestions for revision should be directed to the Secretary, IEEE-SA Standards Board, 445 Hoes Lane, Piscataway, NJ 08855, and should be received no later than 16 October 2021.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Participants

At the time this IEEE trial-use standard was completed, the P2430 Working Group had the following membership:

Talmon Ben-Cnaan, *Chair*

Bill Curtis
Victoria (Vicky) Hailey
Annette Reilly
Robert Schaaf
Roopali Thapar
Charley Tichenor
Altaz Valani
Kiran Yeole

The following members of the individual balloting committee voted on this trial-use standard. Balloters may have voted for approval, disapproval, or abstention.

Robert Aiello
Talmon Ben-Cnaan
Juris Borzovs
Pieter Botman
Demetrio Bucaneg Jr.
Paul Cardinal
Lawrence Catchpole
Jan de Liefde
Yaacov Fenster
Andrew Fieldsend
David Fuschi
Julian Gomez
Randall Groves
Victoria (Vicky) Hailey
Mark Henley
Werner Hoelzl
Noriyuki Ikeuchi
Atsushi Ito
Srinivasa Rao
Kanneganti
Piotr Karocki
David Leciston
Johnny Marques
LaMont McAliley
Rajesh Murthy
Nick S.A. Nikjoo
Mark Paulk
Annette Reilly
Saurabh Saxena
Robert Schaaf
Stephen Schwarm
Carl Singer
Kendall Southwick
Friedrich Stallinger
Thomas Starai
Walter Struppler
Marcy Stutzman
Sachin Thakur
Roopali Thapar
Charley Tichenor
John Vergis
Scott Willy
Steven Woodward
Jian Yu
Oren Yuen
Janusz Zalewski

When the IEEE-SA Standards Board approved this trial-use standard on 13 June 2019, it had the following membership:

Gary Hoffman, *Chair*
Ted Burse, *Vice Chair*
Jean-Philippe Faure, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Stephen D. Dukes
J.Travis Griffith
Guido Hiertz
Christel Hunter
Thomas Koshy
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
David J. Law
Joseph Levy
Howard Li
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Annette D. Reilly
Dorothy Stanley
Sha Wei
Phil Wennblom
Philip Winston
Howard Wolfman
Feng Wu
Jingyi Zhou

*Member Emeritus

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Introduction

This introduction is not part of IEEE Std 2430-2019, IEEE Trial-Use Standard for Software Non-Functional Sizing Measurements.

Having both software functional size and non-functional size provides significant information for the management of software product development. The functional size is quantifiable and represents a good measure of the functional project/application size. Providing a quantifiable measure for the non-functional requirements (NFR) allows organizations to build historical data repositories that can be referenced to assist in decision making for the technical and/or quality aspects of applications.

Non-functional sizing assists organization in multiple ways (see Annex A). It provides insight into projects and applications to assist in effort and cost estimating and in the analysis of quality and productivity. Used in conjunction with function-point analysis, non-functional sizing provides information that can identify additional software size, which may impact quality and productivity in a positive or negative way. Having this information enables software professionals to:

— Better plan, schedule, and estimate projects.

— Identify areas of process improvement.

— Assist in determining future technical strategies.

— Quantify the impacts of the current technical strategies.

— Improve quality: Analyzing non-functional size may assist in identifying implicit requirement and may assist in analyzing the components of the solution to meet the NFR by looking at the sizing attributes.

By learning the methodology as described in this standard and by performing the non-functional sizing together with functional sizing, the added time and effort to size the NFR is small.

Acknowledgments

The author thanks the International Electrotechnical Commission (IEC) for permission to reproduce information from its international standards. All such extracts are copyright of IEC Geneva, Switzerland. All rights reserved. Further information on the IEC is available from www.iec.ch. IEC has no responsibility for the placement and context in which the extracts and contents are reproduced by the author, nor is IEC in any way responsible for the other content or accuracy therein.

Portions of the Software Non-functional Assessment Process (SNAP), Assessment Practices Manual, Release 2.4, reprinted with permission from IFPUG, ©2017.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Contents

1. Overview ... 10
1.1 Scope ... 10
1.2 Purpose ... 10
1.3 Word usage ... 11

2. Normative references ... 11

3. Definitions, acronyms, and abbreviations ... 12
3.1 Definitions ... 12
3.2 Abbreviations ... 16

4. Introductory Information ... 17
4.1 Non-Functional Software Size Measurement (NFSSM) introduction ... 17
4.2 Software-intensive system and software product ... 18
4.3 Software domains ... 18
4.4 The relations between non-functional requirements (NFR) definition and functional user requirements (FUR) ... 18
4.5 Key features of the NFSSM ... 20
4.6 Future evolution of NFR ... 21
4.7 Objectives and benefits ... 22

5. Non-functional size: Categories and sub-categories ... 23
5.1 Category 1: Data operations ... 23
5.2 Category 2: Interface design ... 32
5.3 Category 3: Technical environment ... 41
5.4 Category 4: Architecture ... 47
5.5 Sizing code data ... 52

6. The sizing process ... 55
6.1 The timing of the non-functional sizing ... 56
6.2 Non-functional sizing and FSM ... 56
6.3 Steps to determine the non-functional size ... 57
6.4 Calculating the Non-functional size ... 63

7. Complementarity of the functional and the non-functional sizes ... 66
7.1 Requirements involving functional and non-functional requirements ... 67

8. Use of non-functional software sizing ... 77
8.1 Functional size and non-functional size ... 77
8.2 Project management ... 77
8.3 Performance management ... 81

Annex A (informative) NFSSM strengths ... 84

Annex B (informative) Bibliography ... 85

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

IEEE Trial-Use Standard for Software Non-Functional Sizing Measurements

1. Overview

1.1 Scope

This standard defines a method for the sizing of non-functional software requirements. It complements ISO/IEC 20926:2009, which defines a method for the sizing of functional user requirements (FUR).[1]

This standard also describes the complementarity of functional and non-functional sizes, so that sizing both functional and non-functional requirements (NFR) do not overlap. It also describes how non-functional size, together with functional size, should be used for measuring the performance of software projects, setting benchmarks, and estimating the cost and duration of software projects.

In general, there are many types of non-functional software requirements. Moreover, non-functional aspects evolve over time and may include additional aspects in the as technology advances. This standard does not intend to define the type of NFR for a given context. Users may choose ISO 25010:2011 or any other standard for the definition of NFR. It is assumed that users will size the NFR based on the definitions they use.

This standard covers a subset of non-functional types. It is expected that, with time, the state of the art can improve and that potential future versions of this standard can define an extended coverage. The ultimate goal is a version that, together with ISO/IEC 20926:2009, covers every aspect that may be required of any prospective piece of software, including aspects such as process and project directives that are hard or impossible to trace to the software's algorithm or data. The combination of functional and non-functional size would then correspond to the total size necessary to bring the software into existence.

Calculating the effort and duration of the implementation of the NFR is outside the scope of this standard.

1.2 Purpose

The purpose of this standard is to define the method of sizing of NFR and describe how to use this size, alongside with the functional size.

[1]Information on references can be found in Clause 2.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals is required to).[2,3]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (should equals is recommended that).

The word *may* is used to indicate a course of action permissible within the limits of the standard (may equals is permitted to).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (can equals is able to).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ISO/IEC 14143-1:2007, Information Technology—Software measurement—Functional size measurement.[4,5]

ISO/IEC 20926:2009, Software and systems engineering—Software measurement—IFPUG functional size measurement method 2009.

ISO/IEC 25010:2011, Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models.

ISO/IEC TR 14143-5:2004 Information technology—Software measurement—Functional size measurement—Part 5: Determination of functional domains for use with functional size measurement.

ISO/IEC/IEEE 24765:2017, Systems and software engineering—Vocabulary.[6,7]

[2]The use of the word *must* is deprecated and shall not be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.
[3]The use of *will* is deprecated and shall not be used when stating mandatory requirements, *will* is only used in statements of fact.
[4]ISO publications are available from the International Organization for Standardization (https://www.iso.org/) and the American National Standards Institute (https://www.ansi.org/).
[5]IEC publications are available from the International Electrotechnical Commission (https://www.iec.ch/). IEC publications are also available in the United States from the American National Standards Institute (http://www.ansi.org).
[6]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.
[7]IEEE publications are available from The Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (https://standards.ieee.org/).

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. For terms not defined in this clause, consult ISO/IEC/IEEE 24765:2017 or the *IEEE Standards Dictionary Online*. [8]

adaptive maintenance: Modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment.

NOTE 1 —Adaptive maintenance provides enhancements necessary to accommodate changes in the environment in which a software product must operate. These changes are those that must be made to keep pace with the changing environment. For example, the operating system might be upgraded, and some changes may be made to accommodate the new operating system.[9]

NOTE 2 —From ISO/IEC 14764:2006 ed 2.0, Copyright © 2006 IEC Geneva, Switzerland. www.iec.ch. [B8].[10]

application: A cohesive collection of automated procedures and data supporting a business objective; it consists of one or more components, modules, or subsystems. Examples: accounts payable, accounts receivable, payroll, procurement, shop production, assembly line control, air search radar, target tracking, weapons firing, flight line scheduling, and passenger reservations.

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

boundary (application boundary): A conceptual interface between the software under study and its users. The boundary (also referred to as application boundary) defines what is external to the application; it indicates the border between the software being measured and the user; it acts as a "membrane" through which data processed by transactions pass into and out of the application; it is dependent on the user's external business view of the application; it is independent of non-functional and/or implementation considerations. The boundary is identical when assessing the functional size and the non-functional size

NOTE —From ISO/IEC 14143-1:2007 ed 2.0, Copyright © 2007 IEC Geneva, Switzerland. www.iec.ch.

code data: A type of data entities, used for software sizing (in addition to Business Data and Reference Data). According to ISO/IEC 20926:2009 [IFPUG function point analysis (FPA)], "code data" usually exists to satisfy non-functional requirements (NFR) from the user (for quality requirements, physical implementation and/or a technical reason). Code data provides a list of valid values that a descriptive attribute may have. Typically, the attributes of the code data are Code, Description and/or other "standard" attributes describing the code; e.g., standard abbreviation, effective date, termination date, audit trail data, etc. The different categories of data are outlined below to assist in identification.

corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems [B8].

NOTE 1—The modification repairs the software product to satisfy requirements.

NOTE 2 —From ISO/IEC 14764:2006 ed 2.0, Copyright © 2006 IEC Geneva, Switzerland. www.iec.ch.

[8] *IEEE Standards Dictionary Online* is available at: h*ttp://dictionary.ieee.org*.
[9] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.
[10] The numbers in brackets correspond to those of the bibliography in Annex B.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

data element type (DET): A unique, non-repeated attribute, which can be in Business Data, Reference Data, or code data. The number of different types of Data Elements of all tables is the number of DETs. Instances of control information are also counted as a DET, such as the "Enter" key to facilitate an external input (EI).

database view: The result set of a stored query on the data, which the database users can query just as they would in a persistent database collection object. This pre-established query command is kept in the database dictionary. Unlike ordinary base tables in a relational database, it is a virtual table computed or collated dynamically from data in the database, when access to that view is requested. Changes applied to the data in a relevant underlying table are reflected in the data shown in subsequent invocations of the view.

elementary process (EP): The smallest unit of activity that is meaningful to the user(s).

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

external input (EI): An elementary process (EP) that processes data or control information sent from outside the boundary.

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

external inquiry (EQ): An elementary process (EP) that sends data or control information outside the boundary.

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

external interface file (EIF): user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is maintained within the boundary of another application.

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

external output (EO): An elementary process (EP) that sends data or control information outside the boundary and includes additional processing logic beyond that of an External Inquiry.

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

family of platforms: Software Platforms that are serving the same purpose. for example, operating systems; browsers. S*ee also:* **platforms**.

file type referenced (FTR): A data function read and/or maintained by a transactional function. A file type referenced includes: An internal logical file read or maintained by a transactional function, or an external interface file read by a transactional function. Code data is grouped into one additional FTR.

function points (FPs): A unit of measurement to express the amount of functionality an information system (as a product) provides to a user. In this standard, function points (FPs).

NOTE —From ISO/IEC 20926:2009 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch.

functional sizing measurements (FSM): The process of measuring functional size.

NOTE —From ISO/IEC 14143-1:2007 ed 2.0, Copyright © 2007 IEC Geneva, Switzerland. www.iec.ch.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

functional user requirements (FUR): A sub-set of the user requirements. Requirements that describe what the software shall do, in terms of tasks and services.

NOTE 1—FUR relate to, but are not limited to:

— Data transfer (for example: input customer data; send control signal)

— Data transformation (for example: calculate bank interest; derive average temperature)

— Data storage (for example: store customer order; record ambient temperature over time)

— Data retrieval (for example: list current employees; retrieve latest aircraft position)

NOTE 2—Examples of user requirements that are not FUR include, but are not limited to:

— Quality constraints (for example: usability, reliability, efficiency and portability)

— Organizational constraints (for example: locations for operation, target hardware and compliance to standards)

— Environmental constraints (for example: interoperability, security, privacy and safety)

— Implementation constraints (for example: development language, delivery schedule)

NOTE 3—From ISO/IEC 14143-1:2007 ed 2.0, Copyright © 2007 IEC Geneva, Switzerland. www.iec.ch.

internal logical file (ILF): User recognizable group of logically related data or control information maintained within the boundary of the application being measured.

logical file: A logical group of data as seen by the user. A logical file is made up of one or more data entities. A data function represents functionality provided to the user to meet internal and external data storage requirements. A data function is either an internal logical file or an external interface file. Grouping of data into logical files is the result of the combined effect of two grouping methods: Method a) is process driven, based on the user transactions in the application. Method b) is data-driven, based on the business rules.

multiple instance approach: The case where each method of delivery of same functionality is counted separately S*ee also:* **single instance approach**.

non-functional size: Size of the software derived by quantifying the non-functional requirements (NFR), defined by a set of rules.

non-functional requirements (NFR): Any requirement for a software-intensive system or for a software product, including how it should be developed and maintained, and how it should perform in operation, except any functional user requirement for the software. Non-functional requirements (NFR) concern: the software system or software product quality, the environment in which the software system or software product must be implemented and which it must serve, and the processes and technology to be used to develop and maintain the software system or software product and the technology to be used for their execution. *See also:* **functional user requirements**.

number of DETs: The sum of all DETs which are part of the input/output/query of the elementary process (EP), plus the data elements which are read or updated internally to the boundary.

partition: A set of software functions within an application boundary that share homogeneous criteria and values. A partition requires development effort, which may not be reflected when sizing the functional aspect of the project/product, using FPA. Partition is designed to improve non-functional perspective of the users,

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

such as maintainability, portability, or installability, (and is not based on technical or physical considerations). Partitions do not overlap.

perfective maintenance: modification of a software product after delivery to detect and correct latent faults in the software product before they are manifested as failures.

NOTE 1—Perfective maintenance provides enhancements for users, improvement of program documentation, and recoding to improve software performance, maintainability, or other software attributes (from ISO/IEC 14764:2006 ed 2.0, Copyright © 2006 IEC Geneva, Switzerland. www.iec.ch).

NOTE 2—Contrast with adaptive maintenance, corrective maintenance.

platform: **(A)** Type of computer or hardware device and/or associated operating system, or a virtual environment, on which software can be installed or run. **(B)** A collection of hardware and software components that are needed for a CASE tool to operate. **(C)** Combination of an operating system and hardware that makes up the operating environment in which a program runs.

NOTE 1—A platform is distinct from the unique instances of that platform, which are typically referred to as devices or instances.

NOTE 2—(A) is from ISO/IEC 19770-1:2017 ed 3.0, 3.9, Copyright © 2017 IEC Geneva, Switzerland. www.iec.ch. (B) is from IEEE Std 1175.2-2006 [B3]. (C) is from ISO/IEC 26513:2017 ed 2.0, 3.26 and ISO/IEC/IEEE 24765:2017, Copyright © 2017 IEC Geneva, Switzerland. www.iec.ch

precautionary principle: The principle that the introduction of a new product or process whose ultimate effects are unknown or disputed should be resisted until such risks are appropriately mitigated. The precautionary principle denotes a duty to prevent harm, when it is within our power to do so, even when all the evidence is not in. This principle has been codified in several international treaties and in international law.

primary intent: intent that is first in importance.

record element type (RET): User recognizable sub-group of data element types (DETs) within a data function, and code data group as defined in the "code data" paragraph.

single instance approach: The case where each method of delivery of same functionality is counted only once. *See also:* **multiple instance approach**.

Software Non-functional Assessment Process (SNAP) category: A group of components, processes or activities that are used in order to meet the non-functional requirement. Categories classify the non-functional requirements (NFR), and are generic enough to allow for future technologies. Categories are divided into sub-categories, so that the SNAP category groups the sub-categories based on the same level of operations and/or similar type of sizing activities.

Software Non-functional Assessment Process (SNAP) counting unit (SCU): A component or activity, in which non-functional complexity and size are measured. The SCU is identified according to the nature of each sub-category. It may contain both functional and non-functional characteristics; therefore, sizing an SCU should be performed for its functional sizing, using function point analysis (FPA), and for its non-functional sizing, using SNAP.

Software Non-functional Assessment Process (SNAP) points (SP): A unit of measurement to express the size of a non-functional requirement. The non-functional size of the software non-functional assessment process (SNAP) counting units (SCUs) is identified in a sub-category. After identifying all the SCUs that are provided to meet the non-functional requirements (NFR), the SNAP points (SPs) of all SCUs are added together to obtain the calculated SPs.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Software Non-functional Assessment Process (SNAP) Sub-category: A component, a process or an activity executed within the project, to meet a non-functional requirement. (A non-functional requirement may consume more than one sub-category).

Software product: A set of computer programs, procedures, and possibly associated documentation and data. [B7].

NOTE —From ISO/IEC/IEEE 12207:2017 ed 2.0, Copyright © 2009 IEC Geneva, Switzerland. www.iec.ch [B7].

software boundary (also referred to as "application boundary"): A conceptual interface between the software under study and its users. The boundary defines what is external to the application; it indicates the border between the software being measured and the user; it acts as a "membrane" through which data processed by transactions pass into and out of the application. Software boundary is dependent on the user's external business view of the application.

software project: The set of work activities, both technical and managerial, required to satisfy the terms and conditions of a project agreement. A software project has specific starting and ending dates, well-defined objectives and constraints, established responsibilities, a budget and a schedule. A software project may be self-contained or may be part of a larger project. In some cases, a software project may span only a portion of the software development cycle. In other cases, a software project may span many years and consist of numerous subprojects, each being a well-defined and self-contained software project.

software-intensive system: A system for which software is a major technical challenge and is perhaps the major factor that affects system schedule, cost, and risk. In the most general case, a software-intensive system is comprised of hardware, software, people, and manual procedures [B2].

NOTE —From IEEE Std 1062-2015 [B2].

user: any person or thing that communicates or interacts with the software at any time.

NOTE 1—Examples of "thing" include, but are not limited to, software applications, animals, sensors, or other hardware.

NOTE 2—From ISO/IEC 14143-1:2007 ed 2.0, Copyright © 2007 IEC Geneva, Switzerland. www.iec.ch.

Web Content Accessibility Guidelines (WCAG): part of a series of web accessibility guidelines published by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C®). They are a set of guidelines that specify how to make content accessible, primarily for people with disabilities—but also for all user agents, including highly limited devices, such as mobile phones.

NOTE—WCAG 2.0 was published in December 2008 by the the World Wide Web Consortium (W3C©[11]) and became an ISO standard, ISO/IEC 40500:2012 in October 2012. WCAG 2.1 [B15] was approved on 05 June 2018.

3.2 Abbreviations

DET	data element type
EDI	electronic data interchange
EI	external input
EO	external output

[11]W3C is trademarks or registered trademarks of the W3C®, (registered in numerous countries) World Wide Web Consortium. Marks of W3C are registered and held by its host institutions: Massachusetts Institute of Technology (MIT), European Research Consortium for Information and Mathematics (ERCIM), and Keio University, Japan.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

EP	elementary process
EQ	external inquiry
FSM	functional size measurement
FUR	functional user requirements
FP	function point
FPA	function point analysis
GUI	graphical user interface
NFR	non-functional requirements
NFSSM	non-functional software size measurement
RET	record element type
SCU	SNAP counting unit
SOA	service-oriented architecture
SNAP	Software Non-functional Assessment Process
SP	SNAP Point
WCAG	Web Content Accessibility Guidelines

4. Introductory Information

4.1 Non-Functional Software Size Measurement (NFSSM) introduction

In software engineering, size is an important attribute of software. For instance, the size of a piece of software-under-planning is used to calculate the resources (including time) necessary to develop the software, provided historical or benchmark productivity data is available. The size of a piece of software once finished is used for the normalization of certain software performance indicators (e.g., the number of defects found in the software) or for the calculation of the development organization's performance, including its productivity. This information is used to estimate the effort of future software projects.

The size of software can be measured in several ways, with different units of measure. For software that already exists, size can be measured by counting the source statements that form the software's algorithm and data. However, one major disadvantage of “source statement” as a unit of measure is that early in the development of software there are no statements yet to count. The substitute for estimating the number of source statements needed to satisfy the software requirements is quite imprecise due to a dependency on implementation specifics typically decided later in the project. This imprecision can be avoided by basing the sizing activity on present software requirements instead of future source statements.

Several sizing methods have been put forward that are based on software requirements. One of these methods is defined in ISO/IEC 20926:2009, which focuses on a subtype of the software requirements, the FUR, and assigns “function points” (FPs) (the unit of measure) to elements of FUR.

In addition to functional and non-functional size, other project-constraint attributes can be considered to provide perspective to an appropriate level of detail. This includes, for example, development methodology, maturity level of the organization, tools, experience, quality, and collaboration effectiveness between team members.

This standard complements ISO/IEC 20926:2009 by focusing on non-functional (software) requirements. Instead of FPs, this standard assigns SNAP points (SPs) to NFR.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

4.2 Software-intensive system and software product

Along with software, a software-intensive system may include hardware, data, and other components, such as documents and training. The NFR of the software-intensive system may be met by the software product or by the other components of the software-intensive system. For example, requirements for improving the response time of a system may be met by advanced hardware, by improving the efficiency of the software, or by both. NFR may be also met by the changes to the data maintained or used by the software product, and not by the software product.

Non-functional software size measures the software product, it does not measure the NFR that are met by non-software components of the software-intensive system.

4.3 Software domains

ISO 14143-1:2007 defines software functional domain as a "class of software based on the characteristics of Functional User Requirements." The term "software domains" replaces the phrase "software types" (which have been loosely defined) and is used to differentiate between categories of software products, so that performance indicators can be compared within similar types of software products.

ISO/IEC 14143-1:2007 requires that a Functional Size Measurement (FSM) Method shall describe the functional domain(s) to which it can be applied. ISO/IEC TR 14143-5:2004 defines the method to determine the functional domain.

A non-functional software size measurement (NFSSM) should use the same method of functional domains, as described in ISO/IEC 14143-1:2007 when comparing the performance of the non-functional aspects of software products or software projects.

4.4 The relations between non-functional requirements (NFR) definition and functional user requirements (FUR)

4.4.1 NFR

The boundary between functional requirements and NFR does not have a universally-agreed definition. ISO/IEC/IEEE 24765:2017 [B10] defines NFR as "a software requirement that describes not what the software will do, but how the software will do it." ISO/IEC 25010:2011 defines a product quality model and categorizes product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability). ISO/IEC 25010:2011 supersedes ISO/IEC 9126:1991, which identified six quality characteristics, which were intended to be exhaustive. As the software capabilities evolve, the definition of NFR and their boundary with functional requirements may change, and a new standard may supersede ISO/IEC 25010:2011 in the future.

This standard defines how to size the NFR of the software product irrespective of the way NFR are defined. Instead, a non-functional requirement is mapped to the category and sub-category of the software non-functional sizing method. SNAP sub-categories do not define or describe NFR; they classify how these requirements are met within the software product.

4.4.2 The relations between NFR and SNAP sub-categories

The implementation of a non-functional requirement, which can be defined using ISO/IEC 25010:2011 or any other standard, should be sized using SNAP sub-categories. SNAP measures the size of the implementation, and not the non-functional requirement.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

The categories and sub-categories and the method to size the sub-categories are defined in Clause 5. They are as follows::

a) Category 1—Data operations
   1) Sub-Category 1.1—Data entry validation
   2) Sub-Category 1.2—Logical and mathematical operations
   3) Sub-Category 1.3—Data formatting
   4) Sub-Category 1.4—Internal data movements
   5) Sub-Category 1.5—Delivering added value to users by data configuration
b) Category 2: Interface design
   1) Sub-Category 2.1—User interfaces
   2) Sub-Category 2.2—Help methods
   3) Sub-Category 2.3—Multiple input methods
   4) Sub-Category 2.4—Multiple output methods
c) Category 3: Technical environment
   1) Sub-Category 3.1—Multiple platforms
   2) Sub-Category 3.2—Database technology
   3) Sub-Category 3.3—Batch processes
d) Category 4: Architecture
   1) Sub-Category 4.1—Component-based software
   2) Sub-Category 4.2—Multiple input/output interfaces

A non-functional requirement may be implemented in the software product by using one or more SNAP sub-categories. Accordingly, a sub-category may cater for several non-functional characteristics, as defined by these standards, for example, a requirement to improve the way system recovers from a crash. Using ISO/IEC 25010:2011, this requirement falls under the area of reliability, and the sub characteristic is recoverability.

The software product is designed as follows:

— An algorithm is added to identify corrupted data in specific fields
— Time stamps are added to database records
— An algorithm is written to reconstruct corrupted data using uncorrupted record

The design involves the following SNAP sub-categories:

— Database technology (adding time stamp)
— Logical and mathematical operations

In this example, the “recoverability” type of non-functional requirement is mapped to two SNAP sub-categories, “Database Technology” and “Logical and Mathematical Operations.”

More examples are illustrated in Figure 1.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

ISO/IEC 25010:2011 Sub-characteristics (Partial list)		Snap Categories and sub-categories													
		Data Operation					Interface Design				Technical Environment			Archi-tecture	
		Data Entry Validations	Logical and Mathematical Operations	Data Formatting	Internal Data Movements	Delivering Added Value to Users by Data	User Interfaces	Help Methods	Multiple Input Methods	Multiple Output Methods	Multiple Platforms	Database Technology	Batch Processes	Component-based Software	Multiple Input/Output Interfaces
		1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	4.2
Portability	Replaceability														
	Installability														
	Adaptability														
Security	Accountability														
	Authenticity	Ex. 2													
Reliability	Fault Tolerance														
	Recoverability														
Usability	Accessibility														
	User Error Protection														
	Operability								Ex. 4						
	User Interface Aesthetics						Ex. 1								
	Learnability														
	Appropriateness Recognizability														
Performance Efficiency	Capacity														
	Resource Utilization														
	Time Behavior											Ex. 3			
Functional Suitability	Functional Completeness														
	Functional Appropriateness														
	Functional Correctness														

Figure 1—Mapping ISO/IEC 25010:2011 characteristics to SNAP sub-categories

Example 1: Improving the understandability and learnability by adding pop-up help menus (Sub-category 2.1) and rearranging the screens (Sub-category 2.2).

Example 2: Improving security by adding more validations to the authentication process, using Sub-category 1.1 "Data Entry Validations" and Sub-category 1.2 "Logical and Mathematical Operations."

Example 3: Improving performance by adding indices to the database and improving queries (Sub-category 3.2).

Example 4: Adding barcode reading as additional input method (Sub-category 2.3).

4.5 Key features of the NFSSM

A non-functional size assists organization in multiple ways. It provides insight into the delivery of software projects and maintenance of software applications to assist in effort estimating and in the analysis of key performance indicators, such as quality and productivity. Used in conjunction with FSM, the non-functional size provides information that can identify items impacting quality and productivity in a positive or negative way.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

Having this information enables software professionals to do the following:

a) Better plan and estimate projects

b) Communicate the impact of NFR on the project with users and customers

c) Compare project and analyze trends of improvement, and compare it to benchmarks

d) Identify areas of improvement

e) Assist in determining future non-functional strategies

f) Quantify the impacts of the current non-functional strategies

g) Provide specific data when communicating non-functional issues to various audiences

h) A method to help users determine the benefit of an application package to their organization by assessing portions or categories that specifically match their requirements

i) Determine the non-functional size of a purchased application package

Other key features include the following:

j) NFSSM is independent of the way NFR are defined.

k) This standard contains rules on how to use ISO/IEC 20926:2009 (IFPUG FSM) and NFSSM together, so that there should be no gaps and no overlaps between functional size and non-functional size. A requirement that contains both functional and non-functional aspects should be sized using ISO/IEC 20926:2009 for its functional aspects and SNAP for its non-functional aspects.

l) FSM and NFSSM together should provide a complete view of the size of the software product.[12]

m) This standard contains rules on how to use both functional size and non-functional size to calculate project performance (such as quality and productivity).

4.6 Future evolution of NFR

4.6.1 The challenge

As systems evolve, new variations on the types of requirements emerge from the enhanced capabilities of software. Machine learning, artificial intelligence, and the Internet of Things generate the need for new types of requirements, such as ethics, social responsibility, privacy, and sustainability. Similar to other NFR, such as security and accountability, these attributes shape the functional behaviors of the system and its components.

These requirements may fail to gain sufficient notice by software development teams until after a user finds a fault or a logic error or the system in use has had unintended consequences on user communities. Then everyone takes notice, with software development teams working to find corrective and preventive mitigations to address what may not fit the current definitions of functional requirements and the classification of NFR.

Similar to other requirements that are intended to shape system behaviors through the assurance of adherence to defined values, emerging requirements should reflect stakeholder values, some of which are often not explicitly defined or beyond the immediate access of the development environment (e.g., robotics as prosthetics). Software development teams need to understand the system's context of use, the risks of harm, then the application of the "precautionary principle" as the risk mitigation technique necessary to anticipate and mitigate against undesirable consequences such as harm to life of any kind and quality of life, the environment, a species, or a community.

[12]Future non-functional requirements may evolve in time. See also 1.1 for possible future enhancements of SNAP.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

4.6.2 Current classification of these NFR

In ISO/IEC 25010:2011terms, these emerging social requirements falls within the category of Freedom from risk. It defines Freedom from risk as:

> The quality in use of a system characterizes the impact that the product (system or software product) has on stakeholders. It is determined by the quality of the software, hardware and operating environment, and the characteristics of the users, tasks and social environment. All these factors contribute to the quality in use of the system.

Systems should be evaluated by the following criteria:

— How risk free is the use of the system?

— How risk free does updating the content of the system need to be?

— How risk free does making maintenance changes to the system or porting the system need to be?

— How risk free does using the output from the system need to be?

4.6.3 Sizing quality-in-use requirements

Meeting these user requirements consumes time, effort, and other resources that should be evaluated and considered when planning and managing a software project. Therefore, like other user requirements, they should be sized.

SNAP can be used to size quality-in-use requirements. For example, algorithms to analyze if a certain situation may impose a risk and assessing the probability and the impact of such a risk can be sized using Sub-category 1.2—Logical and Mathematical Operations. Storing data for this purpose may be sized using Sub-category 3.2—Database Technology. However, the current set of sub-categories may be enhanced in the future to cover more aspects of these new requirements.

4.7 Objectives and benefits

4.7.1 Objectives

SNAP measures software by quantifying the size of NFR. With this in mind, the objectives of SNAP are to:

a) Measure the non-functional size of the software that the user requests and receives

b) Demonstrate the full economic value of the application, including its functional aspects as well as the non-functional aspects (have the non-functional application size in addition to the functional application size, to demonstrate the full economic value).

c) Measure software development and maintenance projects based on the NFR

d) Size technical projects, to which FPA is not applicable

In addition to meeting the above objectives, the process of sizing NFR should be:

e) Simple enough to minimize the overhead of the measurement process.

f) A consistent measure among various projects and organizations. SNAP allows determining (by counting each of the four categories, from each one of the sub-characteristics) the possibility of sizing and, therefore, better estimate a project with/without FPs, according to the set of user requirements received for a project.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021

4.7.2 Benefits

Non-functional sizing assists organizations in multiple ways. It provides insight into the delivery of projects and maintenance of applications to assist in estimating and in the analysis of quality and productivity. Used in conjunction with FSM, the non-functional sizing provides information that can identify items impacting quality and productivity in a positive or negative way.

Having this information enables software professionals to:

a) Better plan and estimate projects

b) Identify areas of process improvement

c) Assist in determining future non-functional strategies

d) Quantify the impacts of the current non-functional strategies

e) Provide specific data when communicating non-functional issues to various audiences

Organizations can apply SNAP as:

— A method to measure the non-functional size of a software product to support quality and productivity analysis

— A method to estimate cost and resources required for software development and maintenance

— A method to measure cost reduction for software development and maintenance, in addition to FPA

— A normalization factor for software comparison

— A method to determine the non-functional size of a purchased application package by sizing all the portions and categories included in the package

— A method to help users determine the benefit of an application package to their organization by sizing portions or categories that specifically match their requirements

NOTE 1—"Function Points + SPs" are not equal to the overall product size:

As of the date of this publication, the size of a software application is considered to have two distinct parts: the size of the FURs and the size of the NFRs. For example, if an application's functional size is 700 FPs and non-functional size is 200 SPs, then the entire size could be stated as "700 FPs, and 200 SPs." The two sizes do not sum up to "900 points."[13]

The ISO/IEC 20926:2009 FSM (IFPUG functional sizing methodology) does not change when measuring the NFR using SNAP.

NOTE 2—A project may have zero Function Points and a nonzero number of SPs, or 0 SPs and a nonzero number of Function Points, or any combination of Function Points and SPs

5. Non-functional size: Categories and sub-categories

5.1 Category 1: Data operations

The data operations category relates to how data is processed within the SNAP counting unit (SCU) to meet the NFR in the application.

[13]See 8.2.4 for information on how "equivalent size" is calculated.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Full document details: ISO/IEC/IEEE 32430-2021