

Australian/New Zealand Standard™

**Systems and software engineering—
Systems and software Quality
Requirements and Evaluation
(SQuaRE)—Evaluation process**



AS/NZS ISO/IEC 25040:2013

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT-015, Software and Systems Engineering. It was approved on behalf of the Council of Standards Australia on 6 May 2013 and on behalf of the Council of Standards New Zealand on 29 April 2013.
This Standard was published on 24 May 2013.

The following are represented on Committee IT-015:

Australian Computer Society
Australian Society for Technical Communication, NSW
Charles Sturt University
Department of Defence, Australia
Griffith University
Quantitative Enterprise Software Performance
La Trobe University
National Association of Testing Authorities Australia
National ICT Australia
New Zealand Organisation for Quality
NSW Business Chamber
Systems Engineering Society of Australia
University of Auckland
University of Technology, Sydney
Vendor Interests, New Zealand

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about joint Australian/New Zealand Standards can be found by visiting the Standards Web Shop at www.saiglobal.com.au or Standards New Zealand web site at www.standards.co.nz and looking up the relevant Standard in the on-line catalogue.

For more frequent listings or notification of revisions, amendments and withdrawals, Standards Australia and Standards New Zealand offer a number of update options. For information about these services, users should contact their respective national Standards organization.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Please address your comments to the Chief Executive of either Standards Australia or Standards New Zealand at the address shown on the back cover.

This Standard was issued in draft form for comment as DR AS/NZS ISO/IEC 25040.

Australian/New Zealand Standard™

Systems and software engineering— Systems and software Quality Requirements and Evaluation (SQuaRE)—Evaluation process

First published as AS/NZS ISO/IEC 25040:2013.

COPYRIGHT

© Standards Australia Limited/Standards New Zealand

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher, unless otherwise permitted under the Copyright Act 1968 (Australia) or the Copyright Act 1994 (New Zealand).

Jointly published by SAI Global Limited under licence from Standards Australia Limited, GPO Box 476, Sydney, NSW 2001 and by Standards New Zealand, Private Bag 2439, Wellington 6140.

ISBN 978 1 74342 466 7

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software and Systems Engineering.

The objective of this Standard is to provide requirements and recommendations for the evaluation of software product quality and clarifies the general concepts. It provides a process description for evaluating software product quality and states the requirements for the application of this process.

This Standard is identical with, and has been reproduced from ISO/IEC 25040:2011, *Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Evaluation process*.

As this Standard is reproduced from an International Standard, the following applies:

- (a) In the source text ‘this International Standard’ should read ‘this Australian/New Zealand Standard’.
- (b) A full point substitutes for a comma when referring to a decimal marker.

The term ‘informative’ has been used in this Standard to define the application of the annex to which it applies. An ‘informative’ annex is only for information and guidance.

CONTENTS

1	Scope	1
2	Conformance	1
3	Normative references	1
4	Terms and definitions	1
5	Software product quality evaluation reference model	10
5.1	Reference model - general	10
5.2	Reference model - evaluation processes.....	11
5.3	Roles	13
5.4	Quality in the life cycle.....	13
5.5	Support for the evaluation.....	13
6	Software product quality evaluation process	14
6.1	General requirements	14
6.2	Documentation	14
6.3	Establish the evaluation requirements	15
6.4	Specify the evaluation.....	17
6.5	Design the evaluation	19
6.6	Execute the evaluation.....	20
6.7	Conclude the evaluation	21
	Annex A (informative) Evaluation levels.....	25
	Annex B (informative) Evaluation methods.....	29
	Annex C (informative) Example of Cost-Effectiveness Ranking of Evaluation Methods	34
	Annex D (informative) Relationships between software product quality evaluation process reference model and software and system life cycle processes	35
	Annex E (informative) Evaluation report template	37
	Annex F (informative) Diagrams of inputs, outcomes, constraints and resources for activities	39
	Bibliography.....	44

INTRODUCTION

- ISO/IEC 2502n - *Quality Measurement Division*,
- ISO/IEC 2503n - *Quality Requirements Division*, and
- ISO/IEC 2504n - *Quality Evaluation Division*.

Annex A provides an explanation on levels of evaluation, aspects to be considered when defining evaluation levels and suggestions on evaluation techniques to be applied according to the rank of evaluation level.

Annex B provides examples of evaluation methods.

Annex C provides a table showing relationships between some evaluation methods, possible cost rank and effectiveness per software quality characteristics.

Annex D provides relationships between the software product quality evaluation process reference model and the software and system life cycle processes.

Annex E provides an example template of an evaluation report.

Annex F provides the diagrams of inputs, outcomes, constraints and resources for each evaluation activity.

Figure 1 illustrates the organization of the SQuaRE series representing families of standards, further called Divisions.

Quality Requirements Division 2503n	Quality Model Division 2501n	Quality Evaluation Division 2504n
	Quality Management Division 2500n	
	Quality Measurement Division 2502n	
Extension Division 25050 - 25099		

Figure 1 — Organization of the SQuaRE series of International Standards

The Divisions within the SQuaRE model are as follows.

- **ISO/IEC 2500n - Quality Management Division.** The International Standards that form this division define all common models, terms and definitions referred to by all other standards from the SQuaRE series. Referring paths (guidance through SQuaRE documents) and high-level practical suggestions in applying proper standards to specific application cases offer help to all types of users. The division also provides requirements and guidance for a supporting function which is responsible for the management of software product requirements, specification and evaluation.

- **ISO/IEC 2501n - Quality Model Division.** The International Standard that forms this division presents detailed quality models for software, quality in use and data. Practical guidance on the use of the quality model is also provided.
- **ISO/IEC 2502n - Quality Measurement Division.** The International Standards that form this division include a software product quality measurement reference model, mathematical definitions of quality measures, and practical guidance for their application. This division presents internal measures of software quality, external measures of software quality and quality in use measures. Quality measure elements (QME) forming foundations for the latter measures are defined and presented.
- **ISO/IEC 2503n - Quality Requirements Division.** The International Standard that forms this division helps specifying quality requirements. These quality requirements can be used in the process of quality requirements, elicitation for a software product to be developed or as inputs for an evaluation process. The requirements definition process is mapped to technical processes defined in ISO/IEC 15288.
- **ISO/IEC 2504n - Quality Evaluation Division.** The International Standards that form this division provide requirements, recommendations and guidelines for software product evaluation, whether performed by independent evaluators, acquirers or developers. The support for documenting a measure as an evaluation module is also presented.

ISO/IEC 25050 to ISO/IEC 25099 are reserved to be used for SQuaRE extension International Standards and/or Technical Reports.

This International Standard is part of the 2504n series on quality evaluation division that currently consists of the following International Standards:

- **ISO/IEC 25040 - Evaluation process:** contains general requirements for specification and evaluation of software quality and clarifies the general concepts. Provides a process description for evaluating quality of software product and states the requirements for the application of this process. The evaluation process is the basis for software product quality evaluation for different purposes and approaches. Therefore, the process can be used for the evaluation of quality in use, external measure of software quality and internal measure of software quality and can be applied to evaluate the quality of pre-developed software or custom software during its development process. The software product quality evaluation can be conducted, for instance, by an acquirer, a developer organization, or an independent evaluator.
- **ISO/IEC 25041 - Evaluation guides for developers, acquirers and evaluators:** contains specific requirements and recommendations for developers, acquirers and evaluators.
- **ISO/IEC 25042 - Evaluation modules:** defines the structure and content of the documentation to be used to describe an evaluation module. These evaluation modules contain the specification of the quality model (i.e. characteristics, subcharacteristics and corresponding internal, external or quality in use measures), the associated data and information about the planned application of the model and the information about its actual application. Appropriate evaluation modules are selected for each evaluation. In some cases it may be necessary to develop new evaluation modules. Guidance for developing new evaluation modules is found in ISO/IEC 25042. This International Standard can also be used by organizations producing new evaluation modules.
- **ISO/IEC 25045 - Evaluation module for recoverability:** provides the specification to evaluate the subcharacteristic of recoverability defined under the characteristic of reliability of the quality model. It determines the external measures of software quality of resiliency and autonomic recovery index when the information system composed of one or more software products' execution transactions is subjected to a series of disturbances. A disturbance could be an operational fault (e.g. an abrupt shutdown of an OS process that brings down a system) or an event (e.g. a significant increase of users to the system).

ISO/IEC 25040 is a revised version and replaces the current ISO/IEC 14598-1.

NOTES

AUSTRALIAN/NEW ZEALAND STANDARD

Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Evaluation process**1 Scope**

This International Standard contains requirements and recommendations for the evaluation of software product quality and clarifies the general concepts. It provides a process description for evaluating software product quality and states the requirements for the application of this process. The evaluation process can be used for different purposes and approaches. The process can be used for the evaluation of the quality of pre-developed software, commercial-off-the-shelf software or custom software and can be used during or after the development process.

This International Standard establishes the relationship of the evaluation reference model to the SQuaRE documents as well as shows how each SQuaRE document should be used during the activities of the evaluation process.

It is intended for those responsible for software product evaluation and is appropriate for developers, acquirers and independent evaluators of software products. These three different approaches are detailed in ISO/IEC 14598-3, ISO/IEC 14598-4, and ISO/IEC 14598-5.

It is not intended for evaluation of other aspects of software products (such as functional requirements, process requirements, business requirements, etc.).

2 Conformance

Evaluation of software product quality conforms to this International Standard if it complies with the requirements of Clause 6.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

There are no normative references in this document.

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1**acquirer**

individual or organization that acquires or procures a system, software product or software service from a supplier

NOTE Adapted from ISO/IEC 12207:2008.

4.2**analysis model**

algorithm or calculation combining one or more base and/or derived measures with associated decision criteria

4.3**attribute**

inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

NOTE 1 Adapted from ISO/IEC 15939:2007.

NOTE 2 ISO 9000 distinguishes two types of attributes: a permanent characteristic existing inherently in something; and an assigned characteristic of a product, process or system (e.g. the price of a product, the owner of a product). The assigned characteristic is not an inherent quality characteristic of that product, process or system.

4.4**attribute for quality measure**

attribute that relates to software product itself, to the use of the software product or to its development process

NOTE Attributes for quality measure are used in order to obtain quality measure elements.

4.5**base measure**

measure defined in terms of an attribute and the method for quantifying it

NOTE 1 A base measure is functionally independent of other measures.

NOTE 2 Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

[ISO/IEC 15939:2007]

4.6**commercial-off-the-shelf software product**

software product defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users

4.7**context of use**

users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used

[ISO 9241-11:1998]

4.8**custom software**

software product developed for a specific application from a user requirements specification

4.9**data**

collection of values assigned to base measures, derived measures and/or indicators

[ISO/IEC 15939:2007]

4.10**decision criteria**

thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result.

[ISO/IEC 15939:2007]

4.11**derived measure**

measure that is defined as a function of two or more values of base measures

[ISO/IEC 15939:2007]

NOTE 1 Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

NOTE 2 A transformation of a base measure using a mathematical function can also be considered as a derived measure.

4.12**developer**

individual or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the software life-cycle process

NOTE Adapted from the definition in ISO/IEC 12207:2008.

4.13**division of standards**

division forms a family of standards serving complementary purposes

4.14**end user**

individual person who ultimately benefits from the outcomes of the system

NOTE The end user can be a regular operator of the software product or a casual user such as a member of the public.

4.15**entity**

object that is to be characterized by measuring its attributes

EXAMPLE An object can be a process, product, project, or resource.

[ISO/IEC 15939:2007]

4.16**evaluation**

systematic determination of the extent to which an entity meets its specified criteria

[ISO/IEC 12207:2008]

4.17**evaluation coverage**

degree to which the evaluation covers the specified software product quality requirements

4.18**evaluation level**

rigour to be applied during the evaluation that defines the depth or thoroughness of the evaluation in terms of evaluation techniques to be applied and evaluation results to be achieved

4.19**evaluation method**

procedure describing actions to be performed by the evaluator in order to obtain results for the specified measurement applied to the specified product components or on the product as a whole

4.20**evaluation module**

package of evaluation technology for measuring software quality characteristics, subcharacteristics or attributes

NOTE The package includes evaluation methods and techniques, input to be evaluated, data to be measured and collected and supporting procedures and tools.

4.21**evaluation records**

documented objective evidence of all activities performed and of all results achieved within the evaluation process

4.22**evaluation requester**

person or organization that requests an evaluation

4.23**evaluation tool**

instrument that can be used during evaluation to collect data, to perform interpretation of data or to automate part of the evaluation

NOTE Examples of such tools are source code analysers to compute code metrics, CASE tools to produce formalized models, test environments to run the executable programs, checklists to collect inspection data or spreadsheets to produce syntheses of measures.

4.24**evaluation stringency**

degree required for the software product quality characteristics and subcharacteristics to fulfil the expected use criticality of the software product

4.25**evaluator**

individual or organization that performs an evaluation

4.26**failure**

termination of the ability of a product to perform a required function or its inability to perform within previously specified limits

NOTE Adapted from IEEE 610.12-1990.

4.27**fault**

incorrect step, process or data definition in a computer program

[IEEE 610.12-1990]

4.28**functional requirement**

requirement that specifies a function that a system or system component must be able to perform

[IEEE 610.12-1990]

NOTE The software quality characteristic “functionality” can be used to specify or evaluate the suitability, accuracy, interoperability, security and compliance of a function.

4.29**implied needs**

needs that may not have been stated but are actual needs

NOTE Some implied needs only become evident when the software product is used in particular conditions.

EXAMPLE Implied needs include: needs not stated but implied by other stated needs and needs not stated because they are considered to be evident or obvious.

4.30**independent evaluator**

individual or organization that performs an evaluation independently from developers and acquirers

NOTE The individual or organization acting as developer or acquirer for the target system to be evaluated cannot become the independent evaluator for the system. The independent evaluator can be an organization. Independent evaluators can belong to the same organization as the developers as long as they are independent from developers and acquirers.

4.31**indicator**

measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs

[ISO/IEC 15939:2007]

NOTE In ISO/IEC 14598-1 this definition was “a measure that can be used to estimate or predict another measure”.

4.32**information need**

insight necessary to manage objectives, goals, risks, and problems

[ISO/IEC 15939:2007]

4.33**information product**

one or more indicators and their associated interpretations that address an information need

EXAMPLE A comparison of a measured defect rate to planned defect rate along with an assessment of whether or not the difference indicates a problem.

[ISO/IEC 15939:2007]

4.34**information system needs**

needs that can be specified as quality requirements by external measures and sometimes by internal measures

4.35**intermediate software product**

product of the software development process that is used as input to another stage of the software development process

EXAMPLE Intermediate software products can include static and dynamic models, other documents and source code.

4.36**intermediate software product needs**

needs that can be specified as quality requirements by internal measures

4.37**maintainer**

individual or organization that performs maintenance activities

NOTE Adapted from ISO/IEC 12207:2008.

4.38**measure**, noun

variable to which a value is assigned as the result of measurement

NOTE 1 The term “measures” is used to refer collectively to base measures, derived measures, and indicators.

NOTE 2 Adapted from ISO/IEC 14598-1:1999.

4.39**measure**, verb

make a measurement

[ISO/IEC 14598-1:1999]

4.40**measurement**

set of operations having the object of determining a value of a measure

[ISO/IEC 15939:2007]

NOTE 1 Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

NOTE 2 Measurement can include assigning a qualitative category such as the language of a source program (ADA, C, COBOL, etc.).

4.41**measurement function**

algorithm or calculation performed to combine two or more base measures

[ISO/IEC 15939:2007]

4.42**measurement method**

logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale

[ISO/IEC 15939:2007]

NOTE Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

4.43**measurement procedure**

set of operations, described specifically, used in the performance of a particular measurement according to a given method

[ISO/IEC 15939:2007]

NOTE Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

4.44**measurement process**

process for establishing, planning, performing and evaluating software measurement within an overall project or organizational measurement structure

NOTE Adapted from ISO/IEC 15939:2007.

4.45**observation**

instance of applying a measurement procedure to produce a value for a base measure

[ISO/IEC 15939:2007]

4.46**operator**

individual or organization that operates the system

NOTE Adapted from ISO/IEC 12207:2008.

4.47**process**

system of activities, which uses resources to transform inputs into outputs

NOTE Adapted from ISO 9000:2005.

4.48**quality in use (measure)**

the extent to which a product used by specific users meets the users' needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

4.49**quality measure elements**

measure, which is either a base measure or a derived measure, that is used for constructing software quality measures

NOTE The software quality characteristic or subcharacteristic of the entity is derived afterwards by calculating a software quality measure.

4.50**quality model**

defined set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality

4.51**rating**

action of mapping the measured value to the appropriate rating level and used to determine the rating level associated with the software product for a specific quality characteristic

4.52**rating level**

scale point on an ordinal scale, which is used to categorize a measurement scale

NOTE 1 The rating level enables the software product to be classified (rated) in accordance with the stated or implied needs.

NOTE 2 Appropriate rating levels may be associated with the different views of quality, i.e. users', managers' or developers'.

4.53**requirements**

expression of a perceived need that something be accomplished or realized

NOTE The requirements can be specified as part of a contract, or specified by the development organization, as when a product is developed for unspecified users, such as consumer software, or the requirements can be more general, as when a user evaluates products for comparison and selection purpose.

**4.54
scale**

ordered set of values, continuous or discrete, or a set of categories to which the attribute is mapped

[ISO/IEC 15939:2007]

NOTE Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

EXAMPLE Types of scales are: a nominal scale which corresponds to a set of categories; an ordinal scale which corresponds to an ordered set of scale points; an interval scale which corresponds to an ordered scale with equidistant scale points; and a ratio scale which not only has equidistant scale points, but also possesses an absolute zero. Measures using nominal or ordinal scales produce qualitative data, and measures using interval and ratio scales produce quantitative data.

**4.55
software product**

set of computer programs, procedures, and possibly associated documentation and data

[ISO/IEC 12207:2008]

NOTE 1 Products include intermediate products, and products intended for users such as developers and maintainers.

NOTE 2 In SQuaRE standards, software quality has the same meaning as software product quality.

**4.56
software product evaluation**

technical operation that consists of producing an assessment of one or more characteristics of a software product according to a specified procedure

**4.57
software quality**

degree to which the software product satisfies stated and implied needs when used under specified conditions

NOTE 1 This definition differs from the ISO 9000:2005 quality definition because the software quality definition refers to the satisfaction of stated and implied needs, while the ISO 9000 quality definition refers to the satisfaction of requirements.

NOTE 2 Adapted from the ISO/IEC 25000:2005 definition, rephrased as "degree to which".

**4.58
software quality characteristic**

category of software quality attributes that bears on software quality

NOTE Software quality characteristics can be refined into multiple levels of subcharacteristics and finally into software quality attributes.

**4.59
software quality evaluation**

systematic examination of the extent to which a software product is capable of satisfying stated and implied needs

**4.60
software quality in use**

capability of the software product to enable specific users to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

NOTE Before the product is released, quality in use can be specified and measured in a test environment for the intended users, goals and contexts of use. Once in use, it can be measured for actual users, goals and contexts of use. The actual needs of users might not be the same as those anticipated in requirements, so actual quality in use might be different from quality in use measured earlier in a test environment.

4.61**software quality measure**

internal measure of software quality, external measure of software quality or software quality in use measure

NOTE Internal measure of software quality, external measure of software quality or software quality in use measure are described in the quality model in ISO/IEC 25010.

4.62**stakeholder**

a party having a right, share or claim in a system or in its possession of characteristics that meet that party's needs and expectations

NOTE 1 Adapted from ISO/IEC 15288:2008.

NOTE 2 Stakeholders include, but are not limited to, end users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, supplier organizations and regulatory bodies.

4.63**supplier**

individual or organization that enters into a contract with the acquirer for the supply of a system, software product or software service under the terms of the contract

NOTE Adapted from ISO/IEC 12207:2008.

4.64**system**

a combination of interacting elements organized to achieve one or more stated purposes

NOTE 1 A system can be considered as a product or as the services it provides.

NOTE 2 In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word system can be substituted simply by a context dependent synonym, e.g. aircraft, though this can then obscure a system principles' perspective.

[ISO/IEC 15288:2008]

4.65**target of process**

software product or task executed by software product to which a measurement or evaluation process is applied

4.66**unit of measurement**

particular quantity, defined and adopted by convention, with which a other quantities of the same kind are compared in order to express their magnitude relative to that quantity

[ISO/IEC 15939:2007]

NOTE Adapted from the *International Vocabulary of Basic and General Terms in Metrology*, 1993.

4.67**user**

individual or group that interacts with a system or benefits from a system during its utilization

NOTE Users can include operators, recipients of the results of the software, or developers or maintainers of software.

4.68**validation**

confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled

NOTE 1 “Validated” is used to designate the corresponding status.

[ISO 9000:2005]

NOTE 2 In design and development, validation concerns the process of examining a product to determine conformity with user needs.

NOTE 3 Validation is normally performed on the final product under defined operating conditions. It can be necessary in earlier stages.

NOTE 4 Multiple validations can be carried out if there are different intended uses.

4.69

value

number or category assigned to an attribute of an entity by making a measurement

4.70

verification

confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

NOTE 1 “Verified” is used to designate the corresponding status.

[ISO 9000:2005]

NOTE 2 In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.

5 Software product quality evaluation reference model

5.1 Reference model - general

Software product quality evaluation process includes the inputs and outcomes, constraints and resources for the software product quality evaluation process, as shown in Figure 2.

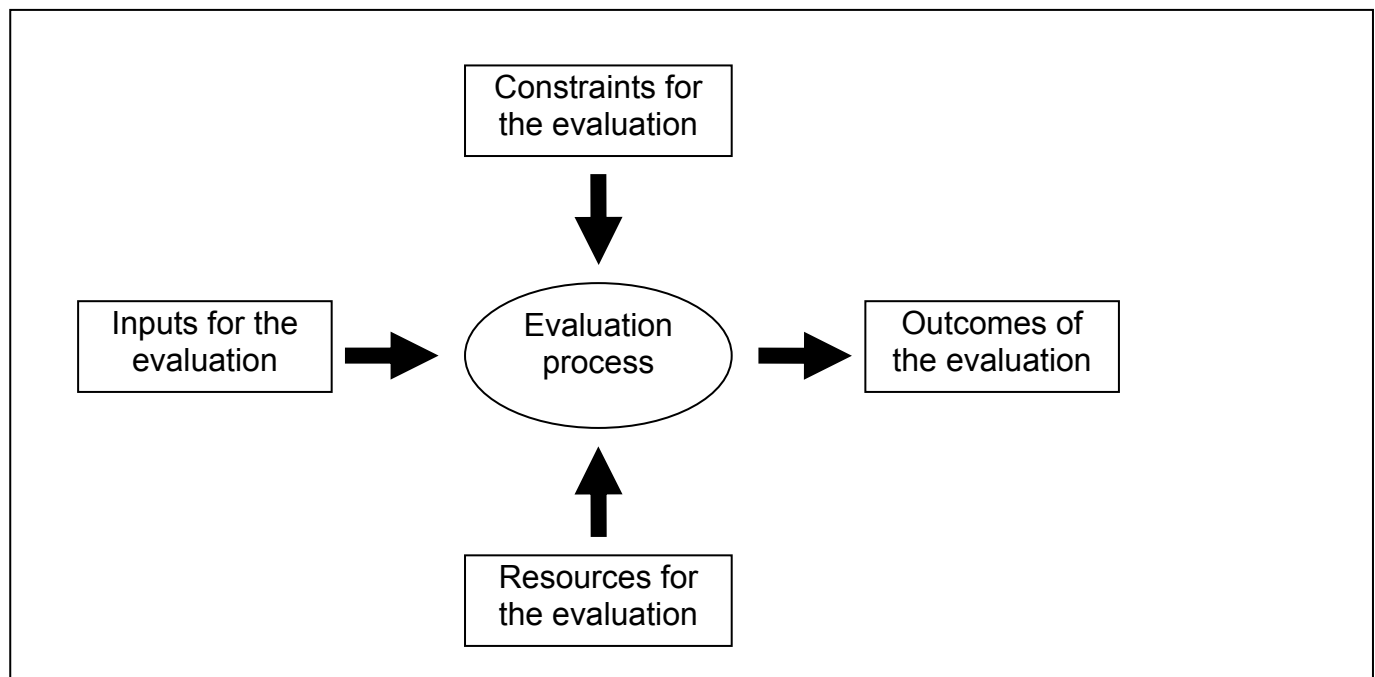


Figure 2 — Overview of software product quality evaluation