FINAL
DRAFT

TECHNICAL
REPORT

ISO/IEC DTR 7052

ISO/IEC JTC **1**/SC **7**

Secretariat: **BIS**

Voting begins on:
2023-06-15

Voting terminates on:
2023-08-10

Software engineering – Controlling frequently occurring risks during development and maintenance of custom software

Ingéniérie du logiciel – Contrôle des risques fréquents au cours du développement et de la maintenance d'un logiciel sur mesure

Reference number
ISO/IEC DTR 7052:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword ..... **iv**
Introduction ..... **v**
1 Scope **1**
2 Normative references **1**
3 Terms and definitions **1**
4 Abbreviated terms **11**
5 Explanatory note on terms **11**
5.1 The term risk 11
5.2 The term control 11
6 Risks **12**
6.1 General 12
6.2 Product-related risks 13
6.2.1 General 13
6.2.2 Risk 01: The quality of the software is reduced because of modifications to it 13
6.2.3 Risk 02: The quality of the software is reduced because of modifications to the environment in which it runs 13
6.3 Project-related risks 14
6.3.1 General 14
6.3.2 Risk 03: Planned functionality is not completed on time because of underestimation of the amount of work involved 14
6.3.3 Risk 04: The product is not delivered on time and within budget because the scope is changed 14
6.3.4 Risk 05: The software does not meet the requirements laid down because the team does not have the required expertise 15
6.3.5 Risk 06: The product does not offer the right functionality because of inadequate management of the work 15
6.3.6 Risk 07: The product lacks the right non-functional properties because functional requirements were given too much priority 16
6.3.7 Risk 08: Misunderstandings occur because the communication between stakeholders is poor 16
6.3.8 Risk 09: Custom software does not (demonstrably) meet obligations because development, use and maintenance were not sufficiently auditable 17
6.3.9 Risk 10: The product is not delivered on time because a great deal of time was needed to set up for software development 17
7 Controls **18**
7.1 General 18
7.2 Project-related controls 18
7.2.1 General 18
7.2.2 Project preparation 19
7.2.3 Project execution 23
7.2.4 Completion of development and/or maintenance 27
7.3 Organization-related controls 28
7.3.1 Control 16: Supporting teams with specialist knowledge and tools 28
7.3.2 Control 17: Continuous risk management 29
Annex A (informative) **Overview of risks and controls** **31**
Annex B (informative) **Assessment tool** **33**
Bibliography **35**

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 document 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).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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.

This document was prepared 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.

Introduction

Information and communication technology (ICT) projects run many risks. ICT projects often have to contend with delay, budget overruns, and an end result of low quality.

ICT projects in which custom software is developed and/or maintained often run extra risks, on top of the risks that are part and parcel of ICT projects in general[31]. This seems to be caused by the sheer size and complexity of such custom software projects, and by a failure to mitigate the risks inherent to software development in general, despite the fact that they are well known, and that there are suitable controls for their management.

This document describes controls for some of the risks inherent in custom software development. The purpose of this document is that during the development of custom software stakeholders can avail themselves of a collection of suitable controls. The controls included are common of themselves, making this collection of controls a logical starting point for assuring the quality of custom software development. Controls were selected for inclusion based on the experience and opinion of the subject matter experts contributing to this document.

Two target groups are important when mitigating risks during the development of custom software:

a) the software development acquirers and suppliers;

b) the end users and maintainers of the software developed.

This document details risks and controls specific to custom software development. Risk management in the context of software development is covered in ISO/IEC/IEEE 12207 and its elaboration standard ISO/IEC/IEEE 16085. Generic risk management is covered by ISO 31000 and its related standards.

This document is based on NPR 5326 developed by Royal Netherlands Standardization Institute Foundation (NEN, https://www.nen.nl/).

Software engineering – Controlling frequently occurring risks during development and maintenance of custom software

1 Scope

This document:

— describes frequently occurring risks during development and maintenance of custom software;

— describes possible controls for frequently occurring risks;

— describes the related:

  — activities, facilities and roles typically used for these controls;

  — properties of products and processes;

  — standards, measurements, testing and assessment of the properties of products and processes.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
acceptance test-driven development
ATDD
development method where team members with various backgrounds [*developers* (3.18), testers and business analysts] jointly write the acceptance tests prior to development of the relevant functionality

Note 1 to entry: Although the name of the method, acceptance test-driven development, suggests that ATDD is a specialization of *test-driven development* (3.46) (TDD), this is not the case. Rather, TDD focuses on driving development by writing *unit tests* (3.48) first and ATDD focuses on driving development by writing acceptance tests first.

[SOURCE: NEN NPR 5326:2019, 3.1, modified —Added note 1 to entry.]

3.2
acquirer
stakeholder that acquires or procures a product or service from a *supplier* (3.44)

Note 1 to entry: Other terms commonly used for an acquirer are buyer, customer, owner, purchaser or internal/organizational sponsor.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.1]

3.3
agile development
development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback

Note 1 to entry: Any use of the word “agile” in this document refers to methodology.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.1, modified — In note 1 to entry, removed the reference to ISO/IEC/IEEE 26515:2018, Annex A.]

3.4
backlog
collection of agile features or stories of both functional and non-functional requirements that are typically sorted in an order based on value priority

Note 1 to entry: This can be used as a list of product requirements and *deliverables* (3.13) not part of current work, to be prioritized and completed.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.4, modified — Removed note 1 to entry.]

3.5
behaviour-driven development
BDD
development method where team members with various backgrounds [*developers* (3.18), testers and business analysts] jointly describe the behaviour of the intended functionality prior to development of the relevant functionality

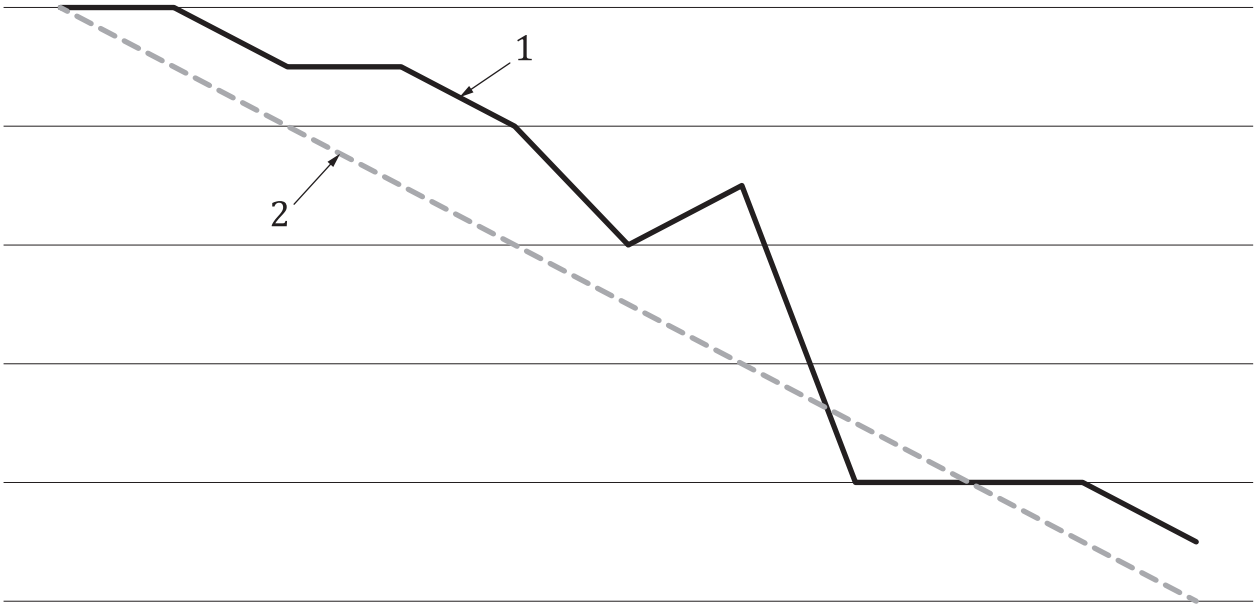
[SOURCE: NEN NPR 5326:2019, 3.5]

3.6
burndown
indicator of the work completed and estimate of remaining work to be completed or remaining effort needed to complete a product development iteration cycleNote 1 to entry: Work is measured as all work done to deliver story points, stories, features, functions, *function points* (3.25), *user stories* (3.50), *use cases* (3.49), or requirements during a product development iteration.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.437, modified — Restructured into definition and note 1 to entry.]

3.7
burndown chart
graph that represents the work remaining to do on a project

Note 1 to entry: See Figure 1 for an example of a burndown chart.



Key

1 solid line, representing the actual work-remaining

2 stippled line, representing the ideal burndown of the work

Figure 1 — Example of a burndown chart with time on the horizontal axis and work-remaining on the vertical axis

Note 2 to entry: This can be used as a chart to show the amount of the work done versus the amount of the work still to do.

Note 3 to entry: This can be presented per *sprint* (3.42), as well as per release or iteration.

Note 4 to entry: For example, user *story points* (3.51) or *function points* (3.25) can be used to measure the amount.

[SOURCE: ISO/IEC/IEEE 26511:2018, 3.1.6, modified — Added the example figure and notes to entry.]

3.8
business impact analysis
process of analysing the impact over time of a disruption on the organization

Note 1 to entry: The outcome is a statement and justification of business continuity requirements.

[SOURCE: ISO 22300:2021, 3.1.24]

3.9
code review
activity where one or more *developers* (3.18) establish the *quality* (3.35) of (part of) the *source code* (3.41) by going through it

Note 1 to entry: There are a variety of ways to conduct code reviews which range from formal to very informal and from discrete to continuous.

[SOURCE: NEN NPR 5326:2019, 3.10, modified — Modified definition and added note 1 to entry.]

3.10
configuration management database
CMDB
database that is used by an organization to store information about the hardware and software in use

[SOURCE: NEN NPR 5326:2019 3.11]

3.11
consequence
outcome of an *event* (3.23) affecting *objectives* (3.30)

Note 1 to entry: An event can lead to a range of consequences.

Note 2 to entry: A consequence can be certain or uncertain and can have positive or negative effects on objectives.

Note 3 to entry: Consequences can be expressed qualitatively or quantitatively.

Note 4 to entry: Initial consequences can escalate through knock-on effects.

[SOURCE: ISO Guide 73:2009, 3.6.1.3]

3.12
control
measure that is modifying *risk* (3.37)

Note 1 to entry: Controls include any process, policy, device, practice, or other actions which modify risk.

Note 2 to entry: Controls cannot always exert the intended or assumed modifying effect.

[SOURCE: ISO Guide 73:2009, 3.8.1.1, modified — In note 2 to entry, changed “may not” to “cannot”.]

3.13
custom software
software product developed for a specific application from a user requirements specification

[SOURCE: ISO/IEC 25000:2014, 4.3]

3.14
data protection impact assessment
DPIA
tool described in the *General Data Protection Regulation* (3.27) (GDPR) to assess in advance the privacy risks of data processing and then to be able to implement *controls* (3.12) to reduce the *risks* (3.37)

Note 1 to entry: See https://gdpr.eu/article-35-impact-assessment/.

[SOURCE: NEN NPR 5326:2019 3.12, modified — Changed “take measures” to “implement controls”. Added note 1 to entry with link to the GDPR Article 35.]

3.15
deliverable
any unique and verifiable product, result, or capability to perform a service that must be produced to complete a process, phase, or project

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1098, definition 1]

3.16
Delphi method
information-gathering technique used as a way to reach consensus of experts on a subject

Note 1 to entry: The Delphi method is applied as consensus tool for determining weights of indicators/sub-indicators in this document.

Note 2 to entry: A facilitator uses a questionnaire to solicit ideas about the important project points related to the subject. The responses are summarized and are then recirculated to the experts for further comment. Consensus can be reached in a few rounds of this process.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1102, modified — Restructured into definition and notes to entry.]

3.17
design thinking
methodology for solving (very complex) problems where these are defined from the human needs

Note 1 to entry: In design thinking solutions are determined with brainstorming sessions, where *prototypes* (3.34) are produced to test the intended solution in practice.

[SOURCE: NEN NPR 5326:2019, 3.15]

3.18
developer
individual or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the system or software life-cycle process

Note 1 to entry: Activities of the developer of *custom software* (3.13) include setup and analysis of functional and non-functional system and software requirements, design, programming and testing.

Note 2 to entry: Designers, programmers and testers are therefore all developers.

[SOURCE: ISO/IEC 25000:2014, 4.6. modified — Added notes to entry.]

3.19
development pipeline
build pipeline
set of tools aimed at the managed and controlled build of a package with which an application can be taken into use

Note 1 to entry: As part of this various tests are carried out to determine and assess *quality* (3.35).

[SOURCE: NEN NPR 5326:2019, 3.30]

3.20
development team
team that develops and/or maintains software

Note 1 to entry: The *acquirer* (3.2), *product owner* (3.33) and future maintainer can form part of the development team.

Note 2 to entry: Development teams can be broken down by function (e.g. a team of designers, a team of programmers, a team of testers) or be multidisciplinary (each team has e.g. design, programming and test expertise).

[SOURCE: NEN NPR 5326:2019, 3.31]

3.21
DevOps
set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, and operating software and systems products and services, and continuous improvements in all aspects of the life cycle

[SOURCE: IEEE 2675-2021 3.1]

3.22
distributed denial of service attack
DDoS attack
unauthorized access to a system resource or the delaying of system operations and functions in the way of compromising multiple systems to flood the bandwidth or resources of the targeted system, with resultant loss of availability to authorized users

[SOURCE: ISO/IEC 27039:2015, 3.7]

3.23
event
occurrence or change of a particular set of circumstances

Note 1 to entry: An event can be one or more occurrences and can have several causes.

Note 2 to entry: An event can consist of something not happening.

Note 3 to entry: An event can sometimes be referred to as an 'incident' or 'accident'.

Note 4 to entry: An event without *consequences* (3.11) can also be referred to as a 'near miss', 'accident', 'near hit' or 'close call'.

[SOURCE: ISO Guide 73:2009, 3.5.1.3]

3.24
extreme programming
XP
form of agile software development where procedures for iterative planning and working are combined with technical procedures, such as test-driven design and pair programming

[SOURCE: NEN NPR 5326:2019, 3.20]

3.25
function point
FP
unit of measure for functional size

[SOURCE: ISO/IEC 20926:2009, 3.35, modified — removed "as defined within this International Standard".]

3.26
function point analysis
FPA
method for measuring functional size

[SOURCE: ISO/IEC 20926:2009, 3.36, modified — removed "as defined within this International Standard".]

3.27
General Data Protection Regulation
GDPR
European Regulation that standardizes the rules for processing personal data by private businesses and government bodies in the whole European Union

Note 1 to entry: See https://gdpr.eu.

[SOURCE: NEN NPR 5326:2019, 3.3]

3.28
integration testing
testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.2034, modified — Removed note 1 to entry.]

3.29
minimum viable product
MVP
version of a work product with just enough features and requirements to satisfy early customers and/or provide feedback for future development

[SOURCE: IEEE 2675-2021, 3.1]

3.30
objective
predetermined result to be achieved

Note 1 to entry: To achieve an objective several tools and activities are generally needed, one of which can be *custom software* (3.13).

[SOURCE: NEN NPR 5326:2019, 3.18]

3.31
performance test
test type conducted to evaluate the degree to which a test item accomplishes its designated functions within given constraints of time and other resources

Note 1 to entry: A distinction is often made between load, stress and endurance tests. Load tests simulate a normal load on the system. Stress tests allow the load to increase to determine the maximum load at which the system still functions. Endurance tests load the system for a longer period in order to discover memory leaks or other problems which only manifest themselves after some time.

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.58, modified — Changed the term from "performance testing" to "performance test"; changed "type of testing" to "test type"; added note 1 to entry.]

3.32
product breakdown structure
PBS
decomposition of the product into its components

Note 1 to entry: The PBS begins with the complete product at the top of the hierarchy and below this the main components, which can also each be broken down again into components, etc.

[SOURCE: ISO 21511:2018, 3.7, modified — Added the abbreviated term and note 1 to entry.]

3.33
product owner
stakeholder (3.16) responsible for the capabilities, acceptance and use of a product

Note 1 to entry: The product owner shares the product vision, required features and their priorities, and acceptance criteria.

[SOURCE: ISO/IEC TR 24587:2021, 3.12]

3.34
prototype
model or preliminary implementation suitable for evaluation of system design, performance, and production potential, or for better understanding or determination of the requirements

[SOURCE: ISO/IEC 2382:2015, 2122670, modified — Removed notes to entry.]

3.35
quality
ability of a product, service, system, component, or process to meet customer or user needs, expectations, or requirements

Note 1 to entry: For the quality of software and IT systems, ISO/IEC 25010 offers a breakdown into aspects, such as: