

Terminologi for test av programvare

Oversettelse til norsk av

**Standard glossary of terms used in Software Testing
Version 2.1**

**Produced by the ‘Glossary Working Party’
International Software Testing Qualifications Board**

**Norsk versjon:
Norwegian Testing Board
International Software Testing Qualification Board**

Versjon 2.1.08N

Editor: Erik van Veenendaal (engelsk),
Ernst von Düring, Hans Schaefer, Thomas Borchsenius, Lena
Kleven, Frans Dijkman (norsk)

Copyright Notice

Dette dokument kan kopieres fritt, dersom kilden blir nevnt.

Bidragsyttere til den engelske originalversjonen

Rex Black (USA)	Dale Perry (USA)
Enst Düring (Norway)	Horst Pohlmann (Germany)
Sigrid Eldh (Sweden)	Meile Posthuma (The Netherlands)
Isabel Evans (UK)	Erkki Pöyhönen (Finland)
David Fuller (Australia)	Maaret Pyhäjärvi (Finland)
Annu George (India)	Andy Redwood (UK)
Dorothy Graham (UK)	Stuart Reid (UK)
Mats Grindal (Sweden)	Piet de Roo (The Netherlands)
Matthias Hamburg (Germany)	Steve Sampson (UK)
Julian Harty (UK)	Shane Saunders (UK)
David Hayman (UK)	Hans Schaefer (Norway)
Bernard Homes (France)	Jurriën Seubers (The Netherlands)
Ian Howles (UK)	Dave Sherratt (UK)
Juha Itkonen (Finland)	Mike Smith (UK)
Paul Jorgensen (US)	Andreas Spillner (Germany)
Vipul Kocher (India)	Lucjan Stapp (Poland)
Fernando Lamas de Oliveira (Portugal)	Richard Taylor (UK)
Tilo Linz (Germany)	Geoff Thompson (UK)
Gustavo Marquez Sosa (Spain)	Stephanie Ulrich (Germany)
Peter Morgan (UK)	Matti Vuori (Finland)
Thomas Müller (Switzerland)	Gearrel Welvaart (The Netherlands)
Avi Ofer (Israel)	Pete Williams (UK)

Norsk versjon

Oversettelse: Hans Schaefer, Lena Kleven og Ernst von Düring

Norsk versjon 2.0 gransket av: Hans Schaefer, Monika Stöcklein-Olsen, Lillann Solberg, Kåre Ivar Gauksrud

Norsk versjon 2.1: Hans Schaefer, Ernst von Düring, Lena Kleven, Monique Pruess, Thomas Borchsenius samt bidrag av flere i Norwegian Testing Board

Endringslogg for norsk versjon

Norsk versjon	Dato	Beskrivelse	Ansvarlig (bidragsyttere)
1.2	2006-01-12	Basert på engelsk versjon 1.1	Hans Schaefer
2.0	2008-01-14	Basert på engelsk versjon 2.0	Ernst von Düring (se også over)
2.0.01	2009-01-02	Norsk ”mellomversjon”. Gjennomgang i forbindelse med utarbeidelse av ”strippet versjon”. Rettet noen feil og komplettert en del mangler.	Ernst von Düring (Hans Schaefer, Monika Stöcklein-Olsen, Trude Rosendal)
2.1.08N	2010-9-14	Oppdatert i forhold til ISTQB syllabus versjon 2010. Utført konsistenssjekk, granskning og lagt inn kommentarer/fotnote ved inkrementell utvikling og ved sikkerhet	Ernst von Düring (Hans Schaefer, Monique Pruess, Thomas Borchsenius, Lena Kleven, Frans Dijkman)

Se <http://www.istqb.no/dokumentasjon/> for siste engelske og norske utgave samt en strippet versjon engelsk-norsk.

Innholdsfortegnelse

Foreword.....	5
1. Introduction.....	5
2. Scope.....	5
3. Arrangement	5
4. Normative references	6
5. Trademarks	6
6. Andre referanser.....	7
7. Definisjoner.....	7
A.....	7
B.....	9
C.....	13
D.....	20
E.....	25
F.....	27
G.....	30
H.....	31
I.....	32
J.....	36
K.....	36
L.....	36
M.....	38
N.....	42
O.....	43
P.....	44
Q.....	49
R.....	50
S.....	55
T.....	62
U.....	74
V.....	75
W.....	76
Annex A (Informative)	77
Annex B (Method of commenting on this glossary).....	79

Foreword

In compiling this glossary the working party has sought the views and comments of as broad a spectrum of opinion as possible in industry, commerce and government bodies and organizations, with the aim of producing an international testing standard which would gain acceptance in as wide a field as possible. Total agreement will rarely, if ever, be achieved in compiling a document of this nature. Contributions to this glossary have been received from the testing communities in Australia, Belgium, Finland, France, Germany, India, Israel, The Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom, and USA.

Many (software) testers have used BS 7925-1 since its original publication in 1998. It has served also as a major reference for the Information Systems Examination Board (ISEB) qualification at both Foundation and Practitioner level. The standard was initially developed with a bias towards component testing, but, since its publication, many comments and proposals for new definitions have been submitted to both improve and expand the standard to cover a wider range of software testing. The ISTQB testing glossary has incorporated many of these suggested updates. It is used as a reference document for the International Software Testing Qualifications Board (ISTQB) software testing qualification scheme.

1. Introduction

Much time and effort is wasted both within and between industry, commerce, government and professional and academic institutions when ambiguities arise as a result of the inability to differentiate adequately between such terms as ‘statement coverage’ and ‘decision coverage’; ‘test suite’, ‘test specification’ and ‘test plan’ and similar terms which form an interface between various sectors of society. Moreover, the professional or technical use of these terms is often at variance with different meanings attributed to them.

2. Scope

This document presents concepts, terms and definitions designed to aid communication in (software) testing and related disciplines.

3. Arrangement

The glossary has been arranged in a single section of definitions ordered alphabetically. Some terms are preferred to other synonymous ones, in which case, the definition of the preferred term appears, with the synonymous ones referring to that. For example *structural testing* refers to *white box testing*. For synonyms, the “See” indicator is used

“See also” cross-references are also used. They assist the user to quickly navigate to the right index term. “See also” cross-references are constructed for relationships such as broader term to a narrower term, and overlapping meaning between two terms.

4. Normative references

At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based upon this Standard are encouraged to investigate the possibility of applying the most recent edition of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

- BS 7925-2:1998. Software Component Testing.
- DO-178B: 1992. Software Considerations in Airborne Systems and Equipment Certification, Requirements and Technical Concepts for Aviation (RTCA SC167).
- IEEE 610.12:1990. Standard Glossary of Software Engineering Terminology.
- IEEE 829:1998. Standard for Software Test Documentation.
- IEEE 1008:1993. Standard for Software Unit Testing.
- IEEE 1012:2004. Standard for Verification and Validation Plans
- IEEE 1028:1997. Standard for Software Reviews and Audits.
- IEEE 1044:1993. Standard Classification for Software Anomalies.
- IEEE 1219:1998. Software Maintenance.
- ISO/IEC 2382-1:1993. Data processing - Vocabulary - Part 1: Fundamental terms.
- ISO 9000:2005. Quality Management Systems – Fundamentals and Vocabulary.
- ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality characteristics and sub-characteristics.
- ISO/IEC 12207:1995. Information Technology – Software Life Cycle Processes.
- ISO/IEC 14598-1:1999. Information Technology – Software Product Evaluation - Part 1: General Overview.
- ISO 15504-9: 1998. Information Technology – Software Process Assessment – Part 9: Vocabulary

5. Trademarks

In this document the following trademarks are used:

- CMM, and CMMI and IDEAL are registered trademarks of Carnegie Mellon University
- EFQM is a registered trademark of the EFQM Foundation
- Rational Unified Process is a registered trademark of Rational Software Corporation
- STEP is a registered trademark of Software Quality Engineering
- TMap, TPA and TPI are registered trademarks of Sogeti Nederland BV
- TMM is a registered service mark of Illinois Institute of Technology
- TMMi is a registered trademark of the TMMi Foundation

6. Andre referanser

- tidl. DND terminologiliste: DND faggruppen software testing utarbeidet en ordliste innen software testing på 1990-tallet. De fleste termer har blitt tatt inn i denne listen, der de ikke var direkte oversettelser av engelske termer.

7. Definisjoner

Rekkefølgen etter som i den engelske originalen, dvs. alfabetisk etter engelske termer. Noen begreper er ikke oversatt. Disse brukes bare på de høyere nivå sertifikeringene i ISTQB. De er merket spesielt. Forslag til oversettelse imøteses av leserne.

A

abstract test case: See <i>high-level test case</i> .	abstrakt testtilfelle: Se <i>høynivå testtilfelle</i>
acceptance: See <i>acceptance testing</i> .	akseptanse (godkjenning): Se <i>akseptansetesting</i> .
acceptance criteria: The exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. [IEEE 610]	akseptanseskriterier: Sluttkriteriene som en komponent eller et system må oppfylle for å bli godkjent av en kunde, bruker eller en annen autorisert enhet. [IEEE 610]. Se også: <i>godkjenningskriterier (pass/fail criteria)</i>
acceptance testing: Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. [IEEE 610]	akseptansetesting: Formell testing med hensyn til brukerbehov, brukerkrav, myndighetskrav og kundens arbeidsprosesser som utføres for å avklare om et system oppfyller akseptanseskriterier eller ikke. [IEEE 610]
accessibility testing: Testing to determine the ease by which users with disabilities can use a component or system. [Gerrard]	testing av tilgjengelighet: Testing for å avgjøre hvor lett det er for personer med handikap å bruke en komponent eller system.
accuracy: The capability of the software product to provide the right or agreed results or effects with the needed degree of precision. [ISO 9126] See also <i>functionality testing</i> .	nøyaktighet: Programvareproduktets evne til å sørge for de rette eller avtalte resultater eller virkninger med den nødvendige grad av presisjon. [ISO 9126].
action word driven testing: See <i>keyword driven testing</i> .	Se <i>nøkkelord drevet testing</i> .
actual outcome: See <i>actual result</i> .	virkelig resultat.

actual result: The behaviour produced/observed when a component or system is tested.	virkelig resultat: Testobjektets produserte eller observerte oppførsel under test. Også <i>faktisk resultat</i> .
ad hoc review: See <i>informal review</i> .	ad-hoc review: Se <i>uformell granskning</i>
ad hoc testing: Testing carried out informally; no formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity.	ad-hoc testing: Testing som blir utført uformelt. Ingen formell testforberedelse finner sted. Ingen anerkjent testdesignteknikk blir brukt, det finnes ingen nedskrevne forventede resultater. Tilfeldighet styrer testutførelsen.
adaptability: The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered. [ISO 9126] See also <i>portability</i> .	tilpassningsevne: Programvareproduktets evne til å bli tilpasset for ulike spesifiserte omgivelser uten å bruke andre aksjoner eller midler enn de som er inkludert for dette formål.[ISO 9126]. Se også <i>portabilitet</i> .
agile testing: Testing practice for a project using agile methodologies, such as extreme programming (XP), treating development as the customer of testing and emphasizing the test-first design paradigm. See also <i>test driven development</i> .	smidig testing: Testing ved et prosjekt som bruker agile (smidige) metoder som for eksempel ekstrem programmering (XP), der utviklingen blir behandlet som en kunde av testingen og der test-først design blir brukt. <i>Se også testdrevet utvikling</i>
algorithm test [TMap]: See <i>branch testing</i> .	<i>Se forgreningstesting.</i>
alpha testing: Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organization. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing.	alfatesting: Simulert eller virkelig operasjonell testing som gjennomføres av potensielle brukere eller kunder eller et uavhengig testteam i utviklingsmiljøet, dog uavhengig av utviklingsorganisasjonen. Dette blir ofte brukt som en form for intern akseptansetesting.
analyzability: The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified. [ISO 9126] See also <i>maintainability</i> .	analyserbarhet: Programvareproduktets evne til å bli diagnostisert for feil, mangler e.l. eller årsaker for feil i programvaren, eller muligheten til å modifisere deler slik at de kan identifiseres. [ISO 9126] Se også <i>vedlikeholdbarhetstesting</i> .
analyzer: See <i>static analyzer</i> .	analysator: Se <i>statisk analysator</i>
anomaly: Any condition that deviates from expectation based on requirements specifications, design documents, user documents, standards, etc. or from someone's perception or experience. Anomalies may be found during, but not limited to, reviewing, testing, analysis, compilation, or use of software products or applicable documentation. [IEEE 1044] See also <i>defect, deviation, error, fault, failure, incident, problem</i> .	anomali: Ethvert forhold som avviker fra forventningene som er basert på kravspesifikasjoner, design dokumenter, brukerdokumenter, standarder etc. eller avvik fra noens syn eller erfaring. Anomalier kan finnes under testing, analyse, kompilering, eller bruk av programvareprodukter eller tilhørende dokumentasjon. [IEEE 1044]. Se også <i>defekt, avvik, feil, problem</i> .

arc testing: See <i>branch testing</i> .	forgreningstesting: Se også (eng.) <i>branch testing</i> ..
attack: Directed and focused attempt to evaluate the quality, especially reliability, of a test object by attempting to force specific failures to occur.	angrep: Styrt og konsentrert innsats for å evaluere kvaliteten, spesielt påliteligheten eller sikkerheten, til et testobjekt ved å forsøke tvinge frem spesifikke problem.
attractiveness: The capability of the software product to be attractive to the user. [ISO 9126] See also <i>usability</i> .	attraktivitet: Programvareproduktets evne til å være attraktiv for brukeren. [ISO 9126] Se også <i>brukervennlighet</i> .
audit: An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and/or procedures based on objective criteria, including documents that specify: (1) the form or content of the products to be produced (2) the process by which the products shall be produced (3) how compliance to standards or guidelines shall be measured. [IEEE 1028]	revisjon: En uavhengig evaluering av programvareproduktene eller prosessene som brukes til å sikre samsvar til standarder, retningslinjer, spesifikasjoner, og/eller prosedyrer og som baserer seg på objektive kriterier, inklusive dokumenter som spesifiserer: (1) formen eller innholdet av produktene som skal produseres (2) prosessen som skal brukes for å produsere produktene (3) hvordan samsvar med standarder eller retningslinjer skal måles. [IEEE 1028]
audit trail: A path by which the original input to a process (e.g. data) can be traced back through the process, taking the process output as a starting point. This facilitates defect analysis and allows a process audit to be carried out. [TMap]	revisjonsspør: Foreløpig ikke oversatt da dette bare brukes i ISTQB-syllabi på høyere nivå.
Automated testware: Testware used in automated testing, such as tool scripts.	automatisert testvare: Testmateriale som blir brukt innen automatisert testing, som for eksempel scripter for testverktøy.
availability: The degree to which a component or system is operational and accessible when required for use. Often expressed as a percentage. [IEEE 610]	tilgjengelighet: Til hvilken grad en komponent eller system er klart til bruk og kan fås kontakt med når det skal brukes. Ofte uttrykt som prosent. [IEEE 610]

B

back-to-back testing: Testing in which two or more variants of a component or system are executed with the same inputs, the outputs compared, and analyzed in cases of discrepancies. [IEEE 610]	sammenlignende test: Test der to eller flere varianter av en komponent eller et system blir utført med de samme inndata, der utdata og resultater blir sammenlignet og analysert når avvik oppstår. [IEEE 610]
---	---

<p>baseline: A specification or software product that has been formally reviewed or agreed upon, that thereafter serves as the basis for further development, and that can be changed only through a formal change control process. [IEEE 610]</p>	<p>basislinje, baseline: En spesifikasjon eller programvareprodukt som har blitt formelt gjennomgått eller akseptert og som deretter tjener som basis for videre utvikling, og som kan bare endres gjennom en formell endringsprosess. [IEEE 610]</p>
<p>basic block: A sequence of one or more consecutive executable statements containing no branches. Note: A node in a control flow graph represents a basic block.</p>	<p>basisblokk: En sekvens av et eller flere etterfølgende utførbare programinstruksjoner som ikke inneholder forgreninger. NB: En node i en kontrollflytgraf representerer en basisblokk.</p>
<p>basis test set: A set of test cases derived from the internal structure of a component or specification to ensure that 100% of a specified coverage criterion will be achieved.</p>	<p>basis testsett: (1) En samling testtilfeller som er avledet fra komponentens interne struktur eller spesifikasjonen og som sikrer at 100% av et spesifisert dekningskriterium blir oppnådd. (2) Et begrenset sett med testtilfeller som kan for eksempel brukes til ”smoke test”.</p>
<p>bebugging: See <i>fault seeding</i>. [Abbott]</p>	<p>Se (eng.) <i>fault seeding</i> Foreløpig ikke oversatt da dette bare brukes i ISTQB-syllabi på høyere nivå.</p>
<p>behavior: The response of a component or system to a set of input values and preconditions.</p>	<p>oppførsel: Komponentens eller systemets svar på et sett med inputverdier og forbetninger.</p>
<p>benchmark test: (1) A standard against which measurements or comparisons can be made. (2) A test that is used to compare components or systems to each other or to a standard as in (1). [IEEE 610]</p>	<p>referansetest: (1) En standard som målinger eller sammenligninger gjøres mot. (2) En test som brukes for å sammenligne komponenter eller systemer mot hverandre eller til en standard som i (1). [IEEE 610]</p>
<p>bespoke software: Software developed specifically for a set of users or customers. The opposite is off-the-shelf software.</p>	<p>spesialutviklet programvare: Programvare som er spesifikt utviklet for en samling brukere eller kunder. Det motsatte av ”off-the-shelf” programvare eller standardprogrammer.</p>
<p>beta testing: Operational testing by potential and/or existing users/customers at an external site not otherwise involved with the developers, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for off-the-shelf software in order to acquire feedback from the market.</p>	<p>betatesting: Operasjonell testing ved potensielle og/eller eksisterende brukere eller kunder på en ekstern plattform som ikke er involvert i utviklingen. Betatest skal finne ut om en komponent eller et system oppfyller brukerens eller kundens behov og passer inn i forretningsprosessen. Betatesting blir ofte brukt som en form for ekstern ”akseptansetesting av standardprogrammer” for å få tilbakemelding fra markedet.</p>

<p>big-bang testing: A type of integration testing in which software elements, hardware elements, or both are combined all at once into a component or an overall system, rather than in stages. [IEEE 610] See also <i>integration testing</i>.</p>	<p>big bang testing: En type integrasjonstesting der alle programvareelementer, hardwareelementer eller begge to blir kombinert på en gang til en komponent eller et helt system, istedenfor å bli kombinert trinnvis. [IEEE 610] Se også <i>integrasjonstesting</i> <i>Anmerkning av oversetter: Et naturlig norsk begrep for dette mangler. Derfor brukes det engelske ordet.</i></p>
<p>black-box technique: See <i>black box test design technique</i>.</p>	<p>svart boks teknikk: Se <i>svart boks testdesignteknikk</i>.</p>
<p>black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.</p>	<p>svart boks testing: Funksjonell eller ikke funksjonell testing uten referanse til den interne strukturen i komponenten eller systemet. Test av systemets oppførsel sett utenfra. En lager testtilfeller kun ved å se på kravene til testobjektet. Kunnskap om interne detaljer (design, kode) brukes (i prinsipp) ikke. Se også <i>strukturell test</i>.</p>
<p>black-box test design technique: Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.</p>	<p>svart boks testdesignteknikk: Framgangsmåte til å avlede og velge testtilfelle basert på en analyse av spesifikasjonen, enten funksjonell eller ikke funksjonell, av en komponent eller et system uten referanse til den interne strukturen.</p>
<p>blocked test case: A test case that cannot be executed because the preconditions for its execution are not fulfilled.</p>	<p>blokkert testtilfelle: Et testtilfelle som ikke kan bli utført fordi forbetingelsene for dets utførelse ikke er oppfylt.</p>
<p>bottom-up testing: An incremental approach to integration testing where the lowest level components are tested first, and then used to facilitate the testing of higher-level components. This process is repeated until the component at the top of the hierarchy is tested. See also <i>integration testing</i>.</p>	<p>nedenfra-opp-testing: En trinnvis metode for integrasjonstesting der de komponentene som er på laveste nivå testes først. De blir deretter brukt til testing av høyere nivå komponenter. Denne prosessen repeteres inntil komponenten på toppen av hierarkiet er testet. Se også <i>integrasjonstesting</i>.</p>
<p>boundary value: An input value or output value, which is on the edge of an equivalence partition or at the smallest incremental distance on either side of an edge, for example the minimum or maximum value of a range.</p>	<p>grenseverdi: En input eller output verdi som er på grensen av en ekvivalensinndeling eller som er i den minst mulige avstanden til en slik grense, for eksempel minimum eller maksimum verdi av et verdiområde.</p>

boundary value analysis: A black box test design technique in which test cases are designed based on boundary values. See also <i>boundary value</i> .	grenseverdianalyse: En svart boks testdesignteknikk der testtilfelle blir laget basert på grenseverdier. <i>Se også grenseverdi</i>
boundary value coverage: The percentage of boundary values that have been exercised by a test suite.	grenseverdidekning: Prosenten grenseverdier som er blitt utført av en testsuite.
boundary value testing: See <i>boundary value analysis</i> .	grenseverditesting: Se <i>grenseverdianalyse</i> .
branch: A basic block that can be selected for execution based on a program construct in which one of two or more alternative program paths are available, e.g. case, jump, go to, if-then-else.	forgrening: En basisblokk i et program som kan bli valgt for utføring basert på en programkonstruksjon der en av to eller flere alternative programveier er tilgjengelig, for eksempel case, jump, go to, if-then-else.
branch condition: See <i>condition</i> .	<i>Se betingelse.</i>
branch condition combination coverage: See <i>multiple condition coverage</i> .	betingelseskombinasjonsdekning: Se <i>multiple condition coverage (eng.)</i>
branch condition combination testing: See <i>multiple condition testing</i> .	betingelseskombinasjonstesting: Se <i>sammensatt betingelses testing</i> .
branch condition coverage: See <i>condition coverage</i> .	betingelsesdekning: Et strukturelt testkriterium der det blir krevd at hver individuell betingelse i et program blir utført minst en gang i hver retning. Dekningen er 100% hvis dette er oppfylt. Se (eng.) <i>condition coverage</i> .
branch coverage: The percentage of branches that have been exercised by a test suite. 100% branch coverage implies both 100% decision coverage and 100% statement coverage.	forgreningsdekning: Prosenten av forgreninger som har blitt utført av en testsuite. 100% forgreningsdekning impliserer 100% beslutningsdekning og 100% programinstruksjonsdekning.
branch testing: A white box test design technique in which test cases are designed to execute branches.	forgreningstesting: En hvit boks testdesignteknikk der testtilfeller blir laget for å utføre forgreninger. Se også <i>strukturell testing</i> .
buffer: A device or storage area used to store data temporarily for difference in rates of flow, time or occurrence of events, or amounts of data that can be handled by the devices or processes involved in the transfer or use of the data. [IEEE 610]	buffer: En innretning eller et lagringsområde som brukes til midlertidig lagring av data på grunn av forskjellig hastighet i dataflyt, tid eller rekkefølge på hendelser, eller mengden data som kan håndteres av innretningene eller prosessene involvert i utveksling og bruk av data. [IEEE 610]

buffer overflow: A memory access defect due to the attempt by a process to store data beyond the boundaries of a fixed length buffer, resulting in overwriting of adjacent memory areas or the raising of an overflow exception. See also <i>buffer</i> .	bufferoverskridelse: En feil der et program skriver utenfor et definert område i minnet som resulterer i overskriving av andre deler i minne.
bug: See <i>defect</i> .	programfeil: Se <i>defekt</i> .
bug report: See <i>defect report</i> .	programfeilrapport: Se <i>feilrapport</i> .
bug taxonomy: See <i>defect taxonomy</i> .	taksonomi av feil : Se (eng.) <i>defect taxonomy</i>
bug tracking tool: See <i>defect management tool</i> .	Se <i>feilhåndteringsverktøy</i> .
business process-based testing: An approach to testing in which test cases are designed based on descriptions and/or knowledge of business processes	forretningsprosessetesting: Også prosessetesting. En testmåte der testtilfelle blir konstruert basert på beskrivelser og/eller kunnskap om forretningsprosesser.

C

Capability Maturity Model (CMM): A five level staged framework that describes the key elements of an effective software process. The Capability Maturity Model covers best-practices for planning, engineering and managing software development and maintenance. [CMM] See also <i>Capability Maturity Model Integration (CMMI)</i> .	Capability Maturity Model (CMM): Et rammeverk i fem nivåer som beskriver nøkkelementene i en effektiv software prosess. Modellen dekker teknikker for å planlegge, utvikle og styre software utvikling og vedlikehold. Også kalt modenhetsmodell. Beskriver en organisasjons modenhet. [CMM] Se også <i>Capability Maturity Model Integration (CMMI)</i> .
Capability Maturity Model Integration (CMMI): A framework that describes the key elements of an effective product development and maintenance process. The Capability Maturity Model Integration covers best-practices for planning, engineering and managing product development and maintenance. CMMI is the designated successor of the CMM. [CMMI] See also <i>Capability Maturity Model (CMM)</i> .	Capability Maturity Model Integration (CMMI): Et rammeverk som beskriver nøkkelementene i en effektiv produktutviklings- og vedlikeholdsprosess. Modellen dekker teknikker for planlegging, utvikling og styring av produktutvikling og vedlikehold. CMMI er utpekt som etterfølger av CMM. [CMMI] Se også <i>Capability Maturity Model (CMM)</i> .
capture/playback tool: A type of test execution tool where inputs are recorded during manual testing in order to generate automated test scripts that can be executed later (i.e. replayed). These tools are often used to support automated regression testing.	verktøy for innspilling og avspilling: En type testutførelsesverktøy der inndata og handlinger blir logget under manuell testing for å generere automatiske test scripter som kan utføres senere (ved avspilling). Disse verktøy blir ofte brukt for å understøtte automatisert regresjonstesting.
capture/replay tool: See <i>capture/playback tool</i> .	Se <i>verktøy</i> for innspilling og avspilling

CASE: Acronym for Computer Aided Software Engineering.	CASE: Forkortelse for Computer Aided Software Engineering
CAST: Acronym for Computer Aided Software Testing. See also <i>test automation</i> .	CAST: Forkortelse for Computer Aided Software Testing. Se også <i>testautomatisering</i>
cause-effect analysis: See <i>cause-effect graphing</i> .	Se <i>årsak-virkningsanalyse</i> .
cause-effect graph: A graphical representation of inputs and/or stimuli (causes) with their associated outputs (effects), which can be used to design test cases.	årsak-virkningsgraf: En grafisk representasjon av inputs eller årsaker med deres tilhørende outputs eller effekter som kan brukes for å lage testtilfelle.
cause-effect graphing: A black box test design technique in which test cases are designed from cause-effect graphs. [BS 7925/2]	årsak-virkningsanalyse: En svart boks testdesignteknikk der testtilfelle blir laget fra årsak-virkningsgrafer. [BS 7925/2]
cause-effect decision table: See <i>decision table</i> .	årsak-virknings beslutningstabell: Se <i>beslutningstabell</i>
certification: The process of confirming that a component, system or person complies with its specified requirements, e.g. by passing an exam.	sertifisering: Prosessen der en fastslår at en komponent, et system eller en person samsvarer med spesifiserte krav, for eksempel ved å bestå en eksamen.
changeability: The capability of the software product to enable specified modifications to be implemented. [ISO 9126] See also <i>maintainability</i> .	endringsvennlighet: Programvareproduktets evne til å muliggjøre implementasjon av spesifiserte modifikasjoner. [ISO 9126] Se også <i>vedlikeholdbarhet</i>
change control: See <i>configuration control</i> .	endringskontroll: Se <i>konfigurasjonskontroll</i> .
change control board: See <i>configuration control board</i> .	endringskontrollråd: Se <i>configuration control board (eng.)</i> .
checker: See <i>reviewer</i> .	kontrollør: Se <i>reviewer (eng.)</i> .
Chow's coverage metrics: See <i>N-switch coverage</i> . [Chow]	Chow's dekningsmål: Se <i>N-switch coverage</i> . [Chow]
classification tree: A tree showing equivalence partitions hierarchically ordered, which is used to design test cases in the classification tree method. See also <i>classification tree method</i> .	klassifikasjonstre: En trestruktur som viser ekvivalensklasser ordnet i et hierarki, som brukes til å lage testtilfeller i klassifikasjonstremetoden. Se også <i>klassifikasjonstremetode</i> .
classification tree method: A black box test design technique in which test cases, described by means of a classification tree, are designed to execute combinations of representatives of input and/or output domains. [Grochtmann]	klassifikasjonstremetode: En svart boks testdesignteknikk der problemet blir beskrevet ved hjelp av et klassifikasjonstre. Testtilfelle blir laget for å utføre kombinasjoner av representanter for input eller output domener. [Grochtmann]
code: Computer instructions and data definitions expressed in a programming language or in a form output by an assembler, compiler or other translator. [IEEE 610]	kode: Instruksjoner og datadefinisjoner uttrykt i et programmeringsspråk eller som resultat fra assembler, kompilator eller annen oversetter. [IEEE 610]
code analyzer: See <i>static code analyzer</i> .	kodeanalysator: Se <i>statisk kodeanalysator</i>

code coverage: An analysis method that determines which parts of the software have been executed (covered) by the test suite and which parts have not been executed, e.g. statement coverage, decision coverage or condition coverage.	kodedekning: En analysemetode som måler hvilke deler av programvaren har blitt utført (dekket) av en testsuite og hvilke deler har ikke blitt utført, for eksempel programinstruksjonsdekning, beslutningsdekning eller betingelsesdekning.
code-based testing: See <i>white box testing</i> .	kodebasert test: Se <i>strukturell testing</i> , <i>hvit boks testing</i> .
co-existence: The capability of the software product to co-exist with other independent software in a common environment sharing common resources. [ISO 9126] See also <i>portability testing</i> .	koeksistens: Programvareproduktets evne å være sammen med annen uavhengig programvare i en felles omgivelse der en deler felles ressurser. [ISO 9126] Se også <i>portabilitetstesting</i> .
commercial off-the-shelf software: See <i>off-the-shelf software</i> .	standardprogramvare: Se (eng.) <i>off-the-shelf software</i> .
comparator: See <i>test comparator</i> .	komparator, sammenligningsverktøy. Se <i>test komparator</i> .
compatibility testing: See <i>interoperability testing</i> .	kompatibilitetstesting: Se <i>interoperabilitetstest</i> .
compiler: A software tool that translates programs expressed in a high order language into their machine language equivalents. [IEEE 610]	kompilator: Softwareverktøy som oversetter programmer uttrykt i høynivå programmeringsspråk til maskinkode. [IEEE 610]
complete testing: See <i>exhaustive testing</i> .	fullstendig test: Se (eng.) <i>exhaustive testing</i> .
completion criteria: See <i>exit criteria</i> .	sluttkriterier: Se (eng.) <i>exit criteria</i> .
complexity: The degree to which a component or system has a design and/or internal structure that is difficult to understand, maintain and verify. See also <i>cyclomatic complexity</i> .	kompleksitet: Egenskap som en komponent eller et system har i sin design eller sin interne struktur som gjør det vanskelig å forstå, vedlikeholde og verifisere. Se også norsk oversettelse av (eng.) <i>cyclomatic complexity</i> .
compliance: The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions. [ISO 9126]	samsvar: Programvareproduktets evne til å oppfylle standarder, konvensjoner eller regler i lovverk eller andre liknende forskrifter. [ISO 9126]
compliance testing: The process of testing to determine the compliance of the component or system.	samsvartesting: Testing av systemets eller komponentens samsvar med standarder, konvensjoner o.l. Se <i>samsvar</i> .
component: A minimal software item that can be tested in isolation.	komponent: Et minimalt stykke software som kan testes isolert. Også modul, enhet.
component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.	komponentintegrasjonstest: Test som utføres for å finne feil i grensesnittene og samspillet mellom integrerte komponenter. Se også <i>modulintegrasjonstest</i> , <i>integrasjonstest</i> .

component specification: A description of a component's function in terms of its output values for specified input values under specified conditions, and required non-functional behavior (e.g. resource-utilization).	komponentspesifikasjon: En beskrivelse av en komponents funksjon ved å oppgi output for spesifisert input under spesifiserte betingelser samt påkrevde ikke funksjonelle egenskaper (for eksempel bruk av ressurser).
component testing: The testing of individual software components. [IEEE 610]	komponenttesting: Test av individuelle programvarekomponenter [IEEE 610] Se også <i>modultesting, enhetstesting.</i>
compound condition: Two or more single conditions joined by means of a logical operator (AND, OR or XOR), e.g. 'A>B AND C>1000'.	sammensatt betingelse: To eller flere enkle betingelser som er sammensatt ved hjelp av logiske operatører (AND, OR eller XOR), for eksempel 'A>B AND C>1000'.
concrete test case: See <i>low level test case.</i>	konkret testtilfelle: Se <i>lavnivå testtilfelle.</i>
concurrency testing: Testing to determine how the occurrence of two or more activities within the same interval of time, achieved either by interleaving the activities or by simultaneous execution, is handled by the component or system. [IEEE 610]	samtidighetstest: Test for å bestemme hvordan forekomsten av to eller flere aktiviteter innen samme tidsintervall, oppnådd enten ved å veve aktivitetene sammen eller ved å utføre de samtidig, blir håndtert av en komponent eller et system. [IEEE 610]
condition: A logical expression that can be evaluated as True or False, e.g. A>B. See also <i>test condition.</i>	betingelse: Et logisk uttrykk som kan evalueres til rett eller feil, for eksempel A>B. Se også <i>testbetingelse.</i>
condition combination coverage: See <i>multiple condition coverage.</i>	betingelses-kombinasjon dekning: Se <i>sammensatt betingelses dekning.</i>
condition combination testing: See <i>multiple condition testing.</i>	betingelses-kombinasjon testing: Se <i>sammensatt betingelses testing.</i>
condition coverage: The percentage of condition outcomes that have been exercised by a test suite. 100% condition coverage requires each single condition in every decision statement to be tested as True and False.	betingelsesdekning: Prosenten av betingelsesresultater (rett og feil) som har blitt utført av en testsuite. 100% betingelsesdekning krever at hver enkel betingelse i hver forgrening i programmet blir testet for sann og feil.
condition determination coverage: The percentage of all single condition outcomes that independently affect a decision outcome that have been exercised by a test case suite. 100% condition determination coverage implies 100% decision condition coverage.	Foreløpig ikke oversatt da dette bare brukes i ISTQB-syllabi på høyere nivå. Erklæring: Prosenten av alle betingelsesresultater, som hvert for seg berører et beslutningsresultat, som ble utført av en testsuite.
condition determination testing: A white box test design technique in which test cases are designed to execute single condition outcomes that independently affect a decision outcome.	Foreløpig ikke oversatt da dette bare brukes i ISTQB-syllabi på høyere nivå. Erklæring: En strukturell testdesignteknikk hvor testtilfeller lages for å utføre betingelsesresultater som hvert for seg har innflytelse på et beslutningsresultat.

condition testing: A white box test design technique in which test cases are designed to execute condition outcomes.	betingelsestesting: En strukturell (hvit boks) testdesignteknikk der testtilfeller blir konstruert for å utføre resultater av betingelser. Se også <i>strukturell testing</i> .
condition outcome: The evaluation of a condition to True or False.	betingelsesresultat: Evalueringen av en betingelse til sann eller feil
confidence test: See <i>smoke test</i> .	konfidenstest: Se (eng.) <i>smoke test</i>
configuration: The composition of a component or system as defined by the number, nature, and interconnections of its constituent parts.	konfigurasjon: Sammenstilling av en komponent eller et system definert ved antall, arten og forbindelsene av dens tilhørende deler.
configuration auditing: The function to check on the contents of libraries of configuration items, e.g. for standards compliance. [IEEE 610]	konfigurasjonsrevisjon: Funksjonen som skal sjekke innholdet av bibliotekene av konfigurasjonselementer, for eksempel for å kontrollere samsvar med standarder. [IEEE 610]
configuration control: An element of configuration management, consisting of the evaluation, co-ordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [IEEE 610]	konfigurasjonskontroll: Et element av konfigurasjonsstyringen, som består av å evaluere, koordinere, godkjenne eller ikke godkjenne, og implementere endringer til konfigurasjonselementer etter formell etablering av deres konfigurasjonsidentifikasjon. [IEEE 610]
configuration control board (CCB): A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. [IEEE 610]	konfigurasjonskontrollråd: En gruppe mennesker som er ansvarlige for å vurdere og evt. godkjenne foreslåtte endringer til konfigurasjonselementer, og for å sikre implementering av godkjente endringer. [IEEE 610]
configuration identification: An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. [IEEE 610]	konfigurasjonsidentifikasjon: Et element av konfigurasjonsstyringen, som består av å velge konfigurasjonselementene for et system og å bokføre deres funksjonelle og fysiske egenskaper i en teknisk dokumentasjon. [IEEE 610]
configuration item: An aggregation of hardware, software or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE 610]	konfigurasjonselement: En samling hardware og/eller software som er bestemt for konfigurasjonsstyring og blir behandlet som en enkel enhet i konfigurasjonsstyringsprosessen. [IEEE 610]

configuration management: A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [IEEE 610]	konfigurasjonsstyring: En disiplin der en bruker tekniske og administrative midler og overvåkning for å: Identifisere og dokumentere de funksjonelle og fysiske egenskaper av en konfigurasjonsenhet, for å styre endringer til disse egenskapene, for å bokføre og rapportere endringshåndtering og implementeringsstatus, og for å verifisere samsvar med spesifiserte krav. [IEEE 610]
configuration management tool: A tool that provides support for the identification and control of configuration items, their status over changes and versions, and the release of baselines consisting of configuration items.	konfigurasjonsstyringsverktøy: Verktøy som gir støtte ved identifikasjon og kontroll av konfigurasjonselementer, deres status gjennom endringer og over versjoner, samt frigivelser som består av et eller flere konfigurasjonselementer.
configuration testing: See <i>portability testing</i> .	konfigurasjonstesting: Se <i>portabilitetstesting</i>
confirmation testing: See <i>re-testing</i> .	<i>Se retest</i>
conformance testing: See <i>compliance testing</i> .	overensstemmelsestesting <i>Se samsvartesting</i>
consistency: The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a component or system. [IEEE 610]	konsistens / motsigelsesfrihet: Graden av enhetlighet, standardiserhet, og motsigelsesfrihet mellom dokumenter eller deler av en komponent eller et system. [IEEE 610]
continuous representation: A capability maturity model structure wherein capability levels provide a recommended order for approaching process improvement within specified process areas. [CMMI]	kontinuerlig representasjon: 1 - En struktur for en modenhetsmodell der modenhetsnivåer angir en anbefalt rekkefølge for å forbedre prosessene i et spesifisert prosessområde. 2 - En måte å bruke en modenhetsmodell på, der det gis en individuell evaluering av graden for hvert prosessområde, men ingen samlet grad. Se også (eng.) <i>staged representation</i> .
control flow: A sequence of events (paths) in the execution through a component or system.	kontrollflyt: En følge av hendelser (stier) i utførelsen gjennom en komponent eller et system.
control flow analysis: A form of static analysis based on a representation of sequences of events (paths) in the execution through a component or system.	kontrollflytanalyse: En form for statisk analyse basert på en representasjon av sekvenser av hendelser (stier) i utførelsen gjennom en komponent eller et system.
control flow graph: An abstract representation of all possible sequences of events (paths) in the execution through a component or system.	kontrollflytgraf: En abstrakt presentasjon av følgen av hendelser (stier) i utførelsen gjennom en komponent eller et system.
control flow path: See <i>path</i> .	sti (i programmet)

conversion testing: Testing of software used to convert data from existing systems for use in replacement systems.	konverteringstesting: Test av programvare som brukes for å konvertere data fra et eksisterende system for bruk i et system som erstatter dette.
cost of quality: The total costs incurred on quality activities and issues and often split into prevention costs, appraisal costs, internal failure costs and external failure costs.	kvalitetskostnad: Den totale summen av kostnader ved kvalitetsarbeid og feilsituasjoner. Deles ofte inn i forebyggende kostnad, kontrollkostnad og interne og eksterne feilkostnader.
COTS: Acronym for Commercial Off-The-Shelf software. See <i>off-the-shelf software</i> .	COTS: akronym for ”Commercial Off-The-Shelf” programvare; Umiddelbart tilgjengelig handelsvare fra lager (hylevare programvare).
coverage: The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite.	testdekning: Graden, uttrykt som prosent, til hvilken en spesifisert testdekningsenhet har blitt oppnådd av en testsuite. For eksempel i hvilken grad strukturen eller koden er utført under testen.
coverage analysis: Measurement of achieved coverage to a specified coverage item during test execution referring to predetermined criteria to determine whether additional testing is required and if so, which test cases are needed.	dekningsanalyse: Mål av oppnådd testdekning for en spesifisert testdekningsenhet under utførelsen av testen. Analysen refererer til et forhåndsbestemt kriterium for å slå fast om ekstra testing er nødvendig og hvis så er, hvilke testtilfeller som behøves.
coverage item: An entity or property used as a basis for test coverage, e.g. equivalence partitions or code statements.	testdekningsenhet: Et objekt eller en egenskap som blir brukt som basis for testdekning, for eksempel ekvivalensklasser eller programinstruksjoner.
coverage measurement tool: See <i>coverage tool</i> .	Se <i>testdekningsverktøy</i> .
coverage tool: A tool that provides objective measures of what structural elements, e.g. statements, branches have been exercised by the test suite.	testdekningsverktøy: Et verktøy som gir objektive mål av hva slags strukturelle elementer, for eksempel programinstruksjoner, forgreninger etc. som har blitt utført av en testsuite.
custom software: See <i>bespoke software</i> .	spesialutviklet programvare: Se (eng.) <i>bespoke software</i>
cyclomatic complexity: The number of independent paths through a program. Cyclomatic complexity is defined as: $L - N + 2P$, where: L = the number of edges/links in a graph N = the number of nodes in a graph P = the number of disconnected parts of the graph (e.g. a calling graph and a subroutine). [After McCabe]	syklomatisk kompleksitet: Antall uavhengige veier gjennom et program. Den er definert som: $L - N + 2P$, med: L: antall kanter i en graf N: antall noder i en graf P antall ikke sammenhengende deler av en graf (for eksempel kallgraf eller kontrollflygraf) [etter McCabe]

cyclomatic number: See <i>cyclomatic complexity</i> .	syklomatisk tall: Se <i>syklomatisk kompleksitet</i> .
--	---

D

daily build: a development activity where a complete system is compiled and linked every day (usually overnight), so that a consistent system is available at any time including all latest changes.	daglig bygging: En utvikling der et fullstendig system blir compilert og sammensatt hver dag (vanligvis over natt), slik at et fungerende system er tilgjengelig når som helst, inklusive de siste endringene.
data definition: An executable statement where a variable is assigned a value.	datadefinisjon: En utførbar programinstruksjon der en variabel får en verdi. (Anmerkning: Ikke en deklarasjon, men tilordningen av en verdi!)
data driven testing: A scripting technique that stores test input and expected results in a table or spreadsheet, so that a single control script can execute all of the tests in the table. Data driven testing is often used to support the application of test execution tools such as capture/playback tools. [Fewster and Graham] See also keyword driven testing.	datadrevet test: En teknikk for å programmere automatiske tester der inputverdier og forventede resultater blir lagret i tabeller eller regneark, slik at et enkelt testscript kan utføre alle testene i tabellen. Datadrevet test blir ofte brukt for å støtte bruk av testutførelsesverktøy (testdrivere eller roboter). [Fewster og Graham] Se også (eng.) <i>keyword driven testing</i> .
data flow: An abstract representation of the sequence and possible changes of the state of data objects, where the state of an object is any of: creation, usage, or destruction. [Beizer]	dataflyt: En abstrakt representasjon av sekvensen og mulig endring av tilstanden til dataobjekter, der tilstandene er initialisering, bruk eller sletting. [Beizer]
data flow analysis: A form of static analysis based on the definition and usage of variables.	dataflytanalyse: En form av statisk analyse basert på definisjon og bruk av variablene.
data flow coverage: The percentage of definition-use pairs that have been exercised by a test suite.	dataflytdekning: Prosenten av par for datadefinisjon og bruk (eng. definition-use-pairs) som har blitt utført av en samling testtilfelle.
data flow testing: A white box test design technique in which test cases are designed to execute definition and use pairs of variables.	dataflyttest: En strukturell testdesignteknikk der testtilfelle blir konstruert for å utføre par av definisjon og bruk av variable.
data integrity testing: See <i>database integrity testing</i> .	test av dataintegritet: Se <i>test av databaseintegritet</i> .

database integrity testing: Testing the methods and processes used to access and manage the data (base), to ensure access methods, processes and data rules function as expected and that during access to the database, data is not corrupted or unexpectedly deleted, updated or created.	test av databaseintegritet: Test av metodene og prosessene som brukes for adgang og management av en database eller data. Testingen skal kontrollere at tilgangsmetoder, prosesser og dataregler fungerer som forventet og at det under tilgang til data(basen) ikke forekommer korrupsjon av data eller at data uforventet blir ødelagt, oppdatert eller opprettet.
dead code: See <i>unreachable code</i> .	ubrukt ("død") kode: Kode som ikke kan utføres uten å manipulere programmets tekst eller verdier av variable. <i>Se også: ikke utførbar kode</i>
debugger: See <i>debugging tool</i> .	debugger: Se (eng.) <i>debugging tool</i> .
debugging: The process of finding, analyzing and removing the causes of failures in software.	debugging: Prosessen med å finne, analysere og ta bort årsaken av problemer med software (altså feil).
debugging tool: A tool used by programmers to reproduce failures, investigate the state of programs and find the corresponding defect. Debuggers enable programmers to execute programs step by step, to halt a program at any program statement and to set and examine program variables.	debuggingsverktøy: Et verktøy brukt av programmerere for å reprodusere feil, analysere tilstanden til programmet og finne den tilhørende feil. Slike verktøy gjør det mulig for programmerere å utføre programmet linje for linje eller instruksjon for instruksjon, stoppe programmet hvor som helst og å analysere og manipulere verdiene til variable.
decision: A program point at which the control flow has two or more alternative routes. A node with two or more links to separate branches.	beslutning: Et punkt i programmet der kontrollflyt har to eller flere alternative veier. En plass med to eller flere linker til forskjellige grener i programmet.
decision condition coverage: The percentage of all condition outcomes and decision outcomes that have been exercised by a test suite. 100% decision condition coverage implies both 100% condition coverage and 100% decision coverage.	Beslutnings-betingelsesdekning: Prosenten av alle betingelsesresultater og beslutningsresultater som er utført av en testsuite. 100% beslutnings-betingelsesdekning impliserer både 100% betingelsesdekning og 100% beslutningsdekning.
decision condition testing: A white box test design technique in which test cases are designed to execute condition outcomes and decision outcomes.	Beslutnings-betingelsestesting: En strukturell testdesignteknikk der testtilfelle blir konstruert for å utføre resultatene av betingelser og beslutninger.
decision coverage: The percentage of decision outcomes that have been exercised by a test suite. 100% decision coverage implies both 100% branch coverage and 100% statement coverage.	beslutningsdekning, forgreningsdekning: Prosenten av beslutningsresultater som har blitt utført av en testsuite. 100% beslutningsdekning impliserer både 100% forgreningsdekning og 100% programinstruksjonsdekning.

decision outcome: The result of a decision (which therefore determines the branches to be taken).	beslutningsresultat: Resultat av en beslutning i et program, som dermed bestemmer hvilken vei utførelsen fortsetter.
decision table: A table showing combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects), which can be used to design test cases.	beslutningstabell: En tabell som viser kombinasjoner av inputs og/eller hendelser (årsaker) med deres tilhørende output og/eller virkninger (aksjoner), som kan brukes for å konstruere testtilfelle.
decision table testing: A black box test design techniques in which test cases are designed to execute the combinations of inputs and/or stimuli (causes) shown in a decision table. [Veenendaal] See also <i>decision table</i> .	beslutningstabelltesting: En svart boks testdesignteknikk der testtilfeller blir konstruert for å utføre kombinasjoner av input og/eller hendelser (årsaker) som vises i en beslutningstabell [Veenendaal]
decision testing: A white box test design technique in which test cases are designed to execute decision outcomes.	beslutningstesting: En strukturell (hvit boks) testdesignteknikk der testtilfeller blir konstruert for å utføre beslutningsresultater.
defect: A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system.	defekt: En mangel eller feil i en komponent eller system som kan forårsake at komponenten eller systemet ikke oppfyller sin påkrevde funksjon, for eksempel en feil programinstruksjon, programlinje eller datadefinisjon. En defekt, hvis koden der den befinner seg utføres, kan forårsake et problem i komponenten eller systemet. Også kalt feil .
defect based technique: See <i>defect based test design technique</i> .	feilbasert teknikk: Se feilbasert testdesignteknikk
defect based test design technique: A procedure to derive and/or select test cases targeted at one or more defect categories, with tests being developed from what is known about the specific defect category. See also <i>defect taxonomy</i> .	feilbasert testdesignteknikk: Prosedyre for å utlede og/eller velge testtilfeller rettet inn mot en eller flere kategorier av feil, hvor testtilfeller utarbeides fra kunnskap om den spesifikke kategorien av feil. Se også <i>taksonomi av defekter</i> .
defect density: The number of defects identified in a component or system divided by the size of the component or system (expressed in standard measurement terms, e.g. lines-of-code, number of classes or function points).	feiltetthet: Antall defekter eller feil identifisert i en komponent eller et system dividert med størrelsen på komponenten eller systemet (uttrykt i standardiserte størrelser, for eksempel kodelinjer, antall klasser eller funksjonspunkter).
Defect Detection Percentage (DDP): the number of defects found by a test phase, divided by the number found by that test phase and any other means afterwards.	feilfinningsprosent (DDP): Antall feil funnet av en testfase eller et testnivå, dividert med antallet funnet (samme antall som før nevnt) pluss antall feil funnet senere.

defect management: The process of recognizing, investigating, taking action and disposing of defects. It involves recording defects, classifying them and identifying the impact. [IEEE 1044]	feilhåndtering/feilstyring: Prosessen med å oppdage, undersøke og behandle feil. Dette inkluderer å beskrive feil, klassifisere de og identifisere deres alvorlighet. [IEEE 1044]
defect management tool: A tool that facilitates the recording and status tracking of defects and changes. They often have workflow-oriented facilities to track and control the allocation, correction and re-testing of defects and provide reporting facilities. See also <i>incident management tool</i> .	feilhåndteringsverktøy: Et verktøy som gir hjelp til å registrere og følge opp feil og endringer. Slike verktøy har ofte muligheter til å følge opp arbeidsgangen med å rapportere feil og forfølge arbeidsgangen med å undersøke og rette disse, teste deres rettelse og rapportere omkring de. Se også (eng.) <i>incident management tool</i> .
defect masking: An occurrence in which one defect prevents the detection of another. [IEEE 610]	maskering av feil: En hendelse der en feil forhindrer at en annen feil blir oppdaget. [IEEE 610]
defect report: A document reporting on any flaw in a component or system that can cause the component or system to fail to perform its required function. [IEEE 829]	feilrapport: Et dokument som rapporterer hvilken som helst feil i en komponent eller et system. [IEEE 829]
defect taxonomy: A system of (hierarchical) categories designed to be a useful aid for reproducibly classifying defects.	taksonomi av defekter: Et (hierarkisk) system av kategorier med hensikt å være et effektivt hjelpemiddel for gjentagende klassifisering av defekter.
defect tracking tool: See <i>incident management tool</i> .	feilforvaltningsverktøy: Se (eng.) <i>incident management tool</i> .
definition-use pair: The association of the definition of a variable with the use of that variable. Variable uses include computational (e.g. multiplication) or to direct the execution of a path (“predicate” use).	definisjon-bruk-par: Definisjonen av en variabel sammen med dets neste bruk. Bruken kan være for å beregne et resultat (« beregningsbruk ») eller for å velge hvilken vei en skal utføre videre (« predikatbruk »)
deliverable: Any (work) product that must be delivered to someone other than the (work) product’s author.	leveranse: Et produkt som må bli levert til noen annen enn de som arbeider med utviklingen av produktet.
design-based testing: An approach to testing in which test cases are designed based on the architecture and/or detailed design of a component or system (e.g. tests of interfaces between components or systems).	designbasert testing: En testdesignmetode der testtilfeller blir konstruert basert på arkitekturen og/eller detaljert design av en komponent eller et system (for eksempel tester av grensesnitt mellom komponenter eller systemer).
desk checking: Testing of software or specification by manual simulation of its execution. See also <i>static analysis</i> .	skrivebordstesting: Test av programvare ved å simulere den manuelt. Se også <i>statisk analyse</i>

development testing: Formal or informal testing conducted during the implementation of a component or system, usually in the development environment by developers. [IEEE 610]	utviklingstesting: Formell eller informell testing som utføres under utvikling av en komponent eller et system, vanligvis i utviklingsomgivelsen og av utviklere. [IEEE 610]
deviation: See <i>incident</i> .	avvik: Se (eng.) <i>incident</i>
deviation report: See <i>incident report</i> .	avviksrapport: Se <i>feilrapport</i> .
dirty testing: See <i>negative testing</i> .	skitten testing: se <i>negativ testing</i> .
documentation testing: Testing the quality of the documentation, e.g. user guide or installation guide.	dokumentasjonstest: Test av kvaliteten på dokumentasjonen, for eksempel brukermanual eller installasjonsanvisning.
domain: The set from which valid input and/or output values can be selected.	domene: Mengden av gyldige input og/eller output verdier.
driver: A software component or test tool that replaces a component that takes care of the control and/or the calling of a component or system. [TMap]	driver: En programvarekomponent eller testverktøy som erstatter en komponent som styrer eller kaller opp en komponent eller et system under test. Også kalt testdriver . [TMap]
dynamic analysis: The process of evaluating behavior, e.g. memory performance, CPU usage, of a system or component during execution. [IEEE 610]	dynamisk analyse: Prosessen der en evaluerer testobjektets oppførsel, for eksempel bruk av minne, bruk av CPU under testutførelse. [IEEE 610]
dynamic analysis tool: A tool that provides run-time information on the state of the software code. These tools are most commonly used to identify unassigned pointers, check pointer arithmetic and to monitor the allocation, use and de-allocation of memory and to flag memory leaks.	dynamisk analyseverktøy: Verktøy som gir aktuell informasjon om tilstanden av programkoden under kjøring. Slike verktøy blir som oftest brukt for å identifisere pekere som ikke er tilordnet, kontrollere pekeraritmetikk og følge med allokering, bruk og frigivelse av minne, og avmerke minnelekkasjer.
dynamic comparison: Comparison of actual and expected results, performed while the software is being executed, for example by a test execution tool.	dynamisk sammenligning: Sammenligning av virkelig og forventet resultat, utført mens programvaren blir utført, for eksempel ved hjelp av testverktøy.
dynamic testing: Testing that involves the execution of the software of a component or system.	dynamisk test: Testutføring av en programvarekomponent eller et system.

E

efficiency: The capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions. [ISO 9126]	yteevne: Et programvareprodukts evne til å stille til rådighet adekvat ytelse, i forhold til mengden av ressurser når det blir brukt under spesifiserte betingelser. [ISO 9126]
efficiency testing: The process of testing to determine the efficiency of a software product.	test av yteevne: Prosessen der en tester et produkt for å bestemme dets effektivitet eller yteevne.
elementary comparison testing: A black box test design techniques in which test cases are designed to execute combinations of inputs using the concept of condition determination coverage. [TMap]	norsk term ikke definert (for spesiell)
emulator: A device, computer program, or system that accepts the same inputs and produces the same outputs as a given system. [IEEE 610] See also <i>simulator</i> .	emulator: Et apparat, program eller system som mottar den samme input og produserer den samme output som et gitt system. [IEEE 610] Se også <i>simulator</i> .
entry criteria: the set of generic and specific conditions for permitting a process to go forward with a defined task, e.g. test phase. The purpose of entry criteria is to prevent a task from starting which would entail more (wasted) effort compared to the effort needed to remove the failed entry criteria. [Gilb and Graham]	startkriterier: Mengden generiske og spesifikke betingelser for å tillate en prosess til å starte en definert oppgave, for eksempel testutførelsen. Formålet med startkriteriene er å forhindre at en oppgave blir startet når dette fører til mer bortkastet arbeid, sammenlignet med arbeidet med å gjøre noe med de feilede startkriteriene. [Gilb og Graham]
entry point: The first executable statement within a component.	startpunkt: Den første utførbare programinstruksjonen i en komponent.
equivalence class: See <i>equivalence partition</i> .	ekvivalensklasse.
equivalence partition: A portion of an input or output domain for which the behavior of a component or system is assumed to be the same, based on the specification.	ekvivalensklasse: En del av en input- eller outputdomene for hvilken komponent eller systemet antas å ville oppføre seg likedan, basert på kunnskap om spesifikasjonen.
equivalence partition coverage: The percentage of equivalence partitions that have been exercised by a test suite.	ekvivalensklassedekning: Prosenten av ekvivalensklassene som er blitt utført av en testsuite.
equivalence partitioning: A black box test design technique in which test cases are designed to execute representatives from equivalence partitions. In principle test cases are designed to cover each partition at least once.	ekvivalensklasseinndeling: En svart boks testdesignteknikk der testtilfeller blir konstruert for å utføre representanter for ekvivalensklasser. I prinsipp blir testtilfellene konstruert for å utføre hver klasse minst en gang..

error: A human action that produces an incorrect result. [IEEE 610]	feil: En menneskelig handling som fører til et galt resultat. [IEEE 610] På norsk ofte brukt for å betegne programfeil, dvs. at et program ikke oppfører seg som spesifisert eller forventet. Se også <i>avvik</i> .
error guessing: A test design technique where the experience of the tester is used to anticipate what defects might be present in the component or system under test as a result of errors made, and to design tests specifically to expose them.	feilgjetting: En teknikk for testdesign der erfaringen til testeren blir brukt for å forutse hva slags feil som kan være tilstede i komponenten eller systemet under test som et resultat av feil som er gjort, og for å lage tester som finner disse.
error seeding: See <i>fault seeding</i>	introduksjon av kunstige feil: Se (eng.) <i>fault seeding</i> .
error seeding tool: See <i>fault seeding tool</i> .	verktøy for introduksjon av kunstige feil: Se (eng.) <i>fault seeding tool</i> .
error tolerance: The ability of a system or component to continue normal operation despite the presence of erroneous inputs. [IEEE 610].	feiltoleranse: Systemets eller komponentens evne til å fortsette normal drift selv om feilaktige input blir gitt. [IEEE 610]
establishing (IDEAL): The phase within the IDEAL model where the specifics of how one will reach its destination are planned. The establishing phase consists of the activities: set priorities, develop approach and plan actions. See also <i>IDEAL</i> .	etablering (IDEAL): En fase i IDEAL der man planlegger hvordan man skal nå sine mål. Fasen består i å prioritere, utvikle en fremgangsmåte og planlegge handlinger. Se også <i>IDEAL</i> .
evaluation: See <i>testing</i> .	evaluering: se <i>testing</i> .
exception handling: Behavior of a component or system in response to erroneous input, from either a human user or from another component or system, or to an internal failure.	unntakshåndtering eller feilhåndtering: Oppførselen til en komponent eller et system som svar på feil input fra enten en menneskelig bruker eller fra en annen komponent eller system, eller pga. en intern feil.
executable statement: A statement which, when compiled, is translated into object code, and which will be executed procedurally when the program is running and may perform an action on data.	utførbar programinstruksjon: En programinstruksjon som, når den er compilert, blir oversatt til objektkode, og som vil bli utført når programmet kjører og som kan utføre en aksjon på data.
exercised: A program element is said to be exercised by a test case when the input value causes the execution of that element, such as a statement, decision, or other structural element.	utført: Et programelement er utført av et testtilfelle når dens inputverdier sørger for at elementet blir utført. Elementet kan være en programinstruksjon, programlinje, beslutning eller et annet strukturelement.
exhaustive testing: A test approach in which the test suite comprises all combinations of input values and preconditions.	fullstendig testing: En framgangsmåte der testsuiten omfatter alle kombinasjoner av inputverdier og forbetingelser.

exit criteria: The set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed. The purpose of exit criteria is to prevent a task from being considered completed when there are still outstanding parts of the task which have not been finished. Exit criteria are used by testing to report against and to plan when to stop testing. [Gilb and Graham]	sluttkriterier: Generiske og spesifikke betingelser for å tillate at en prosess blir offisielt avsluttet og som de ulike interessentene er blitt enige om. Målet med sluttkriterier er å forhindre at en oppgave blir ansett som avsluttet når det fortsatt finnes utestående deler av oppgaven som ikke er avsluttet. Sluttkriterier blir brukt i testingen for å rapportere mot og for å planlegge når en skal stoppe testingen. [Gilb og Graham]
exit point: The last executable statement within a component.	sluttpunkt: Siste utførbare programinstruksjon i en komponent.
expected outcome: See <i>expected result</i> .	forventet resultat.
expected result: The behavior predicted by the specification, or another source, of the component or system under specified conditions.	forventet resultat: Oppførselen som blir forutsagt av spesifikasjonen eller en annen kilde, når en komponent eller et system blir utført under spesifiserte betingelser.
experience-based technique: See <i>experience-based test design technique</i> .	erfaringsbasert teknikk: Se <i>erfaringsbasert testdesignteknikk</i> .
experience-based test design technique: Procedure to derive and/or select test cases based the tester's experience, knowledge and intuition.	erfaringsbasert testdesignteknikk: Prosedyre for å utlede og/eller velge testtilfeller basert på testerens erfaring, kunnskap og intuisjon.

F

fail: A test is deemed to fail if its actual result does not match its expected result.	å feile: En test blir kalt feilet hvis det aktuelle resultat avviker fra det forventede resultat.
failure: Deviation of the component or system from its expected delivery, service or result. [Fenton]	problem: Også kalt feil eller avvik: Fravikelse av en komponent eller et system fra dens forventede resultat, tjeneste eller output. [Fenton]; Hendelse der systemet eller komponenten ikke klarer å utføre sin spesifiserte funksjon eller å utføre den innenfor de begrensninger som er spesifisert. [IEEE 829]
failure mode: The physical or functional manifestation of a failure. For example, a system in failure mode may be characterized by slow operation, incorrect outputs, or complete termination of execution. [IEEE 610]	feilmodus: Den fysiske eller funksjonelle manifestasjon av et problem. Et system i feilmodus kan f.eks. være unormalt tregt, gi feil utdata, eller kjøringen kan være fullstendig terminert. [IEEE 610]

Failure Mode and Effect Analysis (FMEA): A systematic approach to risk identification and analysis of identifying possible modes of failure and attempting to prevent their occurrence. See also <i>Failure Mode, Effect and Criticality Analysis (FMECA)</i> .	Failure Mode and Effect Analysis (FMEA): Systematisk framgangsmåte for å identifisere risiko og for å analysere mulige måter for et program å feile på og for å forbygge deres opptreden. Se også (eng.) <i>Failure Mode, Effect and and Criticality Analysis</i> .
Failure Mode, Effect and Criticality Analysis (FMECA): An extension of FMEA, as in addition to the basic FMEA, it includes a criticality analysis, which is used to chart the probability of failure modes against the severity of their consequences. The result highlights failure modes with relatively high probability and severity of consequences, allowing remedial effort to be directed where it will produce the greatest value. See also <i>Failure Mode and Effect Analysis (FMEA)</i> .	Failure Mode, Effect and Criticality Analysis (FMECA): En utvidelse av FMEA, som i tillegg inkluderer analyse av kritikalitet, brukt til å kartlegge sannsynligheten for avvik mot alvorlighetsgraden av konsekvensene. Resultatet peker ut feilområder med relativt høy sannsynlighet og alvorlighetsgrad av konsekvensene, og gjør det mulig å rette innsatsen mot områdene som gir størst verdi. Se også <i>Failure Mode and Effect Analysis</i> .
failure rate: The ratio of the number of failures of a given category to a given unit of measure, e.g. failures per unit of time, failures per number of transactions, failures per number of computer runs. [IEEE 610]	feilrate: Forholdstall mellom antall feil av en gitt kategori og en annen måleenhet. For eksempel feil pr. tidsenhet, feil pr. antall transaksjoner, feil pr. antall utførelser. [IEEE 610]
false-fail result: A test result in wich a defect is reported although no such defect actually exist in the test object.	feilaktig underkjent resultat: Testresultat som feilaktig fører til en feilrapport.
false-negative result: See <i>false-fail result</i> .	feilaktig negativt resultat: Se (eng.) <i>false-fail result</i>
false-pass result: A test result, which fails to identify the presence of a defect that is actually present in the test object.	feilaktig godkjent resultat: Testresultat som feilaktig godkjenner en test, og mislykkes i å identifisere eksisterende feil i testobjektet.
false-positive result: See <i>false-pass result</i> .	feilaktig positivt resultat: Se (eng.) <i>false-pass result</i>
fault: See <i>defect</i> .	defekt: se <i>feil</i> , se (eng.) <i>bug, defect</i> . Resultat av en feil gjort av en utvikler i et program. En defekt kan forårsake et problem eller et avvik. Kalles også mangel . [IEEE 829]
fault attack: See <i>attack</i>	Se (eng.) <i>attack</i>
fault density: See <i>defect density</i> .	feiltetthet: se (eng.) <i>defect density</i>
Fault Detection Percentage (FDP): See <i>Defect Detection Percentage (DDP)</i> .	feilfinningsprosent: Se (eng.) <i>defect detection percentage</i> .
fault masking: See <i>defect masking</i> .	feilmasking: Se (eng.) <i>defect masking</i>

fault seeding: The process of intentionally adding known defects to those already in the component or system for the purpose of monitoring the rate of detection and removal, and estimating the number of remaining defects. [IEEE 610]	feilsåing, introduksjon av kunstige feil: Prosessen der en med vilje legger inn kunstige feil i et program med den hensikt å måle hvor mange av disse som blir funnet under en senere evaluering eller test. Dette for å estimere gjenværende antall feil.
fault seeding tool: A tool for seeding (i.e. intentionally inserting) faults in a component or system.	feilsåingsverktøy: Verktøy for å legge inn kunstige defekter i en komponent eller et system.
fault tolerance: The capability of the software product to maintain a specified level of performance in cases of software faults (defects) or of infringement of its specified interface. [ISO 9126] See also <i>reliability, robustness</i> .	feiltoleranse: Evnen til et programvareprodukt til å opprettholde et spesifisert nivå av funksjon og ytelse når feil er tilstede eller når dens spesifiserte grensesnitt ikke fungerer som forutsatt. [ISO 9126]
Fault Tree Analysis (FTA): A technique used to analyze the causes of faults (defects). The technique visually models how logical relationships between failures, human errors, and external events can combine to cause specific faults to disclose.	feiltreanalyse: En teknikk som brukes til å analysere de mulige virkningene av feil (defekter). Viser hvordan logiske forbindelser mellom avvik og menneskelige feil kan samvirke slik at spesifikke feil oppstår.
feasible path: A path for which a set of input values and preconditions exists which causes it to be executed.	utførbar sti: En sti eller vei gjennom et program som det eksisterer inputverdier og forbetingelser til som fører til at den blir utført.
feature: An attribute of a component or system specified or implied by requirements documentation (for example reliability, usability or design constraints). [IEEE 1008]	kjennetegn: En egenskap ved et system eller en komponent som er spesifisert eller implisert av kravspesifikasjonen (for eksempel pålitelighet, brukbarhet eller design restriksjoner). [IEEE 1008]
field testing: See <i>beta testing</i> .	felttest: se <i>beta test</i> .
finite state machine: A computational model consisting of a finite number of states and transitions between those states, possibly with accompanying actions. [IEEE 610]	tilstandsmaskin: En modell som består av et begrenset antall tilstander og overganger mellom disse, muligens med tilhørende aksjoner. [IEEE 610]
finite state testing: See <i>state transition testing</i> .	tilstandsbasert testing: Se (eng.) <i>state transition testing</i> .
formal review: A review characterized by documented procedures and requirements, e.g. inspection.	formell granskning: En granskning karakterisert ved dokumenterte prosedyrer og krav, for eksempel inspeksjon. IEEE Standard 1028 beskriver formelle granskninger. Også kalt formell gjennomgang.
frozen test basis: A test basis document that can only be amended by a formal change control process. See also <i>baseline</i> .	frosset testbasis: Et dokument som er basis for testen som bare kan endres gjennom en formell endringshåndteringsprosess. Se også <i>basislinje</i> .

Function Point Analysis (FPA): Method aiming to measure the size of the functionality of an information system. The measurement is independent of the technology. This measurement may be used as a basis for the measurement of productivity, the estimation of the needed resources, and project control.	funksjonspunktanalyse (FPA): En metode som er bestemt for å måle omfanget av funksjonalitet av et informasjonssystem, Målingen er uavhengig av teknologi. Denne målingen kan brukes som en basis for måling av produktivitet, estimering av nødvendige ressurser, og prosjektstyring.
functional integration: An integration approach that combines the components or systems for the purpose of getting a basic functionality working early. See also <i>integration testing</i> .	funksjonell integrasjon: En integrasjonsmetode som kombinerer komponentene eller systemene med den hensikt å få basisfunksjonaliteten til å virke tidlig. Se også <i>integrasjonstesting</i> .
functional requirement: A requirement that specifies a function that a component or system must perform. [IEEE 610]	funksjonelt krav: Et krav som spesifiserer en funksjon som et system eller en komponent må utføre. [IEEE 610]
functional test design technique: Procedure to derive and/or select test cases based on an analysis of the specification of the functionality of a component or system without reference to its internal structure. See also <i>black box test design technique</i> .	funksjonell testdesignteknikk: Dokumentert prosedyre for å avlede og velge ut testtilfelle basert på en analyse av spesifikasjonen av funksjonaliteten til en komponent eller et system uten å se på den interne strukturen. Se også <i>svart boks testing</i> (eng. <i>black box testing</i>).
functional testing: Testing based on an analysis of the specification of the functionality of a component or system. See also <i>black box testing</i> .	funksjonell testing: Testing basert på en analyse av spesifikasjonen av funksjonaliteten av en komponent eller system. Se også <i>svart boks testing</i> (eng. <i>black box testing</i>).
functionality: The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions. [ISO 9126]	funksjonalitet: Evnen til et programvareprodukt til å muliggjøre funksjoner som oppfyller nedskrevne eller impliserte behov når programmet brukes under spesifiserte betingelser. [ISO 9126]
functionality testing: The process of testing to determine the functionality of a software product.	funksjonstesting: Prosessen med å teste for å bestemme funksjonaliteten til et programvareprodukt.

G

glass box testing: See <i>white box testing</i> .	glassbokstesting: se <i>strukturell testing</i> , <i>hvit boks testing</i>
--	---

H

<p>hazard analysis: A technique used to characterize the elements of risk. The result of a hazard analysis will drive the methods used for development and testing of a system. See also <i>risk analysis</i>.</p>	<p>fareanalyse: En teknikk for å beskrive risikoelementene. Resultatet vil påvirke metodene for utvikling og testing av et system.</p>
<p>heuristic evaluation: A static usability test technique to determine the compliance of a user interface with recognized usability principles (the so-called “heuristics”).</p>	<p>heuristisk evaluering: En statisk testteknikk for brukbarhet for å undersøke samsvar mellom et brukergrensesnitt og anerkjente prinsipper for brukbarhet (brukervennlighet) (de så kalte ”heuristikker”, tommelfingerregler).</p>
<p>high level test case: A test case without concrete (implementation level) values for input data and expected results. Logical operators are used; instances of the actual values are not yet defined and/or available. See also <i>low level test case</i>.</p>	<p>høynivå testtilfelle: Et testtilfelle uten konkrete verdier for input data og forventede resultater. Når en velger konkrete verdier blir det til lavnivå testtilfelle eller konkret testtilfelle. Også kalt ”abstrakt testtilfelle”.</p> <p>Eksempel:</p> <ul style="list-style-type: none"> - ”Søk etter en bussrute med rette input”. Dette blir senere til det konkrete testtilfelle: - ”Fra Bergen”, ”Til Oslo”, ”i dag”.
<p>horizontal traceability: The tracing of requirements for a test level through the layers of test documentation (e.g. test plan, test design specification, test case specification and test procedure specification or test script).</p>	<p>horisontal sporbarhet: Sporing av krav for et testnivå gjennom lagene i testdokumentasjonen (for eksempel testplan, testdesign, testtilfelle og testprosedyre eller testscript).</p>
<p>hyperlink: A pointer within a web page that leads to other web pages.</p>	<p>hyperlenke: En peker innenfor en webside eller et dokument som leder til andre websider eller dokumenter.</p>
<p>hyperlink tool: A tool used to check that no broken hyperlinks are present on a web site.</p>	<p>hyperlenkeverktøy: Verktøy som sjekker at det ikke er noen brutte hyperlenker på en webside.</p>

I

<p>IDEAL: An organizational improvement model that serves as a roadmap for initiating, planning, and implementing improvement actions. The IDEAL model is named for the five phases it describes: initiating, diagnosing, establishing, acting and learning</p>	<p>IDEAL: En organisatorisk modell som fungerer som en veiviser for å initiere, planlegge og implementere prosessforbedringer. IDEAL-modellen har navnet sitt etter de engelske ordene ”initiating, diagnosing, establishing, acting and learning”.</p>
<p>impact analysis: The assessment of change to the layers of development documentation, test documentation and components, in order to implement a given change to specified requirements.</p>	<p>konsekvensanalyse: Bedømmelse av nødvendig endring til nivåene i utviklingsdokumentasjonen, testdokumentasjonen og komponentene, for å implementere en gitt endring i henhold til spesifiserte krav.</p>
<p>incident: Any event occurring that requires investigation. [IEEE 1008]</p>	<p>hendelse: Enhver hendelse som skjer under testutførelse som krever nærmere undersøkelse. [IEEE 1008]</p> <p>Ethvert avvik mellom faktisk og forventet resultat, eller forhold som representerer et problem under testgjennomføringen, dokumenteres som testobservasjoner. Nærmere undersøkelser kan føre til at observasjonen klassifiseres som feil eller defekt.</p>
<p>incident logging: Recording the details of any incident that occurred, e.g. during testing.</p>	<p>hendelsesregistrering (også feilregistrering): Registrering av alle hendelser som krever nærmere undersøkelse, f.eks. under testing.</p>
<p>incident management: The process of recognizing, investigating, taking action and disposing of incidents. It involves logging incidents, classifying them and identifying the impact. [IEEE 1044]</p>	<p>feilhåndtering: Prosessen med å oppdage, undersøke, behandle og disponere feil. Dette inkluderer å beskrive feil, klassifisere de og identifisere deres alvor. [IEEE 1044]</p>
<p>incident management tool: A tool that facilitates the recording and status tracking of incidents. They often have workflow-oriented facilities to track and control the allocation, correction and re-testing of incidents and provide reporting facilities. See also <i>defect management tool</i>.</p>	<p>feilhåndteringsverktøy: Et verktøy som hjelper til å holde styr på hendelser og feil oppdaget under testing (og annetsteds). Slike verktøy har ofte muligheter å følge opp arbeidsgangen med å rapportere feil og forfølge arbeidsgangen med å undersøke og rette disse, teste deres rettelse og rapportere omkring de.</p>
<p>incident report: A document reporting on any event that occurred, e.g. during the testing, which requires investigation. [IEEE 829]</p>	<p>problemrapport: også hendelsesrapport. Et dokument som rapporterer en hendelse som krever undersøkelse. [IEEE 829]</p>

<p>incremental development model: A development life cycle where a project is broken into a series of increments, each of which delivers a portion of the functionality in the overall project requirements. The requirements are prioritized and delivered in priority order in the appropriate increment. In some (but not all) versions of this life cycle model, each subproject follows a ‘mini V-model’ with its own design, coding and testing phases.</p>	<p>trinnvis (inkrementell) utviklingsmodell: En utviklingsmodell der et prosjekt blir delt i en serie av del-leveranser (inkremitter). Hver av disse leverer en del av funksjonaliteten av de totale prosjektkrav. Kravene blir prioritert og del-leveransene implementerer og leverer disse i prioritert rekkefølgen. I noen (men ikke alle) versjoner av denne livssyklusmodellen følger hvert delprosjekt sin egen ”mini-V-modell” med sin egen design-, implementasjons- og testfase.¹</p>
<p>incremental testing: Testing where components or systems are integrated and tested one or some at a time, until all the components or systems are integrated and tested.</p>	<p>inkrementell testing: Tester der komponentene eller systemene blir integrert og testet en eller noen av gangen, inntil alle komponenter eller systemer er integrert og testet.</p>
<p>independence of testing: Separation of responsibilities, which encourages the accomplishment of objective testing. [DO-178B]</p>	<p>testingens uavhengighet: Atskillelse av ansvar som bidrar til at testingen blir objektiv. [DO-178B]</p>
<p>infeasible path: A path that cannot be exercised by any set of possible input values.</p>	<p>umulig sti (eller vei): En vei gjennom programmet ikke kan utføres med mulige inputverdier.</p>
<p>informal review: A review not based on a formal (documented) procedure.</p>	<p>uformell granskning: En granskning som ikke baserer seg på en formell (dokumentert) prosedyre.</p>

¹ A common mistake is to consider "iterative" and "incremental" as synonyms, which they are not. In software/systems development, however, they typically go hand in hand. The basic idea is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing the developer to take advantage of what was learned during the development of earlier portions or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. Med følgende referanser: This page is extensively based on: Dr. Alistair Cockburn (May 2008). "Using Both Incremental and Iterative Development". STSC CrossTalk (USAF Software Technology Support Center) 21 (5): 27?30. ISSN d0000089. <http://www.stsc.hill.af.mil/crosstalk/2008/05/0805Cockburn.html>. Retrieved 2009-01-10. Craig Larman, Victor R. Basili (June 2003). "Iterative and Incremental Development: A Brief History". IEEE Computer (IEEE Computer Society) 36 (6): 47?56. doi:10.1109/MC.2003.1204375. ISSN 0018-9162. <http://c2.com/cgi/wiki/wiki?HistoryOfIterative>. Retrieved 2009-01-10.

Norsk forklaring av begrepene:

- (1) Inkrementell: vi utvikler deler av funksjonalitet (funksjonalitet øker over tid)
- (2) Iterativt: vi forbedrer hva vi har utviklet tidligere.

initiating (IDEAL): The phase within the IDEAL model where the groundwork is laid for a successful improvement effort. The initiating phase consists of the activities: set context, build sponsorship and charter infrastructure. See also <i>IDEAL</i> .	initiering (IDEAL): Fasen i IDEAL der grunnlaget for en vellykket forbedringsjobb gjøres. I denne fasen å bestemmes konteksten for forbedringen og det tilrettelegges for forbedringsprosessen. Se også <i>IDEAL</i> .
input: A variable (whether stored within a component or outside) that is read by a component.	inndata: En variabel (lagret i eller utenfor en komponent) som leses av en komponent.
input domain: The set from which valid input values can be selected. See also <i>domain</i> .	inndata domene: Mengden gyldige inndataverdier. Se også <i>domene</i> .
input value: An instance of an input. See also <i>input</i> .	inndataverdi: Et konkret eksempel for inndata.
inspection: A type of peer review that relies on visual examination of documents to detect defects, e.g. violations of development standards and non-conformance to higher-level documentation. The most formal review technique and therefore always based on a documented procedure. [IEEE 610, IEEE 1028] See also <i>peer review</i> .	inspeksjon: En type granskning som baserer seg på visuell undersøkelse av dokumenter for å finne feil, for eksempel avvik fra standarder eller spesifikasjoner eller avvik fra høyere nivå dokumentasjon. Den mest formelle granskningsteknikken. Alltid basert på dokumenterte prosedyrer. [IEEE 610, IEEE 1028]
inspection leader: See <i>moderator</i> .	inspeksjonsleder: Se <i>moderator</i>
inspector: See <i>reviewer</i> .	inspektør: En person som analyserer et dokument for å finne feil.
installability: The capability of the software product to be installed in a specified environment [ISO 9126]. See also <i>portability</i> .	installerbarhet: Et programvareprodukts evne til å bli installert i en spesifisert omgivelse [ISO 9126] Se også <i>portabilitet</i> .
installability testing: The process of testing the installability of a software product. See also <i>portability testing</i> .	installasjonstesting: Prosessen med å teste installerbarheten av et programvareprodukt. Se også <i>portabilitetstesting</i> .
installation guide: Supplied instructions on any suitable media, which guides the installer through the installation process. This may be a manual guide, step-by-step procedure, installation wizard, or any other similar process description.	installasjonsanvisning: Medleverte instruksjoner på et hvilket som helst medium som skal hjelpe den som installerer gjennom installasjonsprosessen. Dette kan være direkte anvisninger, en trinnvis prosedyre, en automatisert installasjonsprosedyre ("wizard") eller en liknende prosessbeskrivelse.
installation wizard: Supplied software on any suitable media, which leads the installer through the installation process. It normally runs the installation process, provides feedback on installation results, and prompts for options.	installasjonsprosedyre: Medlevert programvare på et hvilket som helst medium som leder den som installerer gjennom installasjonsprosessen. Normalt utfører denne installasjonsprosessen, gir tilbakemelding om dens resultater og spør om valg.

instrumentation: The insertion of additional code into the program in order to collect information about program behavior during execution, e.g. for measuring code coverage.	instrumentering: Innlegging av ekstra kode i et program for å samle informasjon om programmets oppførsel under utføring eller for å samle data om utføringen.
instrumenter: A software tool used to carry out instrumentation.	instrumenterer: Et verktøy som utfører instrumentering.
intake test: A special instance of a smoke test to decide if the component or system is ready for detailed and further testing. An intake test is typically carried out at the start of the test execution phase. See also <i>smoke test</i> .	inntakstest: En spesiell type test som er til for å finne ut om komponenten eller systemet er klar for detaljert og videre testing. Denne testen blir typisk utført ved starten av en tests eksekveringsfase. Se også (eng.) <i>smoke test</i> .
integration: The process of combining components or systems into larger assemblies.	integrasjon: Prosessen der en setter sammen komponenter eller systemer til større enheter.
integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also <i>component integration testing</i> , <i>system integration testing</i> .	integrasjonstest: Test som utføres for å finne feil i grensesnitt og samspillet mellom integrerte komponenter eller systemer. Se også <i>komponentintegrasjonstesting</i> og <i>systemintegrasjonstesting</i> .
integration testing in the large: See <i>system integration testing</i> .	systemintegrasjonstesting
integration testing in the small: See <i>component integration testing</i> .	komponentintegrasjonstesting
interface testing: An integration test type that is concerned with testing the interfaces between components or systems.	grensesnitt-testing: Se <i>integrasjonstesting</i> .
interoperability: The capability of the software product to interact with one or more specified components or systems. [ISO 9126] See also <i>functionality</i> .	interoperabilitet: Et programvareprodukts evne til å samarbeide med en eller flere spesifiserte komponenter eller systemer. [ISO 9126] Se også <i>funksjonalitet</i> .
interoperability testing: The process of testing to determine the interoperability of a software product. See also <i>functionality testing</i> .	interoperabilitetstest: Prosessen der en tester for å måle samarbeidsevnen av et programvareprodukt. Se også <i>funksjonalitetstesting</i> , <i>funksjonstesting</i> .
invalid testing: Testing using input values that should be rejected by the component or system. See also <i>error tolerance</i> , <i>negative testing</i> .	negativ testing: Testing som bruker verdier som burde bli avvist av komponenten eller systemet under test. Se også <i>feiltoleranse</i> , <i>negativ testing</i> .
Ishikawa diagram: See <i>cause/effect diagram</i> .	Ishikawa diagram: Se <i>årsak-virkningsgraf</i> .
isolation testing: Testing of individual components in isolation from surrounding components, with surrounding components being simulated by stubs and drivers, if needed.	isolert testing: Test av individuelle komponenter isolert fra deres omgivelse, dvs. andre komponenter, der disse komponentene blir simulert av stubber og drivere, hvis nødvendig.
item transmittal report: See <i>release note</i> .	frigivelsesrapport: Se (eng.) <i>release note</i>

<p>iterative development model: A development life cycle where a project is broken into a, usually large, numbers of iterations. An iteration is a complete development loop resulting in a release (internal or external) of an executable product, a subset of the final product under development, which grows from iteration to iteration to become the final product.</p>	<p>iterativ utviklingsmodell: En utviklingsmodell hvor et prosjekt deles opp i iterasjoner ("gjentakelser"). En iterasjon er en komplett utviklingssyklus hvor resultatet er en (intern eller eksternt) leveranse av et kjørbart produkt som inneholder deler av funksjonaliteten som skal utvikles, og systemet blir mer og mer komplett for hver iterasjon inntil endelig ferdig produkt.</p>
---	--

J

Ingen aktuelle engelske terminologier.

K

<p>key performance indicator: See <i>performance indicator</i>.</p>	<p>nøkkelindikator for ytelsen: Se (eng.) <i>performance indicator</i>.</p>
<p>keyword driven testing: A scripting technique that uses data files to contain not only test data and expected results, but also keywords related to the application being tested. The keywords are interpreted by special supporting scripts that are called by the control script for the test. See also <i>data driven testing</i>.</p>	<p>nøkkelord drevet testing: En teknikk for automatisering av testen som bruker datafiler som inneholder ikke bare test data og forventede resultater, men også nøkkelord som er relatert til applikasjonen under test. Nøkkelordene blir interpretert som kommandoer av spesielle scripter som kalles opp av testautomatiseringsprogrammer. Se også <i>datadrevet testing</i>.</p>

L

<p>LCSAJ: A Linear Code Sequence And Jump, consisting of the following three items (conventionally identified by line numbers in a source code listing): the start of the linear sequence of executable statements, the end of the linear sequence, and the target line to which control flow is transferred at the end of the linear sequence.</p>	<p>LCSAJ: Et kompleks testdekningskriterium som støttes av et engelsk produsert testdekningsverktøy og ikke er mye i bruk ellers. Se engelsk definisjon.</p>
--	---

LCSAJ coverage: The percentage of LCSAJs of a component that have been exercised by a test suite. 100% LCSAJ coverage implies 100% decision coverage.	LCSAJ-dekning: Se <i>LCSAJ</i>
LCSAJ testing: A white box test design technique in which test cases are designed to execute LCSAJs.	LCSAJ-testing: En strukturell testdesignteknikk som har som mål å utføre alle LCSAJ i et program.
lead-assessor: The person who leads an assessment. In some cases, for instance CMMi and TMMi when formal assessments are conducted, the lead-assessor must be accredited and formally trained.	ledende revisor: Personen som leder en undersøkelse. Ved kontroll av kvalitetssystem eller undersøkelse av en organisasjons modenhet er denne personen vanligvis formelt utdannet, sertifisert og akkreditert.
learnability: The capability of the software product to enable the user to learn its application. [ISO 9126] See also <i>usability</i> .	lærbarhet: Et programvareprodukts evne til å gjøre det mulig for brukeren å lære dets bruk [ISO 9126] Se også <i>brukbarhet</i> .
learning (IDEAL): The phase within the IDEAL model where one learns from experiences and improves one's ability to adopt new processes and technologies in the future. The learning phase consists of the activities: analyze and validate, and propose future actions. See also <i>IDEAL</i> .	læring (IDEAL modellen): Fasen i IDEAL modellen der en lærer av erfaringer og forbedrer ens evne til å ta i bruk nye prosesser og teknologier i framtiden. Lærefasen består av følgende aktiviteter: Analysere og validere og foreslår framtidige handlinger. Se også <i>IDEAL</i> .
level test plan: A test plan that typically addresses one test level. See also <i>test plan</i> .	testplan for et testnivå: En testplan som beskriver et testnivå.
lifecycle model: A partitioning of the life of a product or project into phases. [CMMI] See also <i>software lifecycle</i> .	livssyklusmodell: En oppdeling av livet til et produkt eller et prosjekt i faser. Se også <i>programvare-livssyklus</i> .
link testing: See <i>component integration testing</i> .	linktesting: Se <i>komponentintegrasjonstesting</i>
load profile: A specification of the activity which a component or system being tested may experience in production. A load profile consists of a designated number of virtual users who process a defined set of transactions in a specified time period and according to a predefined operational profile. See also <i>operational profile</i> .	belastningsprofil: Spesifikasjon av hvilken aktivitet en komponent eller et system som testes kan bli utsatt for i produksjon. En belastningsprofil består av et bestemt antall virtuelle brukere som utfører et definert sett transaksjoner innenfor en spesifisert tidsperiode ifølge en forhåndsdefinert bruksprofil. Se også <i>bruksprofil</i> .
load testing: A type of performance testing conducted to evaluate the behavior of a component or system with increasing load, e.g. numbers of parallel users and/or numbers of transactions to determine what load can be handled by the component or system. See also performance testing, <i>stress testing</i>	belastningstest: En type ytelse test som måler oppførselen av en komponent eller et system med økende belastning, for eksempel antall parallelle brukere og/eller antall transaksjoner for å bestemme hva slags belastning testobjektet kan håndtere.
load testing tool: See <i>performance testing tool</i> .	verktøy for ytelsestest

logic-coverage testing: See <i>white box testing</i> . [Myers]	hvit boks testing
logic-driven testing: See <i>white box testing</i> .	logikkdrevet testing: Testmetode som har som mål å dekke logikken i programmet som testes. (Avviker fra engelsk definisjon)
logical test case: See <i>high-level test case</i> .	logisk testtilfelle: Se <i>høynivå testtilfelle</i> eller <i>abstrakt testtilfelle</i> .
low level test case: A test case with concrete (implementation level) values for input data and expected results. Logical operators from high-level test cases are replaced by actual values that correspond to the objectives of the logical operators. See also <i>high-level test case</i> .	lavnivå testtilfelle: Et testtilfelle med konkrete verdier for input data og forventede resultater. Også <i>konkret testtilfelle</i> .

M

maintainability: The ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment. [ISO 9126]	vedlikeholdbarhet: Egenskapen som uttrykker hvor lett det er for et programvareprodukt å bli endret for å rette feil, møte nye krav, gjøre framtidig vedlikehold enklere, eller bli tilpasset en endret omgivelse. [ISO 9126]
maintainability testing: The process of testing to determine the maintainability of a software product.	vedlikeholdbarhetstesting: Testprosessen for å bestemme et programvareprodukts vedlikeholdbarhet.
maintenance: Modification of a software product after delivery to correct defects, to improve performance or other attributes, or to adapt the product to a modified environment. [IEEE 1219]	vedlikehold: Endring av et programvareprodukt etter levering for å korrigere feil, for å forbedre ytelsen eller andre egenskaper, eller for å tilpasse produktet til en endret omgivelse. [IEEE 1219]
maintenance testing: Testing the changes to an operational system or the impact of a changed environment to an operational system.	testing under vedlikehold: Testing av endringene til et system i drift eller test av følgene av endringer i omgivelsen til et system i drift.

<p>management review: A systematic evaluation of software acquisition, supply, development, operation, or maintenance process, performed by or on behalf of management that monitors progress, determines the status of plans and schedules, confirms requirements and their system allocation, or evaluates the effectiveness of management approaches to achieve fitness for purpose. [IEEE 610, IEEE 1028]</p>	<p>ledelsesgjennomgang: En systematisk evaluering av prosessen for innkjøp, leveranse, utvikling, drift eller vedlikehold (av programvare), utført ved eller for ledelsen som forfølger framdrift, bestemmer status vedrørende planer og tider, bekrefter krav og deres allokering, eller evaluerer effektiviteten til ledelsen mht. å oppnå kvalitet. [IEEE 610, IEEE 1028]</p>
<p>manufacturing-based quality: A view of quality, whereby quality is measured by the degree to which a product or service conforms to its intended design or requirements. Quality arises from the process(es) used. [Garvin] See also <i>product-based quality, transcendent-based quality, user-based quality, value-based quality.</i></p>	<p>Fremstillingsbasert kvalitet: Et syn på kvalitet der kvalitet blir målt ved graden, hvorvidt produktet eller tjenesten oppfyller dets opprinnelig oppsatte design eller spesifisering. Kvalitet kommer av prosessen(e) som er brukt [Garvin]. Se også (eng.) <i>product-based quality, transcendent-based quality, user-based quality, value-based quality.</i></p>
<p>master test plan: A test plan that typically addresses multiple test levels or all test levels in a project. See also <i>test plan.</i></p>	<p>hovedtestplan, overordnet testplan, strategisk testplan: En testplan som adresserer flere eller alle nivåer av testingen. Se også <i>prosjekttestplan.</i></p>
<p>maturity: (1) The capability of an organization with respect to the effectiveness and efficiency of its processes and work practices. See also <i>Capability Maturity Model, Test Maturity Model.</i> (2) The capability of the software product to avoid failure as a result of defects in the software. [ISO 9126] See also <i>reliability.</i></p>	<p>modenhet: (1) En organisasjons evne mht. effektiviteten og effektiviteten av dens prosesser og arbeidsmetoder. Se også CMM, TMM. (2) Et programvareprodukts evne til å unngå å feile på grunn av feil i det. [ISO 9126], Se også <i>pålitelighet.</i></p>
<p>maturity level: Degree of process improvement across a predefined set of process areas in which all goals in the set are attained. [TMMi]</p>	<p>modenhetsnivå: Graden av prosessforbedring i forhold til et på forhånd definert sett av prosessområder der alle mål i settet er oppnådd.</p>
<p>maturity model: A structured collection of elements that describe certain aspects of maturity in an organization, and aids in the definition and understanding of an organization's processes. A maturity model often provides a common language, shared vision and framework for prioritizing improvement actions.</p>	<p>modenhetsmodell: En strukturert sammenstilling av elementer som beskriver aspekter av modenhet i en organisasjon, som hjelper med å definere og forstå organisasjonens prosesser. En modenhetsmodell gir ofte en felles terminologi, felles forståelse og et rammeverk for å prioritere forbedringsinnsats.</p>

Mean Time Between Failures (MTBF): The arithmetic mean (average) time between failures of a system. The MTBF is typically part of a reliability growth model that assumes the failed system is immediately repaired, as a part of a defect fixing process. See also <i>reliability growth model</i> .	Ikke oversatt fordi begrepet bare brukes på høyere nivå ISTQB-syllabi.
Mean Time To Repair: The arithmetic mean (average) time a system will take to recover from any failure. This typically includes testing to insure that the defect has been resolved.	Ikke oversatt fordi begrepet bare brukes på høyere nivå ISTQB-syllabi.
measure: The number or category assigned to an attribute of an entity by making a measurement [ISO 14598].	mål: Tallet eller kategorien tilordnet en egenskap når en måler. [ISO 14598]
measurement: The process of assigning a number or category to an entity to describe an attribute of that entity. [ISO 14598]	måling: Prosessen med å tilordne et tall eller en kategori til et måleobjekt for å beskrive en egenskap av objektet. [ISO 14598]
measurement scale: A scale that constrains the type of data analysis that can be performed on it. [ISO 14598]	målestokk: En skala som begrenser typen av dataanalyse som kan utføres på dataverdiene. [ISO 14598]
memory leak: A defect in a program's dynamic store allocation logic that causes it to fail to release memory after it has finished using it, eventually causing the program to fail due to lack of memory.	minnelekkasje: En feil i et programs dynamiske minneallokering som fører til at det ikke tilbakeleverer minne som det ikke bruker lengre. Det kan før eller siden føre til at programmet feiler pga. mangel på minne. Se også <i>ressurslekkasje</i> .
metric: A measurement scale and the method used for measurement. [ISO 14598]	mål, metrikk: En skala og metoden som brukes for å måle [ISO 14598]
migration testing: See <i>conversion testing</i> .	migrasjonstesting: Se <i>konverteringstesting</i>
milestone: A point in time in a project at which defined (intermediate) deliverables and results should be ready.	milepæl: Et tidspunkt i et prosjekt der definerte leveranser og resultater bør være ferdige.
mind-map: A diagram used to represent words, ideas, tasks, or other items linked to and arranged around a central key word or idea. Mind maps are used to generate, visualize, structure, and classify ideas, making, and writing.	Tenkekart: Et diagram som brukes for å representere ord, ideer, oppgaver eller andre ting som er forbundet med og arrangert rundt et sentralt stikkord eller en idé. Det brukes til å generere, vise, strukturere og klassifisere ideer, konstruksjoner og skriving.
mistake: See <i>error</i> .	feil: Se (eng.) <i>error, fault, failure, incident</i>
modelling tool: A tool that supports the validation of models of the software or system [Graham].	modelleringsverktøy: Verktøy som støtter validering av modeller av programvare eller system. [Graham]
moderator: The leader and main person responsible for an inspection or other review process.	moderator: Leder for en inspeksjon eller en annen granskningsprosess.

modified condition decision coverage: See <i>condition determination coverage</i> .	modifisert betingelsesbeslutningsdekning: Se (eng.) <i>condition determination coverage</i> .
modified condition decision testing: See <i>condition determination testing</i> .	modifisert betingelsesbeslutningstesting: En testdesignteknikk der en oppnår modifisert betingelsesbeslutningsdekning.
modified multiple condition coverage: See <i>condition determination coverage</i> .	modifisert sammensatt betingelsesdekning: Se (eng.) <i>condition determination coverage</i> .
modified multiple condition testing: See <i>condition determination testing</i> .	modifisert sammensatt betingelsestesting: Se (eng.) <i>condition determination coverage</i> .
module: See <i>component</i> .	modul: Se <i>komponent</i> .
module testing: See <i>component testing</i> .	modultesting: Test av individuelle moduler. Se også <i>komponenttesting, enhetstesting</i> .
monitor: A software tool or hardware device that runs concurrently with the component or system under test and supervises, records and/or analyses the behavior of the component or system. [IEEE 610]	monitor: Et programvareverktøy eller apparat som arbeider parallelt med komponenten eller systemet under test og overvåker, måler og/eller analyserer oppførselen til det som testes. [IEEE 610]
monitoring tool: See <i>monitor</i> .	Overvåkingsverktøy
monkey testing: Testing by means of a random selection from a large range of inputs and by randomly pushing buttons, ignorant on how the product is being used.	tilfeldig testing: Testing ved å velge tilfeldig fra en stor mengde mulig input og tilfeldig trykke på knapper, uten hensyn til hvordan produktet skal brukes.
MTBF: See <i>Mean Time Between Failures</i> .	Ikke oversatt fordi begrepet bare brukes på høyere nivå ISTQB-syllabi.
MTTR: See <i>Mean Time To Repair</i> .	Ikke oversatt fordi begrepet bare brukes på høyere nivå ISTQB-syllabi.
multiple condition: See <i>compound condition</i> .	sammensatt betingelse: sammensatt betingelse (med AND, OR)
multiple condition coverage: The percentage of combinations of all single condition outcomes within one statement that have been exercised by a test suite. 100% multiple condition coverage implies 100% condition determination coverage.	sammensatt betingelsesdekning: Prosenten av kombinasjoner av alle enkle betingelsesresultater i en programinstruksjon som har blitt utført av en testsuite. 100% slik dekning impliserer 100% (eng.) condition determination coverage.
multiple condition testing: A white box test design technique in which test cases are designed to execute combinations of single condition outcomes (within one statement).	sammensatt betingelsesdekning: En strukturelltestdesignteknikk der testtilfelle blir konstruert for å utføre kombinasjoner av resultatene av enkle betingelser (i en programinstruksjon). Se også <i>strukturell testing</i> .
mutation analysis: A method to determine test suite thoroughness by measuring the extent to which a test suite can discriminate the program from slight variants (mutants) of the program.	mutasjonsanalyse: En metode for å bestemme effektiviteten av en testsuite ved å måle graden til hvilken denne testsuite kan skille programmet fra andre programmer med små endringer (mutanter).

mutation testing: See <i>back-to-back testing</i> .	Mutasjonstesting, sammenlignende test
--	--

N

N-switch coverage: The percentage of sequences of N+1 transitions that have been exercised by a test suite. [Chow]	N-switch dekning: Prosenten av sekvenser av N+1 overganger i et tilstandsdiagram som har blitt utført av en testsuite. [Chow]
N-switch testing: A form of state transition testing in which test cases are designed to execute all valid sequences of N+1 transitions. [Chow] See also <i>state transition testing</i> .	N-switch testing: En form for tilstandsbasert testing der testtilfelle blir laget for å dekke alle gyldige sekvenser av N+1 overganger. Se også <i>tilstandsbasert testing</i> .
negative testing: Tests aimed at showing that a component or system does not work. Negative testing is related to the testers' attitude rather than a specific test approach or test design technique, e.g. testing with invalid input values or exceptions. [Beizer].	negativ testing: (1) Tester som skal vise at en komponent eller et system ikke virker. Negativ testing er relatert til testerens holdning heller enn til en spesifikk framgangsmåte eller testdesignteknikk [Beizer]. (2) Test der en bruker feil input eller unntak.
non-conformity: Non-fulfilment of a specified requirement. [ISO 9000]	avvik: Faktum at et spesifikt krav ikke er oppfylt. [ISO 9000]
non-functional requirement: A requirement that does not relate to functionality, but to attributes such as reliability, efficiency, usability, maintainability and portability.	ikke-funksjonelt krav: Et krav som ikke går på funksjonalitet, men på egenskaper som for eksempel pålitelighet, effektivitet, brukbarhet, vedlikeholdbarhet eller portabilitet etc.
non-functional testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, maintainability and portability.	ikke-funksjonell testing: Testing av alt annet enn funksjonalitet.
non-functional test design techniques: Procedure to derive and/or select test cases for non-functional testing based on an analysis of the specification of a component or system without reference to its internal structure. See also <i>black box test design technique</i>	ikke-funksjonelle testdesignteknikker: Metoder for å konstruere og velge tester for ikke-funksjonell testing basert på analyse av spesifisering, komponent eller system uten å se på den interne strukturen. Se også <i>svart boks testdesignteknikk</i> .

O

<p>off-the-shelf software: A software product that is developed for the general market, i.e. for a large number of customers, and that is delivered to many customers in identical format.</p>	<p>standardprogram: Et programvareprodukt som er utviklet for det generelle marked, altså for et stort antall kunder, og som blir levert til mange kunder i identisk format.</p>
<p>operability: The capability of the software product to enable the user to operate and control it. [ISO 9126] See also <i>usability</i>.</p>	<p>opererbarhet: Et programvareprodukts evne til å muliggjøre for brukeren å benytte og styre det [ISO 9126] Se også <i>brukbarhet</i>.</p>
<p>operational acceptance testing: Operational testing in the acceptance test phase, typically performed in a (simulated) operational environment by operator and/or administrator focusing on operational aspects, e.g. recoverability, resource-behavior, installability and technical compliance. See also <i>operational testing</i>.</p>	<p>driftsakseptansetesting: Driftstesting i akseptansetesting. Utføres i et simulert driftsmiljø av operatør og/eller administrator med fokus på aspekter driften, for eksempel gjenopprettelse, ressursbruk, installerbarhet og oppfyllelse av tekniske krav. Se også <i>driftstesting</i>.</p>
<p>operational environment: Hardware and software products installed at users' or customers' sites where the component or system under test will be used. The software may include operating systems, database management systems, and other applications.</p>	<p>driftsmiljø: Hardware- og softwareprodukter installert på brukerens eller kundens plattform der komponenten eller systemet under test vil bli brukt. Programvaren kan inkludere operativsystemer, databasesystemer og andre applikasjoner.</p>
<p>operational profile: The representation of a distinct set of tasks performed by the component or system, possibly based on user behavior when interacting with the component or system, and their probabilities of occurrence. A task is logical rather than physical and can be executed over several machines or be executed in non-contiguous time segments.</p>	<p>bruksprofil: Fremstilling av et avgrenset sett oppgaver utført av en komponent eller et system, som kan være basert på brukeres atferd ved interaksjon med komponenten eller systemet, og sannsynligheten for at de inntreffer. En oppgave er logisk, heller enn fysisk, og kan bli utført over flere maskiner eller i uavhengige tidssegmenter.</p>
<p>operational profile testing: Statistical testing using a model of system operations (short duration tasks) and their probability of typical use. [Musa]</p>	<p>test basert på bruksprofil: Statistisk test der en bruker en modell av systemets drift (som består av korte oppgaver) og deres sannsynlighet under typisk bruk. [Musa]</p>
<p>operational testing: Testing conducted to evaluate a component or system in its operational environment. [IEEE 610]</p>	<p>driftstesting: (1) Test som kjøres i driftsomgivelsen. [IEEE 610] Også SIT – Site Acceptance Testing (2) De tester som utføres for å godkjenne om driftsrutinene er gode nok. Testene utformes med henblikk på å prøve ut både manuelle og maskinelle driftsrutiner samt driftsdokumentasjon.</p>
<p>oracle: See <i>test oracle</i>.</p>	<p>orakel: Se <i>testorakel</i></p>

orthogonal array: A 2-dimensional array constructed with special mathematical properties, such that choosing any two columns in the array provides every pair combination of each number in the array.	ortogonal matrise: En matrise med spesielle matematiske egenskaper som inneholder alle parvise kombinasjoner av verdier for hver parameter. Matrisen inneholder hvert par like mange ganger.
orthogonal array testing: A systematic way of testing all-pair combinations of variables using orthogonal arrays. It significantly reduces the number of all combinations of variables to test all pair combinations. See also <i>pairwise testing</i> .	testing ved hjelp av ortogonale matriser: Systematisk måte å lage testtilfeller med alle mulige parvise kombinasjoner av input parametere ved å bruke ortogonale matriser. Gir en betydelig reduksjon av antall kombinasjoner for å teste alle parvise kombinasjoner. Se også <i>parvis testing</i> .
outcome: See <i>result</i> .	utgang: Se <i>resultat</i> .
output: A variable (whether stored within a component or outside) that is written by a component.	utdata: En variabel (lagret i en komponent eller utenfor) som skrives av en komponent.
output domain: The set from which valid output values can be selected. See also <i>domain</i> .	utdatadomene: Mengden med gyldige utdataverdier. Se også <i>domene</i> .
output value: An instance of an output. See also <i>output</i> .	utdataverdi: En konkret verdi for utdata.

P

pair programming: A software development approach whereby lines of code (production and/or test) of a component are written by two programmers sitting at a single computer. This implicitly means ongoing real-time code reviews are performed.	parprogrammering: En utviklingsmetode der koden blir skrevet av to programmerere som sitter ved en datamaskin sammen. Dette betyr at en kontinuerlig gjennomgår den koden som skrives.
pair testing: Two persons, e.g. two testers, a developer and a tester, or an end-user and a tester, working together to find defects. Typically, they share one computer and trade control of it while testing.	partesting: To testere arbeider sammen for å finne feil. Typisk deler de en datamaskin og skifter på hvem som kjører testtilfelle.
pairwise testing: A black box test design technique in which test cases are designed to execute all possible discrete combinations of each pair of input parameters. See also <i>orthogonal array testing</i> .	parvis testing: En svart boks testdesignteknikk hvor testtilfeller lages for adskilt testing av alle mulige parvise kombinasjoner av verdier for ulike input parametere. Metoden garanterer ikke at alle par forekommer like mange ganger. Se også <i>orthogonal array testing</i> .

pareto analysis: A statistical technique in decision-making that is used for selection of a limited number of areas that produce significant overall effect. In terms of quality improvement, a large majority of problems (80%) are produced by a few key causes (20%).	paretoanalyse: En statistisk teknikk som brukes for å velge et begrenset antall områder som produserer en signifikant virkning på helheten. Når det gjelder kvalitetsforbedring blir et stort flertall av problemene (80%) laget av noen få nøkkelårsaker (20%).
partition testing: See <i>equivalence partitioning</i> . [Beizer]	partisjonstesting: Se ekvivalensklasseinndeling [Beizer]
pass: A test is deemed to pass if its actual result matches its expected result.	godkjent: En test blir godkjent når dens aktuelle resultat stemmer med det forventede resultat.
pass/fail criteria: Decision rules used to determine whether a test item (function) or feature has passed or failed a test. [IEEE 829]	godkjenningskriterier: Beslutningsregler som brukes for å bestemme om et testobjekt eller en feature har klart eller ikke klart en test (skal betraktes som godkjent). [IEEE 829]
path: A sequence of events, e.g. executable statements, of a component or system from an entry point to an exit point.	sti (rute): En sekvens av hendelser, for eksempel eksekverbare programinstruksjoner av en komponent eller et system fra et startpunkt til et slutt punkt.
path coverage: The percentage of paths that have been exercised by a test suite. 100% path coverage implies 100% LCSAJ coverage.	stidekning: Prosenten stier som har blitt utført av en testsuite. 100% stidekning betyr at alle andre strukturelle testdekningskriterier er oppnådd. Se også <i>strukturell testing</i> .
path sensitizing: Choosing a set of input values to force the execution of a given path.	stivalg: Å velge et sett med inputverdier som skal sørge for at en bestemt sti blir utført.
path testing: A white box test design technique in which test cases are designed to execute paths.	stitesting: En strukturell testdesignteknikk der testtilfelle blir konstruert for å utføre stier. Se også <i>strukturell testing</i> .
peer review: A review of a software work product by colleagues of the producer of the product for the purpose of identifying defects and improvements. Examples are inspection, technical review and walkthrough.	kollegagjennomgang: Se <i>teknisk gjennomgang</i> .
performance: The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate. [IEEE 610] See also <i>efficiency</i> .	ytelse: Graden til hvilken et system eller en komponent oppfyller sine bestemte funksjoner innen gitte restriksjoner når det gjelder behandlingstid og kapasitet. [IEEE 610] Se også <i>effektivitet</i> .
performance indicator: A high level metric of effectiveness and/or efficiency used to guide and control progressive development, e.g. lead-time slip for software development. [CMMI]	ytelsesindikator: En høynivåmåling av effektivitet som brukes for å styre videre utvikling. For eksempel prosenten feil som er funnet under testing. [CMMI]

performance profiling: Definition of user profiles in performance, load and/or stress testing. Profiles should reflect anticipated or actual usage based on an operational profile of a component or system, and hence the expected workload. See also <i>load profile, operational profile</i> .	ytelsesprofilering: Definisjon av brukerprofiler i test av ytelse, belastning og/eller stress. Bør gjenspeile forventet eller virkelig bruk basert på en bruksprofil til en komponent eller et system, og derav forventet belastning. Se også <i>belastningsprofil, bruksprofil</i> .
performance testing: The process of testing to determine the performance of a software product. See also <i>efficiency testing</i> .	ytelsestesting: Testing for å måle ytelsen til et programvareprodukt. Måling av systemets svartider eller testing av systemets kapasitet under gitte krav til svartider.
performance testing tool: A tool to support performance testing that usually has two main facilities: load generation and test transaction measurement. Load generation can simulate either multiple users or high volumes of input data. During execution, response time measurements are taken from selected transactions and these are logged. Performance testing tools normally provide reports based on test logs and graphs of load against response times.	ytelsestestverktøy: Et verktøy som understøtter ytelsestesting som vanligvis har to hovedkomponenter: Lastgenerering og måling av testtransaksjoner. Lastgenerering kan enten simulere flere brukere eller store mengder inputdata. Under utførelsen blir responstiden målt for utvalgte transaksjoner og disse blir logget. Slike verktøy gir vanligvis rapporter og grafer som viser responstid og kapasitet mot belastning og som er basert på loggene.
phase test plan: A test plan that typically addresses one test phase. See also <i>test plan</i> .	testplan for et testnivå: En testplan som gjelder et testnivå.
pointer: A data item that specifies the location of another data item; for example, a data item that specifies the address of the next employee record to be processed. [IEEE 610]	peker: Dataelement som angir plasseringen til et annet dataelement; for eksempel et dataelement som angir adressen til neste record som skal behandles. [IEEE 610]
portability: The ease with which the software product can be transferred from one hardware or software environment to another. [ISO 9126]	portabilitet: Enkelheten med hvilken et programvareprodukt kan bli overført fra en hardware- og softwareomgivelse til en annen. [ISO 9126]
portability testing: The process of testing to determine the portability of a software product.	portabilitetstesting: Prosessen med å teste et program for å bestemme dens portabilitet.
postcondition: Environmental and state conditions that must be fulfilled after the execution of a test or test procedure.	post-condition: Betingelser i omgivelsen og tilstanden som må være oppfylt etter utføring av en test eller en samling tester. Resultater som programmet garanterer. Også etterbetingelse .
post-execution comparison: Comparison of actual and expected results, performed after the software has finished running.	sammenligning etter utførelsen: Sammenligning av aktuelle og forventede resultater, utført etter at programvaren har blitt kjørt.
post project meeting: See <i>retrospective meeting</i> .	prosjekt-oppsummeringsmøte: Se (eng.) <i>retrospective meeting</i> .

precondition: Environmental and state conditions that must be fulfilled before the component or system can be executed with a particular test or test procedure.	forbetingelse: Betingelser i omgivelsen og tilstanden som må være oppfylt før komponenten eller systemet kan bli utført med en bestemt test eller samling tester. Også: forutsetninger.
predicted outcome: See <i>expected result.</i>	forutsagt utgang: Se <i>forventet resultat.</i>
pretest: See <i>intake test.</i>	Se (eng.) smoke test
priority: The level of (business) importance assigned to an item, e.g. defect.	prioritet: Nivået for viktighet som bestemmes for en sak.
probe effect: The effect on the component or system by the measurement instrument when the component or system is being measured, e.g. by a performance testing tool or monitor. For example performance may be slightly worse when performance testing tools are being used.	instrumenteringseffekt: Effekten en ”probe” (sonde, måleinstrument) har på komponenten eller systemet som måles. F.eks. kan aktiv måling av ytelse med måleverktøy påvirke ytelsen negativt, forhåpentligvis marginalt. Også <i>probeeffekt.</i>
problem: See <i>defect.</i>	problem: Se <i>feil.</i>
problem management: See <i>defect management.</i>	problemhåndtering: Se (eng.) <i>defect management.</i>
problem report: See <i>defect report.</i>	problemrapport: Se <i>feilrapport.</i>
procedure testing: Testing aimed at ensuring that the component or system can operate in conjunction with new or existing users’ business procedures or operational procedures.	prosedyretesting: Testing for å ivareta at komponenten eller systemet kan kjøre i samsvar med nye eller eksisterende brukeres forretningsprosedyrer eller driftsprosedyrer.
process: A set of interrelated activities, which transform inputs into outputs. [ISO 12207]	prosess: En samling sammenhengende aktiviteter som overfører inputs til outputs. [ISO 12207]
process assessment: A disciplined evaluation of an organization’s software processes against a model compatible with the reference model. [ISO 15504]	prosessvurdering: En systematisk evaluering av organisasjonens softwareprosesser mot en modell som er kompatibel med referansemodellen. [ISO 15504]
process cycle test: A black box test design technique in which test cases are designed to execute business procedures and processes. [TMap] See also <i>procedure testing.</i>	prosess-syklustest: En svart boks testdesignteknikk hvor testtilfeller utarbeides for å gjennomføre forretningsprosedyrer og prosesser [TMap]. Se også <i>prosedyretesting.</i>
process improvement: A program of activities designed to improve the performance and maturity of the organization’s processes, and the result of such a program. [CMMI]	prosessforbedring: Program av aktiviteter beregnet på å forbedre ytelsen og modenheten i prosessene til en organisasjon, samt resultatene av et slikt program. [CMMI]
process model: A framework wherein processes of the same nature are classified into a framework, e.g. a test improvement model.	Prosessmodell: Et rammeverk der prosesser av samme art er klassifisert mot rammeverket, for eksempel en testforbedringsmodell.

<p>product-based quality: A view of quality, wherein quality is based on a well-defined set of quality attributes. These attributes must be measured in an objective and quantitative way. Differences in the quality of products of the same type can be traced back to the way the specific quality attributes have been implemented. [Garvin] See also <i>manufacturing-based quality, quality attribute, transcendent-based quality, user-based quality, value-based quality.</i></p>	<p>Produktbasert kvalitet: Et synspunkt på kvalitet der kvalitet er basert på et veldefinert sett med kvalitetsegenskaper. Disse må måles objektivt og kvantitativt. Forskjeller i kvaliteten når det gjelder produkter av samme type kan da føres tilbake til måten de spesifikke kvalitetsegenskapene har blitt implementert. [Garvin]</p>
<p>product risk: A risk directly related to the test object. See also <i>risk.</i></p>	<p>produkt risiko: Risiko som er direkte relatert til testobjektet. Se også <i>risiko.</i></p>
<p>production acceptance testing: See <i>operational acceptance testing.</i></p>	<p>Se (eng.) <i>operational acceptance testing.</i></p>
<p>program instrumenter: See <i>instrumenter.</i></p>	<p>Instrumenteringsverktøy</p>
<p>program testing: See <i>component testing.</i></p>	<p>programtesting: Se <i>komponenttesting.</i></p>
<p>project: A project is a unique set of coordinated and controlled activities with start and finish dates undertaken to achieve an objective conforming to specific requirements, including the constraints of time, cost and resources. [ISO 9000]</p>	<p>prosjekt: En mengde koordinerte og styrte aktiviteter med start og sluttdatoer som brukes for å oppnå et mål som oppfyller spesifikke krav, under hensyn til begrensninger i tid, ressursbruk og kostnader. [ISO 9000]</p>
<p>project retrospective: A structured way to capture lessons learned and to create specific action plans for improving on the next project or next project phase.</p>	<p>Prosjektretrospektiv: En strukturert måte å samle ting en har lært og til å lage spesifikke aksjonsplaner for å forbedre det neste prosjekt eller den neste prosjektfasen.</p>
<p>project risk: A risk related to management and control of the (test) project, e.g. lack of staffing, strict deadlines, changing requirements, etc. See also <i>risk.</i></p>	<p>prosjekt risiko: Risiko relatert til ledelse og styring av et (test) prosjekt, for eksempel mangel på personell, knappe tidsfrister, endrede krav, etc.. Se også <i>risiko.</i></p>
<p>project test plan: See <i>master test plan.</i></p>	<p>prosjekttestplan: En testplan som adresserer flere eller alle nivåer av testingen. Se også <i>hovedtestplan.</i></p>
<p>pseudo-random: A series, which appears to be random but is in fact generated according to some prearranged sequence.</p>	<p>pseudo-random: En serie (tall eller tegn) som ser ut til å være tilfeldig (random) men er i virkeligheten generert i en på forhånd fastlagt rekkefølge.</p>

Q

<p>qualification: The process of demonstrating the ability to fulfil specified requirements. Note the term ‘qualified’ is used to designate the corresponding status. [ISO 9000]</p>	<p>kvalifisering: Prosessen å vise evne til å tilfredsstille spesifiserte krav. Uttrykket ’kvalifisert’ betegner den tilsvarende status. Se også <i>sertifisering</i>.</p>
<p>quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations. [IEEE 610]</p>	<p>kvalitet: Graden med hvilken en komponent, et system eller en prosess oppfyller spesifiserte krav og/eller bruker/kundebehov og forventninger. [IEEE 610] Se også [ISO 9000], [ISO 8402]</p>
<p>quality assurance: Part of quality management focused on providing confidence that quality requirements will be fulfilled. [ISO 9000]</p>	<p>kvalitetssikring: Del av kvalitetsledelse, fokuserer på å generere tiltro til at kvalitetskrav vil bli oppfylt. [ISO 9000]</p>
<p>quality attribute: A feature or characteristic that affects an item’s quality. [IEEE 610]</p>	<p>kvalitetsegenskap: En egenskap som har med et objekts kvalitet å gjøre. [IEEE 610]</p>
<p>quality characteristic: See <i>quality attribute</i>.</p>	<p>Se (eng.) <i>quality attribute</i>.</p>
<p>quality gate: A special milestone in a project. They are located between those phases of a project strongly depending on the outcome of a previous phase. A quality gate includes a formal check of the documents of the previous phase.</p>	<p>kvalitets-milepæl (foreløpig valgt term, i påvente av tilbakemelding. Termen brukes bare på de høyere nivåene ISTQB-syllabi): En spesiell milepæl i et prosjekt. Slike milepæler ligger mellom slike faser i et prosjekt der neste fase er sterkt avhengig av kvaliteten av resultatet i en tidligere fase. En slik milepæl omfatter en formell kontroll av dokumentene i en tidligere fase.</p>
<p>quality management: Coordinated activities to direct and control an organization with regard to quality. Direction and control with regard to quality generally includes the establishment of the quality policy and quality objectives, quality planning, quality control, quality assurance and quality improvement. [ISO 9000]</p>	<p>kvalitetsledelse: Se norsk definisjon i [ISO 9000]. Koordinerte aktiviteter for å lede og styre en organisasjon mht. kvalitet. Dette inkluderer opprettelsen av en kvalitetspolicy og kvalitetsmål, kvalitetsplanlegging, styring, sikring og forbedring.</p>

R

<p>random testing: A black box test design technique where test cases are selected, possibly using a pseudo-random generation algorithm, to match an operational profile. This technique can be used for testing non-functional attributes such as reliability and performance.</p>	<p>tilfeldig testing: En svart boks testdesignteknikk der testtilfelle blir valgt tilfeldig, vanligvis gjennom en pseudo-random genereringsalgoritme. Ofte blir det brukt et bruksprofil. Dette blir mest brukt for å teste pålitelighet og ytelse.</p>
<p>Rational Unified Process: A proprietary adaptable iterative software development process framework consisting of four project lifecycle phases: inception, elaboration, construction and transition.</p>	<p>Rational Unified Process: En proprietær programvareutviklingsprosess som bruker en gjentagende metodologi, med røtter i spiralmodellen, for å sikre et resultat med høyt kvalitetsnivå. Den består av fasene Inception (Innledning), Elaboration (Utforming), Construction (Bygging), Transition (Overgang) [wikipedia.no]</p>
<p>recorder: See <i>scribe</i>.</p>	<p>sekretær: Den som skriver protokoll i en gjennomgang eller granskning.</p>
<p>record/playback tool: See <i>capture/playback tool</i>.</p>	<p>capture/playback verktøy: Se (eng.) <i>capture/playback tool</i>.</p>
<p>recoverability: The capability of the software product to re-establish a specified level of performance and recover the data directly affected in case of failure. [ISO 9126] See also <i>reliability</i>.</p>	<p>evnen til gjenopprettelse: Evnen til et programvareprodukt å reetablere et spesifisert ytelsesnivå og gjenopprette data som var affisert av et sammenbrudd. [ISO 9126] Se også <i>pålitelighet</i>.</p>
<p>recoverability testing: The process of testing to determine the recoverability of a software product. See also <i>reliability testing</i>.</p>	<p>test av evnen til gjenopprettelse: Prosessen med å teste evnen til gjenopprettelse av et programvareprodukt. Se også <i>test av pålitelighet</i>.</p>
<p>recovery testing: See <i>recoverability testing</i>.</p>	<p>test av evnen til gjenopprettelse</p>
<p>regression testing: Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.</p>	<p>regresjonstesting: (1) Test av et program som var testet før etter at modifikasjoner er utført for å sikre at feil ikke har blitt introdusert eller er aktivert i uendrete områder i produktet, som et resultat av endringen. Regresjonstest blir utført etter endringer i programvaren eller omgivelsen. (2) Gjentakelse av en test for å verifisere at feilrettinger eller endringer ikke har introdusert nye feil eller uønskede effekter, og at systemet eller komponenten (testobjektet) fremdeles tilfredsstillter spesifiserte krav. (etter tidl. DND terminologiliste).</p>

regulation testing: See <i>compliance testing</i>	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
release note: A document identifying test items, their configuration, current status and other delivery information delivered by development to testing, and possibly other stakeholders, at the start of a test execution phase. [IEEE 829]	frigivelsesdokument: Et dokument som identifiserer testobjekter, deres konfigurasjon, aktuell status og annen leveringsinformasjon som blir levert av utviklingen til testingen og muligens andre interessenter, på starten av en testutføring [IEEE 829]. Også <i>testobjekt overleveringsrapport</i> .
reliability: The ability of the software product to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations. [ISO 9126]	pålitelighet: Et programvareprodukts evne til å utføre dets påkrevde funksjoner under spesifiserte betingelser for en spesifisert tidsperiode eller for et spesifisert antall handlinger, når det blir brukt under spesifiserte betingelser. [ISO 9126]
reliability growth model: A model that shows the growth in reliability over time during continuous testing of a component or system as a result of the removal of defects that result in reliability failures.	pålitelighetsmodell: Modell som viser endringen over tid med hensyn til pålitelighet som resultat av kontinuerlig testing av en komponent eller et system og fjerning av feilene som gir brudd i påliteligheten.
reliability testing: The process of testing to determine the reliability of a software product.	pålitelighetstesting: Prosessen med å teste for å måle påliteligheten til et programvareprodukt.
replaceability: The capability of the software product to be used in place of another specified software product for the same purpose in the same environment. [ISO 9126] See also <i>portability</i> .	erstattbarhet: Et programvareprodukts evne til å bli brukt istedenfor et annet spesifisert programvareprodukt for samme mål i samme omgivelse. [ISO 9126], Se også <i>portabilitet</i> .
requirement: A condition or capability needed by a user to solve a problem or achieve an objective that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. [IEEE 610]	krav: En betingelse eller evne som behøves av en bruker for å løse et problem eller oppnå et mål som må bli oppfylt eller må eies av et system eller en komponent for å oppfylle en kontrakt, spesifikasjon eller et annet formelt pålagt dokument. [IEEE 610]
requirements-based testing: An approach to testing in which test cases are designed based on test objectives and test conditions derived from requirements, e.g. tests that exercise specific functions or probe non-functional attributes such as reliability or usability.	kravbasert testing: En måte å teste på der testtilfelle blir laget basert på testmål og betingelser som blir avledet fra krav, for eksempel at testene utfører bestemte funksjoner eller prøver ikke-funksjonelle egenskaper som pålitelighet og brukbarhet.

<p>requirements management tool: A tool that supports the recording of requirements, requirements attributes (e.g. priority, knowledge responsible) and annotation, and facilitates traceability through layers of requirements and requirements change management. Some requirements management tools also provide facilities for static analysis, such as consistency checking and violations to pre-defined requirements rules.</p>	<p>kravhåndteringsverktøy: Et verktøy som understøtter formulering, analysering, prioritering og vedlikehold av kravene.</p>
<p>requirements phase: The period of time in the software life cycle during which the requirements for a software product are defined and documented. [IEEE 610]</p>	<p>kravspesifikasjonsfase: Tidsperioden i programvarens livssyklus der kravene til produktet blir definert og dokumentert. [IEEE 610]</p>
<p>resource leak (not in official English dictionary)</p>	<p>ressurslekkasje: En feil i et program der programmet reserverer ressurser men ikke frigir disse etter at det er ferdig med å bruke de. Eksempel: minnelekkasje.</p>
<p>resource utilization: The capability of the software product to use appropriate amounts and types of resources, for example the amounts of main and secondary memory used by the program and the sizes of required temporary or overflow files, when the software performs its function under stated conditions. [ISO 9126] See also <i>efficiency</i>.</p>	<p>ressursbruk: Et programvareprodukts evne til å bruke passende mengder og typer av ressurser, for eksempel minne, diskplass eller prosessortid. [ISO 9126] Se også <i>ytelse</i>.</p>
<p>resource utilization testing: The process of testing to determine the resource-utilization of a software product. See also <i>efficiency testing</i>.</p>	<p>ressursbrukstesting: Prosessen med å teste for å måle ressursbruken til et programvareprodukt.</p>
<p>result: The consequence/outcome of the execution of a test. It includes outputs to screens, changes to data, reports, and communication messages sent out. See also <i>actual result</i>, <i>expected result</i>.</p>	<p>resultat: Konsekvenser og tilstander etter utføringen av en test. Dette inkluderer utdata til skjerm, endringer av data, rapporter, kommunikasjonsmeldinger og alle utdata til alle kanaler som er aktuelle. Se også <i>aktuelt resultat</i>, <i>forventet resultat</i>.</p>
<p>resumption criteria: The testing activities that must be repeated when testing is re-started after a suspension. [IEEE 829]</p>	<p>kriterier for å fortsette testen: Testaktivitetene som må gjentas når testingen blir startet om igjen etter at det har blitt avbrutt. [IEEE 829]</p>
<p>re-testing: Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions.</p>	<p>retest: Testing som kjører det testtilfellet som feilet siste gang testen ble kjørt, for å kontrollere at en feil er rettet.</p>

retrospective meeting: A meeting at the end of a project during which the project team members evaluate the project and learn lessons that can be applied to the next project.	erfaringsmøte: Møte i slutten av et prosjekt hvor medlemmene evaluerer prosjektet og summerer opp erfaringer for bruk i neste prosjekt.
review: An evaluation of a product or project status to ascertain discrepancies from planned results and to recommend improvements. Examples include management review, informal review, technical review, inspection, and walkthrough. [IEEE 1028]	granskning, gjennomgang, review: Evaluering av et produkt eller prosjektstatus for å finne avvik fra planlagte resultater og for å anbefale forbedringer. Eksempler er ledelsesgjennomganger, uformelle gjennomganger, tekniske gjennomganger, inspeksjoner og walkthrough. [IEEE 1028]
review tool: A tool that provides support to the review process. Typical features include review planning and tracking support, communication support, collaborative reviews and a repository for collecting and reporting of metrics.	verktøy for granskning: Et verktøy som understøtter granskningsprosessen. Typiske funksjoner er støtte til planlegging og oppfølging av granskningen, støtte til kommunikasjon, samarbeid og et sentralt sted for å samle og rapportere måldata.
reviewer: The person involved in the review that identifies and describes anomalies in the product or project under review. Reviewers can be chosen to represent different viewpoints and roles in the review process.	revisor: Også kontrollør. Personen som er involvert i en granskning og som skal identifisere og beskrive avvik i produktet eller prosjektet under granskning.
risk: A factor that could result in future negative consequences; usually expressed as impact and likelihood.	risiko: Et forhold som kan resultere i framtidige negative konsekvenser, vanligvis beskrevet ved kostnad og sannsynlighet.
risk analysis: The process of assessing identified risks to estimate their impact and probability of occurrence (likelihood).	risikoanalyse: Prosessen der identifiserte risikoer blir analysert for å estimere deres kostnader og sannsynligheter.
risk-based testing: An approach to testing to reduce the level of product risks and inform stakeholders of their status, starting in the initial stages of a project. It involves the identification of product risks and the use of risk levels to guide the test process.	risikobasert testing: Metode for testing som tar sikte på å redusere produktrisiko og informere interessentene om risikoenes status, allerede fra oppstart av et prosjekt. Innebærer å identifisere produktrisikoen og bruken av disse til å styre testprosessen.
risk category: <i>See risk type</i>	risikotype
risk control: The process through which decisions are reached and protective measures are implemented for reducing risks to, or maintaining risks within, specified levels.	risikostyring: Prosessen der beslutninger blir tatt og forbyggende aksjoner blir implementert for å redusere risiko eller holde de innenfor spesifiserte nivåer.
risk identification: The process of identifying risks using techniques such as brainstorming, checklists and failure history.	risikoidentifikasjon: Prosessen med å identifisere risiko ved hjelp av teknikker som brainstorming, sjekklister og feilhistorie.

risk level: The importance of a risk as defined by its characteristics impact and likelihood. The level of risk can be used to determine the intensity of testing to be performed. A risk level can be expressed either qualitatively (e.g. high, medium, low) or quantitatively.	risikonivå: Betydningen til en risiko definert ut fra dens effekt og sannsynlighet. Risikonivå kan brukes til å bestemme grundigheten i testingen som skal utføres. Risikonivå kan uttrykkes enten kvantitativt (for eksempel høy, middels, lav) eller kvantitativt.
risk management: Systematic application of procedures and practices to the tasks of identifying, analyzing, prioritizing, and controlling risk.	risikoleidelse: Systematisk anvendelse av prosedyrer og praksis til risikoidentifisering, prioritering og styring.
risk mitigation: See <i>risk control</i> .	risikoreduksjon: Se <i>risikostyring</i>
risk type: A set of risks grouped by one or more common factors such as a quality attribute, cause, location, or potential effect of risk;. A specific set of product risk types is related to the type of testing that can mitigate (control) that risk type. For example the risk of user-interactions being misunderstood can be mitigated by usability testing.	risikotype: Ett sett med risikoer gruppert ved å bruke en eller flere felles egenskaper som kvalitetsegenskap, årsak, sted eller potensiell effekt av risikoen: En spesifikk samling av produktisikotyper er relatert til typen testing som kan utføres for risikostyring innenfor risikotypen. For eksempel kan en risiko som 'misforstått brukerinteraksjon' håndteres med brukbarhetstesting.
robustness: The degree to which a component or system can function correctly in the presence of invalid inputs or stressful environmental conditions. [IEEE 610] See also <i>error tolerance</i> , <i>fault-tolerance</i> .	robusthet: Graden som en komponent eller et system kan fungere korrekt når ugyldige inputs eller vanskelige betingelser i omgivelsen er tilstede. [IEEE 610], Se også <i>feiltoleranse</i> .
robustness testing: Testing to determine the robustness of the software product.	test for robusthet: Testing for å bestemme et produkts robusthet.
root cause: A source of a defect such that if it is removed, the occurrence of the defect type is decreased or removed. [CMMI]	årsak: Kilden til en defekt, slik at om den fjernes vil forekomsten av denne typen defekter avta eller forsvinne. [CMMI]
root cause analysis: An analysis technique aimed at identifying the root causes of defects. By directing corrective measures at root causes, it is hoped that the likelihood of defect recurrence will be minimized.	årsaksanalyse: Analyseteknikk med mål å identifisere den grunnleggende årsaken til defekter. Ved å rette korrigerende tiltak mot årsaken, er håpet at gjentakelse av feil minimeres.
RUP: See <i>Rational Unified Process</i> .	RUP: Se (eng.) <i>Rational Unified Process</i> .

S

safety: The capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property or the environment in a specified context of use. [ISO 9126]	sikkerhet: Programvareprodukts evne til å oppnå aksepterte nivåer av risiko for skade til folk, forretning, programvare, eiendom eller miljøet i en spesifisert kontekst for bruk. [ISO 9126]. På norsk skilles der ikke mellom ”safety” og ”security”.
safety critical system: A system whose failure or malfunction may result in death or serious injury to people, or loss or severe damage to equipment, or environmental harm.	sikkerhetskritisk system: Et system som kan forvolde død eller alvorlig skade på personer, tap eller betydelig skade på utstyr, eller skade på miljøet hvis det feiler.
safety testing: Testing to determine the safety of a software product.	sikkerhetstest: Test for å evaluere sikkerheten av et programvareprodukt.
sanity test: See <i>smoke test</i> .	smoke test
scalability: The capability of the software product to be upgraded to accommodate increased loads. [After Gerrard]	skalerbarhet: Et programvareprodukts evne til å bli oppgradert for å klare økende belastninger. [Gerrard]
scalability testing: Testing to determine the scalability of the software product.	skalerbarhetstesting: Prosessen med å evaluere et produkts skalerbarhet.
scenario testing: See <i>use case testing</i> .	scenariotest: Se <i>use case testing</i> .
scorecard: A representation of summarized performance measurements representing progress towards the implementation of long-term goals. A scorecard provides static measurements of performance over or at the end of a defined interval. See also <i>balanced scorecard</i> , <i>dashboard</i> .	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
scribe: The person who has to record each defect mentioned and any suggestions for improvement during a review meeting, on a logging form. The scribe should ensure that the logging form is readable and understandable.	sekretær: Personen som skal protokollere alle feilene nevnt og forslagene for forbedring gjort under et gjennomgangsmøte. Sekretæren skal sørge for at protokollen er lesbar og forståelig.
scripted testing: Test execution carried out by following a previously documented sequence of tests.	scriptbasert testing: Test gjennomføres ved å følge en forhåndsdefinert sekvens av tester.
scripting language: A programming language in which executable test scripts are written, used by a test execution tool (e.g. a capture/playback tool).	scriptspråk: Et programmeringsspråk for å lage utførbare testskripter som brukes av et testautomatiseringsverktøy.
SCRUM: An iterative incremental framework for managing projects commonly used with agile software development.	SCRUM: et inkrementelt og iterativ rammeverk for prosjektstyring som i hovedsak benyttes innenfor smidig programvareutvikling.

security: Attributes of software products that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data. [ISO 9126] See also <i>functionality</i> .	sikkerhet: Et programvareprodukts evne til å forebygge ikke autorisert adgang, tilfeldig eller med hensikt, til programmer og data. [ISO 9126]. Se også funksjonalitet.
security testing: Testing to determine the security of the software product. See also <i>functionality testing</i> .	sikkerhetstest: Testing som har til hensikt å avdekke sårbarheter i sikkerheten til et programvareprodukt. Se også funksjonell testing.
security testing tool: A tool that provides support for testing security vulnerabilities.	sikkerhetstestverktøy: Et verktøy som gir støtte til testing av sårbarheter i sikkerhet.
security tool: A tool that supports operational security.	sikkerhetsverktøy: Et verktøy for støtte til driftssikkerhet.
serviceability testing: See <i>maintainability testing</i> .	vedlikeholdbarhetstesting
session-based test management: A method for measuring and managing session-based testing, e.g. exploratory testing.	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
session-based testing: An approach to testing in which test activities are planned as uninterrupted sessions of test design and execution, often used in conjunction with exploratory testing.	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
severity: The degree of impact that a defect has on the development or operation of a component or system. [IEEE 610]	alvorlighetsgrad: Graden for innvirkning som en feil har på utviklingen eller driften av en komponent eller et system. [IEEE 610]
simulation: The representation of selected behavioral characteristics of one physical or abstract system by another system. [ISO 2382/1]	simulering: Representasjon av utvalgte oppførselsegenskaper av et konkret eller abstrakt system gjennom et annet system. [ISO 2382/1]
simulator: A device, computer program or system used during testing, which behaves or operates like a given system when provided with a set of controlled inputs. [IEEE 610, DO178b] See also <i>emulator</i> .	simulator: Et apparat, program eller system brukt under testing, som oppfører seg som et gitt system når det blir gitt en mengde med spesifikke inputs. [IEEE 610, DO-178b] Se også <i>emulator</i> .
site acceptance testing: Acceptance testing by users/customers at their site, to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes, normally including hardware as well as software.	site akseptansetesting: Akseptansetesting som utføres på kundens/brukerens plattform for å avgjøre om en komponent eller et system tilfredsstiller kundenes/brukerens behov og forretningsprosesser. Inkluderer vanligvis både hardware og programvare.

smoke test: A subset of all defined/planned test cases that cover the main functionality of a component or system, to ascertaining that the most crucial functions of a program work, but not bothering with finer details. A daily build and smoke test is among industry best practices. See also <i>intake test</i> .	pretest: Et utvalg av alle definerte eller planlagte testtilfeller som dekker hovedfunksjonaliteten av testobjektet, for å sikre at de viktigste funksjonene i testobjektet virker, men uten å ta hensyn til mindre detaljer. Daglig build av en ny release og pretest imellom er blant beste industripraksis. Se også inntakstest.
software: Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system [IEEE 610]	programvare.
software attack: See <i>attack</i> .	angrep på et program: Se <i>angrep</i> .
Software Failure Mode and Effect Analysis (SFMEA): See <i>Failure Mode and Effect Analysis (FMEA)</i> .	Se (eng.) <i>Failure Mode and Effect Analysis (FMEA)</i>
Software Failure Mode Effect, and Criticality Analysis (SFMECA): See <i>Failure Mode, Effect, and Criticality Analysis (FMECA)</i> .	Se (eng.) <i>Failure Mode and Effect, and Criticality Analysis (FMECA)</i>
Software Fault Tree Analysis (SFTA): See <i>Fault Tree Analysis (FTA)</i> .	Se (eng.) <i>Fault Tree Analysis (FTA)</i> .
software feature: See <i>feature</i> .	software feature: Se (eng.) <i>feature</i> .
software life cycle: The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. Note these phases may overlap or be performed iteratively.	livssyklus til programvare: Tidsperioden som begynner når ideen om et programvareprodukt oppstår og slutter når produktet ikke lenger er tilgjengelig. Typiske faser karakteriseres av idé, krav, design, implementering, test, installasjon og etterprøving, drift og vedlikehold, og noen ganger, avvikling. Disse fasene kan overlappe hverandre eller utføres iterativt.
Software Process Improvement: A program of activities designed to improve the performance and maturity of the organization's software processes and the results of such a program. [CMMI]	prosessforbedring for programvare: Et program bestående av aktiviteter som sikter på å forbedre ytelsen og modenheten av organisasjonens prosesser mht. programvare og resultatet av et slikt program. [CMMI]
software product characteristic: See <i>quality attribute</i> .	egenskap for programvareprodukt: Se <i>kvalitetsegenskap</i>
software quality: The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs. [ISO 9126]	programvarekvalitet: Helheten av funksjonalitet og egenskaper av et programvareprodukt som sørger for dets evne til å oppfylle oppgitt eller impliserte behov. [ISO 9126]
software quality characteristic: See <i>quality attribute</i> .	egenskap for programvarekvalitet: Se <i>kvalitetsegenskap</i> .

software test incident: <i>See incident.</i>	testhendelse: <i>Se hendelse.</i>
software test incident report: <i>See incident report.</i>	hendelsesrapport: <i>Se også feilrapport.</i>
Software Usability Measurement Inventory (SUMI): A questionnaire-based usability test technique for measuring software quality from the end user's point of view. [Veenendaal04]	SUMI: Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
source statement: <i>See statement.</i>	kildeinstruksjon: <i>Se (eng.) statement.</i>
specification: A document that specifies, ideally in a complete, precise and verifiable manner, the requirements, design, behavior, or other characteristics of a component or system, and, often, the procedures for determining whether these provisions have been satisfied. [IEEE 610]	spesifikasjon: <i>Se kravspesifikasjon.</i>
specification-based testing: <i>See black box testing.</i>	spesifikasjonsbasert testing: <i>Se svart boks testing.</i>
specification-based technique: <i>See black box test design technique.</i>	spesifikasjonsbasert teknikk: <i>Se svart boks testdesignteknikk.</i>
specification-based test design technique: <i>See black box test design technique.</i>	spesifikasjonsbasert testdesignteknikk: <i>Se svart boks testdesignteknikk.</i>
specified input: An input for which the specification predicts a result.	spesifisert input: En input for hvilken spesifisering forutsier et resultat.
SPI: <i>See Software Process Improvement.</i>	SPI: <i>Se Software prosess forbedring</i>
stability: The capability of the software product to avoid unexpected effects from modifications in the software. [ISO 9126] <i>See also maintainability.</i>	stabilitet: Programvareproduktets evne til å unngå uventede følger av endringer i programvaren. [ISO 9126] <i>Se også vedlikeholdbarhet.</i>
staged representation: A model structure wherein attaining the goals of a set of process areas establishes a maturity level; each level builds a foundation for subsequent levels. [CMMI]	trinnvis fremstilling: En modellstruktur der oppnåelse av mål for et sett av prosessområder etablerer et samlet modenhetsnivå. Hvert nivå bygger basis for etterfølgende nivåer. [CMMI]
standard: Formal, possibly mandatory, set of requirements developed and used to prescribe consistent approaches to the way of working or to provide guidelines (e.g., ISO/IEC standards, IEEE standards, and organizational standards). [CMMI]	standard: Formell, muligens obligatorisk, samling av krav som er utviklet og brukt for å foreskrive konsistente framgangsmåter når det gjelder måten å arbeide på eller for å gi retningslinjer. (eks. ISO/IEC standarder, IEEE standarder og organisasjonsstandarder) [CMMI]
standard software: <i>See off-the-shelf software.</i>	standardprogram
standards testing: <i>See compliance testing.</i>	test mot standarder: <i>Se (eng.) compliance testing.</i>

state diagram: A diagram that depicts the states that a component or system can assume, and shows the events or circumstances that cause and/or result from a change from one state to another. [IEEE 610]	tilstandsdiagram: Et diagram som viser tilstandene som en komponent eller et system kan anta, og som viser hendelsene eller omstendighetene som fører til og/eller resulterer fra en endring fra en tilstand til en annen. [IEEE 610]
state table: A grid showing the resulting transitions for each state combined with each possible event, showing both valid and invalid transitions.	tilstandstabell: En tabell som viser overgangene for hver tilstand for hver hendelse som viser gyldige og ugyldige overganger.
state transition: A transition between two states of a component or system.	tilstandsovergang: En overgang mellom to tilstander av en komponent eller et system.
state transition testing: A black box test design technique in which test cases are designed to execute valid and invalid state transitions. See also <i>N-switch testing</i> .	tilstandsbasert testing: En svart boks testdesignteknikk der testtilfeller blir laget for å utføre gyldige og ugyldige tilstandsoverganger. Se også (eng.) <i>N-switch-testing</i> .
statement: An entity in a programming language, which is typically the smallest indivisible unit of execution.	programinstruksjon: En enhet i et programmeringsspråk som typisk er den minste udelelige enheten for utførelse.
statement coverage: The percentage of executable statements that have been exercised by a test suite.	programinstruksjonsdekning: Prosenten utførbare programinstruksjoner som er blitt utført av en testsuite.
statement testing: A white box test design technique in which test cases are designed to execute statements.	programinstruksjonstesting: En strukturell testdesignteknikk der testtilfelle blir laget for å utføre programinstruksjoner.
static analysis: Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software artifacts. Static analysis is usually carried out by means of a supporting tool.	statisk analyse: Analyse av programvareprodukter, for eksempel krav eller kode, som gjøres uten å utføre disse. Vanligvis utføres statisk analyse ved hjelp av et verktøy.
static analysis tool: See <i>static analyzer</i> .	statisk analyseverktøy: Se <i>statisk analysator</i> .
static analyzer: A tool that carries out static analysis.	statisk analysator: Et verktøy som utfører statisk analyse.
static code analysis: Analysis of source code carried out without execution of that software.	statisk kodeanalyse: Analyse av kildekoden som utføres uten å utføre koden.
static code analyzer: A tool that carries out static code analysis. The tool checks source code, for certain properties such as conformance to coding standards, quality metrics or data flow anomalies.	statisk kodeanalysator: Et verktøy som utfører statisk kodeanalyse. Verktøyet kontrollerer kildekoden for bestemte egenskaper, som for eksempel samsvar med kodestandarder, kvalitetsmål og dataflytanomalier.

static testing: Testing of a component or system at specification or implementation level without execution of that software, e.g. reviews or static analysis.	statisk testing: Testing av en komponent eller et system, for eksempel krav eller kode, som gjøres uten å utføre programvare, for eksempel gjennomgang eller statisk analyse.
statistical testing: A test design technique in which a model of the statistical distribution of the input is used to construct representative test cases. See also <i>operational profile testing</i> .	statistisk testing: En testdesignteknikk der en bruker en modell av den statistiske fordelingen av input for å lage representative testtilfelle. Typisk vil denne fordelingen følge en bruksprofil.
status accounting: An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of the approved changes. [IEEE 610]	statusoppfølging: Et element av konfigurasjonsstyringen som består av å beskrive informasjonen som behøves for å styre en konfigurasjon effektivt. Informasjonen inkluderer en liste over godkjent konfigurasjonsidentifikasjon, status til forslåtte endringer til konfigurasjonen, og implementeringsstatus av de endringer som er godkjent. [IEEE 610]
STEP: See <i>Systematic Test and Evaluation Process</i> .	STEP: Se (eng.) <i>Systematic Test and Evaluation Process</i> .
storage: See <i>resource utilization</i> .	minne, lagringsplass: Se <i>ressursbruk</i> .
storage testing: See <i>resource utilization testing</i> .	minnetesting: Se <i>ressursbrukstesting</i> .
stress testing: A type of performance testing conducted to evaluate a system or component at or beyond the limits of its anticipated or specified work loads, or with reduced availability of resources such as access to memory or servers. [IEEE 610] See also <i>performance testing, load testing</i> .	stresstesting: En type ytelsestesting som brukes for å evaluere et system eller en komponent på eller utenfor grensene av dets antatte eller spesifiserte belastning, eller med redusert tilgjengelighet av ressurser som minne eller servere. [IEEE 610] Se også <i>ytelsestesting, belastningstesting</i> .
stress testing tool: A tool that supports stress testing.	stresstestverktøy: Et verktøy som støtter stresstesting.
structural coverage: Coverage measures based on the internal structure of a component or system.	strukturell dekning: Testdekningsmål som baserer seg på den interne strukturen av et testobjekt.
structural test design technique: See <i>white box test design technique</i> .	strukturell testdesignteknikk: Testdesignteknikk som baserer seg på den interne strukturen av et testobjekt.
structural testing: See <i>white box testing</i> .	strukturell testing: Test som baserer seg på den interne strukturen av et testobjekt. Typisk måles den oppnåelse av strukturelle testdekningskriterier.
structure-based test design technique: See <i>white box test design technique</i> .	strukturbasert teknikk: Se <i>hvit boks test design teknikk</i> .
structure based testing: See <i>white box testing</i> .	strukturbasert testing: Se <i>hvit boks testing</i> .

structured walkthrough: See <i>walkthrough</i> .	strukturert gjennomgang
stub: A skeletal or special-purpose implementation of a software component, used to develop or test a component that calls or is otherwise dependent on it. It replaces a called component. [IEEE 610]	stubb: Et program som simulerer et program som testobjektet kaller opp. Stubber brukes i inkrementell integrasjonstesting. [IEEE 610]
subpath: A sequence of executable statements within a component.	understi: En følge av utførbare programinstruksjoner i en komponent.
suitability: The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives. [ISO 9126] See also <i>functionality</i> .	velegnethet
suitability testing: The process of testing to determine the suitability of a software product.	test av velegnethet: Test for å bestemme om et produkt er velegnet.
suspension criteria: The criteria used to (temporarily) stop all or a portion of the testing activities on the test items. [IEEE 829]	avbruddskriterier: Kriteriene som brukes for (midlertidig) å stoppe hele testen eller en del av den på testobjektet. [IEEE 829]
syntax testing: A black box test design technique in which test cases are designed based upon the definition of the input domain and/or output domain.	syntakstesting: En svart boks testdesignteknikk der testtilfelle blir konstruert basert på definisjonen av protokoller eller syntaksen i kommandospråk eller input eller output.
system: A collection of components organized to accomplish a specific function or set of functions. [IEEE 610]	system: En samling komponenter organisert for å oppnå en spesifikk funksjon eller et sett av funksjoner. [IEEE 610]
system integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).	systemintegrasjonstest: (1) Testingen av integrasjonen av systemet med sin omgivelse, for eksempel eksterne programmer. (2) En formell test for å avklare om hele ”storsystemet” virker, både med hensyn til interne og eksterne grensesnitt. Fokuserer på å avdekke feil i samspillet mellom systemer [tidl. DND terminologiliste].
system of systems: Multiple heterogeneous, distributed systems that are embedded in networks at multiple levels and in multiple domains interconnected addressing large-scale inter-disciplinary common problems and purposes.	system av systemer: Flere heterogene, distribuerte systemer som er lagt i nettverk på flere nivåer og i ulike anvendelsesområder og som er forbundet for å løse store interdisiplinære felles problemer og mål.

<p>system testing: The process of testing an integrated system to verify that it meets specified requirements. [Hetzel]</p>	<p>systemtest: (1) Prosessen med å teste et integrert system for å verifisere at det oppfyller spesifiserte krav. [Hetzel] (2) En formell test for å avklare om et system virker som beskrevet i kravspesifikasjonen. Kan innbefatte spesielle ikke-funksjonelle testtyper for å få testet ifølge kravspesifikasjonen /forretningsprosesser. For eksempel volumtest, stresstest, ytelsestest, sikkerhetstest, installasjonstest, ”online-hjelp”-/brukermanualtest etc. (etter tidl. DND terminologiliste).</p>
<p>Systematic Test and Evaluation Process: A structured testing methodology, also used as a content-based reference model for improving the testing process. Systematic Test and Evaluation Process (STEP) does not require that improvements occur in a specific order.</p>	<p>Se (eng.) STEP. Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.</p>

T

<p>technical review: A peer group discussion activity that focuses on achieving consensus on the technical approach to be taken. A technical review is also known as a peer review. [Gilb and Graham, IEEE 1028] See also <i>peer review</i></p>	<p>teknisk gjennomgang: En diskusjon i en gruppe av medarbeidere som fokuserer på å oppnå konsensus på den tekniske løsningen. [Gilb and Graham, IEEE 1028] Se også (eng.) <i>peer review</i>.</p>
<p>test: A set of one or more test cases [IEEE 829]</p>	<p>test: En samling bestående av et eller flere testtilfeller. [IEEE 829] Se også <i>testsuite</i>.</p>
<p>test approach: The implementation of the test strategy for a specific project. It typically includes the decisions made that follow based on the (test) project’s goal and the risk assessment carried out, starting points regarding the test process, the test design techniques to be applied, exit criteria and test types to be performed.</p>	<p>tilnærming til test, testmetode: Implementeringen av teststrategien for et spesifikt prosjekt. Dette inkluderer typisk beslutningene som tas basert på testprosjektets mål og risikovurderingen, starttidspunktet for testprosessen, testdesignteknikkene som skal brukes, sluttkriterier og testtyper som skal anvendes.</p>
<p>test automation: The use of software to perform or support test activities, e.g. test management, test design, test execution and results checking.</p>	<p>testautomatisering: Bruk av programvare for å utføre eller støtte testaktiviteter, for eksempel testledelse, testdesign, testutføring og kontroll av resultater.</p>

<p>test basis: All documents from which the requirements of a component or system can be inferred. The documentation on which the test cases are based. If a document can be amended only by way of formal amendment procedure, then the test basis is called a frozen test basis. [TMap]</p>	<p>testbasis: Alle dokumenter som en kan avlede krav til systemet eller komponenten som skal testes. Dokumentasjonen som testen blir basert på, for eksempel krav- eller designspesifikasjon. Hvis et dokument bare kan endres gjennom en formell prosess, blir det kalt frosset test basis. [TMap]</p>
<p>test bed: See <i>test environment</i>.</p>	<p>testomgivelse: Se <i>testmiljø</i>.</p>
<p>test case: A set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement. [IEEE 610]</p>	<p>testtilfelle: (1) En beskrivelse av forutsetninger, aksjoner, inndata og forventede resultater utviklet for å verifisere en spesiell del av et program eller for å teste om et testobjekt oppfyller et bestemt krav. (tidl. DND terminologiliste) (2) Et sett med inputverdier, forutsetninger for utførelsen, forventede resultater og etterbetingelser for utføringen, som er utviklet for et bestemt mål eller en testbetingelse, som for eksempel å utføre en bestemt sti gjennom et program eller å verifisere samsvar med et spesifikt krav. [IEEE 610].</p>
<p>test case design technique: See <i>test design technique</i>.</p>	<p>testdesignteknikk</p>
<p>test case specification: A document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item. [IEEE 829]</p>	<p>testtilfelle spesifikasjon: Et dokument som spesifiserer et sett med testtilfeller (mål, forutsetninger, input, aksjoner, forventede resultater) for en test. [IEEE 829]</p>
<p>test case suite: See <i>test suite</i>.</p>	<p>suite av testtilfelle: Se <i>testsuite</i>.</p>
<p>test charter: A statement of test objectives, and possibly test ideas about how to test. Test charters are used in exploratory testing. See also <i>exploratory testing</i>.</p>	<p>testoppgave: En liste av testmål og eventuelt ideer om hvordan en skal teste disse. Testoppgaver brukes for eksempel ofte i eksplorativ testing. Se også <i>eksplorativ testing</i>.</p>
<p>test closure: During the test closure phase of a test process data is collected from completed activities to consolidate experience, test ware, facts and numbers. The test closure phase consists of finalizing and archiving the test ware and evaluating the test process, including preparation of a test evaluation report. See also <i>test process</i>.</p>	<p>testavslutning: Fase i en testprosess der data blir samlet fra de utførte aktivitetene for å oppsummere erfaringer, testmateriale, fakta og tall. Denne fasen består av å slutføre og arkivere testvare (testmaterialet) og å evaluere testprosessen som er gjennomført, og å lage en evalueringsrapport om testingen. Se også (eng.) <i>test process</i>.</p>
<p>test comparator: A test tool to perform automated test comparison of actual results with expected results.</p>	<p>sammenligningsverktøy: Et verktøy for automatisk sammenligning av faktisk resultat mot forventet resultat.</p>

<p>test comparison: The process of identifying differences between the actual results produced by the component or system under test and the expected results for a test. Test comparison can be performed during test execution (dynamic comparison) or after test execution.</p>	<p>testsammenligning: Prosessen med å identifisere forskjeller mellom aktuelle resultater som er produsert av et testobjekt under test og forventede resultater for denne testen. Testsammenligning kan bli utført under testutføringen (dynamisk sammenligning) eller etterpå.</p>
<p>test completion criteria: See <i>exit criteria</i>.</p>	<p>sluttkriterier</p>
<p>test condition: An item or event of a component or system that could be verified by one or more test cases, e.g. a function, transaction, quality attribute, or structural element. (i.e. “something to be tested”)</p>	<p>testbetingelse: En del av et testobjekt som kan bli verifisert av en eller flere testtilfelle, for eksempel en funksjon, transaksjon, kvalitetsegenskap eller strukturelement.</p> <p>Et kriterium som må testes for å verifisere om et logisk behandlingsløp i en funksjon fungerer i henhold til spesifikasjonene (tidl. DND terminologiliste).</p> <p>Kort: Noe som skal testes.</p>
<p>test coordinator (not in official English dictionary)</p>	<p>testkoordinator: En person som koordinerer de personene som utfører test og testaktivitetene (tidl. DND terminologiliste).</p>
<p>test control: A test management task that deals with developing and applying a set of corrective actions to get a test project on track when monitoring shows a deviation from what was planned. See also <i>test management</i>.</p>	<p>teststyring: En testledelsesoppgave som omhandler å utvikle og bruke et sett med korrigerende tiltak for å få et testprosjekt til å komme på rett spor igjen når en ser at avvik fra planen har skjedd. Se også testledelse.</p>
<p>test coverage: See <i>coverage</i>.</p>	<p>testdekning: I hvilken grad data faktisk verifiserer et spesifikt krav for et gitt testobjekt, eller i hvilken grad all kode er utført under testen (tidl. DND terminologiliste).</p>
<p>test cycle: Execution of the test process against a single identifiable release of the test object.</p>	<p>testsyklus:</p> <ol style="list-style-type: none"> 1. Utføringen av testprosessen mot en enkel identifiserbar frigivelse av et testobjekt. 2. Utføringen av en serie testtilfelle. 3. Utføringen av den generelle testprosessen for nøyaktig en versjon av testobjektet. På slutten av dette har en oppdrag for reparasjon av feil eller endringer til utvikleren.

<p>test data: Data that exists (for example, in a database) before a test is executed, and that affects or is affected by the component or system under test.</p>	<p>testdata: Data som eksisterer (for eksempel i en database eller fil) før en test blir utført og som har innflytelse på eller blir influert av testobjektet under testen. Data spesielt konstruert på forhånd for å gjøre det mulig å gjennomføre handlingene i testtilfelle osv. Testdata kan kopieres fra produksjon eller andre steder, og eventuelt manipuleres, eller de kan konstrueres (tidl. DND terminologiliste).</p>
<p>test data preparation tool: A type of test tool that enables data to be selected from existing databases or created, generated, manipulated and edited for use in testing.</p>	<p>verktøy for å lage testdata: Et testverktøy som tillater å hente data fra eksisterende databaser, eller genererer, manipulerer eller endrer disse for bruk i testingen.</p>
<p>test deliverable: Any test (work) product that must be delivered to someone other than the test (work) product's author. See also <i>deliverable</i>.</p>	<p>testleveranse: Et arbeidsprodukt av testarbeid som skal leveres til en annen enn produktets forfatter. Se også leveranse.</p>
<p>test design: (1) See <i>test design specification</i>. (2) The process of transforming general testing objectives into tangible test conditions and test cases.</p>	<p>testdesign: (1) Se testdesignspesifikasjon. (2) Aktiviteten med å spesifisere detaljene i en test.</p>
<p>test design specification: A document specifying the test conditions (coverage items) for a test item, the detailed test approach and identifying the associated high level test cases. [IEEE 829] (i.e. a document specifying WHAT is to be tested)</p>	<p>testdesignspesifikasjon: Et dokument som spesifiserer testbetingelsene (som skal dekkes) for et testobjekt, den detaljerte måten å teste på og identifiserer de tilhørende høynivå testtilfelle. [IEEE 829]. Et dokument som detaljer teststrategien for en egenskap eller kombinasjon av egenskaper og som identifiserer de testene som skal brukes for å verifisere disse egenskapene (tidl. DND terminologiliste).</p>
<p>test design technique: Procedure used to derive and/or select test cases.</p>	<p>teknikk for testdesign, testdesignteknikk: Prosedyre som brukes for å utlede og/eller velge testtilfelle.</p>
<p>test design tool: A tool that supports the test design activity by generating test inputs from a specification that may be held in a CASE tool repository, e.g. requirements management tool, from specified test conditions held in the tool itself, or from code.</p>	<p>testdesign verktøy: Et verktøy som understøtter testdesign ved å generere inputs fra en spesifikasjon som kan være i et CASE verktøy eller fra spesifiserte testbetingelser som er i verktøyet selv.</p>
<p>test driven development: A way of developing software where the test cases are developed, and often automated, before the software is being developed to run those test cases.</p>	<p>testdrevet utvikling: Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.</p>

test driver: See <i>driver</i> .	testdriver: Se <i>driver</i>
test environment: An environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test. [IEEE 610]	testmiljø: Et miljø som består av hardware, instrumentering, simulatorer, programvareverktøy og andre støtteelementer som behøves for å utføre en test. [IEEE 610]
test estimation: The calculated approximation of a result related to various aspects of testing (e.g. effort spent, completion date, costs involved, number of test cases, etc.) which is usable even if input data may be incomplete, uncertain, or noisy.	test estimering: Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
test evaluation report: A document produced at the end of the test process summarizing all testing activities and results. It also contains an evaluation of the test process and lessons learned.	testevalueringsrapport: Et dokument som blir produsert på slutten av testprosessen og som oppsummerer alle testaktiviteter og resultater. Det omfatter også en evaluering av testprosessen og erfaring.
test execution: The process of running a test on the component or system under test, producing actual result(s).	testgjennomføring: Prosessen med å utføre en test av et testobjekt der en produserer faktiske resultater.
test execution automation: The use of software, e.g. capture/playback tools, to control the execution of tests, the comparison of actual results to expected results, the setting up of test preconditions, and other test control and reporting functions.	automatisering av testgjennomføring: Bruk av automatiske verktøy for å utføre en eller flere testtilfeller, sammenligne faktiske resultater med forventede resultater og sette opp forutsetningene for testen samt andre funksjoner for å styre og rapportere.
test execution phase: The period of time in a software development life cycle during which the components of a software product are executed, and the software product is evaluated to determine whether or not requirements have been satisfied. [IEEE 610]	testgjennomføringsfase: Tidsperioden i programvarelivssyklusen der komponentene av et programvareprodukt blir utført og der produktet evalueres for å bestemme om krav har blitt oppfylt eller ikke. [IEEE 610]
test execution schedule: A scheme for the execution of test procedures. The test procedures are included in the test execution schedule in their context and in the order in which they are to be executed.	kjøreplan: En plan for utføringen av testprosedyrene som viser i hvilken rekkefølge og når disse skal utføres.
test execution technique: The method used to perform the actual test execution, either manually or automated.	testutføringsteknikk: Metoden som brukes for å utføre testen, enten manuelt eller automatisk.
test execution tool: A type of test tool that is able to execute other software using an automated test script, e.g. capture/playback. [Fewster and Graham]	testutføringsverktøy: En type testverktøy som kan utføre annen programvare ved hjelp av et test script. [Fewster og Graham]
test fail: See <i>fail</i> .	feilet test: En test der forventet og aktuelt resultat ikke stemmer overens.

test generator: See <i>test data preparation tool</i> .	testgenerator: Et verktøy som genererer test input data for en test.
test harness: A test environment comprised of stubs and drivers needed to execute a test.	testramme: Et testmiljø som består av stubber og drivere for å utføre en test.
test implementation: The process of developing and prioritizing test procedures, creating test data and, optionally, preparing test harnesses and writing automated test scripts.	testimplementering: Prosessen med å utvikle og prioritere testprosedyrer, laging av testdata og, valgfritt, forberedelse av testrammer og skriving av automatiske testscript.
test incident: See <i>incident</i> .	hendelse: Se <i>problem</i> .
test incident report: See <i>incident report</i> .	problemrapport
test infrastructure: The organizational artifacts needed to perform testing, consisting of test environments, test tools, office environment and procedures.	testinfrastruktur: De nødvendige ting som trenges for å utføre testingen, dvs. testmiljøer, verktøy, kontorer og prosedyrer.
test input: The data received from an external source by the test object during test execution. The external source can be hardware, software or human.	testinput: Data som testobjekt mottar fra en ekstern kilde under gjennomføring av test. Kilden kan være maskinvare, programvare eller personer.
test item: The individual element to be tested. There usually is one test object and many test items. See also <i>test object</i> .	testelement: Det individuelle elementet som skal testes. Vanligvis er det et testobjekt og flere testelementer. Se også testobjekt.
test item transmittal report: See <i>release note</i> .	leveringsrapport: Se (eng.) <i>release note</i> .
test leader: See <i>test manager</i> .	Se <i>testleder</i> .
test level: A group of test activities that are organized and managed together. A test level is linked to the responsibilities in a project. Examples of test levels are component test, integration test, system test and acceptance test. [TMap]	testnivå: (1) En gruppe med testaktiviteter som er organisert og ledet sammen. Et testnivå hører vanligvis til et abstraksjonsnivå i prosjektet (utviklingsmodellen). Eksempler er enhetstest eller modultest, integrasjonstest, systemtest og akseptansetest. Nivåene som skal brukes, tilpasset det gjeldende system, spesifiseres i teststrategien (hovedtestplan). (2) De nivåer eller avgrensede tester som et system eller endringer bør testes gjennom etter gjeldende utviklingsmodell. Nivåene som skal brukes, tilpasset gjeldende testobjekt, spesifiseres i teststrategien . Innenfor hvert enkelt testnivå er det et sett med anbefalte testtyper. (tidl. DND terminologiliste)
test log: A chronological record of relevant details about the execution of tests. [IEEE 829]	testlogg: En kronologisk oversikt av relevante detaljer om testgjennomføring. [IEEE 829]
test logging: The process of recording information about tests executed into a test log.	testlogging: Prosessen med å skrive en testlogg.

test management: The planning, estimating, monitoring and control of test activities, typically carried out by a test manager.	testledelse: Planlegging, estimering, oppfølging og styring av testaktiviteter, typisk utført av en testleder.
test management tool: A tool that provides support to the test management and control part of a test process. It often has several capabilities, such as testware management, scheduling of tests, the logging of results, progress tracking, incident management and test reporting.	testledelsesverktøy: Et verktøy som gir støtte til testledelse og styring av testprosessen. Vanligvis har slike verktøy flere funksjoner om styring av testmaterialet, tidsbestemmelse av tester, resultatlogging, framdriftsoppfølging, feilhåndtering og testrapportering.
test manager: The person responsible for project management of testing activities and resources, and evaluation of a test object. The individual, who directs, controls, administers plans and regulates the evaluation of a test object.	testleder: Personen i prosjektledelsen som har ansvar for testaktiviteter og ressurser og evaluering av et testobjekt. Personen som planlegger, styrer, følger opp og bistår til evaluering av testobjekt.
Test Maturity Model (TMM): A five level staged framework for test process improvement, related to the Capability Maturity Model (CMM) that describes the key elements of an effective test process.	test modenhetsmodell (TMM): Et fem trinns rammeverk for test prosess forbedring som er relatert til CMM og som beskriver nøkkelementene i en effektiv testprosess.
Test Maturity Model Integrated (TMMi): A five level staged framework for test process improvement, related to the Capability Maturity Model Integration (CMMI) that describes the key elements of an effective test process.	test modenhetsmodell, integrert (TMMi): Et fem trinns rammeverk for test prosess forbedring som er relatert til CMMI og som beskriver nøkkelementene i en effektiv testprosess.
test monitoring: A test management task that deals with the activities related to periodically checking the status of a test project. Reports are prepared that compare the actual results to that which was planned. See also <i>test management</i> .	testoppfølging: En testledelsesoppgave som omhandler at en periodevis sjekker status av testprosjektet. En sammenligner de virkelige resultatene til det som var planlagt. Se også <i>testledelse</i> .
test object: The component or system to be tested. See also <i>test item</i> .	testobjekt: (1) Komponentene eller systemet som er gjenstand for testing. (2) En programvarekomponent eller en samling av programvare som er gjenstand for testing. (tidl. DND terminologiliste).
test objective: A reason or purpose for designing and executing a test.	testmål: Grunnen eller mål for å utvikle og utføre en test.
test oracle: A source to determine expected results to compare with the actual result of the software under test. An oracle may be the existing system (for a benchmark), other software, a user manual, or an individual's specialized knowledge, but should not be the code. [Adrion]	testorakel: En mekanisme, kilde eller verktøy for å bestemme forventede resultater som skal brukes til å sammenligne med aktuelle resultater når programvare utføres under testen. Et orakel kan være det eksisterende system (for referanse), et annet system som gjør liknende eller motsatte oppgaver, bruksanvisningen, eller en persons spesielle kunnskap, men burde ikke være koden selv.

test outcome: <i>See result.</i>	testresultat: <i>Se resultat.</i>
test organization: (not in official English terminology)	testorganisasjon: En måte å beskrive roller knyttet til testarbeide. (tidl. DND terminologiliste)
test pass: <i>See pass.</i>	godkjent test: <i>Se godkjent.</i>
test performance indicator: A high level metric of effectiveness and/or efficiency used to guide and control progressive test development, e.g. Defect Detection Percentage (DDP).	testkvalitetsindikator: En måleverdi, vanligvis på et høyt nivå, som indikerer til hvilken grad et mål eller kriterium er oppfylt. Ofte relatert til test prosess forbedring. Eksempelvis feilfinningsprosenten.
test phase: A distinct set of test activities collected into a manageable phase of a project, e.g. the execution activities of a test level. [Gerrard]	testfase: Et sett med testaktiviteter som blir satt sammen til en fase i et prosjekt, for eksempel testutførelsen av et visst testnivå. [Gerrard]
test plan: A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. [IEEE 829]	testplan: Et dokument som beskriver omfang (hva som skal testes), tilnærming til test, ressurser og tidsplan for planlagte testaktiviteter. Planen identifiserer testobjekter, hva som skal testes, testoppgavene og hvem som skal utføre disse, testernes grad av uavhengighet, testmiljøet, testdesignteknikker og testmåleteknikker som skal brukes og begrunnelsen for deres valg, og beskriver risikoene og planene for deres inntreden. Testplanen er en dokumentasjon av testplanleggingen. [IEEE 829]
test planning: The activity of establishing or updating a test plan.	testplanlegging: Aktivitet som innebærer etablering eller oppdatering av en testplan.
Test Point Analysis (TPA): A formula based test estimation method based on function point analysis. [TMap]	testpunktanalyse: En modell for testestimering basert på funksjonspunktanalyse. [TMap]
test policy: A high level document describing the principles, approach and major objectives of the organization regarding testing.	testpolicy: Et høynivå dokument som beskriver prinsippene, framgangsmåten og organisasjonens hovedmål angående testing.
test procedure: <i>See test procedure specification.</i>	testprosedyre: Også testmanuskript
test procedure specification: A document specifying a sequence of actions for the execution of a test. Also known as test script or manual test script. [IEEE 829]	testprosedyre: Et dokument som spesifiserer en sekvens av handlinger med tilhørende kontroller relatert til testtilfeller for å utføre en test. En beskrivelse av hvordan testen skal gjennomføres. Dette er også kjent som test script (for automatisk utførelse) eller manuelt test script eller testinstruksjon. [IEEE 829]. Også testmanuskript .

test process: The fundamental test process comprises test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities.	testprosess: Den fundamentale testprosessen inneholder planlegging og styring, analyse og design, implementering og gjennomføring, evaluering av sluttkriterier og rapportering samt etterarbeid.
Test Process Improvement (TPI): A continuous framework for test process improvement that describes the key elements of an effective test process, especially targeted at system testing and acceptance testing.	testprosessforbedring: Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
test process improvement manifesto: A statement that echoes the agile manifesto, and proposes defines values for improving the testing process. The values are: <ul style="list-style-type: none"> - flexibility over detailed processes - best Practices over templates - deployment orientation over process orientation - reviews over quality assurance (departments) - business driven over model driven. [Veenendaal08]	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
test process improver: A person implementing improvements in the test process based on a test improvement plan.	Testprosess pådriver: En person som implementerer forbedringer i testprosessen basert på en plan til testforbedring.
test progress report: A document summarizing testing activities and results, produced at regular intervals, to report progress of testing activities against a baseline (such as the original test plan) and to communicate risks and alternatives requiring a decision to management.	test framdriftsrapport: Et dokument som oppsummerer testaktiviteter og resultater, som er produsert med visse mellomrom, for å rapportere framdriften av testaktiviteter mot planlagte (som for eksempel den originale testplanen) og for å kommunisere risiko og valgmuligheter som krever en beslutning av ledelsen.
test record: See <i>test log</i> .	testdagbok: Se <i>testlogg</i> .
test recording: See <i>test logging</i> .	testlogging
test report: See <i>test summary report</i> and <i>test progress report</i> .	testrapport: Se <i>testsluttrapport</i> .
test reproducibility: An attribute of a test indicating whether the same results are produced each time the test is executed.	gjentagbarhet av en test: En egenskap av en test som indikerer om en kan produsere de samme resultatene hver gang en utfører denne testen.
test requirement: See <i>test condition</i> .	testkrav.
test result: See <i>result</i> .	testresultat: Se <i>resultat</i> .
test rig: See <i>test environment</i> .	testomgivelse: Se <i>testmiljø</i> .

test run: Execution of a test on a specific version of the test object.	testkjøring: Utførelse av en test på en spesifikk versjon av testobjektet.
test run	testløp: Se <i>testkjøring</i> .
test run log: See <i>test log</i> .	Se <i>testlogg</i> .
test scenario: See <i>test procedure specification</i> .	testscenario
test schedule: A list of activities, tasks or events of the test process, identifying their intended start and finish dates and/or times, and interdependencies.	test fremdriftsplan: En liste aktiviteter, oppgaver eller hendelser i testprosessen, med identifikasjon av deres planlagte tid og/eller dato for start og slutt, samt deres gjensidige avhengigheter.
test script: Commonly used to refer to a test procedure specification, especially an automated one.	testscript: En testprosedyre, vanligvis for automatisk utførelse av testen.
test session: An uninterrupted period of time spent in executing tests. In exploratory testing, each test session is focused on a charter, but testers can also explore new opportunities or issues during a session. The tester creates and executes test cases on the fly and records their progress. See also <i>exploratory testing</i> .	testperiode: Et uavbrutt tidsrom som er brukt til å utføre tester. I eksplorativ testing er hver testperiode fokusert på en oppgave (charter), men testere kan også undersøke nye muligheter eller problemer under en slik testperiode. Testeren lager og utfører testtilfeller underveis og dokumenterer deres fremskritt. Se også <i>eksplorativ testing</i> .
test set: See <i>test suite</i> .	testsett: Se <i>testsuite</i> .
test situation: See <i>test condition</i> .	testsituasjon
test specification: A document that consists of a test design specification, test case specification and/or test procedure specification.	testspesifikasjon: Et dokument som beskriver hva som skal testes samt inkluderer eller refererer tilhørende testdesign, testtilfelle og testprosedyre.
test specification technique: See <i>test design technique</i> .	teknikk for testspesifikasjon
test stage: See <i>test level</i> .	testtrinn: Se <i>testnivå</i> .
test strategy: A high-level description of the test levels to be performed and the testing within those levels for an organization or program (one or more projects).	teststrategi: Et høynivådokument som definerer testnivåene som skal utføres og testingen innenfor disse nivåene for et eller flere prosjekter.
test suite: A set of several test cases for a component or system under test, where the post condition of one test is often used as the precondition for the next one.	testsuite: En samling av flere testtilfeller for et testobjekt der etterbetingelsen til en test ofte blir brukt som forutsetning for neste test.
test summary report: A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria. [IEEE 829]	testsluttrapport: Et dokument som oppsummerer testaktivitetene og resultatene. Det inneholder også en evaluering av testobjektet mot sluttkriteriene. [IEEE 829]
test target: A set of exit criteria.	testmål: En mengde med sluttkriterier.
test technique: See <i>test design technique</i> .	testteknikk: Teknikk som benyttes for å få konstruert en test på en hensiktsmessig måte.

<p>test tool: A software product that supports one or more test activities, such as planning and control, specification, building initial files and data, test execution and test analysis. [TMap] See also <i>CAST</i>.</p>	<p>testverktøy: Et produkt som understøtter en eller flere testaktiviteter, som planlegging, spesifikasjon, design, å lage testdata filer, utførelse av test, styring og testanalyse. Se også (eng.) <i>CAST</i>.</p>
<p>test type: A group of test activities aimed at testing a component or system regarding one or more interrelated quality attributes. A test type is focused on a specific test objective, i.e. reliability test, usability test, regression test etc., and may take place on one or more test levels or test phases. [TMap]</p>	<p>testtype: En gruppe testaktiviteter som har som mål å teste et testobjekt vedrørende et eller flere sammenhengende kvalitetsegenskaper. En testtype er fokusert på et spesifikt testmål, som pålitelighetstest, brukbarhetstest, ytelsestest, stresstest, volumtest, sikkerhetstest, regresjonstest etc. Testtyper kan utføres på et eller flere testnivåer. [TMap]</p>
<p>testability: The capability of the software product to enable modified software to be tested. [ISO 9126] See also <i>maintainability</i>.</p>	<p>testbarhet: (1) Et programvareprodukts evne til å muliggjøre modifisert programvare å bli testet. [ISO 9126] Se også <i>vedlikeholdbarhet</i>. (2) Vanskeligheten for feil å forbli skjult i et produkt. (3) Enkeltheten for testeren å teste et program. Til dette hører at testeren har adgang til å styre alle inputverdier som brukes av testobjektet samt at han kan observere alle outputs som testobjektet kan generere.</p>
<p>testability review: A detailed check of the test basis to determine whether the test basis is at an adequate quality level to act as an input document for the test process. [TMap]</p>	<p>testbarhetsgjennomgang: En detaljert kontroll av testbasis for å bestemme om den har passende kvalitet for å tjene som input dokument til en testprosess.</p>
<p>testable requirements: The degree to which a requirement is stated in terms that permit establishment of test designs (and subsequently test cases) and execution of tests to determine whether the requirements have been met. [IEEE 610]</p>	<p>testbart krav: Graden som et krav er beskrevet slik at det tillater konstruksjon av testtilfelle og utføring av en test for å bestemme om kravet er oppfylt. [IEEE 610]</p>
<p>tester: A skilled professional who is involved in the testing of a component or system.</p>	<p>tester: En faglig kvalifisert person som er involvert i testingen av en komponent eller et system.</p>
<p>testing: The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.</p>	<p>testing: Prosessen som består av alle livssyklusaktiviteter, både statiske og dynamiske, som omfatter planlegging, forberedelse og evaluering av programvareprodukter og relaterte arbeidsresultater for å fastslå om de oppfyller spesifiserte krav og/eller for å demonstrere at de er brukbare i praksis og for å finne feil.</p>

test ware: Artifacts produced during the test process required to plan, design, and execute tests, such as documentation, scripts, inputs, expected results, set-up and clear-up procedures, files, databases, environment, and any additional software or utilities used in testing. [Fewster and Graham]	testvare: Alt som trenges for å planlegge, designe eller gjennomføre tester. Kan innebære dokumentasjon, skript, input, forventede resultater, oppsett og oppryddingsprosedyrer, filer, databaser, miljø og alt ytterligere programvare og verktøy som brukes under en testprosess. [Fewster and Graham]
test ware	testmateriell: Alle gjenbrukbare testdokumenter, teststrategier, testplaner, testdesign, testprosedyre, stubber og drivere, filer med testdata osv som blir produsert som et resultat av testprosessen. Det kan være et krav at testmateriell kan gjenbrukes og være mulig å vedlikeholdes. (tidl. DND terminologiliste)
thread testing: A version of component integration testing where the progressive integration of components follows the implementation of subsets of the requirements, as opposed to the integration of components by levels of a hierarchy.	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
time behavior: See <i>performance</i> .	oppførsel vedr. tid: Se <i>ytelse</i> .
top-down testing: An incremental approach to integration testing where the component at the top of the component hierarchy is tested first, with lower level components being simulated by stubs. Tested components are then used to test lower level components. The process is repeated until the lowest level components have been tested. See also <i>integration testing</i> .	opp-ned test: En inkrementell måte å integrasjonsteste et system på, der en tester den komponenten som ligger i toppen først og lavere nivå komponenter simuleres av stubber. Testing går videre til neste lag ned i komponenthierarki. Prosessen gjentas til komponentene på det nederste nivået er testet.
Total Quality Management: An organization-wide management approach centered on quality, based on the participation of all its members and aiming at long-term success through customer satisfaction, and benefits to all members of the organization and to society. Total Quality Management consists of planning, organizing, directing, control, and assurance. [ISO 8402]	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
TPG: See <i>Test Process Group</i> .	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
TQM: See <i>Total Quality Management</i> .	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.

traceability: The ability to identify related items in documentation and software, such as requirements with associated tests. See also <i>horizontal traceability, vertical traceability</i> .	sporbarhet: Muligheten til å identifisere relaterte objekter i dokumentasjon og programvare, som krav med tilhørende tester. Se også <i>horisontal og vertikal sporbarhet</i> .
transactional analysis: The analysis of transactions between people and within people's minds; a transaction is defined as a stimulus plus a response. Transactions take place between people and between the ego states (personality segments) within one person's mind.	transaksjonsanalyse: Analyse av transaksjoner mellom mennesker og på innsiden av menneskesinnet. En transaksjon er definert som en stimulus sammen med en reaksjon. Transaksjoner foregår mellom mennesker og egotilstander (personlighetssegmenter) i personens sinn.
transcendent-based quality: A view of quality, wherein quality cannot be precisely defined, but we know it when we see it, or are aware of its absence when it is missing. Quality depends on the perception and affective feelings of an individual or group of individuals towards a product. [Garvin] See also <i>manufacturing-based quality, product-based quality, user-based quality, value-based quality</i> .	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.

U

understandability: The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use. [ISO 9126] See also <i>usability</i> .	forståelighet: Et programvareprodukts evne til å gi brukeren mulighet til å forstå om produktet er brukbart og hvordan det kan brukes for spesifikke oppgaver og betingelsene som knytter seg til bruken. [ISO 9216] Se også <i>brukbarhet</i> .
unit: See <i>component</i> .	enhet: Se <i>komponent</i> .
unit test framework: A tool that provides an environment for unit or component testing in which a component can be tested in isolation or with suitable stubs and drivers. It also provides other support for the developer, such as debugging capabilities. [Graham]	rammeverk for enhetstesting: Et verktøy som skaffer et miljø for enhetstesting hvor en enhet kan testes isolert ved hjelp av egnede stubber og drivere. Inneholder også andre former for støtte til utviklere, som muligheter for debugging.
unit testing: See <i>component testing</i> .	enhetstesting: Se <i>komponenttesting</i>
unreachable code: Code that cannot be reached and therefore is impossible to execute.	ikke utførbar kode: Kode som ikke kan nås under programutføringen og som derfor er umulig å utføre.

usability: The capability of the software to be understood, learned, used and attractive to the user when used under specified conditions. [ISO 9126]	brukervennlighet: Et programvareprodukts evne til å bli forstått, lært, brukt og å være tilfredsstillende til brukeren når det blir brukt under spesifiserte betingelser. [ISO 9126]
usability testing: Testing to determine the extent to which the software product is understood, easy to learn, easy to operate and attractive to the users under specified conditions. [ISO 9126]	brukbarhetstesting, brukervennlighetstesting, test av brukervennlighet: (på norsk brukes alle tre begreper i samme betydning). Testing for å finne ut i hvilken grad er programvareprodukt lett å lære, å forstå, lett å drifte og tilfredsstillende for brukerne under spesifiserte betingelser. [ISO 9126]
use case: A sequence of transactions in a dialogue between a user and the system with a tangible result.	brukstilfelle: En rekkefølge av transaksjoner i dialogen mellom en bruker og et system med konkrete resultater.
use case testing: A black box test design technique in which test cases are designed to execute user scenarios.	brukstilfelletesting: En svart boks testdesignteknikk der testtilfeller utformes for å utføre brukerscenarier.
user acceptance testing: See <i>acceptance testing</i> .	brukerakseptansetesting: Se <i>akseptansetesting</i> .
user scenario testing: See <i>use case testing</i> .	brukersituasjonstesting: Se <i>use case testing</i> .
user test: A test whereby real-life users are involved to evaluate the usability of a component or system.	brukertest: En test der reelle brukere er involvert for å evaluere brukervennlighet av en komponent eller et system.

V

V-model: A framework to describe the software development life cycle activities from requirements specification to maintenance. The V-model illustrates how testing activities can be integrated into each phase of the software development life cycle.	V-modell: Et rammeverk for å beskrive livssyklusen for programvare fra kravspesifikasjon til vedlikehold. V-modellen illustrerer hvordan testaktiviteter kan bli integrert i hver fase av programvarens livssyklus.
variable: An element of storage in a computer that is accessible by a software program by referring to it by a name.	variabel: Et lagret dataelement i en datamaskin som et program kan ha tilgang til ved å referere det med et navn.
verification: Confirmation by examination and through the provision of objective evidence that specified requirements have been fulfilled. [ISO 9000]	verifisering: Bekreftelse ved hjelp av undersøkelse og samling av objektiv informasjon at spesifiserte krav har blitt oppfylt. [ISO 9000].
version control: See <i>configuration control</i> .	versjonskontroll: Se <i>konfigurasjonskontroll</i> .

vertical traceability: The tracing of requirements through the layers of development documentation to components.	vertikal sporbarhet: Sporing av krav gjennom nivåer av utviklingsdokumentasjonen til komponentene.
volume testing: Testing where the system is subjected to large volumes of data. See also <i>resource-utilization testing</i> .	volumtest: Test av systemets oppførsel ved prosessering av store datamengder. Se også <i>ressursbrukstesting</i> .

W

walkthrough: A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content. [Freedman and Weinberg, IEEE 1028] See also <i>peer review</i> .	gjennomgang: En detaljert presentasjon som gjøres av forfatteren av et dokument for å samle informasjon og synspunkter og for å etablere en felles forståelse av dets innhold.
white-box technique: See <i>white-box test design technique</i> .	hvit boks teknikk: Se også <i>hvit boks testdesignteknikk</i> .
white-box test design technique: Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.	hvit boks testdesignteknikk: Dokumentert prosedyre for å avlede og konstruere testtilfelle basert på en analyse av komponentens eller systemet interne struktur.
white-box testing: Testing based on an analysis of the internal structure of the component or system.	hvit boks testing: Testing basert på analysen av komponentens eller systemets interne struktur. Se også <i>strukturell testing</i> .
Wide Band Delphi: An expert based test estimation technique that aims at making an accurate estimation using the collective wisdom of the team members.	Ikke oversatt til norsk da dette begrepet bare brukes på høyere nivå ISTQB-syllabi.
wild pointer: A pointer that references a location that is out of scope for that pointer or that does not exist. See also <i>pointer</i> .	peker til intet: En peker som peker på et sted utenfor definisjonsområdet for denne pekeren eller til et sted som ikke eksisterer. Se også <i>peker</i> .

Annex A (Informative)

Index of sources; the following non-normative sources were used in constructing this glossary:

- [Abbott] J. Abbot (1986), *Software Testing Techniques*, NCC Publications.
- [Adrion] W. Adrion, M. Branstad and J. Cherniabsky (1982), Validation, Verification and Testing of Computer Software, in: *Computing Surveys*, Vol. 14, No 2, June 1982.
- [Bach] J. Bach (2004), Exploratory Testing, in: E. van Veenendaal, *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.
- [Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442-20672-0
- [Chow] T. Chow (1978), Testing Software Design Modeled by Finite-Sate Machines, in: *IEEE Transactions on Software Engineering*, Vol. 4, No 3, May 1978.
- [CMM] M. Paulk, C. Weber, B. Curtis and M.B. Chrissis (1995), *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison-Wesley, ISBN 0-201-54664-7
- [CMMI] M.B. Chrissis, M. Konrad and S. Shrum (2004), *CMMI, Guidelines for Process Integration and Product Improvement*, Addison Wesley, ISBN 0-321-15496-7
- [Fenton] N. Fenton (1991), *Software Metrics: a Rigorous Approach*, Chapman & Hall, ISBN 0-53249-425-1
- [Fewster and Graham] M. Fewster and D. Graham (1999), *Software Test Automation, Effective use of test execution tools*, Addison-Wesley, ISBN 0-201-33140-3.
- [Freedman and Weinberg] D. Freedman and G. Weinberg (1990), *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, ISBN 0-932633-19-6.
- [Gerrard] P. Gerrard and N. Thompson (2002), *Risk-Based E-Business Testing*, Artech House Publishers, ISBN 1-58053-314-0.
- [Gilb and Graham] T. Gilb and D. Graham (1993), *Software Inspection*, Addison-Wesley, ISBN 0-201-63181-4.
- [Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2
- [Grochtmann] M. Grochtmann (1994), Test Case Design Using Classification Trees, in: *Conference Proceedings STAR 1994*.
- [Hetzel] W. Hetzel (1988), *The complete guide to software testing – 2nd edition*, QED Information Sciences, ISBN 0-89435-242-3.
- [McCabe] T. McCabe (1976), A complexity measure, in: *IEEE Transactions on Software Engineering*, Vol. 2, pp. 308-320.

- [Musa] J. Musa (1998), *Software Reliability Engineering Testing*, McGraw-Hill Education, ISBN 0-07913-271-5.
- [Myers] G. Myers (1979), *The Art of Software Testing*, Wiley, ISBN 0-471-04328-1.
- [TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), *Software Testing, A guide to the TMap Approach*, Addison Wesley, ISBN 0-201-745712.
- [Veenendaal] E. van Veenendaal (2004), *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.

Annex B (Method of commenting on this glossary)

Comments are invited on this document so that the glossary can be further improved to satisfy the needs of the testing community.

When making a comment, be sure to include the following information:

- Your name and contact details;
- The version number of the glossary (currently 2.0);
- Exact part of the glossary;
- Supporting information, such as the reason for a proposed change, or the reference to the use of a term.

You can submit comments in a variety of ways, which in order of preference are as follows:

1. By E-mail to eve@improveqs.nl;
2. By post to Improve Quality Services BV, attn. Mr. E. van Veenendaal, Waalreneweg 39, 5554 HA, Valkenswaard, The Netherlands;
3. By FAX to +31 40 20 21450, marked for the attention of Mr. E. van Veenendaal.
4. **About the Norwegian part: By email to:**
 - **Hans Schaefer:** hans.schaefer@ieee.org
 - or
 - **Ernst von Düring:** evd@bluezone.no