

ISTQB®

Standard glossary of terms used in Software Testing

Pojmovnik osnovnih termina u testiranju softvera

Lokalizirana verzija – hrvatski

Verzija 2.0 (02.12.2007.)

© ISTQB Glossary Working Party

Urednik: Erik van Veenendaal (Nizozemska)

Datum zadnje promjene lokalizirane verzije: 09.04.2011.

Autorska prava:

Dio dokumenta ili dokument u cijelosti može se kopirati pod uvjetom da se navede izvor.

Suradnici

Rex Black (SAD)	Maaret Pyhäjärvi (Finska)
Sigrid Eldh (Švedska)	Andy Redwood (Velika Britanija)
Isabel Evans (Velika Britanija)	Stuart Reid (Velika Britanija)
Dorothy Graham (Velika Britanija)	Piet de Roo (Nizozemska)
Julian Harty (Velika Britanija)	Steve Sampson (Velika Britanija)
David Hayman (Velika Britanija)	Shane Saunders (Velika Britanija)
Juha Itkonen (Finska)	Hans Schaefer (Norveška)
Vipul Kocher (Indija)	Jurriën Seubers (Nizozemska)
Fernando Lamas de Oliveira (Portugal)	Dave Sherratt (Velika Britanija)
Tilo Linz (Njemačka)	Mike Smith (Velika Britanija)
Peter Morgan (Velika Britanija)	Andreas Spillner (Njemačka)
Thomas Müller (Švicarska)	Richard Taylor (Velika Britanija)
Avi Ofer (Izrael)	Geoff Thompson (Velika Britanija)
Dale Perry (SAD)	Stephanie Ulrich (Njemačka)
Horst Pohlmann (Njemačka)	Matti Vuori (Finska)
Meile Posthuma (Nizozemska)	Gearrel Welvaart (Nizozemska)
Erkki Pöyhönen (Finska)	Pete Williams (Velika Britanija)

Radna skupina koja je pripremila lokaliziranu verziju:

mr. sc. Nadica Hrgarek, voditelj radne skupine
Ivana Arar, bacc. oec., član tima
Nikolina Pauk, dipl. inf., član tima
Davor Banović, dipl. teol., član tima

Provjera kvalitete lokalizacije:

Damir Šmigovec

Povijesnost promjena

Verzija 1.3 (31.05.2007)	
<p><i>Dodani novi pojmovi:</i></p> <ul style="list-style-type: none"> - action word driven testing - bug tracking tool - coverage measurement tool - modelling tool - monkey testing - scripted testing - specification-based technique - stress testing tool - structure-based technique - unit test framework - white box technique 	<p><i>Ažurirani pojmovi:</i></p> <ul style="list-style-type: none"> - basic block - control flow graph - defect management tool - independence of testing - project risk - risk-based testing - test comparator - test process
Verzija 2.0 (02.12.2007)	
<p><i>Dodani novi pojmovi:</i></p> <ul style="list-style-type: none"> - attack - buffer - buffer overflow - bug taxonomy - classification tree - control flow analysis - continuous representation - cost of quality - defect based technique - defect based test design technique - defect taxonomy - error seeding tool - Failure Mode, Effect and Criticality Analysis (FMECA) - false-fail result - false-pass result - false-negative result - false-positive result - fault attack - fault seeding - fault seeding tool - hazard analysis - hyperlink - hyperlink tool - load profile - operational acceptance testing - operational profile - orthogonal array - orthogonal array testing - pairwise testing - performance profiling - pointer - procedure testing - process improvement - production acceptance testing - qualification - reliability growth model - retrospective meeting 	<ul style="list-style-type: none"> - risk level - risk type - root cause analysis - safety critical system - software attack - Software Failure Mode and Effect Analysis (SFMEA) - Software Failure Mode Effect and Criticality Analysis (SFMECA) - Software Fault Tree Analysis (SFTA) - software life cycle - staged representation - system of systems - test design - test estimation - test implementation - Test Maturity Model Integration (TMMi) - test progress report - test rig - test schedule - test session - wild pointer <p><i>Ažurirani pojmovi:</i></p> <ul style="list-style-type: none"> - bebugging - error seeding - Failure Mode and Effect Analysis (FMEA) - Fault Tree Analysis (FTA) - modified multiple condition testing - process cycle test - root cause - specification-based technique - stress testing - test charter

Sadržaj

<i>Predgovor</i>	5
<i>Uvod</i>	5
<i>Doseg dokumenta</i>	5
<i>Uređenje dokumenta</i>	5
<i>Normativne reference</i>	5
<i>Zaštićene robne marke</i>	6
<i>Definicije</i>	6
<i>A</i>	6
<i>B</i>	8
<i>C</i>	11
<i>D</i>	17
<i>E</i>	21
<i>F</i>	23
<i>G</i>	26
<i>H</i>	26
<i>I</i>	26
<i>K</i>	29
<i>L</i>	29
<i>M</i>	30
<i>N</i>	33
<i>O</i>	33
<i>P</i>	35
<i>Q</i>	38
<i>R</i>	38
<i>S</i>	42
<i>T</i>	47
<i>U</i>	56
<i>V</i>	57
<i>W</i>	58
<i>Prilog A (informativno)</i>	59
<i>Prilog B (metoda komentiranja ovog pojmovnika)</i>	60

Predgovor

Radna grupa je prilikom sastavljanja ovoga pojmovnika tražila mišljenja i komentare što je moguće šireg opsega iz područja industrije, ekonomije i vladinih tijela i organizacija s ciljem kreiranja međunarodnog standarda testiranja koji će biti prihvaćen u što širem opsegu. Potpuno slaganje biti će rijetko, a možda i nikad ostvareno u sastavljanju dokumenta ovakve vrste. Prilozi za ovaj pojmovnik dobiveni su iz testnih zajednica iz Australije, Belgije, Finske, Njemačke, Indije, Israela, Nizozemske, Portugala, Švedske, Švicarske, Velike Britanije i SAD-a.

Mnogi tester (softvera) koristili su britanski standard BS 7925-1 od njegove izvorne objave u 1998. On je također služio kao glavna referenca za kvalifikaciju Ispitnog odbora za informacijske sustave – *Information Systems Examination Board (ISEB)* kako za *Foundation* tako i za *Practitioner* razinu. Standard je inicijalno razvijen s naglaskom na testiranje komponenti, ali od njegove objave, priloženi su mnogi komentari i prijedlozi za nove definicije u smislu poboljšanja i proširenja standarda kako bi se pokrilo šire područje testiranja softvera. U ovu novu verziju pojmovnika za testiranje uvršteni su mnogi predloženi dodaci. On će biti korišten kao referentni dokument za kreiranje sheme za osposobljavanje u testiranju softvera Međunarodnog odbora za osposobljavanje testera softvera – *International Software Testing Qualification Board (ISTQB)*.

Uvod

Mnogo vremena i napora troši se u industriji i između industrija, trgovini, vladinim, profesionalnim i akademskim institucijama kada se pojave nejasnoće kao rezultat nemogućnosti razlikovanja pojmova kao što su primjerice, „pokrivenost iskaza programskog koda“ (*statement coverage*) i „pokrivenost odluka“ (*decision coverage*); „skup testova“ (*test suite*), „specifikacija testa“ (*test specification*), „plan testiranja“ (*test plan*) te slični pojmovi koji čine sučelje između različitih sektora društva. Štoviše, profesionalno ili tehničko korištenje ovih pojmova je obično inačica različitih značenja koje ih opisuju.

Doseg dokumenta

Ovaj dokument predstavlja koncepte, pojmove i definicije da bi se olakšala komunikacija u testiranju (softvera) i srodnim disciplinama.

Uređenje dokumenta

Ovaj pojmovnik izrađen je kao jednostavan abecedni prikaz definicija. Za riječi koje imaju sinonime pojavljuje se sinonim na kraju objašnjenja. Primjerice, *structural testing* upućuje na *white box testing*. Za sinonime koristi se uputnica „vidi“.

Križne uputnice karakteristične po naputku „vidjeti također“ također se koriste. Ove uputnice pomažu čitatelju da lakše navigira do pravog pojma u indeksu. Križne uputnice „vidjeti također“ koriste se za uspostavljanje veze između šireg pojma prema užem pojmu, kao i za preklapanje značenja između dva pojma.

Normativne reference

Za vrijeme objave ovog dokumenta, vrijedila su navedena izdanja standarda. Svi standardi podliježu reviziji, a radne grupe se potiču da istraže mogućnosti primjene najnovijeg izdanja niže navedenih standarda. Članovi organizacija IEC i ISO održavaju registre međunarodnih standarda koji su trenutno važeći.

- 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

Zaštićene robne marke

U dokumentu se koriste slijedeće robne marke:

- CMM i CMMI su zaštićene robne marke Sveučilišta Carnegie Mellon
- TMap, TPA i TPI su zaštićene robne marke kompanije Sogeti Nederland BV
- TMM je zaštićeni uslužni žig Illinois Institute of Technology
- TMMi je zaštićena robna marka TMMi Foundation

Definicije

A

abstract test case: See *high level test case*.

abstraktni testni slučaj: Vidi *high-level test case*.

acceptance: See *acceptance testing*.

prihvatljivost, prihvaćanje: Vidi *acceptance testing*.

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]

kriteriji za prihvata, kriteriji prihvaćanja: Izlazni kriteriji koje komponenta ili sustav mora zadovoljiti da bi bili prihvaćeni od strane korisnika, klijenta ili drugih ovlaštenih entiteta. [IEEE 610]

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. [After IEEE 610]

testiranje prihvatljivosti, prijamna provjera, prijamni test: Formalno testiranje s obzirom na korisnikove potrebe, zahtjeve i poslovne procese, a koje se provodi da bi se utvrdilo da li sustav zadovoljava kriterije prihvaćanja, te da bi se omogućilo korisniku, klijentu ili drugim ovlaštenim entitetima da odluče da li da prihvate sustav ili ne. [prema IEEE 610]

accessibility testing: Testing to determine the ease by which users with disabilities can use a component or system. [Gerrard]

testiranje pristupačnosti: Testiranje sa ciljem utvrđivanja lakoće kojom korisnik s nedostacima može koristiti komponentu ili sustav. [Gerrard]

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 *functionality testing*.

točnost, ispravnost: Sposobnost programskog proizvoda da osigura točne ili dogovorene rezultate ili učinke sa potrebnim stupnjem preciznosti. [ISO 9126] Vidjeti također *functionality testing*.

action word driven testing: See *keyword driven testing*.

testiranje vođeno ključnim riječima: Vidi *keyword driven testing*.

actual outcome: See *actual result*.

stvarni ishod: Vidi *actual result*.

actual result: The behavior produced/observed when a component or system is tested.

stvarni rezultat: Ponašanje proizvedeno/uočeno nakon što je komponenta ili sustav testiran.

ad hoc review: See *informal review*.

ad hoc pregled: Vidi *informal review*.

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 testiranje: Neformalno testiranje; nema formalne pripreme za testiranje, ne primjenjuje se priznata tehnika oblikovanja testova, nema očekivanja rezultata i aktivnosti provedbe testiranja vođene su samovoljom.

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 *portability*.

prilagodljivost: Sposobnost programskog proizvoda da se prilagodi različitim specificiranim okruženjima bez primjene radnji ili sredstava drugačijih od onih koji su za to predviđeni za softver koji se uzima u obzir. [ISO 9126] Vidjeti također *portability*.

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 *test driven development*.

agilno testiranje: Praksa testiranja na projektu u kojemu se koriste agilne metodike, kao što je ekstremno programiranje (XP), razvoj se tretira kao klijent testiranja i naglašava se paradigma oblikovanja „prvo piši testove“. Vidjeti također *test driven development*.

algorithm test [TMap]: See *branch testing*.

test algoritma [TMap]: Vidi *branch testing*.

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.

alfa-testiranje, alfa-ispitivanje: Simulirano ili stvarno operacijsko testiranje od strane potencijalnih korisnika/klijenata ili od strane neovisnog tima za testiranje na lokaciji razvoja, ali izvan organizacije koja razvija softver. Alfa-testiranje često se koristi za komercijalni off-the-shelf softver kao vrsta internog testiranja prihvaćanja.

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 *maintainability*.

prikladnost za analizu: Sposobnost programskog proizvoda da se može dijagnosticirati s obzirom na nedostatke ili uzroke zatajenja u softveru, ili s obzirom na dijelove za modifikaciju koji se trebaju identificirati. [ISO 9126] Vidjeti također *maintability*.

analyzer: See *static analyzer*.

analizator: Vidi *static analyzer*.

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 *bug, defect, deviation, error, fault, failure, incident, problem*.

anomalija: Svako stanje koje odstupa od očekivanja temeljenih na specifikacijama zahtjeva, dokumentima oblikovanja, korisničkoj dokumentaciji, standardima, itd. ili prema nečijoj percepciji ili iskustvu. Anomalije mogu biti pronađene tijekom pregleda, testiranja, analize, kompilacije ili korištenjem programskog proizvoda ili

odgovarajuće dokumentacije [IEEE 1044] Vidjeti također *bug, defect, deviation, error, fault, failure, incident, problem*.

arc testing: See *branch testing*.

testiranje luka, testiranje grane: Vidi *branch testing*.

attack: Directed and focused attempt to evaluate the quality, especially reliability, of a test object by attempting to force specific failures to occur.

napad: Direktni i fokusirani pokušaj prosudbe kvalitete (posebno pouzdanosti) objekta koji se testira pokušavajući izazvati pojavu specifičnih zatajenja.

attractiveness: The capability of the software product to be attractive to the user. [ISO 9126] See also *usability*.

atraktivnost: Sposobnost programskog proizvoda da se sviđa korisniku. [ISO 9126] Vidjeti također *usability*.

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]

audit, revizija, provjera: Nezavisna prosudba programskih proizvoda ili procesa kako bi se osigurala usklađenost prema standardima, smjernicama, specifikacijama i/ili procedurama koje se temelje na objektivnim kriterijima uključujući dokumente koji specificiraju:

- (1) formu ili sadržaj proizvoda koji se proizvode
- (2) proces po kojemu proizvod treba biti proizveden
- (3) kako će se mjeriti usklađenost prema standardima ili smjernicama. [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. [After TMap]

revizijska sljedivost: Putanja po kojoj možemo izvorni ulaz u proces (npr. podaci) pratiti kroz sam proces tako da izlaz procesa uzimamo kao početnu točku. Na taj se način olakšava analiza grešaka i omogućuje provjera procesa. [prema TMap]

automated testware: Testware used in automated testing, such as tool scripts.

automatizirani testver: Testver koji se koristi za automatizirano testiranje kao što su skripte alata.

availability: The degree to which a component or system is operational and accessible when required for use. Often expressed as a percentage. [IEEE 610]

dostupnost: Stupanj do kojeg je komponenta ili sustav operativan i dostupan za uporabu. Obično se izražava u postotku. [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]

suprotno testiranje: Testiranje u kojemu se dvije ili više varijanti komponente ili sustava izvršavaju sa jednakim ulaznim podacima, pri čemu se izlazi uspoređuju i analiziraju u slučajevima pojave odstupanja. [IEEE 610]

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. [After IEEE 610]

osnovica, temeljni plan: Specifikacija ili programski proizvod koji je formalno pregledan ili ugovoren, a koji kasnije služi kao osnova za daljnji razvoj i koji se može mijenjati samo kroz formalni proces upravljanja promjenama. [prema IEEE 610]

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.

osnovni blok: Slijed jedne ili više uzastopnih izvršnih linija programskog koda bez grananja. Napomena: Čvor u grafu tijeka programa predstavlja osnovni blok.

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.

osnovni skup testova: Skup testnih slučajeva izvedenih iz unutrašnje strukture komponente ili specifikacije kako bi se postigla 100%-tna pokrivenost specificiranog kriterija.

bebugging: See *fault seeding*. [Abbott]

sijanje grešaka, umetanje grešaka: Vidi *fault seeding*. [Abbott]

behavior: The response of a component or system to a set of input values and preconditions.

ponašanje: Odgovor komponente ili sustava na skup ulaznih vrijednosti i preduvjeta.

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). [After IEEE 610]

mjerni test: (1) Standard prema kojemu se poduzimaju mjerenja ili usporedbe. (2) Test koji se koristi kako bi se usporedile komponente ili sustavi međusobno ili prema standardu kao pod (1). [prema IEEE 610]

bespoke software: Software developed specifically for a set of users or customers. The opposite is off-the-shelf software.

softver po narudžbi, softver po mjeri korisnikovih zahtjeva: Softver koji je posebno razvijen za određeni skup korisnika ili kupaca. Oprečni pojam je off-the-shelf softver.

best practice: A superior method or innovative practice that contributes to the improved performance of an organization under given context, usually recognized as 'best' by other peer organizations.

najbolja praksa, najbolji postupak: Superiorna metoda ili inovativna praksa koja pridonosi unapređenju učinkovitosti organizacije u danom kontekstu, obično se prepoznaje kao „najbolja“ u ostalim ravnopravnim organizacijama.

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.

beta-testiranje, beta-ispitivanje: Operativno testiranje od strane potencijalnih i/ili postojećih korisnika/kupaca na vanjskoj lokaciji, a koji nisu bili uključeni sa osobama koje su razvile sustav, kako bi odredili da li komponenta ili sustav zadovoljava potrebe kupca/korisnika i da li se uklapa u poslovne procese. Beta-testiranje obično se koristi kao oblik vanjskog testiranja prihvatljivosti za off-the-shelf softver kako bi dobili povratnu informaciju od tržišta.

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. [After IEEE 610] See also *integration testing*.

testiranje velikog praska, big bang testiranje: Tip integracijskog testiranja u kojemu se elementi softvera, elementi hardvera ili oboje, kombiniraju odjednom u komponentu ili cjelokupni sustav, a ne po fazama. [After IEEE 610] Vidjeti također *integration testing*.

black-box technique: See *black box test design technique*.

tehnika crne kutije: Vidi *black box test design technique*.

black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.

testiranje crne kutije: Testiranje funkcionalnih ili nefunkcionalnih karakteristika bez uzimanja u obzir interne strukture komponente ili sustava.

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.

tehnika oblikovanja testova crne kutije: Procedura pomoću koje se izvode i/ili selektiraju testni slučajevi koji se temelje na analizi specifikacije bilo funkcionalne ili nefunkcionalne, komponente ili sustava bez uzimanja u obzir unutrašnje strukture.

blocked test case: A test case that cannot be executed because the preconditions for its execution are not fulfilled.

blokirani testni slučaj: Testni slučaj koji se ne može izvršiti jer preduvjeti za njegovo izvršenje nisu ispunjeni.

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 *integration testing*.

testiranje odozdo prema gore: Inkrementalni pristup integracijskom testiranju gdje se komponente najniže razine prve testiraju, a zatim se koriste da omoguće testiranje komponenti viših razina. Ovaj proces se ponavlja tako dugo dok je testirana i posljednja komponenta na vrhu hijerarhije. Vidjeti također *integration testing*.

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.

granična vrijednost, rubna vrijednost: Ulazna ili izlazna vrijednost koja se nalazi na granicama ekvivalentne particije ili na najmanjoj inkrementalnoj udaljenosti bilo koje granice ruba, npr. minimalna ili maksimalna vrijednost raspona.

boundary value analysis: A black box test design technique in which test cases are designed based on boundary values. See also *boundary value*.

analiza graničnih vrijednosti, analiza rubnih vrijednosti: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani na temelju graničnih vrijednosti. Vidjeti također *boundary value*.

boundary value coverage: The percentage of boundary values that have been exercised by a test suite.

pokrivenost graničnih vrijednosti: Postotak graničnih vrijednosti koje su provjerene pomoću skupa testnih slučajeva.

boundary value testing: See *boundary value analysis*.

testiranje graničnih vrijednosti: Vidi *boundary value analysis*.

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 is available, e.g. case, jump, go to, if-then-else.

grana, grananje: Osnovni blok koji se može odabrati za izvođenje zasnovan na programskom konstrukt u kojemu su dostupne jedna ili više alternativnih putanji programa, npr. case, jump, go to, if-then-else.

branch condition: See *condition*.

uvjetno grananje: Vidi *condition*.

branch condition combination coverage: See *multiple condition coverage*.

pokrivenost svih kombinacija uvjeta grananja: Vidi *multiple condition coverage*.

branch condition combination testing: See *multiple condition testing*.

testiranje svih kombinacija uvjeta grananja: Vidi *multiple condition testing*.

branch condition coverage: See *condition coverage*.

pokrivenost uvjeta grananja: Vidi *condition coverage*.

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.

pokrivenost grananja: Postotak grananja koji su ispitani pomoću skupa testnih slučajeva. 100%-tna pokrivenost grananja znači 100%-tnu pokrivenost odluka i 100%-tnu pokrivenost linija koda.

branch testing: A white box test design technique in which test cases are designed to execute branches.

testiranje grananja: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani kako bi se izvršile grane.

buffer: A device or storage area used to store data temporarily for differences in rates of data 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]

međuspremnik, međumemorija: Uređaj ili mjesto za pohranu koje se koristi za privremenu pohranu podataka u slučaju razlika u toku podataka, vremenu ili pojavljivanju događaja, te količini podataka koja se može obraditi u uređajima ili procesima uključenim u prijenos ili upotrebu podataka. [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 *buffer*.

preljev međumemorije, prekoračenje kapaciteta međuspremnika: Greška u pristupu memoriji prilikom pokušaja procesa da pohrani podatke izvan granica međuspremnika određene duljine. Posljedica toga je prepisivanje susjednih memorijskih lokacija ili podizanje iznimke preljeva. Vidjeti također *buffer*.

bug: See *defect*.

pogreška, kvar: Vidi *defect*.

bug report: See *defect report*.

izvještaj o pogrešci: Vidi *defect report*.

bug taxonomy: See *defect taxonomy*.

taksonomija pogreške: Vidi *defect taxonomy*.

bug tracking tool: See *defect management tool*.

alat za bilježenje pogrešaka: Vidi *defect management tool*.

business process-based testing: An approach to testing in which test cases are designed based on descriptions and/or knowledge of business processes.

testiranje zasnovano na poslovnim procesima: Pristup testiranju u kojemu su testni slučajevi oblikovani na temelju opisa i/ili znanja o poslovnim procesima.

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 *Capability Maturity Model Integration (CMMI)*.

Capability Maturity Model (CMM): Okvir koji se sastoji od pet razina, a opisuje ključne elemente djelotvornog softverskog procesa. Capability Maturity Model pokriva najbolje prakse za planiranje, inženjerstvo, te upravljanje razvojem softvera i održavanje. Vidjeti također *Capability Maturity Model Integration (CMMI)*.

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 *Capability Maturity Model (CMM)*.

Capability Maturity Model Integration (CMMI): Okvir koji opisuje ključne elemente djelotvornog razvoja proizvoda i procesa održavanja. Capability Maturity Model pokriva najbolje prakse za planiranje, inženjerstvo, te upravljanje razvojem softvera i održavanje. CMMI je određeni nasljednik CMM modela. [CMMI] Vidjeti također *Capability Maturity Model (CMM)*.

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.

alat za hvatanje/reprodukciju: Vrsta alata za izvođenje testova gdje se ulazni podaci snimaju tijekom ručnog testiranja kako bi se generirale automatizirane testne skripte koje se mogu kasnije izvršiti, tj. reproducirati. Ovi alati često se koriste kao podrška automatiziranog regresijskog testiranja.

capture/replay tool: See *capture/playback tool*.

alat za hvatanje/reprodukciju: Vidi *capture/playback tool*.

CASE: Acronym for Computer Aided Software Engineering.

CASE: Akronim za Computer Aided Software Engineering.

CAST: Acronym for Computer Aided Software Testing. See also *test automation*.

CAST: Akronim za Computer Aided Software Testing. Vidjeti također *test automation*.

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.

dijagram uzroka i posljedice: Grafički prikaz ulaza i/ili stimulativnih poticaja (uzroci) i njihovih pridruženih izlaza (posljedice), a koji se mogu koristiti za oblikovanje testnih slučajeva.

cause-effect graphing: A black box test design technique in which test cases are designed from cause-effect graphs. [BS 7925/2]

izrada dijagrama uzroka i posljedice: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani iz dijagrama uzroka i posljedica. [BS 7925/2]

cause-effect analysis: See *cause-effect graphing*.

analiza uzroka i posljedice: Vidi *cause-effect graphing*.

cause-effect decision table: See *decision table*.

tablica odlučivanja uzroka i posljedice: Vidi *decision table*.

certification: The process of confirming that a component, system or person complies with its specified requirements, e.g. by passing an exam.

certifikacija: Proces potvrđivanja da je komponenta, sustav ili osoba u skladu sa specificiranim zahtjevima, npr. polaganje ispita.

changeability: The capability of the software product to enable specified modifications to be implemented. [ISO 9126] See also *maintainability*.

sposobnost promjene: Sposobnost programskog proizvoda da omogući implementaciju specificiranih promjena. [ISO 9126] Vidjeti također *maintainability*.

change control: See *configuration control*.

upravljanje izmjenama, upravljanje promjenama: Vidi *configuration control*.

change control board: See *configuration control board*.

odbor za upravljanje izmjenama, odbor za upravljanje promjenama: Vidi *configuration control board*.

checker: See *reviewer*.

ispitivač: Vidi *reviewer*.

Chow's coverage metrics: See *N-switch coverage*. [Chow]

Chowove metrike pokrivenosti: Vidi *N-switch coverage*. [Chow]

classification tree: A tree showing equivalence partitions hierarchically ordered, which is used to design test cases in the classification tree method. See also *classification tree method*.

klasifikacijsko stablo: Stablo koje prikazuje uređenu hijerarhiju ekvivalentnih particija, a koje se koristi za oblikovanje testnih slučajeva pomoću metode klasifikacijskog stabla. Vidjeti također *classification tree method*.

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]

metoda klasifikacijskog stabla: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi, opisani pomoću klasifikacijskog stabla, oblikovani za izvršavanje kombinacija predstavnika ulaznih i/ili izlaznih domena. [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]

kôd: Računalne instrukcije i definicije podataka izražene u programskom jeziku ili u obliku izlaza od strane asemblera, kompilatora ili drugog prevoditelja. [IEEE 610]

code analyzer: See *static code analyzer*.

analizator koda: Vidi *static code analyzer*.

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.

pokrivenost koda: Metoda analiziranja koja određuje koji dijelovi softvera su izvršeni (pokriveni) pomoću skupa testova, a koji ne; npr. pokrivenost linija koda, pokrivenost odluka ili pokrivenost uvjeta.

code-based testing: See *white box testing*.

testiranje zasnovano na kodu: Vidi *white box testing*.

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 *portability*.

sunazočnost, koegzistencija: Sposobnost programskog proizvoda da istodobno egzistira sa ostalim nezavisnim softverom u zajedničkom okruženju dijeleći zajedničke resurse. [ISO 9126] Vidjeti također *portability*.

commercial off-the-shelf software: See *off-the-shelf software*.

komercijalni off-the-shelf softver: Vidi *off-the-shelf software*.

comparator: See *test comparator*.

uspoređivač, komparator: Vidi *test comparator*.

compatibility testing: See *interoperability testing*.

testiranje kompatibilnosti: Vidi *interoperability testing*.

compiler: A software tool that translates programs expressed in a high order language into their machine language equivalents. [IEEE 610]

programski prevoditelj, kompilator: Programski alat koji prevodi programe napisane na jeziku visoke razine u ekvivalente strojnog jezika. [IEEE 610]

complete testing: See *exhaustive testing*.

potpuno testiranje, cjelovito testiranje: Vidi *exhaustive testing*.

completion criteria: See *exit criteria*.

kriteriji završetka: Vidi *exit criteria*.

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 *cyclomatic complexity*.

složenost: Stupanj do kojega komponenta ili sustav ima dizajn i/ili internu strukturu koju je teško razumjeti, održavati i provjeravati. Vidjeti također *cyclomatic complexity*.

compliance: The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions. [ISO 9126]

usklađenost, sukladnost: Sposobnost programskog proizvoda da se prilagodi standardima, konvencijama ili pravnima odredbama i sličnim propisima. [ISO 9126]

compliance testing: The process of testing to determine the compliance of the component or system.

testiranje u skladu sa standardom: Proces testiranja koji određuje sukladnost komponente ili sustava.

component: A minimal software item that can be tested in isolation.

komponenta: Minimalna jedinica programa koja se može pojedinačno testirati.

component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.

testiranje integracije komponenti: Testiranje koje se provodi s ciljem pronalaženja grešaka u sučeljima i interakciji između integriranih komponenti.

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).

specifikacija komponente: Opis funkcije komponente uključujući njezine izlazne vrijednosti za specificirane ulazne vrijednosti pod specificiranim uvjetima, te zahtijevano nefunkcionalno ponašanje komponente (npr. korištenje resursa).

component testing: The testing of individual software components. [After IEEE 610]

testiranje komponenti: Individualno testiranje komponenti programa. [prema IEEE 610]

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'.

složeni uvjet: Dva ili više pojedinačnih uvjeta povezanih logičkim operatorom (AND, OR ili XOR), npr. 'A>B AND C>1000'.

concrete test case: See *low level test case*.

konkretni testni slučaj: Vidi *low level test case*.

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. [After IEEE 610]

testiranje paralelnog rada, testiranje istovremenosti: Testiranje da bi se odredilo kako se komponenta ili sustav ponaša pri izvođenju dvije ili više aktivnosti u istom vremenskom intervalu, koje se postiže preklapanjem aktivnosti ili simultanim izvršavanjem. [prema IEEE 610]

condition: A logical expression that can be evaluated as True or False, e.g. A>B. See also *test condition*.

uvjet: Logički izraz koji daje vrijednosti istina (*true*) ili laž (*false*), npr. A>B. Vidjeti također *test condition*.

condition combination coverage: See *multiple condition coverage*.

pokrivenost svih kombinacija uvjeta: Vidi *multiple condition coverage*.

condition combination testing: See *multiple condition testing*.

testiranje svih kombinacija uvjeta: Vidi *multiple condition testing*.

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.

pokrivenost uvjeta: Postotak ishoda uvjeta koji su ispitani pomoću skupu testnih slučajeva. 100%-tna pokrivenost uvjeta zahtijeva da svaki pojedinačni uvjet u svakoj naredbi odlučivanja bude testiran kao istina (*true*) i laž (*false*).

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.

pokrivenost odluka uvjeta: Postotak svih pojedinačnih izlaza uvjeta koji nezavisno utječu na ishod odluke koja je ispitana pomoću skupa testnih slučajeva. 100%-tna pokrivenost odluka uvjeta povlači za sobom 100%-tnu pokrivenost uvjeta odluka.

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.

testiranje odluka uvjeta: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani na način da se izvrše pojedinačni uvjeti koji nezavisno utječu na ishod cjelokupne odluke.

condition testing: A white box test design technique in which test cases are designed to execute condition outcomes.

testiranje uvjeta, testiranje izraza: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani tako da se izvrše ishodi uvjeta.

condition outcome: The evaluation of a condition to True or False.

rezultat uvjeta, ishod uvjeta: Prosudba uvjeta kao točno ili laž.

confidence test: See *smoke test*.

test pouzdanosti: Vidi *smoke test*.

configuration: The composition of a component or system as defined by the number, nature, and interconnections of its constituent parts.

konfiguracija: Kompozicija komponente ili sustava definirana brojem, prirodom i međusobnim vezama njezinih sastavnih dijelova.

configuration auditing: The function to check on the contents of libraries of configuration items, e.g. for standards compliance. [IEEE 610]

provjeravanje konfiguracije: Funkcija za provjeravanje sadržaja biblioteka elemenata konfiguracije, npr. za usklađivanje sa standardima. [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]

nadzor konfiguracije, kontrola konfiguracije: Element upravljanja konfiguracijom koji se sastoji od prosudbe, koordinacije, odobravanja ili neodobravanja i implementacije promjena u elemente konfiguracije nakon formalne uspostave njihovih identifikacija u konfiguraciji. [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]

odbor za nadzor konfiguracije: Grupa ljudi odgovorna za prosudbu i odobravanje ili neprihvatanje predloženih promjena na konfiguracijskim elementima, kao i osiguranje implementacije odobrenih promjena. [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]

identifikacija konfiguracije: Element upravljanja konfiguracijom koji se sastoji od odabira elemenata konfiguracije za sustav te zapisivanje njihovih funkcionalnih i fizičkih obilježja u tehničkoj dokumentaciji. [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]

element konfiguracije: Agregacija hardvera, softvera ili oboje, a koja je određena za upravljanje konfiguracijom i tretirana kao zasebna jedinica u procesu upravljanja konfiguracijom. [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]

upravljanje konfiguracijom: Disciplina koja primjenjuje tehničke i administrativne upute i nadzor za: identificiranje i dokumentiranje funkcionalnih i fizičkih obilježja elementa konfiguracije, kontrolu promjena za ta obilježja, zapisivanje i izvještavanje o obradbi promjena te statusu implementacije i provjeravanje usklađenosti sa specificiranim zahtjevima. [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.

alat za upravljanje konfiguracijom: Alat koji omogućava podršku za identifikaciju i nadzor elemenata konfiguracije, njihovog statusa tijekom promjena i verzija, kao i puštanje u rad osnovice koja se sastoji od elemenata konfiguracije.

configuration testing: See *portability testing*.

testiranje konfiguracije: Vidi *portability testing*.

confirmation testing: See *re-testing*.

potvrđno testiranje: Vidi *re-testing*.

conformance testing: See *compliance testing*.

testiranje sukladnosti, testiranje usklađenosti: Vidi *compliance testing*.

consistency: The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a component or system. [IEEE 610]

konzistentnost, dosljednost: Stupanj jednoznačnosti, standardizacije i slobode od proturječnosti u dokumentima ili dijelovima komponente ili sustava. [IEEE 610]

control flow: A sequence of events (paths) in the execution through a component or system.

upravljajući tok: Slijed događaja (putanje) u izvršavanju kroz komponentu ili sustav.

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.

analiza upravljačkog toka: Oblik statičke analize temeljene na reprezentaciji slijeda događaja (putanji) u izvršenju kroz komponentu ili sustav.

control flow graph: An abstract representation of all possible sequences of events (paths) in the execution through a component or system.

graf tijeka programa: Apstraktni prikaz svih mogućih slijedova događaja (putanji) u izvršenju kroz komponentu ili sustav.

control flow path: See *path*.

putanja tijeka programa: Vidi *path*.

continuous representation: A capability maturity model structure wherein capability levels provide a recommended order for approaching process improvement within specified process areas. [CMMI]

kontinuirana reprezentacija, kontinuirani prikaz: Struktura modela procjene zrelosti gdje razine zrelosti daju preporučeni slijed za pristup unaprijeđenju procesa unutar specificiranih područja procesa. [CMMI]

conversion testing: Testing of software used to convert data from existing systems for use in replacement systems.

testiranje konverzije, testiranje pretvorbe: Testiranje softvera koje se koristi za konverziju podataka iz postojećih sustava za uporabu u zamjenskih sustavima.

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.

trošak kvalitete: Ukupni troškovi koji se pojavljuju u aktivnostima kvalitete i problemima kvalitete. Često se dijele na troškove prevencije, troškove procjene, troškove unutrašnjih propusta i troškove vanjskih propusta.

COTS: Acronym for Commercial Off-The-Shelf software. See *off-the-shelf software*.

COTS: Akronim za Commercial Off-The-Shelf softver. Vidi *off-the-shelf software*.

coverage: The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite.

pokrivenost: Stupanj, izražen u postotku, u kojemu je specificirana jedinica pokrivenosti izvršena u skupu testova.

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.

analiza pokrivenosti: Mjera postignute pokrivenosti određene jedinice tijekom izvršavanja testa prema predodređenim kriterijima kako bi se odredilo da li je potrebno dodatno testiranje i ukoliko jest, koji testni slučajevi su potrebni.

coverage measurement tool: See *coverage tool*.

alat za mjerenje pokrivenosti: Vidi *coverage tool*.

coverage item: An entity or property used as a basis for test coverage, e.g. equivalence partitions or code statements.

jedinica pokrivenosti: Jedinica ili svojstvo koje se koristi kao osnova za pokrivenost testa, npr. ekvivalentne particije ili linije koda.

coverage tool: A tool that provides objective measures of what structural elements, e.g. statements, branches have been exercised by a test suite.

alat za pokrivenost: Alat koji omogućuje objektivno mjerenje strukturalnih elemenata, npr. naredbe, grane koje su ispitane u skupu testova.

custom software: See *bespoke software*.

softver po narudžbi: Vidi *bespoke software*.

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 called graph and a subroutine) [After McCabe]

ciklometrička složenost: Broj nezavisnih putova u programu. Ciklometrička složenost definirana je kao: $L - N + 2P$, gdje je:

- L = broj poveznica/veza u grafu,
- N = broj čvorova u grafu,
- P = broj nepovezanih dijelova u grafu (npr. graf koji se poziva i podrutina) [prema McCabe]

cyclomatic number: See *cyclomatic complexity*.

ciklometrički broj: Vidi *cyclomatic complexity*.

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.

dnevno izdanje: Aktivnost razvoja gdje se cjelokupni sustav kompilira i povezuje svaki dan (obično preko noći). Time se omogućuje da je sustav konzistentan i raspoloživ u bilo koje vrijeme uključujući sve zadnje promjene.

data definition: An executable statement where a variable is assigned a value.

definicija podataka: Izvršiva linija gdje je varijabli dodijeljena vrijednost.

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]

testiranje vođeno podacima: Tehnika skriptiranja koja pohranjuje ulaze testiranja i očekivane rezultate u tablicu ili proračunsku tablicu, tako da pomoću jedne upravljačke skripte možemo izvršiti sve testove u tablici. Testiranje vođeno podacima često se koristi kao podrška aplikaciji alata za izvođenje testova kao što su, primjerice, alati za hvatanje/reprodukciju. [Fewster and Graham]

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]

tok podataka, protok podataka: Apstraktni prikaz slijeda i mogućih promjena stanja testnih objekata, gdje stanje objekta može biti: kreiranje, korištenje ili uništavanje. [Beizer]

data flow analysis: A form of static analysis based on the definition and usage of variables.

analiza toka podataka, raščlamba toka podataka: Oblik statičke analize zasnovane na definiciji i korištenju varijabli.

data flow coverage: The percentage of definition-use pairs that have been exercised by a test suite.

pokrivenost toka podataka: Postotak parova „definicija-uporaba“ koji su ispitani skupom testova.

data flow testing: A white box test design technique in which test cases are designed to execute definition and use pairs of variables.

testiranje toka podataka, testiranje protoka podataka: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani za izvršavanje parova varijabli za definiciju i korištenje.

data integrity testing: See *database integrity testing*.

testiranje integriteta podataka: Vidi *database integrity testing*.

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.

testiranje integriteta baze podataka: Testiranje metoda i procesa koji se koriste za pristup i upravljanje podacima (bazama podataka), da bi se osigurao pristup metodama, procesima i funkciji pravila obradbe podataka kao što je očekivano, te da prilikom pristupanja bazi podaci nisu pogrešni ili neočekivano obrisani, ažurirani ili kreirani.

dead code: See *unreachable code*.

mrtvi kôd: Vidi *unreachable code*.

debugger: See *debugging tool*.

program za otkrivanje pogrešaka: Vidi *debugging tool*.

debugging: The process of finding, analyzing and removing the causes of failures in software.

ispravljanje: Proces pronalaženja, analiziranja i uklanjanja uzroka zatajenja u softveru.

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.

alat za otkrivanje pogrešaka: Alat koji koriste programeri za reprodukciju zatajenja, istraživanje stanja programa i pronalaženja odgovarajuće greške. Programi za otkrivanje pogrešaka omogućuju programerima izvršavanje programa korak po korak, zaustavljanje programa u bilo kojoj programskoj liniji, te postavljanje i ispitivanje varijabli programa.

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.

odluka, odlučivanje: Točka programa u kojoj upravljački tok ima dva ili više alternativnih puteva. Čvor sa dvije ili više veza prema zasebnim granama.

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.

pokrivenost uvjeta odluka: Postotak ishoda svih uvjeta i odluka koji su ispitani pomoću skupa testova. 100%-tna pokrivenost uvjeta odluka povlači za sobom 100%-tnu pokrivenost uvjeta i 100%-tnu pokrivenost odluka.

decision condition testing: A white box test design technique in which test cases are designed to execute condition outcomes and decision outcomes.

testiranje uvjeta odluka: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani da se izvrše ishodi uvjeta i odluka.

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.

pokrivenost odluka: Postotak ishoda odluka koje su ispitane pomoću skupa testova. 100%-tna pokrivenost odluka povlači za sobom 100%-tnu pokrivenost grananja i 100%-tnu pokrivenost linija koda.

decision outcome: The result of a decision (which therefore determines the branches to be taken).

ishod odluke, rezultat odluke: Rezultat odluke (njome su određena izvršena grananja).

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.

tablica odlučivanja: Tablica koja pokazuje kombinacije ulaza i/ili stimulativnih poticaja (uzroci) sa njihovim pridruženim ishodima i/ili akcijama (posljedice) koje se mogu koristiti u oblikovanju testnih slučajeva.

decision table testing: A black box test design technique in which test cases are designed to execute the combinations of inputs and/or stimuli (causes) shown in a decision table. [Veenendaal] See also *decision table*.

testiranje tablice odlučivanja: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvršavanje kombinacija ulaza i/ili stimulativnih poticaja (uzroci) prikazanih u tablici odlučivanja. [Veenendaal]. Vidjeti također *decision table*.

decision testing: A white box test design technique in which test cases are designed to execute decision outcomes.

testiranje odluka: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani za izvršavanje ishoda odluka.

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.

nedostatak, pogreška: Pogreška komponente ili sustava koja može uzrokovati neizvršavanje tražene funkcije komponente ili sustava, npr. neispravna linija ili definicija podataka. Ukoliko se pogreška pojavi tijekom izvođenja može uzrokovati ispad komponente ili sustava.

defect-based technique: See *defect-based test design technique*.

tehnika zasnovana na pogreškama: Vidi *defect-based test design technique*.

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.

tehnika oblikovanja testova zasnovana na pogreškama: Procedura kojom se izvode i/ili odabiru testni slučajevi koji ciljaju jednu ili više kategorija pogrešaka pomoću testova koji su razvijeni iz poznatih podataka o specifičnoj kategoriji pogreške.

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).

gustoća pogrešaka, gustoća grešaka: Broj pogrešaka otkrivenih u komponenti ili sustavu podijeljen veličinom komponente ili sustava (izražava se u standardnim pojmovima mjerenja, npr. broj linija koda, broj klasa ili funkcijskih točaka).

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.

postotak otkrivanja pogrešaka: Broj pogrešaka pronađenih u određenoj fazi testiranja podijeljen sa brojem ukupno pronađenih grešaka u toj fazi testiranja i bilo kojim drugim načinom naknadno.

defect management: The process of recognizing, investigating, taking action and disposing of defects. It involves recording defects, classifying them and identifying the impact. [After IEEE 1044]

upravljanje pogreškama: Proces prepoznavanja, istraživanja, poduzimanja radnji i disponiranja pogrešaka. Uključuje evidentiranje grešaka, njihovo razvrstavanje i identificiranje utjecaja. [prema 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.

alat za upravljanje pogreškama: Alat koji omogućuje zapisivanje i praćenje statusa pogrešaka i promjena. Ovi alati često omogućuju automatiziranje tijeka poslovnih aktivnosti za praćenje i kontrolu dodjeljivanja, ispravljanja i ponovnog testiranja pogrešaka, te omogućuju izvještavanje.

defect masking: An occurrence in which one defect prevents the detection of another. [After IEEE 610]

maskiranje pogrešaka: Pojava u kojoj jedna pogreška onemogućava otkrivanje druge. [prema 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. [After IEEE 829]

izvještaj o pogrešci: Dokument izvještavanja o bilo kojoj pogrešci u komponenti ili sustavu koja može izazvati neizvršavanje tražene funkcije komponente ili sustava. [prema IEEE 829]

defect taxonomy: A system of (hierarchical) categories designed to be a useful aid for reproducibly classifying defects.

taksonomija pogrešaka: Sustav (hijerarhijskih) kategorija oblikovanih kao pomoćno sredstvo za razvrstavanje pogrešaka koje se mogu reproducirati.

defect tracking tool: See *defect management tool*.

alat za bilježenje pogrešaka: Vidi *defect management tool*.

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).

par definicija-uporaba: Povezanost definicije varijable sa uporabom te varijable. Varijabla koristi računalno (npr. umnožavanje) ili direktno izvršavanje putanje (korištenje „predikata“).

deliverable: Any (work) product that must be delivered to someone other than the (work) product's author.

isporuka: Bilo koji (rad) proizvod koji mora biti dostavljen bilo kome osim autoru (rada) proizvoda.

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).

testiranje zasnovano na oblikovanju: Pristup testiranju u kojem su testni slučajevi oblikovani na temelju arhitekture i/ili detaljnog dizajna komponente ili sustava (npr. testovi sučelja između komponenti ili sustava).

desk checking: Testing of software or specification by manual simulation of its execution. See also *static analysis*.

ručno testiranje: Testiranje softvera ili specifikacije pomoću ručne simulacije izvršavanja. Vidjeti također *static analysis*.

development testing: Formal or informal testing conducted during the implementation of a component or system, usually in the development environment by developers. [After IEEE 610]

testiranje razvoja: Formalno ili neformalno testiranje koje se obavlja tijekom implementacije komponente ili sustava obično u razvojnom okruženju od strane osobe koja razvija softver. [prema IEEE 610]

deviation: See *incident*.

odstupanje: Vidi *incident*.

deviation report: See *incident report*.

izvješće o odstupanju: Vidi *incident report*.

dirty testing: See *negative testing*.

prljavo testiranje: Vidi *negative testing*.

documentation testing: Testing the quality of the documentation, e.g. user guide or installation guide.

testiranje dokumentacije: Testiranje kvalitete dokumentacije, npr. korisničkog priručnika ili upute za instalaciju.

domain: The set from which valid input and/or output values can be selected.

domena: Skup iz kojeg se odabiru valjane ulazne i/ili izlazne vrijednosti.

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. [After TMap]

pogonski program: Programska komponenta ili alat za testiranje koji zamjenjuje komponentu koja se brine za kontrolu i/ili pozivanje komponente ili sustava. [prema TMap]

dynamic analysis: The process of evaluating behavior, e.g. memory performance, CPU usage, of a system or component during execution. [After IEEE 610]

dinamička analiza: Proces prosudbe ponašanja (npr. ponašanje performansi memorije, korištenje procesora) sustava ili komponente tijekom izvršenja. [prema 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.

alat za dinamičku analizu, dinamički analizator: Alat koji omogućava informacije o stanju programskog koda za vrijeme izvođenja. Ovi alati najčešće se koriste za otkrivanje nedodijeljenih pokazivača, za provjeru aritmetike pokazivača i nadzor dodjeljivanja, korištenja i oslobađanja memorije, te za signaliziranje gubljenja memorije.

dynamic comparison: Comparison of actual and expected results, performed while the software is being executed, for example by a test execution tool.

dinamička usporedba: Usporedba stvarnih i očekivanih rezultata, koja se provodi tijekom izvođenja softvera, primjerice pomoću alata za izvođenje testova.

dynamic testing: Testing that involves the execution of the software of a component or system.

dinamičko testiranje: Testiranje koje uključuje izvršavanje softverske komponente ili sustava.

E

efficiency: The capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions. [ISO 9126]

djelotvornost: Sposobnost programskog proizvoda da omogući odgovarajuću učinkovitost u odnosu na količinu resursa koji se koriste pod danim uvjetima. [ISO 9126]

efficiency testing: The process of testing to determine the efficiency of a software product.

testiranje djelotvornosti: Proces testiranja da bi se odredila djelotvornost programskog proizvoda.

elementary comparison testing: A black box test design technique in which test cases are designed to execute combinations of inputs using the concept of condition determination coverage. [TMap]

osnovno testiranje usporedbe: Tehnika oblikovanja testova crne kutije kod koje su testni slučajevi oblikovani za izvršavanje kombinacija ulaza koristeći koncept pokrivenosti odluka uvjeta. [TMap]

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 *simulator*.

emulator, simulator: Uređaj, računalni program ili sustav koji prihvaća iste ulaze i proizvodi iste izlaze kao dani sustav. [IEEE 610] Vidjeti također *simulator*.

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]

ulazni kriteriji, kriteriji ulaza: Skup generičkih i specifičnih uvjeta za dopuštanje procesu da nastavi sa definiranom radnjom, npr. faza testiranja. Svrha ulaznih kriterija je da onemoguće pokretanje radnje koja će povlačiti za sobom više (izgubljenog) napora u usporedbi sa naporom potrebnim da se uklone neuspjeli kriteriji ulaza. [Gilb and Graham]

entry point: The first executable statement within a component.

ulazna točka, mjesto ulaza: Prva izvršna naredba unutar komponente.

equivalence class: See *equivalence partition*.

klasa ekvivalencije: Vidi *equivalence partition*.

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.

ekvivalentna particija: Dio ulazne ili izlazne domene za koji se pretpostavlja da je ponašanje komponente ili sustava jednako na temelju specifikacije.

equivalence partition coverage: The percentage of equivalence partitions that have been exercised by a test suite.

pokrivenost ekvivalentne particije: Postotak ekvivalentnih particija koje su ispitane pomoću skupa testova.

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.

podjela na ekvivalentne particije: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvođenje predstavnika iz ekvivalentnih particija. U principu testni slučajevi su oblikovani da svaku particiju pokriju barem jednom.

error: A human action that produces an incorrect result. [After IEEE 610]

pogreška, kvar, neispravan rad: Ljudsko djelovanje koje proizvodi neispravan rezultat. [prema IEEE 610]

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.

pogađanje pogrešaka: Tehnika testiranja gdje se iskustvo testera koristi za predviđanje koje pogreške bi se mogle pojaviti u komponenti ili sustavu koji se testira kao rezultat učinjenih pogrešaka, te da bi se oblikovali specifični testovi za njihovo otkrivanje.

error seeding: See *fault seeding*.

umetanje pogrešaka: Vidi *fault seeding*.

error seeding tool: See *fault seeding tool*.

alat za sijanje pogrešaka: Vidi *fault seeding tool*.

error tolerance: The ability of a system or component to continue normal operation despite the presence of erroneous inputs. [After IEEE 610].

tolerancija pogrešaka: Sposobnost sustava ili komponente da nastavi sa normalnim radom bez obzira na prisustvo pogrešnog ulaza. [prema IEEE 610].

evaluation: See *testing*.

procjena, procjenjivanje, prosudba: Vidi *testing*.

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.

upravljanje iznimkama: Ponašanje komponente ili sustava kao odgovor na pogrešan ulaz, bilo da se radi o pogrešci korisnika, pogrešci druge komponente ili sustava ili o internoj pogrešci.

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.

izvršna naredba: Linija koda koja, kada je kompilirana, prevedena je u objektni kôd i koja će se proceduralno izvršiti kada je program pokrenut, te može izvoditi operaciju nad podacima.

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.

ispitan: Za element programa kaže se da je ispitan pomoću testnog slučaja kada ulazna vrijednost uzrokuje izvođenje tog elementa, kao primjerice naredba, odluka ili drugi strukturni element.

exhaustive testing: A test approach in which the test suite comprises all combinations of input values and preconditions.

iscrpno testiranje: Pristup testiranju u kojemu skup testova sadrži sve kombinacije ulaznih vrijednosti i preduvjeta.

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 to report against and to plan when to stop testing. [After Gilb and Graham]

izlazni kriteriji, kriteriji izlaza: Skup generičkih i specifičnih uvjeta koji su dogovoreni sa nositeljima interesa da bi se proces mogao službeno završiti. Svrha izlaznih kriterija je da spriječe da se radnja smatra završenom kada još uvijek postoje neispunjeni dijelovi radnje koji još nisu završeni. Kriteriji izlaza koriste se za izvještavanje te za planiranje kada prekinuti testiranje. [prema Gilb and Graham]

exit point: The last executable statement within a component.

izlazna točka: Posljednja izvršna naredba unutar komponente.

expected outcome: See *expected result*.

očekivani izlaz, očekivani rezultat: Vidi *expected result*.

expected result: The behavior predicted by the specification, or another source, of the component or system under specified conditions.

očekivani rezultat: Ponašanje komponente ili sustava pod određenim uvjetima koje je predviđeno specifikacijom ili nekim drugim izvorom.

experienced-based technique: See *experienced-based test design technique*.

tehnika zasnovana na iskustvu: Vidi *experienced-based test design technique*.

experienced-based test design technique: Procedure to derive and/or select test cases based on the tester's experience, knowledge and intuition.

tehnika oblikovanja testova zasnovana na iskustvu: Procedura za izvođenje i/ili odabir testnih slučajeva na temelju testerovog iskustva, znanja i intuicije.

exploratory testing: An informal test design technique where the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests. [After Bach]

eksplorativno testiranje, istraživačko testiranje: Neformalna tehnika testiranja kod koje tester aktivno kontrolira oblikovanje testova kada su ti testovi provedeni, te koristi dobivene informacije tijekom testiranja za oblikovanje novih i boljih testova. [prema Bach]

F

fail: A test is deemed to fail if its actual result does not match its expected result.

nije prošlo, neprolazno: Test se smatra da nije prošao ukoliko se stvarni rezultat razlikuje od očekivanog rezultata.

failure: Deviation of the component or system from its expected delivery, service or result. [After Fenton]

zatajenje, ispad: Odstupanje komponente ili sustava od očekivane isporuke, usluge ili rezultata. [prema Fenton]

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]

način rada u slučaju kvara, način rada u slučaju ispada: Fizička ili funkcionalna manifestacija zatajenja/ispada. Na primjer, sustav u ovom načinu rada je obilježen usporenim operacijama, neispravnim izlaznim rezultatima ili cjelokupnim prekidom rada. [prema 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 Failure Mode, Effect and Criticality Analysis (FMECA).

analiza potencijalnih neispravnosti i njihovih posljedica: Sustavni pristup identifikaciji rizika i analizi određivanja potencijalnih neispravnosti te pokušaj sprečavanja njihovog nastanka. Vidjeti također *Failure Mode, Effect and Criticality Analysis (FMECA)*.

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 Failure Mode and Effect Analysis (FMEA).

analiza potencijalnih neispravnosti, posljedica i kritičnosti: Proširena FMEA, koja u odnosu na osnovnu FMEA metodu uključuje i analizu kritičnosti, koja se koristi za prikazivanje vjerojatnosti potencijalnih neispravnosti u odnosu na težinu njihovih posljedica. Rezultat označava potencijalne neispravnosti sa relativno velikom vjerojatnošću i težinom posljedica, a omogućava da se dodatni napori usmjere tamo gdje će proizvesti najveću vrijednost. Vidjeti također *Failure Mode and Effect Analysis (FMEA)*.

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]

učestalost kvara, učestalost ispada: Omjer zatajenja odnosno ispada određene kategorije i jedinice mjere, npr. broj zatajenja u jedinici vremena, broj zatajenja po broju transakcija, broj zatajenja po broju pokretanja računala. [prema IEEE 610]

false-fail result: A test result in which a defect is reported although no such defect actually exists in the test object.

lažni rezultat neprolaza: Rezultat testa u kojemu je prijavljena greška koja zapravo ne postoji u objektu testiranja.

false-pass result: A test result which fails to identify the presence of a defect that is actually present in the test object.

lažni rezultat prolaza: Rezultat testa koji nije uspio identificirati prisutnost greške koja je zaista prisutna u objektu testiranja.

false-positive result: See *false-fail result*.

lažni pozitivni rezultat: Vidjeti također *false-fail result*.

false-negative result: See *false-pass result*.

lažni negativni rezultat: Vidjeti također *false-pass result*.

fault: See *defect*.

pogreška, propust: Vidjeti također *defect*.

fault attack: See *attack*.

napad pogreške: Vidjeti također *attack*.

fault density: See *defect density*.

gustoća pogreške: Vidjeti također *defect density*.

Fault Detection Percentage (FDP): See *Defect Detection Percentage (DDP)*.

postotak otkrivanja pogreške: Vidjeti također *Defect Detection Percentage (DDP)*.

fault masking: See *defect masking*.

maskiranje pogreške: Vidjeti također *defect masking*.

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]

umetanje pogrešaka: Proces namjernog dodavanja poznatih grešaka već postojećima u komponenti ili sustavu kako bi se nadzirala učestalost pronalaženja i uklanjanja grešaka te procijenio broj preostalih grešaka. [prema IEEE 610]

fault seeding tool: A tool for seeding (i.e. intentionally inserting) faults in a component or system.

alat za umetanje pogrešaka: Alat za sisanje (tj. namjerno umetanje) pogrešaka u komponentu ili sustav.

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 *reliability, robustness*.

otpornost na pogrešku, otpornost na kvar: Sposobnost programskog proizvoda da održi specificiranu razinu ponašanja u slučajevima programskih grešaka ili odstupanja od specificiranog sučelja. [ISO 9126] Vidjeti također *reliability, robustness*.

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.

analiza stabla greške, analiza stabla kvarova: Tehnika koja se koristi za analizu uzroka grešaka. Tehnika vizualno predočava kako se logičke povezanosti između zatajenja, ljudskih pogrešaka i vanjskih događaja mogu kombinirati te prouzročiti otkrivanje specifičnih grešaka.

feasible path: A path for which a set of input values and preconditions exists which causes it to be executed.
dopustiva putanja: Putanja za koju su definirane ulazne vrijednosti i preduvjeti koji uzrokuju njezino izvršenje.

feature: An attribute of a component or system specified or implied by requirements documentation (for example reliability, usability or design constraints). [After IEEE 1008]

osobina, obilježje, svojstvo, atribut: Atribut komponente ili sustava specificiran ili impliciran u dokumentaciji (na primjer pouzdanost, uporabljivost ili ograničenja dizajna). [prema IEEE 1008]

field testing: See *beta testing*.

testiranje polja: Vidjeti također *beta testing*.

finite state machine: A computational model consisting of a finite number of states and transitions between those states, possibly with accompanying actions. [IEEE 610]

automat konačnih stanja, konačni automat: Računski model koji se sastoji od konačnog broja stanja i prijelaza između tih stanja, po mogućnosti sa pripadajućim akcijama. [prema IEEE 610]

finite state testing: See *state transition testing*.

testiranje konačnih stanja: Vidjeti također *state transition testing*.

formal review: A review characterized by documented procedures and requirements, e.g. inspection.

formalni pregled: Pregled za koji su karakteristične dokumentirane procedure i zahtjevi, npr. inspekcija.

frozen test basis: A test basis document that can only be amended by a formal change control process. See also *baseline*.

zamrznuta osnova testiranja: Osnovni dokument testiranja koji može biti dopunjen samo kroz formalni proces upravljanja promjenama. Vidjeti također *baseline*.

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.

analiza funkcijskih točaka: Metoda čiji je cilj mjerenje veličine funkcionalnosti informacijskog sustava. Mjerenje je neovisno o tehnologiji. Ovo mjerenje može se koristiti kao osnova za mjerenje produktivnosti, procjenu potrebnih resursa i kontrolu projekta.

functional integration: An integration approach that combines the components or systems for the purpose of getting a basic functionality working early. See also *integration testing*.

funkcionalna integracija: Integracijski pristup koji kombinira komponente ili sustave sa svrhom funkcioniranja osnovnih funkcionalnosti u ranoj fazi. Vidjeti također *integration testing*.

functional requirement: A requirement that specifies a function that a component or system must perform. [IEEE 610]

funkcionalni zahtjev: Zahtjev koji specificira funkciju koju komponenta ili sustav mora izvršavati. [prema 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 *black box test design technique*.

tehnika oblikovanja funkcionalnih testova: Procedura za dobivanje i/ili odabiranje testnih slučajeva koji se temelje na analizi specifikacije funkcionalnosti komponente ili sustava bez obzira na njihovu unutrašnju strukturu. Vidjeti također *black box test design technique*.

functional testing: Testing based on an analysis of the specification of the functionality of a component or system. See also *black box testing*.

funkcijsko testiranje, funkcionalno testiranje: Testiranje temeljeno na analizi specifikacije funkcionalnosti komponente ili sustava. Vidjeti također *black box testing*.

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]

funkcionalnost: Sposobnost programskog proizvoda da osigura funkcije koje odgovaraju iskazanim i pretpostavljenim potrebama prilikom korištenja softvera pod određenim uvjetima. [prema ISO 9126]

functionality testing: The process of testing to determine the functionality of a software product.

testiranje funkcionalnosti: Proces testiranja kako bi se utvrdila funkcionalnost programskog proizvoda.

G

glass box testing: See *white box testing*.

testiranje staklene kutije, testiranje prozirne kutije: Vidjeti također *white box testing*.

H

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 *risk analysis*.

analiza opasnosti: Tehnika koja se koristi za uočavanje elemenata rizika. Rezultat te analize će pokrenuti metode koje se koriste u razvoju i testiranju sustava. Vidjeti također *risk analysis*.

heuristic evaluation: A static usability test technique to determine the compliance of a user interface with recognized usability principles (the so-called "heuristics").

heuristička procjena: Statička tehnika testiranja uporabljivosti za određivanje usklađenosti korisničkog sučelja sa priznatim principima uporabljivosti (takozvana „heuristička“).

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 *low level test case*.

testni slučaj više razine: Testni slučaj bez konkretnih (na razini implementacije) vrijednosti za ulazne podatke i očekivane rezultate. Koriste se logički operatori; instance stvarnih vrijednosti još nisu definirane i/ili dostupne. Vidjeti također *low level test case*.

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).

horizontalna sljedivost: Praćenje zahtjeva za razinu testiranja kroz slojeve testne dokumentacije (npr. plan testiranja, specifikacija dizajna testiranja, specifikacija testnih slučajeva, specifikacija testne procedure ili testna skripta).

hyperlink: A pointer within a web page that leads to other web pages.

hiperveza: Pokazivač unutar web stranice koji vodi prema drugim web stranicama.

hyperlink tool: A tool used to check that no broken hyperlinks are present on a web site.

alat za provjeru hiperveza: Alat koji se koristi za provjeru postojanja nevažećih hiperveza na web stranici.

I

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.

analiza utjecaja: Vrednovanje promjene u slojevima razvojne dokumentacije, testne dokumentacije i komponentama, kako bi se implementirala dana promjena prema specificiranim zahtjevima.

incident: Any event occurring that requires investigation. [After IEEE 1008]

incident: Svaki događaj koji se pojavi, a zahtjeva istragu. [prema IEEE 1008]

incident logging: Recording the details of any incident that occurred, e.g. during testing.

zapisivanje incidenata, bilježenje incidenata: Zapisivanje detalja bilo kojeg incidenta koji se pojavi, npr. tijekom testiranja.

incident management: The process of recognizing, investigating, taking action and disposing of incidents. It involves logging incidents, classifying them and identifying the impact. [After IEEE 1044]

upravljanje incidentima: Proces prepoznavanja, istraživanja, poduzimanja akcije i disponiranja grešaka. Obuhvaća bilježenje incidenata, klasifikaciju i definiranje njihovog utjecaja. [prema IEEE 1044]

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 *defect management tool*.

alat za upravljanje incidentima: Alat koji olakšava zapisivanje i praćenje statusa incidenata. Ovi alati često imaju sadržaje orijentirane prema redoslijedu događaja da bi pratili i kontrolirali alokaciju, korekciju i ponovno testiranje incidenata i omogućili izvještavanje. Vidjeti također *defect management tool*.

incident report: A document reporting on any event that occurred, e.g. during the testing, which requires investigation. [After IEEE 829]

izvješće o incidentu: Dokument koji pokazuje događaje koji su se pojavili, npr. tijekom testiranja, a zahtijevaju istragu. [prema IEEE 829]

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.

model inkrementalnog razvoja: Životni ciklus razvoja gdje je projekt podijeljen u serije inkremenata, a svaki od njih nosi dio funkcionalnosti u cjelokupnim zahtjevima projekta. Zahtjevi su prioritizirani i dostavljeni prema prioritetu u odgovarajućem inkreментu. U nekim (ne u svim) verzijama modela životnog ciklusa, svaki podprojekt slijedi „mini V-model“ s vlastitim dizajnom, kodiranjem i fazama testiranja.

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.

inkrementalno testiranje: Testiranje gdje su komponente ili sustavi povezani i testirani pojedinačno ili više njih istodobno tako dugo dok se sve komponente ili sustavi integriraju i testiraju.

independence of testing: Separation of responsibilities, which encourages the accomplishment of objective testing. [After DO-178b]

neovisnost testiranja: Razdvajanje odgovornosti, koje potiče ostvarenje objektivnog testiranja. [Prema DO-178b]

infeasible path: A path that cannot be exercised by any set of possible input values.

neostvariva putanja: Putanja koja se ne može primjenjivati ni s bilo kojim mogućim skupom ulaznih vrijednosti.

informal review: A review not based on a formal (documented) procedure.

neformalni pregled: Izvještaj koji se ne temelji na propisanoj (dokumentiranoj) proceduri.

input: A variable (whether stored within a component or outside) that is read by a component.

ulaz: Varijabla (pohranjena u komponenti ili eksterno) koju čita komponenta.

input domain: The set from which valid input values can be selected. See also *domain*.

ulazna domena: Skup iz kojeg se mogu selektirati valjane ulazne vrijednosti. Vidjeti također *domain*.

input value: An instance of an input.

ulazna vrijednost, vrijednost ulaza: Instanca ulazne vrijednosti.

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. [After IEEE 610, IEEE 1028] See also *peer review*.

inspekcija: Vrsta neformalnog ravnopravnog pregleda koji se oslanja na vizualni pregled dokumenata kako bi se uočile greške, npr. odstupanja od standarda razvoja i neusklađenost s dokumentacijom više razine. Najformalnija tehnika pregleda, prema tome, uvijek se temelji na dokumentiranoj proceduri. [prema IEEE 610, IEEE 1028] Vidjeti također *peer review*.

inspection leader: See *moderator*.

voditelj inspekcije: Vidjeti također *moderator*.

inspector: See *reviewer*.

inspektor: Vidjeti također *reviewer*.

installability: The capability of the software product to be installed in a specified environment [ISO 9126]. See also *portability*.

sposobnost instaliranja: Sposobnost programskog proizvoda da bude instaliran/postavljen u specificirano okruženje. [prema ISO 9126]. Vidjeti također *portability*.

installability testing: The process of testing the installability of a software product. See also *portability testing*.

testiranje instalacije: Proces testiranja sposobnosti instalacije programskog proizvoda. Vidjeti također *portability testing*.

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.

vodič za instalaciju, upute za instalaciju: Isporučene upute na bilo kojem primjerenom mediju, koje vode osobu koja instalira kroz instalacijski proces. To mogu biti priručnik, procedura korak-po-korak, čarobnjak za instalaciju ili bilo kakav sličan opis procesa.

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.

čarobnjak za instalaciju: Isporučeni program na primjerenom mediju, koji vodi osobu koja instalira kroz instalacijski proces. U pravilu pokreće instalacijski proces, osigurava povratnu informaciju o rezultatima instalacije i traži odabir opcija.

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.

instrumentacija: Ubacivanje dodatnog koda u program kako bi se prikupile informacije o ponašanju programa tijekom izvršenja, npr. mjerenje pokrivenosti koda.

instrumenter: A software tool used to carry out instrumentation.

instrumenter: Programski alat koji se koristi za provedbu instrumentacije.

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 *smoke test*.

test uvođenja: Posebna vrsta dimnog testa za odlučivanje da li je komponenta ili sustav spremna za daljnje detaljno testiranje. Test uvođenja, u pravilu, provodi se na početku faze izvršenja testa. Vidjeti također *smoke test*.

integration: The process of combining components or systems into larger assemblies.

integracija: Proces kombiniranja komponenti ili sustava u veće cjeline.

integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also *component integration testing*, *system integration testing*.

testiranje integracije, integracijska provjera: Testiranje koje se provodi kako bi se uvidjele greške u sučeljima i u interakcijama između integriranih komponenti ili sustava. Vidjeti također *component integration testing*, *system integration testing*.

integration testing in the large: See *system integration testing*.

testiranje integracije na veliko: Vidjeti također *system integration testing*.

integration testing in the small: See *component integration testing*.

testiranje integracije na malo: Vidjeti također *component integration testing*.

interface testing: An integration test type that is concerned with testing the interfaces between components or systems.

testiranje sučelja: Vrsta integracijskog testa koja obuhvaća testiranje sučelja između komponenti ili sustava.

interoperability: The capability of the software product to interact with one or more specified components or systems. [After ISO 9126] See also *functionality*.

interoperabilnost: Sposobnost programskog proizvoda da surađuje s jednom ili više specificiranih komponenti ili sustava. [prema ISO 9126] Vidjeti također *functionality*.

interoperability testing: The process of testing to determine the interoperability of a software product. See also *functionality testing*.

testiranje interoperabilnosti: Proces testiranja s ciljem utvrđivanja interoperabilnosti programskog proizvoda. Vidjeti također *functionality testing*.

invalid testing: Testing using input values that should be rejected by the component or system. See also *error tolerance*.

testiranje neispravnosti: Testiranje u kojem se koriste ulazne vrijednosti koje bi komponenta ili sustav trebale odbiti. Vidjeti također *error tolerance*.

isolation testing: Testing of individual components in isolation from surrounding components, with surrounding components being simulated by stubs and drivers, if needed.

izolacijsko testiranje: Testiranje individualnih komponenti u izolaciji od okolnih komponenti, s okolnim komponentama koje su simulirane preko prividnih modula i pogonskih programa, ukoliko je potrebno.

item transmittal report: See *release note*.

prijenosni izvještaj podataka: Vidjeti također *release note*.

iterative development model: A development life cycle where a project is broken into a usually large number 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.

model iterativnog razvoja: Životni ciklus razvoja u kojem je projekt uobičajeno podijeljen u velik broj iteracija. Iteracija je potpuna petlja razvoja koja rezultira distribucijom izvršnog izdanja (interno ili eksterno izdanje) proizvoda, podskupom konačnog proizvoda u razvoju koji se iz iteracije u iteraciju razvija u konačni proizvod.

K

key performance indicator: See *performance indicator*.

ključni pokazatelj uspješnosti: Vidjeti također *performance indicator*.

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 *data driven testing*.

testiranje vođeno ključnim riječima: Skriptna tehnika koja ne koristi samo datoteke koje sadržavaju testne podatke i očekivane rezultate, već i ključne riječi povezane s aplikacijom koja se testira. Ključne riječi su interpretirane pomoću specijalnih podržanih skripti koje su pozvane od strane kontrolne skripte za testiranje. Vidjeti također *data driven testing*.

L

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.

LCSAJ: *Linear Code Sequence And Jump*, sadržava slijedeća tri elementa (dogovoreno je da se identificira brojem linija u ispisu izvornog koda): početak linearnog slijeda linija za izvršavanje, kraj linearnog slijeda i ciljna linija preko koje kontrolni tijek ide prema kraju linearnog slijeda.

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 pokrivenost: Postotak LCSAJ-a komponente koja je ispitana pomoću skupa testnih slučajeva. 100% -tna LCSAJ pokrivenost implicira 100%-tnu pokrivenost odluka.

LCSAJ testing: A white box test design technique in which test cases are designed to execute LCSAJs.

LCSAJ testiranje: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani za izvršavanje LCSAJ-a (*Linear Code Sequence and Jump*).

learnability: The capability of the software product to enable the user to learn its application. [ISO 9126] See also *usability*.

sposobnost učenja: Sposobnost programskog proizvoda da omogući korisniku učenje kako ga primjeniti. [ISO 9126] Vidjeti također *usability*.

level test plan: A test plan that typically addresses one test level. See also *test plan*.

plan testiranja razine: Plan testiranja koji u pravilu obuhvaća jednu razinu testiranja. Vidjeti također *test plan*.

link testing: See *component integration testing*.

testiranje veza: Vidjeti također *component integration testing*.

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 *operational profile*.

profil opterećenja: Specifikacija aktivnosti koja se može pojaviti prilikom testiranja komponente ili sustava u produkciji. Profil opterećenja čini određeni broj virtualnih korisnika koji obrađuju definirani skup transakcija u točno određenom vremenskom periodu i prema unaprijed definiranom operacijskom profilu. Vidjeti također *operational profile*.

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*, *stress testing*.

testiranje opterećenja: Vrsta testiranja učinkovitosti koja evaluira ponašanje komponente ili sustava s povećanim opterećenjem, npr. broj paralelnih korisnika i/ili broj transakcija, da bi se utvrdilo opterećenje koje komponenta ili sustav može podnijeti. Vidjeti također *performance testing*, *stress testing*.

logic-coverage testing: See *white box testing*. [Myers]

testiranje logičke pokrivenosti: Vidi *white box testing*. [prema Myers]

logic-driven testing: See *white box testing*.

testiranje vođeno logikom: Vidi *white box testing*.

logical test case: See *high level test case*.

logički testni slučaj: Vidi *high level test case*.

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 *high level test case*.

testni slučaj niže razine: Testni slučaj sa konkretnim (razina implementacije) vrijednostima za ulazne podatke i očekivane rezultate. Logički operatori iz testnih slučajeva viših razine su zamijenjeni sa stvarnim vrijednostima koje odgovaraju logičkim operatorima. Vidjeti također *high level test case*.

M

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]

održavanje: Modifikacija programskog proizvoda nakon isporuke kako bi se ispravile greške, poboljšala učinkovitost ili drugi atributi, ili da bi se proizvod prilagodio promijenjenom okruženju. [prema IEEE 1219]

maintenance testing: Testing the changes to an operational system or the impact of a changed environment to an operational system.

testiranje održavanja: Testiranje promjena operacijskog sustava ili utjecaja promijenjene okoline na operacijski sustav.

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]

pogodnost za održavanje, održivost: Lakoća kojom programski proizvod može biti modificiran zbog ispravljanja grešaka, zadovoljavanja novih zahtjeva, olakšavanja budućeg održavanja ili prilagodbe promijenjenoj okolini. [ISO 9126]

maintainability testing: The process of testing to determine the maintainability of a software product.

testiranje održivosti: Proces testiranja kako bi se odredila održivost programskog proizvoda.

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. [After IEEE 610, IEEE 1028]

ocjena uprave: Sistematična evaluacija akvizicije softvera, nabave, razvoja, operacija ili procesa održavanja od strane uprave koja nadzire napredak, odlučuje o statusu planova i rasporeda, potvrđuje zahtjeve i njihovu alokaciju u sustavu ili ocjenjuje učinkovitost upravnog pristupa da bi se postigla sposobnost za primjenu. [prema IEEE 610, IEEE 1028]

master test plan: A test plan that typically addresses multiple test levels. See also *test plan*.

glavni plan testiranja, temeljni plan testiranja: Plan testiranja koji u pravilu upućuje na višestruke razine testiranja. Vidjeti također *test plan*.

maturity: (1) The capability of an organization with respect to the effectiveness and efficiency of its processes and work practices. See also *Capability Maturity Model, Test Maturity Model*. (2) The capability of the software product to avoid failure as a result of defects in the software. [ISO 9126] See also *reliability*.

zrelost: (1) Sposobnost organizacije u odnosu na učinkovitost i djelotvornost vlastitih procesa i radnih praksi. Vidjeti također *Capability Maturity Model, Test Maturity Model*. (2) Sposobnost programskog proizvoda da izbjegne zatajenje kao posljedicu grešaka u softveru. [ISO 9126] Vidjeti također *reliability*.

measure: The number or category assigned to an attribute of an entity by making a measurement. [ISO 14598]

mjera, jedinica mjere: Broj ili kategorija dodijeljena atributu nekog objekta prilikom mjerenja. [ISO 14598]

measurement: The process of assigning a number or category to an entity to describe an attribute of that entity. [ISO 14598]

mjerenje, sustav mjerenja: Proces dodjeljivanja broja ili kategorije nekoj određenoj jedinici da bi opisali atribute te jedinice. [ISO 14598]

measurement scale: A scale that constrains the type of data analysis that can be performed on it. [ISO 14598]

mjerna skala: Skala koja ograničava vrstu analize podataka se može primjeniti na njoj. [ISO 14598]

memory leak: A defect in a program's dynamic store allocation logic that causes it to fail to reclaim memory after it has finished using it, eventually causing the program to fail due to lack of memory.

gubljenje memorije, nestajanje memorije: Greška dinamičke logike pohrane podataka u programu koja uzrokuje pad memorije nakon njenog korištenja, što obično izaziva pad programa zbog nedostatka memorije.

metric: A measurement scale and the method used for measurement. [ISO 14598]

metrika: Mjerna skala i metoda koje se koriste za mjerenje. [ISO 14598]

migration testing: See *conversion testing*.

testiranje migracije: Vidi *conversion testing*.

milestone: A point in time in a project at which defined (intermediate) deliverables and results should be ready.

kontrolna točka: Određen vremenski period u projektu u kojem trebaju biti spremne definirane (prijelazne) isporuke.

mistake: See *error*.

pogreška, zabuna: Vidi *error*.

modelling tool: A tool that supports the validation of models of the software or system [Graham].

alat za modeliranje: Alat koji podržava validaciju modela softvera ili sustava. [Graham].

moderator: The leader and main person responsible for an inspection or other review process.

moderator: Voditelj i glavna osoba odgovorna za inspekciju ili drugi proces pregleda.

modified condition decision coverage: See *condition determination coverage*.

modificirana pokrivenost odluka uvjeta: Vidi *condition determination coverage*.

modified condition decision testing: See *condition determination testing*.

testiranje modificirane pokrivenosti odluka uvjeta: Vidi *condition determination testing*.

modified multiple condition coverage: See *condition determination coverage*.

modificirana pokrivenost višestrukih uvjeta: Vidi *condition determination coverage*.

modified multiple condition testing: See *condition determination testing*.

testiranje modificirane pokrivenosti višestrukih uvjeta: Vidi *condition determination testing*.

module: See *component*.

modul: Vidi *component*.

module testing: See *component testing*.

testiranje modula: Vidi *component testing*.

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. [After IEEE 610]

monitor: Programski alat ili hardverski uređaj koji radi usporedno s komponentom ili sustavom koji se testira. Nadzire, bilježi i/ili analizira ponašanje komponente ili sustava. [prema IEEE 610]

monitoring tool: See *monitor*.

alat za nadzor: Vidi *monitor*.

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.

testiranje ponavljanja: Testiranje prema kriteriju slučajnog odabira iz velikog raspona ulaznih vrijednosti i korištenje nasumično bez obzira na to kako se proizvod inače koristi.

multiple condition: See *compound condition*.

višestruki uvjet: Vidi *compound condition*.

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.

pokrivenost višestrukih uvjeta: Postotak kombinacija svih rezultata jednog uvjeta unutar jedne linije koda, a koje su ispitane pomoću skupa testnih slučajeva. 100%-tna pokrivenost višestrukih uvjeta implicira 100%-tnu pokrivenost odluka uvjeta.

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).

testiranje višestrukih uvjeta: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani s ciljem izvršavanja kombinacija izlaznih rezultata jednog uvjeta (unutar jedne linije koda).

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.

analiza mutacije: Metoda kojom se određuje cjelovitost skupa testnih slučajeva uzimajući u obzir mjeru do koje skup testnih slučajeva može praviti razliku između programa i neznatnih varijanti (mutanata) programa.

mutation testing: See *back-to-back testing*.

testiranje mutacije: Vidi *back to back testing*.

N

N-switch coverage: The percentage of sequences of N+1 transitions that have been exercised by a test suite. [Chow]

pokrivenost višestrukih izbora: Postotak slijedova N+1 prijelaza koji su ispitani skupom testnih slučajeva. [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 *state transition testing*.

testiranje višestrukih izbora: Oblik testiranja prijelaza stanja u kojem su testni slučajevi oblikovani s ciljem izvođenja svih važećih slijedova N+1 prijelaza. [Chow] Vidjeti također *state transition testing*.

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. [After Beizer].

negativno testiranje: Testovi čiji je cilj da pokažu da komponenta ili sustav ne radi. Negativno testiranje više je povezano sa stavom testera nego s posebnim pristupom testiranju ili tehnikom oblikovanja testa, npr. testiranje sa negativnim ulaznim vrijednostima ili iznimkama. [prema Beizer]

non-conformity: Non fulfillment of a specified requirement. [ISO 9000]

nesukladnost, neusklađenost: Neispunjavanje specificiranog zahtjeva. [ISO 9000]

non-functional requirement: A requirement that does not relate to functionality, but to attributes such as reliability, efficiency, usability, maintainability and portability.

nefunkcionalni zahtjev: Zahtjev koji nije povezan s funkcionalnostima nego s atributima kao što su pouzdanost, djelotvornost, uporabljivost, održivost i prenosivost.

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.

nefunkcionalno testiranje: Testiranje atributa komponente ili sustava koji nisu povezani s funkcionalnošću, primjerice, pouzdanost, djelotvornost, uporabljivost, održivost i prenosivost.

non-functional test design techniques: Procedure to derive and/or select test cases for nonfunctional testing based on an analysis of the specification of a component or system without reference to its internal structure. See also *black box test design technique*.

tehnike oblikovanja nefunkcionalnih testova: Procedura za izvođenje i/ili odabir testnih slučajeva za nefunkcionalno testiranje koja se temelji na analizi specifikacije komponente ili sustava bez obzira na unutarnju strukturu komponente ili sustava. Vidjeti također *black box test design technique*.

O

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.

off-the-shelf softver: Programski proizvod koji je razvijen za generalno tržište, tj. za velik broj korisnika, te koji je dostavljen velikom broju korisnika u identičnom formatu.

operability: The capability of the software product to enable the user to operate and control it. [ISO 9126] See also *usability*.

operabilnost: Sposobnost programskog proizvoda da omogući korisniku upravljanje i kontrolu istog. [ISO 9126] Vidjeti također *usability*.

operational acceptance testing: Operational testing in the acceptance test phase, typically performed in a simulated real-life operational environment by operator and/or administrator focusing on operational aspects, e.g. recoverability, resource-behavior, installability and technical compliance. See also *operational testing*.

operacijsko testiranje prihvatljivosti: Operacijsko testiranje u razni testiranja prihvatljivosti, obično izvedeno u simuliranom okruženju koje nalikuje što više produkciji od strane operatora i/ili administratora, a fokusira se na operacijske aspekte, npr. oporavak sustava, ponašanje resursa, sposobnost instalacije i tehnička usklađenost. Vidjeti također *operational testing*.

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.

operacijsko okruženje, operacijska okolina: Hardverski i softverski proizvodi instalirani na lokaciji korisnika ili kupaca na kojoj će se koristiti komponenta ili sustav koji se testira. Softver može obuhvaćati operacijske sustave, sustave za upravljanje bazom podataka i ostale aplikacije.

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.

operacijski profil: Prikaz zasebnih skupova radnji koje izvodi komponenta ili sustav, vjerojatno baziranih na ponašanju korisnika prilikom interakcije s komponentom ili sustavom, te njihova vjerojatnost pojavljivanja. Radnja više pripada logičkim nego fizičkim svojstvima i može se izvršiti na više strojeva ili u nepovezanim vremenskim segmentima.

operational profile testing: Statistical testing using a model of system operations (short duration tasks) and their probability of typical use. [Musa]

testiranje operacijskog profila: Statističko testiranje prilikom kojeg se koristi model sistemskih operacija (kratkotrajne radnje) i vjerojatnost njihove uobičajene uporabe. [Musa]

operational testing: Testing conducted to evaluate a component or system in its operational environment. [IEEE 610]

operacijsko testiranje: Testiranje provedeno s ciljem evaluacije komponente ili sustava u njihovom operacijskom okruženju. [IEEE 610]

oracle: See *test oracle*.

izvor: Vidi *test oracle*.

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.

ortogonalno polje: Dvodimanzionalno polje konstruirano sa specijalnim matematičkim svojstvima na način da izbor bilo kojih dviju kolona u polju daje kombinaciju svih parova svakog broja u polju.

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 *pairwise testing*.

testiranje ortogonalnog polja: Sistematičan način testiranja svih kombinacija parova varijabli koristeći ortogonalna polja. Značajno smanjuje broj svih kombinacija varijabli da bi se testirali svi parovi kombinacija. Vidjeti također *pairwise testing*.

outcome: See *result*.

izlaz, rezultat: Vidi *result*.

output: A variable (whether stored within a component or outside) that is written by a component.

izlaz: Varijabla (pohranjena u komponenti ili izvan nje) koju upisuje komponenta.

output domain: The set from which valid output values can be selected. See also *domain*.

izlazna domena: Skup iz kojeg se mogu selektirati valjane izlazne vrijednosti. Vidjeti također *domain*.

output value: An instance of an output. See also *output*.

izlazna vrijednost: Instanca izlazne vrijednosti. Vidi *output*.

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.

programiranje u paru: Pristup razvoju softvera gdje su linije koda (produksijske i/ili testne) komponente pisane od strane dva programera koji rade u paru na jednom računalu. Implicitno to znači da se provjere koda izvode u stvarnom vremenu.

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.

testiranje u paru: Dvije osobe, npr. dva testera, developer i tester, krajnji korisnik i tester, koji rade zajedno kako bi pronašli greške. U pravilu, oni dijele jedno računalo i kontrolu tijekom testiranja.

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.

usporedno testiranje: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvršavanje svih mogućih diskretnih kombinacija svakog para ulaznih parametara.

partition testing: See *equivalence partitioning*. [Beizer]

testiranje particije: Vidi *equivalence partitioning*. [Beizer]

pass/fail criteria: Decision rules used to determine whether a test item (function) or feature has passed or failed a test. [IEEE 829]

kriterij prolaza/neprolaza: Pravila odlučivanja prema kojima se određuje da li je jedinica testiranja (funkcija) ili svojstvo prošlo ili nije prošlo test. [IEEE 829]

path: A sequence of events, e.g. executable statements of a component or system from an entry point to an exit point.

staza, put, putanja: Slijed događaja, npr. izvršne linije koda komponente ili sustava od ulazne točke do izlazne točke.

path coverage: The percentage of paths that have been exercised by a test suite. 100% path coverage implies 100% LCSAJ coverage.

pokrivenost putanji: Postotak putanji koje su ispitane u skupu testnih slučajeva. 100%-tna pokrivenost putanji implicira 100%-tnu LCSAJ pokrivenost.

path sensitizing: Choosing a set of input values to force the execution of a given path.

senzibiliranje putanje: Odabiranje skupa ulaznih vrijednosti kako bi se ostvarilo izvršenje dane putanje.

path testing: A white box test design technique in which test cases are designed to execute paths.

testiranje putanje, testiranje po putovima: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani za izvršavanje putanji.

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.

ravnopravni pregled: Izveštaj o radu programskog proizvoda od strane kolega proizvođača programskog proizvoda kako bi se identificirale greške i moguća poboljšanja. Primjeri su inspekcija, tehnički pregled i prolazni pregled.

performance: The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate. [After IEEE 610] See also *efficiency*.

učinkovitost: Stupanj do kojeg sustav ili komponenta ispunjava svoje funkcije unutar danih ograničenja s obzirom na vrijeme obrade i stopu propusnosti. [prema IEEE 610] Vidjeti također *efficiency*.

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]

pokazatelj učinkovitosti: Metrika više razine učinkovitosti i/ili djelotvornosti korištena za provedbu i kontrolu progresivnog razvoja, npr. vremenski pomak za razvoj softvera. [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 *load profile*, *operational profile*.

profiliranje učinkovitosti: Definicija korisničkih profila za testiranje učinkovitosti, opterećenja i/ili testiranje pod pritiskom. Profili bi trebali odražavati predviđeno ili stvarno korištenje na temelju operacijskog profila komponente ili sustava, te nadalje očekivani radni učinak. Vidjeti također *load profile*, *operational profile*.

performance testing: The process of testing to determine the performance of a software product.

testiranje učinkovitosti: Proces testiranja kako bi se utvrdila učinkovitost programskog proizvoda.

performance testing tool: A tool to support performance testing and 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.

alat za testiranje učinkovitosti: Alat za podršku testiranja učinkovitosti obično ima dvije osnovne značajke: generiranje opterećenja i mjerenje transakcija testa. Generiranje opterećenja može simulirati ili višestruke korisnike ili visoke obujme ulaznih podataka. Tijekom izvršenja, uzimaju se i bilježe mjerenja vremena odgovora iz odabranih transakcija. Alati za testiranje učinkovitosti uobičajeno omogućuju izvješća bazirana na zapisima testiranja i grafikone opterećenja u odnosu na vremena odgovora.

phase test plan: A test plan that typically addresses one test phase.

plan testiranja faze: Plan testiranja koji obično obuhvaća jednu fazu testiranja.

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]

pokazivač: Podatkovna jedinica koja navodi lokaciju druge podatkovne jedinice; na primjer, podatkovna jedinica koja određuje adresu sljedećeg zapisa o zaposleniku koji će se obrađivati. [IEEE 610]

portability: The ease with which the software product can be transferred from one hardware or software environment to another. [ISO 9126]

prenosivost: Svojstvo programskog proizvoda da se može premjestiti iz jednog hardverskog ili softverskog okruženja u drugo. [ISO 9126]

portability testing: The process of testing to determine the portability of a software product.

testiranje prenosivosti: Proces testiranja s ciljem utvrđivanja prenosivosti programskog proizvoda.

postcondition: Environmental and state conditions that must be fulfilled after the execution of a test or test procedure.

postuvjet, kasniji uvjet: Uvjeti okruženja i stanja koji moraju biti zadovoljeni nakon izvršenja testa ili testne procedure.

post-execution comparison: Comparison of actual and expected results, performed after the software has finished running.

usporedba nakon izvršavanja: Usporedba stvarnih i očekivanih rezultata, koja je izvršena nakon što je softver završio obradu.

precondition: Environmental and state conditions that must be fulfilled before the component or system can be executed with a particular test or test procedure.

preduvjet: Uvjeti okruženja i stanja koji moraju biti zadovoljeni prije nego na komponenti ili sustavu bude proveden određeni test ili testna procedura.

predicted outcome: See *expected result*.

predviđeni rezultat: Vidi *expected result*.

pretest: See *intake test*.

predtest: Vidi *intake test*.

priority: The level of (business) importance assigned to an item, e.g. defect.

prioritet: Razina (poslovne) važnosti dodijeljena nekoj jedinici, npr. greška.

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.

testiranje procedure: Testiranje radi osiguranja da komponenta ili sustav može djelovati zajedno s novim ili postojećim korisničkim poslovnim procedurama ili operacijskim procedurama.

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.

efekt probe: Utjecaj na komponentu ili sustav pomoću mjernog instrumenta prilikom procesa mjerenja komponente ili sustava, npr. utjecaj alata za testiranje učinkovitosti ili nadzora. Na primjer, učinkovitost može biti blago umanjena prilikom korištenja alata za testiranje učinkovitosti.

problem: See *defect*.

problem: Vidi *defect*.

problem management: See *defect management*.

upravljanje problemima: Vidi *defect management*.

problem report: See *defect report*.

izvještaj o problemu: Vidi *defect report*.

process: A set of interrelated activities, which transform inputs into outputs. [ISO 12207]

proces: Skup međusobno povezanih aktivnosti, koje transformiraju ulaze u izlaze. [ISO 12207]

process cycle test: A black box test design technique in which test cases are designed to execute business procedures and processes. [TMap] See also *procedure testing*.

test ciklusa procesa: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvršavanje poslovnih procedura i procesa. [TMap] Vidjeti također *procedure testing*.

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]

poboljšanje procesa, unaprjeđenje procesa: Program aktivnosti oblikovan za unaprjeđenje učinkovitosti i zrelosti procesa organizacije, kao i rezultatata samog programa. [CMMI]

production acceptance testing: See *operational acceptance testing*.

testiranje prihvatljivosti produkcije: Vidi *operational acceptance testing*.

product risk: A risk directly related to the test object. See also *risk*.

rizik proizvoda: Rizik koji je povezan direktno s testnim objektom. Vidjeti također *risk*.

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]

projekt: Skup koordiniranih i kontroliranih aktivnosti s datumom početka i kraja kako bi se postigli ciljevi u skladu sa specifičnim zahtjevima uzimajući u obzir ograničenja vremena, troškova i resursa. [ISO 9000]

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 *risk*.

rizik projekta: Rizik povezan s upravljanjem i kontrolom (testnog) projekta, npr. nedostatak osoblja, strogo određeni rokovi, promjenjivi zahtjevi itd. Vidjeti također *risk*.

program instrumenter: See *instrumenter*.
instrumenter programa: Vidi *instrumenter*.

program testing: See *component testing*.
testiranje programa: Vidi *component testing*.

project test plan: See *master test plan*.
plan testiranja projekta: Vidi *master test plan*.

pseudo-random: A series which appears to be random but is in fact generated according to some prearranged sequence.

pseudoslučajan, lažno slučajan: Serije koje izgledaju kao da su nasumično odabrane, ali zapravo su generirane prema nekim predodređenim slijedovima.

Q

qualification: The process of demonstrating the ability to fulfill specified requirements. Note the term 'qualified' is used to designate the corresponding status. [ISO 9000]

kvalifikacija: Proces demonstracije sposobnosti ispunjavanja specificiranih zahtjeva. Uočite da se pojam 'kvalificiran' koristi za označavanje određenog statusa. [ISO 9000]

quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations. [After IEEE 610]

kvaliteta: Stupanj do kojeg komponenta, sustav ili proces udovoljava specificiranim zahtjevima i/ili korisničkim/kupčevim potrebama i očekivanjima. [prema IEEE 610]

quality assurance: Part of quality management focused on providing confidence that quality requirements will be fulfilled. [ISO 9000]

osiguranje kvalitete, osiguravanje kvalitete: Dio upravljanja kvalitetom fokusiran na uspostavljanje pouzdanosti da će zahtjevi za kvalitetom biti ispunjeni. [ISO 9000]

quality attribute: A feature or characteristic that affects an item's quality. [IEEE 610]

atribut kvalitete, faktor kvalitete, obilježje kvalitete: Svojstvo ili karakteristika koja utječe na kvalitetu neke jedinice. [IEEE 610]

quality characteristic: See *quality attribute*.

karakteristika kvalitete, obilježje kvalitete: Vidi *quality attribute*.

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]

kvaliteta upravljanja, upravljanje kvalitetom: Koordinirane aktivnosti za upravljanje i kontrolu organizacije u odnosu na kvalitetu. Upravljanje i kontrola u odnosu na kvalitetu obično uključuju uspostavljanje politike kvalitete i ciljeva kvalitete, planiranje kvalitete, kontrolu kvalitete, osiguranje kvalitete i unaprjeđenje kvalitete. [ISO 9000]

R

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.

slučajno testiranje, nasumično testiranje: Tehnika oblikovanja testova crne kutije gdje su testni slučajevi odabrani korištenjem pseudo slučajnog algoritma za generiranje da bi odgovarali operacijskom profilu. Ova tehnika može se koristiti za testiranje nefunkcionalnih atributa kao što su pouzdanost i performansa.

recorder: See *scribe*.

zapisničar: Vidi *scribe*.

record/playback tool: See *capture/playback tool*.

alat za snimanje/reprodukciju: Vidi *capture/playback tool*.

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 *reliability*.

spособnost oporavka: Sposobnost programskog proizvoda da ponovo ustanovi određenu razinu izvođenja i oporavi podatke koji su direktno pogođeni u slučaju kvara. [ISO 9126] Vidjeti također *reliability*.

recoverability testing: The process of testing to determine the recoverability of a software product.

testiranje sposobnosti oporavka: Proces testiranja koji utvrđuje sposobnost oporavka programskog proizvoda.

recovery testing: See *recoverability testing*.

testiranje oporavka: Vidi *recoverability testing*.

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.

regresijsko testiranje: Testiranje modifikacija prethodno testiranog programa, kako bi se osiguralo da nisu nastale nove greške u nepromijenjenim dijelovima softvera kao rezultat promjena. Ovo testiranje se izvodi kada se promijeni softver ili njegova okolina.

regulation testing: See *compliance testing*.

testiranje propisa: Vidi *compliance testing*.

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. [After IEEE 829]

napomena uz izdanje, podaci o novom izdanju, podaci o novoj inačici: Dokument koji identificira jedinice testiranja, njihovu konfiguraciju, trenutni status i druge podatke o isporuci koji su isporučeni od strane razvoja i testiranja, kao i od drugih nositelja interesa, na početku faze izvršavanja testova. [prema IEEE 829]

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]

pouzdanost: Sposobnost programskog proizvoda da tijekom utvrđenog vremenskog razdoblja ili za određeni broj operacija izvodi zahtijevane funkcije pod utvrđenim okolnostima. [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.

model rasta pouzdanosti: Model koji pokazuje rast pouzdanosti tijekom kontinuiranog testiranja komponente ili sustava tijekom vremena kao rezultat uklanjanja defekata koji rezultiraju izostancima pouzdanosti.

reliability testing: The process of testing to determine the reliability of a software product.

testiranje pouzdanosti: Proces testiranja kako bi se utvrdila pouzdanost programskog proizvoda.

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 *portability*.

spособnost zamjene: Sposobnost programskog proizvoda da se koristi umjesto drugog određenog programskog proizvoda za istu svrhu i u istoj okolini. [ISO 9126] Vidjeti također *portability*.

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. [After IEEE 610]

zahtjev: Stanje ili sposobnost koje potrebuje korisnik da riješi problem ili postigne cilj, a koji mora zadovoljavati ili posjedovati sustav ili komponenta sustava da zadovolji ugovor, standard, specifikaciju ili drugi formalno predočen document. [prema 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.

testiranje zasnovano na zahtjevima: Pristup testiranju u kojem su testni slučajevi oblikovani na temelju ciljeva testiranja i uvjeta testiranja izvedenih iz zahtjeva, npr. testovi koji izvršavaju specifične funkcije ili ispituju nefunkcionalna obilježja kao što su pouzdanost i uporabivost.

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.

alat za upravljanje zahtjevima: Alat koji podržava dokumentiranje zahtjeva, atributa zahtjeva (npr. prioritet, znanje, odgovorna osoba) i komentara, te olakšava sljedivost kroz slojeve zahtjeva i upravljanje promjenama zahtjeva. Neki alati za upravljanje zahtjevima također imaju mogućnost za statičku analizu kao što je provjera konzistencije i kršenja predefiniраниh uvjeta zahtjeva.

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]

faza zahtjeva: Vremensko razdoblje u životnom ciklusu softvera tijekom kojeg se definiraju i dokumentiraju zahtjevi softverskog proizvoda. [IEEE 610]

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. [After ISO 9126]
See also *efficiency*.

korištenje resursa, uporaba resursa: Mogućnost softverskog proizvoda da, kada izvršava svoju funkciju pod utvrđenim uvjetima, koristi prikladne količine i vrste resursa, npr. količinu glavne i sekundarne memorije koje koristi program i veličinu privremenih datoteka ili datoteka za prekoračenje. [prema ISO 9126] Vidjeti također *efficiency*.

resource utilization testing: The process of testing to determine the resource utilization of a software product. See also *efficiency testing*.

testiranje korištenja resursa, testiranje uporabe resursa: Proces testiranja radi utvrđivanja iskorištenja resursa programskog proizvoda. Vidjeti također *efficiency testing*.

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 *actual result, expected result*.

rezultat: Posljedica/ishod izvršavanja testa. Uključuje izlaz na ekrane, promjene nad podacima, izvještaje i poslana komunikacijske poruke. Vidjeti također *actual result, expected result*.

resumption criteria: The testing activities that must be repeated when testing is re-started after a suspension. [After IEEE 829]

kriteriji za nastavak: Aktivnosti testiranja koje se moraju ponoviti kada se testiranje ponovno pokreće nakon što je bilo obustavljeno. [prema IEEE 829]

re-testing: Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions.

ponovno testiranje: Testiranje koje pokreće testne slučajeve koji zadnji puta kada su bili pokrenuti nisu prošli sa svrhom provjere uspjeha korektivnih akcija.

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.

retrospektivski sastanak: Sastanak na kraju projekta tijekom kojeg članovi projektnog tima vrednuju projekt i sažimaju saznanja iz projekta koja se mogu primijeniti na sljedeći projekt.

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. [After IEEE 1028]

pregled: Vrednovanje statusa proizvoda ili projekta u svrhu provjeravanja postoje li diskrepancije u odnosu na planirane rezultate te da se predlože poboljšanja. Primjeri su ocjena uprave, neformalni pregled, tehnički pregled, inspekcija i prolaz. [prema IEEE 1028]

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.

revizor, ocjenjivač: Osoba uključena u pregled koja identificira i opisuje anomalije u proizvodu ili projektu pod pregledom. Revizori mogu biti birani tako da predstavljaju različita stanovišta i uloge u procesu pregleda.

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.

alat za pregled: Alat koji podržava proces pregleda. Tipična svojstva uključuju: podrška planiranju i praćenju pregleda, podrška komunikaciji, suradnički pregledi i repozitorij za prikupljanje i izvještavanje o metrikama.

risk: A factor that could result in future negative consequences; usually expressed as impact and likelihood.

rizik: Čimbenik koji bi mogao rezultirati negativnim posljedicama u budućnosti, obično se izražava kao utjecaj (*impact*) i vjerojatnost (*likelihood*).

risk analysis: The process of assessing identified risks to estimate their impact and probability of occurrence (*likelihood*).

analiza rizika: Proces procjenjivanja utvrđenih rizika u svrhu procjene njihovog utjecaja (*impact*) i vjerojatnosti učestalosti (*likelihood*).

risk-based testing: An approach to testing to reduce the level of product risks and inform stakeholders on their status, starting in the initial stages of a project. It involves the identification of product risks and their use in guiding the test process.

testiranje zasnovano na rizicima: Pristup testiranju koji u početnim fazama projekta želi reducirati razine rizika vezanih za neki proizvod i informirati nositelje interesa o stanju rizika. Ovaj pristup uključuje identifikaciju rizika vezanih za proizvod i njihovu uporabu u usmjeravanju procesa testiranja.

risk control: The process through which decisions are reached and protective measures are implemented for reducing risks to, or maintaining risks within, specified levels.

kontrola rizika: Proces tijekom kojeg se donose odluke i implementiraju zaštitne mjere za reduciranje rizika ili održavanje rizika unutar specificiranih granica.

risk identification: The process of identifying risks using techniques such as brainstorming, checklists and failure history.

identificiranje rizika: Proces identificiranja rizika korištenjem tehnika kao što su brainstorming (oluja mozgova), liste provjere i povijesnost kvara.

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.

razina rizika: Važnost rizika kako ga definiraju utjecaj ili vjerojatnost njegovih karakteristika. Razina rizika može se koristiti za određivanje intenziteta testiranja kojeg treba poduzeti. Razina rizika može se izraziti kvalitativno (npr. visok, srednji, nizak) ili kvantitativno.

risk management: Systematic application of procedures and practices to the tasks of identifying, analyzing, prioritizing, and controlling risk.

upravljanje rizikom: Sistematska primjena procedura i praksi za radnje identificiranja, analiziranja, prioritiziranja i kontrolu rizika.

risk mitigation: See *risk control*.

ublažavanje rizika: See *risk control*.

risk type: A specific category of risk related to the type of testing that can mitigate (control) that category. For example the risk of user-interactions being misunderstood can be mitigated by usability testing.

vrsta rizika: Specifična kategorija rizika povezana s vrstom testiranja koja može ublažiti (kontrolirati) tu kategoriju. Npr. rizik korisničkih interakcija koje su krivo protumačene može se ublažiti testiranjem uporabivosti.

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 *error-tolerance, fault-tolerance*.

robustnost: Stupanj do kojeg komponenta ili sustav može ispravno funkcionirati u prisustvu nevaljanih ulaznih vrijednosti ili stresnih uvjeta okruženja. [IEEE 610] Vidjeti također *error-tolerance, fault-tolerance*.

robustness testing: Testing to determine the robustness of the software product.

testiranje robustnosti: Testiranje da se utvrdi robusnost programskog proizvoda.

root cause: A source of a defect such that if it is removed, the occurrence of the defect type is decreased or removed. [CMMI]

osnovni uzrok: Izvor pogreške koji je takav da, ako ga se otkloni, pojava te vrste pogreške je smanjena ili uklonjena. [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.

analiza osnovnog uzroka: Tehnika analize sa ciljem identificiranja osnovnog uzroka pogrešaka. Usmjeravanjem korektivnih mjera prema osnovnim uzrocima smatra se da će vjerojatnost ponovnog pojavljivanja pogreške biti umanjena.

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]

sigurnost: Sposobnost programskog proizvoda da u specificiranom kontekstu uporabe dosegne prihvatljive razine rizika štete prema ljudima, poslovanju, softveru, vlasništvu ili okolišu. [ISO 9126]

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.

sigurnosno kritični sustav: Sustav čiji kvar ili neispravnost u radu mogu rezultirati smrću ili ozbiljnim povredama ljudi, gubitkom ili ozbiljnim oštećenjem opreme ili štetom za okoliš.

safety testing: Testing to determine the safety of a software product.

testiranje sigurnosti: Testiranje kako bi se utvrdila sigurnost programskog proizvoda.

sanity test: See *smoke test*.

test zdravlja: Vidi *smoke test*.

scalability: The capability of the software product to be upgraded to accommodate increased loads. [After Gerrard]

sposobnost rasta, skalabilnost: Sposobnost programskog proizvoda za nadograđivanje kako bi podržao povećano opterećenje. [prema Gerrard]

scalability testing: Testing to determine the scalability of the software product.

testiranje sposobnosti rasta, testiranje skalabilnosti: Testiranje u svrhu određivanja skalabilnosti programskog proizvoda.

scenario testing: See *use case testing*.

testiranje scenarija: Vidi *use case testing*.

scribe: The person who records each defect mentioned and any suggestions for process improvement during a review meeting, on a logging form. The scribe has to ensure that the logging form is readable and understandable.

zapisničar: Osoba koja u zapisnički formular bilježi svaku spomenutu pogrešku i svaku sugestiju za unapređenje procesa tijekom sastanka pregleda. Zapisničar mora osigurati da je zapisnički formular čitak i razumljiv.

scripted testing: Test execution carried out by following a previously documented sequence of tests.

skriptno testiranje: Izvršavanje testa koje se provodi na temelju prethodno dokumentiranog slijeda testova.

scripting language: A programming language in which executable test scripts are written, used by a test execution tool (e.g. a capture/playback tool).

skriptorski jezik: Programski jezik u kojem se pišu izvršne testne skripte koje koristi alat za izvršavanje (npr. alat za hvatanje/reprodukciju).

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 *functionality*.

osiguranje: Obilježja programskih proizvoda koja se odnose na njihovu sposobnost sprječavanja neovlaštenog pristupa (bilo slučajan ili namjeran) programima i podacima. [ISO 9126] Vidjeti također *functionality*.

security testing: Testing to determine the security of the software product. See also *functionality testing*.

testiranje osiguranja: Testiranje kako bi se utvrdilo osiguranje programskog proizvoda. Vidjeti također *functionality testing*.

security testing tool: A tool that provides support for testing security characteristics and vulnerabilities.

alat za testiranje osiguranja: Alat koji omogućuje podršku testiranju sigurnosnih karakteristika i ranjivosti.

security tool: A tool that supports operational security.

alat za osiguranje: Alat koji podržava operativno osiguranje.

serviceability testing: See *maintainability testing*.

testiranje servisibilnosti: Vidi *maintainability testing*.

severity: The degree of impact that a defect has on the development or operation of a component or system. [After IEEE 610]

težina: Stupanj utjecaja koji greška ima na razvoj ili operaciju komponente ili sustava. [prema IEEE 610]

simulation: The representation of selected behavioral characteristics of one physical or abstract system by another system. [ISO 2382/1]

simulacija: Prikaz određenih karakteristika ponašanja jednog fizičkog ili apstraktnog sustava od strane drugog sustava. [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. [After IEEE 610, DO178b]

simulator: Uređaj, računalni program ili sustav koji se koristi tijekom testiranja, a koji se, kada mu se pruži skup kontroliranih ulaza, ponaša ili djeluje kao dani sustav. [prema IEEE 610, DO178b]

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.

testiranje prihvatljivosti na lokaciji: Testiranje prihvatljivosti od strane korisnika/kupaca na njihovoj lokaciji kako bi se odredilo zadovoljava li komponenta ili sustav potrebe korisnika/kupca i uklapa li se u poslovni proces što obično uključuje kako hardver tako i softver.

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 *intake test*.

dimni test: Skup svih definiranih/planiranih testnih slučajeva koji pokrivaju glavnu funkciju komponente/sustava u svrhu provjere ključnih funkcija programa, ali bez zamaranja sa finim detaljima. Dnevno izdanje i dimni test spadaju u najbolje prakse širom industrije. Vidjeti također *intake test*.

software: Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. [IEEE 610]

softver, programska oprema, programska podrška: Računalni programi, procedure i eventualno povezana dokumentacija i podaci koji se odnose na rad računalnog sustava. [IEEE 610]

software attack: See *attack*.

softverski napad: Vidi *attack*.

Software Failure Mode and Effect Analysis (SFMEA): See *Failure Mode and Effect Analysis (FMEA)*.
Software Failure Mode and Effect Analysis (SFMEA): Vidi *Failure Mode and Effect Analysis (FMEA)*.

Software Failure Mode Effect, and Criticality Analysis (SFMECA): See *Failure Mode and Effect, and Criticality Analysis (FMECA)*.

Software Failure Mode Effect, and Criticality Analysis (SFMECA): Vidi *Failure Mode and Effect, and Criticality Analysis (FMECA)*.

Software Fault Tree Analysis (SFTA): See *Fault Tree Analysis (FTA)*.
analiza stabla greške softvera: Vidi *Fault Tree Analysis (FTA)*.

software feature: See *feature*.

atribut softvera, obilježje softvera: Vidi *feature*.

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.

životni ciklus softvera: Vrijeme koje počinje kada je programski proizvod koncipiran, a završava kada programski proizvod više nije u uporabi. Životni ciklus softvera obično uključuje fazu koncepcije, fazu definiranja zahtjeva, fazu dizajna, fazu implementacije, fazu testiranja, instalacije i povjere, fazu rada i održavanja i ponekad, fazu umirovljenja. Treba imati u vidu da se ove faze mogu preklapati ili se mogu izvršavati iterativno.

software product characteristic: See *quality attribute*.

obilježje programskog proizvoda, karakteristika programskog proizvoda: Vidi *quality attribute*.

software quality: The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs. [After ISO 9126]

kvaliteta softvera: Sveukupnost funkcionalnosti i svojstava programskog proizvoda koja se temelji na njegovoj mogućnosti zadovoljenja (unaprijed) iskazanih ili impliciranih potreba. [prema ISO 9126]

software quality characteristic: See *quality attribute*.

obilježje kvalitete softvera, atribut kvalitete softvera, faktor kvalitete softvera: Vidi *quality attribute*.

software test incident: See *incident*.

incident testiranja softvera: Vidi *incident*.

software test incident report: See *incident report*.

izvješće o incidentu testiranja softvera: Vidi *incident report*.

Software Usability Measurement Inventory (SUMI): A questionnaire based usability test technique to evaluate the usability, e.g. user-satisfaction, of a component or system. [Veenendaal]

lista mjerljive uporabivosti softvera (SUMI): Tehnika testiranja uporabivosti zasnovana na upitniku u svrhu procjenjivanja uporabivosti komponente ili sustava, npr. zadovoljstvo korisnika. [Veenendaal]

source statement: See *statement*.

izvorna naredba: Vidi *statement*.

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. [After IEEE 610]

specifikacija: Dokument koji određuje, u idealnom slučaju na potpun, precizan i provjerljiv način zahtjeve, dizajn, ponašanje ili ostale karakteristike komponente ili sustava i često procedura za određivanje je li tim odredbama udovoljeno. [prema IEEE 610]

specification-based testing: See *black box testing*.

testiranje zasnovano na specifikaciji: Vidi *black box testing*.

specification-based technique: See *black box test design technique*.

tehnika zasnovana na specifikaciji: Vidi *black box test design technique*.

specification-based test design technique: See *black box test design technique*.

tehnika oblikovanja testova zasnovana na specifikaciji: Vidi *black box test design technique*.

specified input: An input for which the specification predicts a result.

specificirani ulaz: Ulaz za koji specifikacija predviđa rezultat.

stability: The capability of the software product to avoid unexpected effects from modifications in the software. [ISO 9126] See also *maintainability*.

stabilnost: Sposobnost programskog proizvoda da izbjegne neočekivane efekte izazvane promjenama u softveru. [ISO 1926] Vidjeti također *maintainability*.

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]

fazna reprezentacija: Struktura modela koja sadržavajući ciljeve skupa procesnih područja, uspostavlja razinu zrelosti; svaka razina čini temelj za daljnje razine. [CMMI]

standard software: See *off-the-shelf software*.

standardni softver: Vidi *off-the-shelf software*.

standards testing: See *compliance testing*.

testiranje standarda: Vidi *compliance testing*.

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]

dijagram stanja: Dijagram koji prikazuje stanja koja komponenta ili sustav može usvojiti, te pokazuje događaje ili okolnosti koje su prouzročili i/ili su posljedica promjene stanja iz jednog u drugo. [IEEE 610]

state table: A grid showing the resulting transitions for each state combined with each possible event, showing both valid and invalid transitions.

tablica stanja: Mreža koja pokazuje rezultate prijelaza za svako stanje kombinirano sa svakim mogućim događajem, pokazujući i valjane i nevaljane prijelaze.

state transition: A transition between two states of a component or system.

prijelaz stanja: Prijelaz između dva stanja komponentne ili sustava.

state transition testing: A black box test design technique in which test cases are designed to execute valid and invalid state transitions.

testiranje prijelaza stanja: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvršavanje valjanih i nevaljanih prijelaza između stanja.

statement: An entity in a programming language, which is typically the smallest indivisible unit of execution.

naredba: Jedinica u programskom jeziku koja je obično najmanja nedjeljiva jedinica izvršenja.

statement coverage: The percentage of executable statements that have been exercised by a test suite.

pokrivenost linija koda: Postotak izvršnih linija koje su ispitane pomoću skupa testnih slučajeva.

statement testing: A white box test design technique in which test cases are designed to execute statements.

testiranje naredbi: Tehnika oblikovanja testova bijele kutije u kojoj su testni slučajevi oblikovani za izvršavanje naredbi.

static analysis: Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software artifacts.

statička analiza: Analiza softverskih artefakata, npr. zahtjeva ili koda, provedena bez izvršavanja tih artefakata.

static analysis tool: See *static analyzer*.

alat za statičku analizu: Vidi *static analyzer*.

static analyzer: A tool that carries out static analysis.

statički analizator: Alat pomoću kojeg se provodi statička analiza.

static code analysis: Analysis of source code carried out without execution of that software.

statička analiza koda: Analiza izvornog koda provedena bez pokretanja tog softvera.

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.

statički analizator koda: Alat koji provodi statičku analizu koda. Alat provjerava izvorni kôd s obzirom na specifična obilježja kao što su standardi kodiranja, metrike kvalitete ili anomalije u tijeku podataka.

static testing: Testing of a component or system at specification or implementation level without execution of that software, e.g. reviews or static code analysis.

statičko testiranje: Testiranje komponente ili sustava na razini specifikacije ili implementacije bez pokretanja softvera, npr. pregledi ili statička analiza koda.

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 *operational profile testing*.

statističko testiranje: Tehnika oblikovanja testova u kojoj se koristi model statističke raspodjele ulaza da se konstruiraju reprezentativni testni slučajevi. Vidjeti također *operational profile testing*.

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]

evidencija statusa: Element upravljanja konfiguracijom koji se sastoji od snimanja informacija i izvještavanja o informacijama koje su potrebne za učinkovito upravljanje konfiguracijom. Ove informacije uključuju popis odobrenih identifikacija konfiguracije, stanje predloženih promjena u konfiguraciji i implementaciju statusa odobrenih promjena. [IEEE 610]

storage: See *resource utilization*.

pohrana: Vidi *resource utilization*.

storage testing: See *resource utilization testing*.

testiranje pohrane: Vidi *resource utilization testing*.

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. [After IEEE 610] See also *performance testing, load testing*.

testiranje opterećenja, testiranje pod pritiskom: Vrsta testiranja performansi koje se provodi u svrhu evaluiranja sustava ili komponente na ili preko granica predviđenih ili specificiranih radnih opterećenja, ili sa reduciranom dostupnošću resursa kao što su pristup memoriji ili serverima. [prema IEEE 610] Vidjeti također *performance testing, load testing*.

stress testing tool: A tool that supports stress testing.

alat za testiranje opterećenja: Alat koji podržava testiranje opterećenja.

structure-based testing: See *white-box testing*.

testiranje zasnovano na strukturi: Vidi *white-box testing*.

structure-based technique: See *white box test design technique*.

tehnika zasnovana na strukturi: Vidi *white box test design technique*.

structural coverage: Coverage measures based on the internal structure of a component or system.

strukturna pokrivenost: Mjere pokrivenosti zasnovane na internoj strukturi komponente ili sustava.

structural test design technique: See *white box test design technique*.

tehnika oblikovanja strukturnih testova: Vidi *white box test design technique*.

structural testing: See *white box testing*.

strukturno testiranje: Vidi *white box testing*.

structured walkthrough: See *walkthrough*.

strukturni prolaz: Vidi *walkthrough*.

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. [After IEEE 610]

prividni modul, zamjenski element koji se privremeno umeće u program: Skeletna implementacija ili implementacija za posebnu namjenu određene softverske komponente koja se koristi kako bi se razvijala ili testirala komponenta koja poziva ili je na neki drugi način ovisna o toj komponenti. Prividni modul zamjenjuje komponentu koja se poziva. [prema IEEE 610]

subpath: A sequence of executable statements within a component.

podputanja: Slijed izvršnih naredbi unutar komponente.

suitability: The capability of the software product to provide an appropriate set of functions for specified tasks and user objectives. [ISO 9126] See also *functionality*.

prikladnost: Sposobnost programskog proizvoda da omogući adekvatan skup funkcija za određene radnje i korisnikove ciljeve. [ISO 9126] Vidjeti također *functionality*.

suspension criteria: The criteria used to (temporarily) stop all or a portion of the testing activities on the test items. [After IEEE 829]

kriterij za prekid, kriterij obustave: Kriterij korišten da se (privremeno) zaustave sve ili dio aktivnosti testiranja na testiranim objektima. [prema 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.

testiranje sintakse: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani na temelju definicije domene ulaza i/ili domene izlaza.

system: A collection of components organized to accomplish a specific function or set of functions. [IEEE 610]

sustav: Skup komponenti organiziranih u svrhu postizanja određene funkcije ili skupa funkcija. [IEEE 610]

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.

sustav sustava: Višestruki heterogeni, distribuirani sustavi koji su ugrađeni u mreže na mnogim razinama i povezani u mnogim domenama, a primjenju se za međudisciplinarnu opće probleme i svrhe velikih razmjera.

system integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

testiranje integracije sustava: Testiranje integracije sustava i paketa; testiranje sučelja prema vanjskim organizacijama (npr. EDI, Internet).

system testing: The process of testing an integrated system to verify that it meets specified requirements. [Hetzel]

testiranje sustava, provjera sustava: Proces testiranja integriranog sustava u svrhu provjere da li sustav odgovara specificiranim zahtjevima. [Hetzel]

T

technical review: A peer group discussion activity that focuses on achieving consensus on the technical approach to be taken. [Gilb and Graham, IEEE 1028] See also *peer review*.

tehnički pregled: Aktivnost grupne rasprave među kolegama usredotočena na postizanje konsenzusa o tehničkom pristupu kojim treba krenuti. [Gilb and Graham, IEEE 1028] Vidjeti također *peer review*.

test: A set of one or more test cases. [IEEE 829]

test, provjera: Skup jednog ili više testnih slučajeva. [IEEE 829]

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.

pristup testiranju: Implementacija strategije testiranja za određeni projekt. Ona tipično uključuje odluke koje će na temelju cilja (testnog) projekta i provedene procjene rizika, pratiti početne točke glede testnog procesa, tehnike oblikovanja testova koje će se koristiti, kriterije izlaza i vrste testova koji će se izvršavati.

test automation: The use of software to perform or support test activities, e.g. test management, test design, test execution and results checking.

automatiziranje testiranja: Korištenje softvera u svrhu izvođenja ili održavanja aktivnosti testiranja, npr. upravljanje testiranjem, oblikovanje testova i provjera rezultata.

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. [After TMap]

osnovica za testiranje: Svi dokumenti iz kojih se mogu izvesti zahtjevi komponente ili sustava. Dokumentacija na kojoj se temelje testni slučajevi. Ako se dokument može nadopunjavati samo formalnom procedurom amandmana, tada se osnovica za testiranje naziva smrznutom osnovicom za testiranje. [prema TMap]

test bed: See *test environment*.

testna platforma: Vidi *test environment*.

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. [After IEEE 610]

testni slučaj: Skup ulaznih vrijednosti, predujeta za izvršavanje, očekivanih rezultata, te uvjeta koji se pojavljuju nakon izvršenja, razvijenih za određeni cilj ili okolnost testiranja, kao što je provođenje određenog programskog tijeka ili provjera usklađenosti sa specifičnim zahtjevima. [prema IEEE 610]

test case design technique: See *test design technique*.

tehnika oblikovanja testnih slučajeva: Vidi *test design technique*.

test case specification: A document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item. [After IEEE 829]

specifikacija testnih slučajeva: Dokument koji specificira skup testnih slučajeva (ciljevi, ulazi, akcije testiranja, očekivani rezultati i predujete za izvršavanje) za jedinicu koja se testira. [prema IEEE 829]

test case suite: See *test suite*.

skup testnih slučajeva: Vidi *test suite*.

test charter: A statement of test objectives, and possibly test ideas about how to test. Test charters are used in exploratory testing. See also *exploratory testing*.

povelja testiranja: Tvrdnja o ciljevima testiranja i mogućim idejama kako testirati. Povelje testiranja koriste se u eksplorativnom testiranju. Vidjeti također *exploratory testing*.

test closure: During the test closure phase of a test process data is collected from completed activities to consolidate experience, testware, facts and numbers. The test closure phase consists of finalizing and archiving the testware and evaluating the test process, including preparation of a test evaluation report. See also *test process*.

završetak testiranja: Tijekom faze završetka testiranja u procesu testiranja, od završenih aktivnosti prikupljaju se podaci kako bi se objedinila iskustva, testver, činjenice i brojevi. Faza završetka testiranja sastoji se od finaliziranja i arhiviranja testvera i evaluiranja procesa testiranja uključujući i pripreme izvještaja o evaluaciji testova. Vidjeti također *test process*.

test comparator: A test tool to perform automated test comparison of actual results with expected results.

test komparator: Alat testiranja za automatiziranu usporedbu stvarnih rezultata sa očekivanim rezultatima.

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.

usporedba testa: Proces utvrđivanja razlika između stvarnih rezultata koje daje testirana komponenta ili sustav i očekivanih rezultata testa. Usporedba testa može se provesti tijekom izvršavanja testa (dinamička usporedba) ili nakon izvršenja testa.

test completion criteria: See *exit criteria*.

kriterij završetka testiranja: Vidi *exit criteria*.

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, feature, quality attribute, or structural element.

uvjet za testiranje: Jedinica ili događaj komponente ili sustava koji može biti verificiran jednim ili više testnih slučajeva, npr. funkcija, transakcija, svojstvo, atribut kvalitete ili strukturalni element.

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 *test management*.

kontrola testiranja, nadzor testiranja: Radnja upravljanja testiranjem koja se odnosi na razvoj i primjenu niza korektivnih akcija kako bi se testni projekt usmjerio u pravom smjeru kada nadzor pokaže odstupanje u odnosu na ono što je bilo planirano. Vidjeti također *test management*.

test coverage: See *coverage*.

pokrivenost testa: Vidi *coverage*.

test cycle: Execution of the test process against a single identifiable release of the test object.

ciklus testa: Izvršavanje procesa testiranja u odnosu na izdanje testnog objekta koje je moguće jedinstveno identificirati.

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.

ispitni podaci, testni podaci: Podaci koji postoje (na primjer u bazi podataka) prije izvršenja testa i koji utječu ili su pod utjecajem komponente ili sustava koji se testira.

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.

alat za pripremu ispitnih podataka, alat za pripremu testnih podataka: Vrsta alata za testiranje koji omogućuje da se podaci selektiraju iz postojeće baze, ili da se kreiraju, generiraju, da se s njima manipulira i obrađuje ih za uporabu u testiranju.

test design: (1) See *test design specification*. (2) The process of transforming general testing objectives into tangible test conditions and test cases.

oblikovanje testova: (1) Vidi *test design specification*. (2) Proces transformiranja općih ciljeva testiranja u opipljive uvjete testiranja i testne slučajeve.

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. [After IEEE 829]

specifikacija oblikovanja testova: Dokument koji specificira testne uvjete (pokrivenost) za jedinicu koja se testira, detaljni pristup testiranju i s tim povezane testne slučajeve visoke razine. [prema IEEE 829]

test design technique: Procedure used to derive and/or select test cases.

tehnika oblikovanja testova: Procedura koja se koristi da se izvedu i/ili odaberu testni slučajevi.

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.

alat za oblikovanje testova: Alat koji podržava aktivnost oblikovanja testova tako da generira testne ulaze iz specifikacija koje se mogu držati u repozitoriju CASE (Computer Aided Software Engineering) alata, npr. alat za upravljanje zahtjevima, zatim iz specificiranih uvjeta testiranja sadržanih u samome alatu, ili iz koda.

test driven development: A way of developing software where the test cases are developed, and often automated, before the software is developed to run those test cases.

razvoj vođen testiranjem, razvoj usmjeravan testiranjem: Način razvoja softvera gdje su testni slučajevi razvijeni, a često automatizirani, prije nego je razvijen softver za kojeg bi se ti testni slučajevi pokretali.

test driver: See *driver*.

ispitni program, program za ispitivanje: Vidi *driver*.

test environment: An environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test. [After IEEE 610]

okolina za testiranje, testna okolina: Okolina koja uključuje hardver, instrumente, simulatore, alate za testiranje i ostale elemente podrške da se izvede test. [prema IEEE 610]

test estimation: The calculated approximation of a result (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.

procjena testiranja: Kalkulirana procjena rezultata (npr. količina rada, datum završetka, ukupni troškovi, broj testnih slučajeva, itd.) koja je upotrebljiva čak i kada su ulazni podaci nepotpuni, nesigurni ili sadržavaju šum.

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.

izvešće evaluacije testiranja: Dokument koji se izrađuje na kraju procesa testiranja, a sažima sve testne aktivnosti i rezultate. On također sadrži evaluaciju procesa testiranja i korisne spoznaje.

test execution: The process of running a test on the component or system under test, producing actual result(s).

izvršavanje testa: Proces pokretanja testa komponente ili sustava koji se testira, koji proizvodi stvarni(e) rezultat(e).

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.

automatizacija izvršavanja testova: Uporaba softvera, npr. alati za hvatanje/reprodukciju u svrhu kontroliranja izvršavanja testova, usporedbe stvarnih i očekivanih rezultata, postavljanja preduvjeta testiranja i ostalih funkcija kontrole nad testovima i izvještavanja o testovima.

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]

faza izvršavanja testova: Vremensko razdoblje u životnom ciklusu razvoja softvera tijekom kojega se izvršavaju komponente softverskog proizvoda, a softverski se proizvod evaluira da se odredi da li je ili nije udovoljeno zahtjevima. [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.

raspored izvršavanja testova: Nacrt za izvršavanje testnih procedura. Testne procedure uključene su u raspored izvršavanja testova u njihovom kontekstu i u redosljedu kojim će biti izvršavane.

test execution technique: The method used to perform the actual test execution, either manual or automated.

tehnika izvršavanja testova: Metoda koja se koristi za stvarno izvršavanje testa, bilo ručno ili automatski.

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]

alat za izvršenje testa: Vrsta alata za testiranje koji može izvršiti drugi softver koristeći automatiziranu testnu skriptu, npr. hvatanje/reprodukcija. [Fewster and Graham]

test fail: See *fail*.

neuspjeh testa: Vidi *fail*.

test generator: See *test data preparation tool*.

generator testova: Vidi *test data preparation tool*.

test harness: A test environment comprised of stubs and drivers needed to execute a test.

testna sprema: Testno okruženje koje se sastoji od prividnih modula i pogonskih programa koji su potrebni za izvršenje testa.

test incident: See *incident*.

incident testiranja: See *incident*.

test incident report: See *incident report*.

izvješće o incidentu testiranja: See *incident report*.

test implementation: The process of developing and prioritizing test procedures, creating test data and, optionally, preparing test harnesses and writing automated test scripts.

implementacija testa: Proces razvoja i prioritiziranja procedura testiranja, kreiranja testnih podataka i opcionalno pripremanja testne spreme i pisanja automatiziranih testnih skripti.

test infrastructure: The organizational artifacts needed to perform testing, consisting of test environments, test tools, office environment and procedures.

infrastruktura za testiranje: Organizacijski materijal potreban za provođenje testiranja koji se sastoji od testnih okruženja, alata za testiranje, uredskog okruženja i procedura.

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.

ulaz za testiranje: Podaci koje iz vanjskih izvora prima testni objekt tijekom izvršavanja testa. Vanjski izvor može biti hardver, softver ili čovjek.

test item: The individual element to be tested. There usually is one test object and many test items. See also *test object*.

element testa, element testiranja: Individualni element koji se testira. Obično postoji jedan testni objekt i mnogo testnih predmeta. Vidi također *test object*.

test item transmittal report: See *release note*.

prijelazno izvješće o elementu testiranja: Vidi *release note*.

test leader: See *test manager*.

voditelj testiranja: Vidi *test manager*.

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. [After TMap]

razina testiranja: Grupa testnih aktivnosti koje su zajedno organizirane i kojima se zajednički upravlja. Razina testiranja povezana je s odgovornostima na projektu. Primjeri testnih razina su: test komponente, integracijski test, test sustava i test prihvatljivosti. [prema TMap]

test log: A chronological record of relevant details about the execution of tests. [IEEE 829]

zapis o testiranju: Kronološki zapis relevantnih detalja o izvršenju testa. [IEEE 829]

test logging: The process of recording information about tests executed into a test log.

bilježenje testiranja, zapisivanje testiranja: Proces bilježenja informacija o izvršenim testovima u zapis o testiranju.

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.

voditelj testiranja, test menadžer: Osoba zadužena za upravljanje projektom aktivnostima testiranja i resursa te za evaluaciju testnog objekta. To je pojedinac koji upravlja, kontrolira, provodi, planira i regulira evaluaciju testnog objekta.

test management: The planning, estimating, monitoring and control of test activities, typically carried out by a test manager.

upravljanje testiranjem: Planiranje, procjena, nadzor i kontrola aktivnosti testiranja koje provodi voditelj testiranja.

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.

alat za upravljanje testiranjem: Alat koji olakšava upravljanje testiranjem i dio procesa testiranja koji se odnosi na kontrolu. Često ima nekoliko mogućnosti kao što je upravljanje testverom (testnim materijalom), raspoređivanje testova, bilježenje rezultata, praćenje napretka, upravljanje incidentima i izvještavanje o testiranju.

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.

model zrelosti testa: Petorazinski okvir za unapređenje procesa testiranja povezan s modelom razine zrelosti (*Capability Maturity Model – CMM*), a opisuje ključne elemente učinkovitog procesa testiranja.

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.

integrirani model zrelosti testa: Petorazinski okvir za poboljšanje procesa testiranja povezan s integriranim modelom razine zrelosti (*Capability Maturity Model Integrated – CMMI*), a opisuje ključne elemente učinkovitog procesa testiranja.

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 actuals to that which was planned. See also *test management*.

nadzor testiranja: Radnja iz područja upravljanja testiranjem koja se odnosi na aktivnosti koje su u svezi s periodičkom provjerom statusa testnog projekta. Tijekom nadzora testiranja pripremaju se izvještaji koji uspoređuju stvarno s planiranim. Vidjeti također *test management*.

test object: The component or system to be tested. See also *test item*.

objekt testiranja: Komponenta ili sustav koji se testira. Vidjeti također *test item*.

test objective: A reason or purpose for designing and executing a test.

ciljevi testa: Razlog ili svrha za oblikovanje i izvršavanje testa.

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), a user manual, or an individual's specialized knowledge, but should not be the code. [After Adrion]

izvor testa: Izvor iz kojeg se određuje očekivani rezultat kojeg se uspoređuje sa stvarnim rezultatom softvera koji se testira. Oracle može biti postojeći sustav (za mjerenje performansi), korisnički priručnik ili specijalizirana osoba, ali ne bi smio biti kôd. [prema Adrion]

test outcome: See *result*.

rezultat testiranja: Vidi *result*.

test pass: See *pass*.

prolaz testa: Vidi *pass*.

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).

pokazatelj uspješnosti testiranja: Metrika visoke razine za učinkovitost i djelotvornost koja se koristi za vođenje i kontrolu razvoja s progresivnim testiranjem, npr. postotak detekcije defekata (*Defect Detection Percentage – DDP*).

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. [After Gerrard]

faza testiranja: Poseban skup testnih aktivnosti grupiranih u provodivu fazu projekta, npr. aktivnosti vezane za izvršavanje u određenoj razini testiranja. [prema 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. [After IEEE 829]

plan testiranja: Dokument koji opisuje doseg, pristup, resurse i raspored namjeravanih testnih aktivnosti. On među ostalim identificira elemente testiranja, svojstva koja će se testirati, testne radnje, tko će raditi pojedinu radnju, stupanj testerove neovisnosti, testno okruženje, tehnike oblikovanja testova, ulazne i izlazne kriterije koji će se koristiti kao i logičke razloge za njihov izbor te sve rizike koji se odnose na planiranje nepredviđenih situacija. To je zapis o procesu planiranja testa. [prema IEEE 829]

test planning: The activity of establishing or updating a test plan.

planiranje testiranja: Aktivnosti izrade i ažuriranja plana testiranja.

test policy: A high level document describing the principles, approach and major objectives of the organization regarding testing.

politika testiranja: Dokument visoke razine koji opisuje principe, pristupe i glavne ciljeve organizacije glede testiranja.

Test Point Analysis (TPA): A formula based test estimation method based on function point analysis. [TMap]

Test Point Analysis: Metoda procjenjivanja koja se temelji na formuli, a zasnovana je na analizi funkcijskih točaka (*Function Point Analysis – FPA*). [TMap]

test procedure: See *test procedure specification*.

testna procedura: Vidi *test procedure specification*.

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. [After IEEE 829]

specifikacija testne procedure: Dokument koji specificira slijed aktivnosti za izvršenje testa. Također je poznat kao testna skripta ili ručna testna skripta. [prema IEEE 829]

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.

proces testiranja: Osnovni proces testiranja uključuje planiranje i kontrolu testiranja, analizu i dizajn testiranja, implementaciju testa i izvršenje testa, evaluaciju kriterija izlaza i izvještavanje, te aktivnosti završetka testiranja.

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.

unapređenje procesa testiranja: Trajni okvir za unapređenje procesa testiranja koji opisuje ključne elemente učinkovitog procesa testiranja, osobito one koji se odnose na testiranje sustava i testiranje prihvatljivosti.

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.

izvješće o napretku testiranja: Dokument koji sumarizira aktivnosti testiranja i rezultate, kreira se u regularnim intervalima da izvjesti o napretku aktivnosti testiranja u odnosu na zacrtani plan (kao što je izvorni testni plan) i da komunicira rizike i alternative koji traže odluku od strane menadžmenta.

test record: See *test log*.

zapis o testiranju: Vidi *test log*.

test recording: See *test logging*.

bilježenje testiranja, dokumentiranje rezultata testiranja: Vidi *test logging*.

test reproduceability: An attribute of a test indicating whether the same results are produced each time the test is executed.

moгуćnost reproduciranja testa: Atribut testa koji ukazuje jesu li isti rezultati polučeni svaki put kada je test izvršen.

test report: See *test summary report*.

izvješće o testiranju: Vidi *test summary report*.

test requirement: See *test condition*.

zahtjev za test: Vidi *test condition*.

test rig: See *test environment*.

testna oprema: Vidi *test environment*.

test run: Execution of a test on a specific version of the test object.

ispitno izvođenje, pokusno izvođenje: Pokretanje testa na specifičnoj verziji objekta testiranja.

test run log: See *test log*.

zapis o ispitnom izvođenju: Vidi *test log*.

test result: See *result*.

rezultat testiranja, rezultat ispitivanja: Vidi *result*.

test scenario: See *test procedure specification*.

scenarij testiranja: Vidi *test procedure specification*.

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.

raspored testiranja: Lista aktivnosti, radnji i događaja testnog procesa koji identificiraju njihov planirani datum i/ili vrijeme početka i završetka te međuovisnosti.

test script: Commonly used to refer to a test procedure specification, especially an automated one.

skripta za testiranje, testna skripta: Često se koristi kao referenca na specifikaciju testne procedure, posebno automatizirane.

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 *exploratory testing*.

sesija testa: Neprekinuti vremenski period proveden u izvršavanju testova. U eksploratornom testiranju, svaka testna sesija je fokusirana na povelju testiranja, ali tester i mogu također istraživati nove mogućnosti ili zgode tijekom sesije. Tester kreira i izvršava testne slučajeve *ad hoc* i bilježi njihovo napredovanje. Vidjeti također *exploratory testing*.

test set: See *test suite*.

skup testova: Vidi *test suite*.

test situation: See *test condition*.

okolnosti testa: Vidi *test condition*.

test specification: A document that consists of a test design specification, test case specification and/or test procedure specification.

specifikacija testa: Dokument koji se sastoji od specifikacije dizajna testa, specifikacije testnih slučajeva i/ili specifikacije procedure testiranja.

test specification technique: See *test design technique*.
tehnika specifikacije testova: Vidi *test design technique*.

test stage: See *test level*.
razina testiranja: Vidi *test level*.

test strategy: A high-level description of the test levels to be performed and the testing within those levels for an organization or programme (one or more projects).

strategija testiranja: Opis razina testiranja na visokoj razini, koje će se izvesti i testiranje između tih razina za određenu organizaciju ili program (jedan ili više projekata).

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.

skup testova: Skup od nekoliko testnih slučajeva koji se odnose na komponentu ili sustav koji se testira, a gdje se postuvjet jednog testa koristi kao preduvjet za slijedeći.

test summary report: A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria. [After IEEE 829]

sažetak izvješća o testiranju: Dokument koji sažima aktivnosti testiranja i rezultate. Također sadrži i evaluaciju odgovarajućih elemenata testiranja u odnosu na izlazne kriterije. [prema IEEE 829]

test target: A set of exit criteria.
odredište testa, cilj testa: Skup izlaznih kriterija.

test technique: See *test design technique*.
tehnika testiranja: Vidi *test design technique*.

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]

alat za testiranje: Programski proizvod koji podržava jednu ili više aktivnosti testiranja kao što su planiranje i kontrola, specifikacija, izrada početnih datoteka i podataka, izvršavanje testova i analiza testova. [TMap]

test type: A group of test activities aimed at testing a component or system focused on a specific test objective, i.e. functional test, usability test, regression test etc. A test type may take place on one or more test levels or test phases. [After TMap]

vrsta testiranja: Grupa testnih aktivnosti čiji je cilj testiranje komponente ili sustava, a fokusirane su na specifični cilj testiranja, tj. testiranje funkcionalnosti, testiranje uporabivosti, regresijski test, itd. Može se odvijati u jednoj ili više faza testiranja. [prema TMap]

testability: The capability of the software product to enable modified software to be tested. [ISO 9126] See also *maintainability*.

pogodnost za testiranje: Sposobnost programskog proizvoda da omogući testiranje modificiranog softvera. [ISO 9126] Vidjeti također *maintainability*.

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. [After TMap]

pregled pogodnosti za testiranje: Detaljna provjera osnovice za testiranje radi utvrđivanja je li ona na adekvatnoj razini kvalitete kako bi djelovala kao ulazni dokument o procesu testiranja. [prema TMap]

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. [After IEEE 610]

zahtjevi koji se mogu testirati: Stupanj do kojeg je zahtjev obrazložen tako da je moguće započeti oblikovati testove (a posljedično i testne slučajeve) te izvršavati testove da se odredi jesu li zahtjevi dosegnuti. [prema IEEE 610]

tester: A skilled professional who is involved in the testing of a component or system.
tester, ispitivač: Profesionalno osposobljena osoba koja je uključena u testiranje komponente ili sustava.

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.

testiranje, ispitivanje, provjeravanje: Proces koji se sastoji od svih aktivnosti životnog ciklusa (kako statičkih tako i dinamičkih), a odnosi se na planiranje, pripremu i evaluaciju programskih proizvoda i proizvoda koji su u sprezi sa softverom kako bi se odredilo zadovoljavaju li oni specificirane zahtjeve, kako bi se utvrdilo odgovaraju li svrsi te da se pronađu pogreške.

testware: 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. [After Fewster and Graham]

testver: Artefakti nastali tijekom procesa testiranja potrebni za planiranje, dizajn i izvršenje testa, kao primjerice dokumentacija, skripte, ulazne vrijednosti, očekivani rezultati, procedure za instalaciju i deinstaliranje, datoteke, baze podataka, okruženja i bilo koji dodatni softver ili uslužni programi korišteni u testiranju. [prema Fewster and Graham]

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.

testiranje niti: Vrsta integracijskog testiranja komponente gdje progresivna integracija komponenti slijedi implementaciju podskupa zahtjeva nasuprot integraciji komponenti prema razinama hijerarhije.

time behavior: See *performance*.

ponašanje u vremenu: Vidi *performance*.

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 *integration testing*.

testiranje odozgo prema dolje: Inkrementalni pristup integracijskom testiranju gdje se najviša komponenta u hijerarhiji prva testira dok su niže komponente simulirane prividnim modulima. Testirane komponente se zatim koriste za testiranje komponenti nižih razina. Proces se ponavlja sve dok se ne testiranju i komponente najniže razine. Vidjeti također *integration testing*.

traceability: The ability to identify related items in documentation and software, such as requirements with associated tests. See also *horizontal traceability*, *vertical traceability*.

sljedivost: Sposobnost identifikacije povezanih elemenata u dokumentaciji i softveru, kao što su zahtjevi s njima povezanim testovima. Vidjeti također *horizontal traceability*, *vertical traceability*.

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 *usability*.

razumljivost, shvatljivost: Sposobnost programskog proizvoda da omogući korisniku razumjeti da li softver odgovara i kako se može koristiti za određene radnje i uvjete uporabe. [ISO 9126] Vidjeti također *usability*.

unit: See *component*.

jedinica: Vidi *component*.

unit testing: See *component testing*.

testiranje jedinice, testiranje modula: Vidi *component testing*.

unreachable code: Code that cannot be reached and therefore is impossible to execute.

nedostupan kôd: Kôd kojeg je nemoguće dohvatiti i stoga ga je nemoguće izvršiti.

usability: The capability of the software to be understood, learned, used and attractive to the user when used under specified conditions. [ISO 9126]

upotrebljivost, uporabivost, uporabljivost, prilagođenost korisniku: Sposobnost softvera da bude razumljiv, lagan za učenje i korištenje, te atraktivan za korisnika kada se koristi pod specificiranim uvjetima. [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. [After ISO 9126]

testiranje upotrebljivosti, testiranje uporabivosti: Testiranje pomoću kojeg se određuje da li je programski proizvod pod specificiranim uvjetima razumljiv, lagan za učenje i korištenje, te atraktivan za korisnike. [prema ISO 9126]

use case: A sequence of transactions in a dialogue between a user and the system with a tangible result.

slučaj korištenja, slučaj primjene: Slijed transakcija u dijalogu između korisnika i sustava s opipljivim rezultatom.

use case testing: A black box test design technique in which test cases are designed to execute user scenarios.

testiranje slučaja korištenja, testiranje slučaja primjene: Tehnika oblikovanja testova crne kutije u kojoj su testni slučajevi oblikovani za izvršavanje korisničkih scenarija.

user acceptance testing: See *acceptance testing*.

testiranje prihvatljivosti od strane korisnika: Vidi *acceptance testing*.

user scenario testing: See *use case testing*.

testiranje korisničkih scenarija: Vidi *use case testing*.

user test: A test whereby real-life users are involved to evaluate the usability of a component or system.

korisnički test, korisničko testiranje: Test u koji su uključeni stvarni korisnici kako bi evaluirali uporabivost komponente ili sustava.

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]

okvir za testiranje jedinica, okvir za jedinični test: Alat koji omogućuje okolinu za testiranje jedinice ili komponente u kojem se komponenta može testirati u izolaciji ili s odgovarajućim prividnim modulima ili upravljačkim programima. Također omogućuje i podršku programeru kao što je sposobnost otkrivanja pogrešaka. [Graham]

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-model: Okvir koji opisuje aktivnosti životnog ciklusa razvoja softvera od specifikacije zahtjeva do održavanja. V-model ilustrira kako se aktivnosti testiranja mogu integrirati u svaku fazu životnog ciklusa razvoja softvera.

validation: Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled. [ISO 9000]

validacija, provjera ispravnosti, provjera (ocjena) valjanosti: Potvrda kroz ispitivanje i pribavljanje objektivnog dokaza da su ispunjeni zahtjevi za određenu primjenu ili aplikaciju. [ISO 9000]

variable: An element of storage in a computer that is accessible by a software program by referring to it by a name.

varijabla: Element pohrane u računalu kojem softverski program može pristupiti pozivajući se na njegov naziv.

verification: Confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled. [ISO 9000]

verifikacija, ovjera: Potvrda kroz ispitivanje i pribavljanje objektivnih dokaza da li su ispunjeni specificirani zahtjevi. [ISO 9000]

vertical traceability: The tracing of requirements through the layers of development documentation to components.

vertikalna sljedivost: Sljedivost zahtjeva kroz slojeve razvojne dokumentacije prema komponentama.

version control: See *configuration control*.

upravljanje verzijama: Vidi *configuration control*.

volume testing: Testing where the system is subjected to large volumes of data. See also *resource-utilization testing*.

testiranje volumena: Testiranje gdje je sustav suočen s velikim volumenom podataka. Vidjeti također *resource-utilization testing*.

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 *peer review*.

prolaz: Prezentacija po koracima od strane autora dokumenta s ciljem prikupljanja informacija i uspostavljanja općeg razumijevanja njezinog sadržaja. [Freedman i Weinberg, IEEE 1028] Vidjeti također *peer review*.

white-box techniques: See *white-box test design techniques*.

tehnike bijele kutije: Vidi *white-box test design techniques*.

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.

tehnika oblikovanja testova bijele kutije: Procedura za izvođenje i/ili odabir testnih slučajeva koja se temelji na analizi interne strukture komponente ili sustava.

white-box testing: Testing based on an analysis of the internal structure of the component or system.

testiranje bijele kutije: Testiranje zasnovano na analizi interne strukture komponente ili sustava.

Wide Band Delphi: An expert based test estimation technique that aims at making an accurate estimation using the collective wisdom of the team members.

Wideband Delphi: Ekspertno orijentirana tehnika procjene testiranja, a čiji je cilj davanje točne procjene korištenjem kolektivne mudrosti članova tima.

wild pointer: A pointer that references a location that is out of scope for that pointer or that does not exist. See also *pointer*.

pokazivač koji je deklariran, a nije inicijaliziran: Pokazivač koji referencira lokaciju koja je izvan dosega za taj pokazivač ili koja ne postoji. Vidjeti također *pointer*.

Prilog A (informativno)

Indeks izvora; slijedeći izvori, a koji nisu vezani uz norme, korišteni su u razvoju ovoga pojmovnika:

[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 Modelled by Finite-State 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.

Prilog B (metoda komentiranja ovog pojmovnika)

Komentari vezani uz ovaj dokument su dobrodošli da se pojmovnik unaprijedi na dobrobit čitave testne zajednice.

Prilikom slanja komentara, molimo navedite slijedeće informacije:

- Vaše ime i kontakt informacije (uključujući e-mail adresu);
- Broj verzije pojmovnika (trenutno 2.0);
- Točan navod pojma u pojmovniku na koji se komentar odnosi;
- Dodatne informacije, kao što su npr. obrazloženje za predložene promjene i/ili referenca na korišteni izvor.

Komentare možete poslati na engleskome jeziku pomoću više komunikacijskih kanala:

1. E-mail: eve@improveqs.nl;
2. Pošta: Improve Quality Services BV, attn. Mr. E. van Veenendaal, Waalresegweg 39, 5554 HA, Valkenswaard, The Netherlands;
3. FAX: +31 40 20 21450, molimo označite „attn. Mr. E. van Veenendaal“.